

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

1. Beberapa penelitian terdahulu yang dijadikan acuan dalam penelitian ini adalah : Ayu & Fitri (2019), meneliti tentang Perancangan Sistem Informasi Pemesanan Wedding Organizer Online. Bagi beberapa orang yang belum berpengalaman, tidak punya cukup waktu dan tenaga untuk membantu dalam persiapan dan pelaksanaan acara pernikahan, maka membutuhkan jasa untuk melakukan segala aktifitas persiapan dan pelaksanaan pernikahan tersebut. Dengan wedding organizer ini pemangku hajat tidak perlu kesulitan untuk mempersiapkan segala kebutuhan acara pernikahan, jasa ini memberikan informasi dan layanan mengenai berbagai hal yang berhubungan dengan acara pernikahan, mulai dari berbagai macam paket pernikahan, tata rias, busana, koordinasi dekorasi dan hiburan. Metode yang digunakan dalam penelitian ini adalah SDLC. Pembuatan sistem pemesanan wedding organizer online dengan bahasa pemrograman PHP dan database MySQL bertujuan untuk membangun sistem pemesanan wedding online, yang diharapkan dapat mendukung dan memudahkan pemesanan dan promosi wedding organizer. Hasilnya adalah sistem informasi wedding organizer berbasis Web yang dapat memberikan informasi wedding organizer, pemesanan secara online serta menjadi media promosi bagi pemilik wedding organizer.
2. Afriyonga et al., (2014), meneliti tentang Perancangan Sistem Informasi Administrasi Jasa Foto Pernikahan Berbasis Web Pada Euphoria Photo Studio. Euphoria Studio yang merupakan sebuah perusahaan yang bergerak dalam penyediaan jasa dokumentasi pernikahan juga ingin memanfaatkan teknologi internet untuk mempromosikan bisnisnya. Karena selama ini konsumen tidak mendapatkan informasi yang lengkap mengenai paket foto yang ada di Euphoria Studio. Selain itu permasalahan lainnya adalah

dalam penentuan harga terkadang hubungan kekerabatan masih menjadi bahan pertimbangan yang mengakibatkan kurang konsistennya pembayaran penuh. Dari permasalahan tersebut peneliti memberi solusi berupa sistem informasi administrasi jasa foto pernikahan berbasis web agar konsumen bisa langsung melihat informasi lengkap mengenai paket foto dan sekaligus bisa memesan secara online.

3. Ramdhani et al., (2018), meneliti tentang Sistem Informasi Penyewaan Peralatan Event Organizer Berbasis Web pada PT . Adecon Jakarta. Penyewaan Peralatan Event Organizer pada PT. Adecon Jakarta ini masih menggunakan sistem konvensional, oleh karena itu pemesanan, pembayaran dan konfirmasi pembayaran menjadi kurang efektif dan efisien. Dengan menggunakan website sebagai media promosi dan alat bantu pengelolaan dan pengolahan data pemesanan, data pembayaran dan data konfirmasi pembayaran dengan cara online. Metode yang digunakan dalam membangun aplikasi website ini menggunakan metode waterfall yang dimulai dari analisis, desain, kode, pengujian dan pemeliharaan. Perancangan Sistem Informasi Penyewaan Peralatan Event Organizer pada PT. Adecon Jakarta diharapkan bisa memberikan solusi dalam mengelola dan mengolah data pemesanan, data pembayaran dan data konfirmasi pembayaran supaya dapat memberikan kenyamanan dan kemudahan terhadap konsumen.
4. Pradana & Kusniwardi (2020), meneliti tentang Rancang Bangun Sistem Informasi Studio Rental Rekaman Gz Studio Musik Berbasis Web. GZ Studio Musik pelayanan yang bergerak di bidang jasa penyewaan studio musik. Selama ini masih menggunakan sistem booking dan penjadwalan secara manual. Berdasarkan kebutuhan tersebut, dirancang sebuah sistem informasi penyewaan pada GZ Studio Musik berupa aplikasi berbasis web dengan metode waterfall dan menggunakan model Software Development Life Cycles (SDLC). Tujuan dalam penelitian rancang bangun ini membangun sistem informasi berbasis web agar memudahkan Customer dan Pemilik GZ

Studio Musik dalam memperoleh informasi Booking studio, penjadwalan dan administrasi yang baik dan praktis. Penelitian Sistem informasi yang dibangun di GZ Studio Musik mendapat hasil Penilaian User Acceptance Test (UAT) dengan rata rata 90,35%.

5. Angelia et al., (2019), meneliti tentang Sistem Informasi Pemasaran Alat Terapi Berbasis Web pada PT. Jm Therapy Jakarta Barat. Perancangan sistem informasi pemasaran alat-alat terapi ini merupakan salah satu cara untuk memudahkan pihak PT. JM Therapy Jakarta Barat untuk menyebarkan informasi ke masyarakat luas. Website ini dapat membantu mempermudah customer dalam melakukan pemesanan alat terapi secara online. Menggunakan metode waterfall, dengan dibangunnya sistem informasi pemasaran alat terapi, diharapkan dapat mempermudah penyampaian informasi baik dari customer maupun perusahaan dapat dilakukan lebih cepat. Serta aktivitas pengelolaan data dapat terkontrol secara langsung melalui halaman Admin.
6. Rusliyawati, Oktavia and Sucipto (2021), meneliti tentang Rancang Bangun Aplikasi E-Marketplace Untuk Produk Titik Media Reklame Perusahaan Periklanan (Studi Kasus : P3I Lampung). Pada proses pemesanan reklame pelanggan masih dilakukan secara manual yaitu dicatat didalam buku selanjutnya akan direkap kedalam excel sehingga lama dalam proses perekapan dikarenakan harus melihat buku satu persatu, dan tidak adanya laporan secara periode. Pada penelitian ini dibahas bagaimana membuat dan merancang aplikasi marketplace dimulai dari metode pengumpulan data (wawancara, pengamatan dan dokumentasi) menggunakan metode pengembangan Extreme Programming, pembuatan rancangan sistem menggunakan UML dengan model perancangan Usecase Diagram, Activity Diagram, Class Diagram. Hasil pengujian ISO 25010 yang telah dilakukan dengan melibatkan 9 Responden bahwa kesimpulan kualitas

kelayakan perangkat lunak dengan skor 91.62% yang dihasilkan secara keseluruhan mempunyai skala “Sangat Baik”.

7. Isnaini and Sari (2021) meneliti tentang Perancangan Sistem Monitoring Persediaan Stok Es Krim Campina Pada PT Yunikar Jaya Sakti. Dalam aktivitas sehari-hari perusahaan ini melakukan distribusi es krim Campina ke supermarket, minimarket dan toko-toko yang ada di seluruh Lampung. Keluar masuk barang yang terjadi setiap hari dicatat ke dalam tabel posisi stok es krim berupa data excel. Perancangan Sistem Monitoring persediaan stok ES Krim Campina dengan Platform Web menggunakan bahasa pemrograman PHP, dan Framework CodeIgniter dengan toolsadobe dreamweaver sebagai editor penulisan code PHP dan HTML, MySQL sebagai database dengan tools SQLYog. Pencatatan persediaan barang pada PT Yunikar Jaya Sakti yang berjalan masih menggunakan cara menulis pada buku dan faktur-faktur manual yang diarsip per periode waktu tertentu oleh bagian admin, Stok es krim dihitung menggunakan metode FIFO yaitu barang yang masuk pertama akan keluar pertama, Pengujian perangkat lunak berjalan dengan baik sesuai standar ISO 25010 dengan persentase 88,54%

2.2. Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan (Tohari, 2017).

Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware dan software yang saling berinteraksi sebagai kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama (Maniah and Haminidin, 2017).

Dari beberapa kutipan di atas maka penulis dapat menyimpulkan bahwa sistem informasi adalah sistem di dalam suatu instansi atau organisasi perusahaan yang

mempertemukan kebutuhan pengolahan transaksi harian dan memberikan laporan-laporan atau informasi yang dibutuhkan.

2.3. Booking

Booking yaitu pemesanan fasilitas yang diantaranya akomodasi, *meal*, *seat* pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, *night club*, *discoutegue* dan sebagainya (Suartana, 2017) Sedangkan menurut (McLoad, 2014) yang dimaksud *booking* adalah proses, perbuatan, cara memesan (tempat, barang, dan sebagainya) kepada orang lain.

Berdasarkan pengertian para ahli diatas dapat disimpulkan bahwa *booking* adalah suatu aktivitas yang dilakukan oleh konsumen sebelum membeli atau melakukan *service* untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem *booking* yang baik.

2.3.1. Jenis Booking

Booking dibagi menjadi dua jenis yaitu *booking online* dan *booking offline*:

1) *Booking Online.*

Kemajuan teknologi saat ini mengakibatkan sistem *booking* juga mengalami perkembangan kearah sistem *booking online*. *Booking online* bisa diakses oleh siapapun dan dimanapun mereka berada yang memiliki akses internet.

2) *Booking Offline*

Sistem *booking* yang menggunakan pengiriman *booking* langsung ke tempat dengan media *booking* seperti *telepon*, *fax*, *e-mail*, dan *walk in*.

2.4. Multilevel Feedback Queue

Multilevel feedback queue adalah salah satu algoritma yang berdasar pada model antrian Multi Channel Single Server. Kelebihan mendasar yang dimiliki oleh multilevel feedback queue adalah kemungkinan adanya suatu proses berpindah dari satu antrian ke

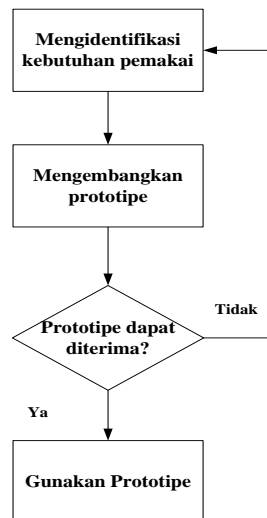
antrian lainnya, misalnya dengan prioritas yang lebih rendah ataupun lebih tinggi. Keunggulan Mendasar yang dimiliki oleh multilevel feedback queue adalah kemungkinan proses pindah dari satu antrian ke Antrian lain, misalnya dengan prioritas lebih rendah atau lebih tinggi (Verawati and Sulistiyono, 2017). Pada *Multilevel Feedback Queue* ada beberapa aturan penting:

1. Jika Prioritas (A) > Prioritas (B), maka A dijalankan (tidak b).
2. Jika algoritma Prioritas (A) = Prioritas (B), A & B berjalan di FSFC (First Come, First Serve).
3. Jika pekerjaan dimasukkan ke dalam sistem, maka pekerjaan masuk ke kategori prioritas tertinggi (antrian paling atas).
4. Jika pekerjaan membutuhkan waktu ekstra saat berjalan, prioritas diturunkan (misal, perpindahan antrian).
5. Jika pekerjaan selesai sebelum waktu, pekerjaan tetap di tingkat yang sama.

2.5. Metode Pengembangan *Prototype*

Menurut *Prototype* adalah suatu versi sistem potensial yang disediakan bagi pengembang dan calon pengguna yang dapat memberikan gambaran bagaimana kira-kira sistem tersebut akan berfungsi bila telah disusun dalam bentuk yang lengkap. Proses dalam memproduksi suatu *prototype* disebut *prototyping* (McLoad, 2014).

Empat Metode *prototyping* adalah menghasilkan *prototype* secepat mungkin, bahkan dalam satu malam dan memperoleh umpan balik dari pengguna yang memungkinkan *prototype* untuk ditingkatkan secepat mungkin. Proses ini diulang beberapa kali sehingga menghasilkan *prototype* yang dianggap sempurna.



Gambar 2. 1 Pengembangan *Prototype*

Sumber: (McLoad, 2014)

Ada empat tahapan dalam pengembangan sistem model *Evolutionary Prototype*, yaitu sebagai berikut:

1. Identifikasi kebutuhan pemakai

Pengembang mengidentifikasi terhadap pemakai untuk memperoleh suatu gagasan mengenai apa yang dibutuhkan dari sistem yang akan digunakan.

2. Mengembangkan *prototype*

Pengembang menggunakan satu atau lebih perkakas *prototyping* untuk mengembangkan satu *prototype*. Contoh perkakas *prototyping* adalah *integrated application generator* (perangkat pembuat aplikasi terintegrasi), yaitu sistem perangkat lunak *prewritten* yang mampu memproduksi semua fasilitas-fasilitas yang diharapkan ada dalam sistem baru serta dengan membuat perancangan sementara yang berfokus pada penyajian kepada pemakai (misalnya dengan membuat *input* dan format *output*).

3. Menentukan apakah *prototype* bisa diterima atau tidak

Tahap ini dilakukan oleh pemakai sistem apakah *prototyping* yang sudah dikembangkan bisa diterima atau tidak. Jika sudah sesuai maka langkah empat akan

diambil, jika tidak *Prototyping* direvisi dengan mengulangi langkah satu, dua, dan tiga dengan pemahaman yang lebih baik mengenai kebutuhan pemakai.

4. Gunakan *prototype*

Tahap ini dilakukan oleh pemakai sistem untuk menggunakan sistem yang telah dibangun.


2.6. Perancangan Sistem UML (*Unified Modeling Language*)



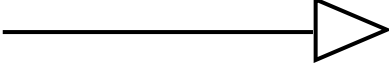

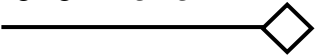
Menurut (Rosa and Shalahudin, 2018) *Unified Modeling Language (UML)* adalah bahasa standar untuk menulis perangkat lunak dalam bentuk gambar. *UML* dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sebuah sistem perangkat lunak. Beberapa jenis diagram *UML* antara lain sebagai berikut:

2.6.1. *Class Diagram*

Menurut (Rosa and Shalahudin, 2018) Unsur-unsur utama dari diagram kelas adalah kotak, yang merupakan ikon yang digunakan untuk mewakili kelas dan *interface*. Setiap kotak dibagi menjadi bagian-bagian horisontal. Bagian atas berisi nama kelas. Bagian tengah berisi daftar atribut kelas. Dan bagian bawah merupakan *operation* dari kelas tersebut menggambarkan simbol-simbol yang ada pada diagram kelas pada tabel class diagram 2.1 di bawah ini :

Tabel 2.1 Simbol *Class Diagram*


Simbol	Deskripsi
Kelas <div style="border: 1px solid black; padding: 2px; width: fit-content;"> nama_kelas +atribut +operasi() </div>	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek






Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2.6.2. Use Case Diagram

Menurut (Rosa and Shalahudin, 2018), *use case* diagram membantu anda menentukan fungsi dan fitur dari perangkat lunak. Dalam diagram ini, gambar yang menyerupai boneka kayu mewakili aktor yang berhubungan dengan kategori dari pengguna. Di dalam diagram *use case*. Para aktor terhubung oleh garis ke *use case* yang mereka kerjakan. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat menggambarkan simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel 2.2 di bawah ini :

Tabel 2.2 Simbol diagram Use Case





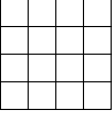


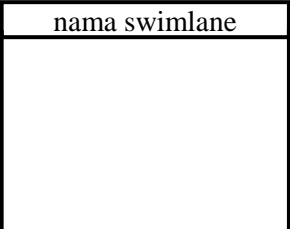
Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata

	kerja di awal frase nama <i>use case</i> .
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi/ <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.
Ekstensi/ <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan .
<< <i>extend</i> >> 	dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan.
Generalisasi/ <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.
Menggunakan/ <i>Include/uses</i> << <i>include</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

2.6.3. Activity Diagram

Menurut (Rosa and Shalahudin, 2018) sebuah diagram *activity* menggambarkan perilaku dinamis dari sistem atau bagian dari sistem melalui aliran kontrol antara tindakan yang sistem lakukan. Hal ini mirip dengan sebuah *flowchart* kecuali bahwa suatu diagram *activity* dapat menunjukkan arus bersamaan. Menggambarkan simbol-simbol yang ada pada *activity diagram* dapat dilihat pada Tabel 2.3 di bawah ini :

Tabel 2.3 Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabungan/ <i>decision</i> 	Asosiasi percabungan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.7. Pengujian ISO 25010

Model ISO-25010 merupakan bagian dari *Software product Quality Requirements and Evaluation (SQuaRE)*, yang merupakan pengembangan dari model kualitas perangkat lunak sebelumnya yaitu ISO-9126. Dalam model ISO-25010 ini digunakan untuk melihat kualitas

suatu perangkat lunak yang digunakan oleh perusahaan, instansi ataupun organisasi. Metode ISO 25010 ini dapat digunakan untuk mengevaluasi kualitas sistem perangkat lunak secara spesifik berdasarkan dua dimensi umum, yaitu dimensi *product quality*, dimana prosesnya mengacu pada karakteristik intrinsik dari sebuah produk perangkat lunak, memiliki beberapa elemen antara lain meliputi *functional suitability*, *reliability*, *operability*, *performance efficiency*, *security*, *compatibility*, *maintainability* dan *transferability*. *Quality in use* dan *product quality*. Sedangkan pada *dimensi quality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user* antara lain *Usability in use*, *Flexibility in use*, dan *Safety*. Adapun untuk mengetahui gambaran kualitas *system* aplikasi *M-Library* Gadjah Mada penulis melakukan analisis berdasarkan model ISO-25010 yang terdiri dari dua dimensi umum, yaitu dimensi *product quality* dan dimensi *quality in use* (Al-Qutaish 2010). Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

- 1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk Merupakan tingkatan dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi spesifik tertentu dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna.
- 2) *Reliability* Merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik dalam hal ini perangkat lunak dapat beroperasi dan siap ketika dibutuhkan untuk digunakan dan juga dapat bertahan pada tingkat kemampuan tertentu terhadap kegagalan, kesalahan serta perangkat lunak kembali pada tingkat tertentu dalam mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak.

- 3) *Performance efficiency* Merupakan tingkatan dimana perangkat lunak dapat memberikan kinerja terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu dalam hal ini *performance efficiency* dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi dan perangkat lunak dapat menggunakan sejumlah sumber daya ketika melakukan aksi dari sebuah fungsi.
- 4) *Operability* Perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik pengguna ketika digunakan dalam hal ini perangkat lunak mudah dipelajari oleh pengguna, perangkat lunak dapat digunakan dan dioperasikan oleh pengguna, perangkat lunak dapat memberikan bantuan ketika pengguna membutuhkan panduan, perangkat lunak dapat menarik perhatian pengguna, perangkat lunak memenuhi kebutuhan pengguna yang memiliki keterbatasan dan perangkat memungkinkan untuk dianalisis oleh pengguna apakah perangkat lunak sudah memenuhi kebutuhan mereka.
- 5) *Security* Merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman atau keganjalan dalam hal ini perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna dan merupakan dari kelengkapan, ketepatan dari sejumlah *asset* yang telah dijaga sehingga aksi atau tindakan yang dilakukan telah terbukti dan hal tersebut tidak dapat ditolak.
- 6) *Compability* Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.
- 7) *Maintainability* Merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian

perangkat lunak untuk dapat berubah pada lingkungan , kebutuhan dan fungsionalitas yang spesifik. Selain itu perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak untuk mengidentifikasi bagian yang dapat dimodifikasi.

- 8) *Transferability*. Merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan satu ke lingkungan yang lain dalam hal ini perangkat lunak dapat beradaptasi dengan cepat pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.