

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka yang serupa dan relevan dengan penelitian bertujuan untuk membuktikan keaslian penelitian, untuk tinjauan pustaka dapat dilihat pada tabel 2.1 dibawah ini:

Tabel 2.1 Penelitian Terdahulu

1.	Judul	Perancangan Sistem Informasi E-Booking Paket Pernikahan Pada Masa Pandemi Covid-19 Berbasis Web (Studi Kasus: Galih Wedding Organizer)
	Penulis	Mulyana Susanti, Deffy
	Tahun	2021
	Metode	RUP (Rational Unified Process)
	Hasil Penelitian	Adanya sistem informasi yang diimplementasikan customer yang ingin membooking hanya perlu membuka halaman website galih wedding organizer dan memilih item yang disukai dan memilih tanggal yang diinginkan serta memasukan data yang diperlukan.
2.	Judul	Pengembangan Sistem Informasi Wedding Organizer Menggunakan Pendekatan Sistem Berorientasi Objek Pada CV Pesta
	Penulis	Aman, Mustar Suroso
	Tahun	2021
	Metode	Kualitatif Deskriptif
	Hasil Penelitian	Hasil penelitian ini diketahui Penerapan Fungsi Planning Wedding Organizer pada CV Pesta cukup baik pelaksanaanya dan penerapan planning ini mulai dari membuat rincian paket harga sesuai dengan budget klien.
3.	Judul	Perancangan Sistem Informasi Manajemen Wedding Organizer Online Menggunakan Scrum
	Penulis	Wulandari, Rosita Setiawan, Ridwan

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

		Mulyani, Asri
	Tahun	2020
	Metode	Agile
	Hasil Penelitian	Hasil dari penelitian ini menjadi sebuah alternatif bagi penyedia jasa pernikahan untuk meningkatkan pemasaran dalam hal produk dan jasanya. Penggunaan sistem informasi ini dapat memudahkan pengguna untuk mengetahui informasi mengenai wedding organizer dan dapat membantu proses transaksi pemesanan wedding organizer.
4	Judul	Analisa Perancangan Sistem Informasi Pemesanan Pada Wedding Organizer Berbasis Web (Studi Kasus: JFS Wedding Organizer Cakung)
	Penulis	Lusti, Hanna Masya, Fajar
	Tahun	2020
	Metode	RAD (Rapid Application Development)
	Hasil Penelitian	Website Wedding Planner Berbasis Web ini dapat memberikan kemudahan dalam pemesanan paket secara online, baik paket yang telah ditentukan maupun pilihan secara manual dan otomatis. Dengan adanya fitur-fitur serta konten yang membantu sharing informasi antara calon pengantin dengan pihak Wedding Organizer.
5	Judul	Rancang Bangun Sistem Informasi Penyewaan Wedding Organizer Berbasis Web Dengan Php Dan Mysql Pada Kiki Rias
	Penulis	Mulyati, Sri Hisyam, Miftahur
	Tahun	2019
	Metode	Waterfall
	Hasil Penelitian	Sistem informasis berbasis web yang diterapkan pada Kiki Rias Wedding Organizer ini dapat lebih efektif dan efisien karna dapat membantu admin mengelola data pelanggan wedding organizer menggunakan komputer, dan juga membantu pemesanan pelanggan yang sesuai diinginkan.

Berdasarkan lima *literature* yang telah dipaparkan, maka penelitian ini memiliki perbedaan dari penelitian-penelitian terdahulu, diantaranya:

1. Pada penelitian sebelumnya dalam proses implementasi sistem belum menerapkan *responsive system* pada desain tampilan antar muka sehingga tidak dapat menyesuaikan spesifikasi layar perangkat keras yang digunakan seperti *Personal Computer (PC)*, Laptop dan *Handpone*.
2. Pada penelitian sebelumnya belum ada yang menggunakan metode penelitian *Extreme Programming (XP)* dalam proses pengembangan sistemnya.

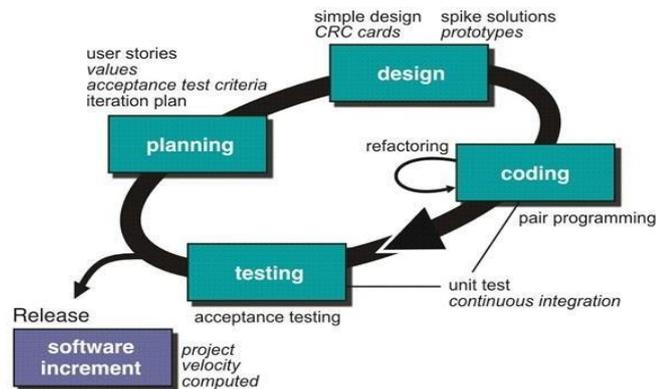
2.2 Aplikasi E-Booking

Aplikasi *E-Booking* adalah aplikasi pemesanan secara elektronik yang digunakan dalam proses pemesanan suatu jasa atau produk memanfaatkan jasa internet dalam pengaplikasiannya. Aplikasi ini dapat digunakan pada perusahaan jasa penyewaan dan penjualan produk. Penggunaan aplikasi *E-Booking*, pelanggan cukup membuka situs web kemudian memesan layanan sesuai dengan tanggal yang mereka inginkan, mengisi formulir, dan melakukan *online payment*. Sistem juga dapat memperbarui data ataupun informasi secara real time sehingga pelanggan tidak perlu menghubungi secara manual untuk menanyakan ketersediaan layanan untuk tanggal yang mereka pesan (Kurnia, 2020).

2.3 Metode Pengembangan Sistem

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria

pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah *testing* (Pressman, 2012).



Gambar 2.1 Model *Extreme Programming* (XP) (Pressman, 2012)

Dibawah ini adalah penjelasan tahapan *Extreme Programming* yaitu :

1. *Planning* (Perencanaan)

Kegiatan Perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan-kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan *fungsi*alitas. Pada perencanaan terdapat *user stories values* yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, *acceptance test criteria iteration plan* melakukan perhitungan kecepatan project selama *development*, *customer* dapat menambah *story*, merubah *value*, membagi *story* atau menghapusnya.

2. *Design* (Perancangan)

Perancangan yang simple, menarik, dan sederhana selalu memberikan hasil yang lebih disukai daripada gambaran-gambaran yang lebih

kompleks. Perancangan XP memberikan panduan implementasi untuk suatu cerita ketika ditulis, tidak kurang, tidak lebih. Terdapat *simple design* *CRC Cards* untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

3. *Coding* (Pengkodean)

Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke pengkodean. Pada tahapan *pair programming* melakukan kerja sama untuk membuat kode dari satu story. Dan *refactoring* adalah proses restrukturisasi kode program komputer yang ada tanpa mengubah perilaku eksternalnya.

4. *Testing* (Pengujian)

Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali. Pada tahapan pengujian yaitu *unit test* *continuous integration* yaitu tahapan pengujian code yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh customer dan focus pada keseluruhan dan fungsional sistem, dan *acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari

software realease. Selanjutnya terdapat tahapan *software increment project velocity computed* yaitu tahapan yang telah diimplementasikan dari *software realease* yang nantinya akan diterapkan dalam suatu sistem

2.4 *Unified Modeling Language (UML)*

Perancangan sistem dalam penelitian ini menggunakan *Unified Modeling Language (UML)*. *UML* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S dan Shalahudin, 2018). Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *UML (Unified Modelling Language)*.

2.4.1 *Use Case Diagram*

A.S dan Shalahudin (2018) *Use Case Diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara salah satu lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol pada *use case diagram* disajikan pada tabel 2.2 dibawah ini:

Tabel 2.2 Simbol *Use Case Diagram* (A.S dan Shalahudin, 2018)

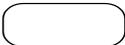
No	Simbol	Deskripsi
1	<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
2	Aktor / <i>actor</i>	Orang, proses atau sistem lain yang

No	Simbol	Deskripsi
	 <p style="text-align: center;">Actor</p>	<p>berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.</p>
3	<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
4	<p>Ekstensi / <i>extend</i></p> <p style="text-align: center;"><<extend>> -----></p>	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.</p>
5	<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
6	<p><i>Include</i></p> <p style="text-align: center;"><<include>> -----></p>	<p>Relasi <i>use case</i> tambahan ke <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini menjalankan fungsionalnya atau sebagai syarat dijalankan <i>use case</i> ini.</p>

2.4.2 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor jadi aktivitas yang dapat dilakukan oleh sistem (A.S & Shalahudin, 2018). Simbol-simbol yang terdapat dalam *Activity Diagram* disajikan pada tabel 2.3 dibawah ini:

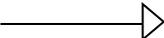
Tabel 2.3 Simbol Activity Diagram (A.S dan Shalahudin, 2018)

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Status akhir yang dilakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.4.3 Class Diagram

Class diagram adalah sebuah *class* yang menggambarkan struktur sistem dari segi pendefinisian kelas kelas yang akan dibuat untuk membangun sistem (A.S & Shalahudin, 2018). *Class diagram* juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan. Simbol-simbol yang terdapat dalam *Activity Diagram* disajikan pada tabel 2.4 dibawah ini:

Tabel 2.4 Simbol Class Diagram (A.S dan Shalahudin, 2018)

No	Simbol	Deskripsi
1	Kelas / <i>class</i> 	Kelas pada struktur sistem.
2	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisai (umum-khusus).

No	Simbol	Deskripsi
6	Agresi / <i>aggregation</i> 	Relasi antar kelas dengan maksa semua bagian (<i>whole-part</i>).

2.5 Wedding Organizer

Wedding Organizer adalah suatu jasa khusus yang secara pribadi membantu calon pengantin dan keluarga dalam perencanaan dan pelaksanaan rangkaian acara pesta pernikahan sesuai dengan jadwal yang telah ditetapkan. *Wedding Organizer* membantu memberikan informasi mengenai berbagai macam yang berhubungan dengan acara pernikahan yaitu mengelola *event* pernikahan mulai dari akad nikah, upacara adat, dekorasi pelaminan, katering, rias pengantin, dokumentasi dan lain sebagainya (Mulyati & Hisyam, 2019).

2.6 E-Booking

Pemesanan adalah perjanjian pemesanan antara dua orang atau lebih proses pemesanan terjadi untuk kebutuhan suatu barang atau jasa secara langsung yang akan digunakan untuk kepentingan perorangan atau kepentingan perusahaan. Sedangkan pemesanan yang dilakukan secara online menggunakan internet dengan mengakses laman web yang disediakan oleh perusahaan disebut juga *E-Booking* (Aman & Suroso, 2021).

2.7 Website

Website adalah kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari

semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan- jaringan halaman (*hyperlink*) (Ismatullah dan Adrian, 2021).

2.8 MYSQL

MySQL merupakan basis data yang bersifat open source sehingga banyak di gunakan untuk media. Walaupun gratis, *MySQL* tetap berkualitas dan sudah cukup memberikan *performance* yang memadai. Penggunaan *PHP MyAdmin* lebih mudah digunakan karena menggunakan *interface* yang lebih mudah dipahami. *MySQL* adalah nama sebuah *database server* yang menangani akses *database* yang selalu dalam bentuk pernyataan *SQL (Structured Query Language)* yaitu suatu bahasa yang digunakan untuk mengakses *database relasional* (Ardian & Fernando, 2020).

2.9 XAMPP

XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dengan menggunakan XAMPP kita tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah disediakan oleh XAMPP. Beberapa paket yang telah disediakan adalah *Apache*, *MySql*, *Php*, *Filezila* dan *Phpmyadmin* (Putera dan Ibrahim, 2018).

2.10 PHP

PHP singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. PHP merupakan

Software Open-Source yang disebar dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya. PHP ditulis menggunakan bahasa C (Hutagalung dan Arif, 2018).

2.11 CPanel

Cpanel adalah salah satu *control panel* berbasis *linux* yang paling banyak digunakan di akun *web hosting*. Melalui panel ini, bisa dengan mudah mengelola semua layanan dalam satu tempat. Saat ini, *cpanel* merupakan panel standar yang paling banyak digunakan dan sebagian besar *web developer* sudah sangat familiar dengan tool ini. Selain intuitif dan mudah digunakan, *cpanel* juga memungkinkan untuk mengelola akun *web hosting* dengan maksimal. Membuat *FTP user* dan alamat *email* baru, memonitor *resource*, membuat *subdomain*, dan *install software* merupakan beberapa kegunaan *cpanel* (Lidar, 2020).

2.12 Framework Codeigniter

Codeigniter adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi *web*. Tujuan utama pengembangan *codeigniter* adalah untuk membantu *developer* dalam pengerjaan aplikasi yang lebih cepat daripada menulis semua *code* dari awal dan *codeigniter* merupakan salah satu *framework* PHP tercepat yang ada saat ini (Romadhon dan Desmulyati, 2019).

Codeigniter memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi *web* secara mudah dan cepat. *Codeigniter* mengizinkan para pengembang untuk menggunakan *framework* secara parsial maupun secara keseluruhan. Adapun keunggulan dari *codeigniter* menurut Romadhon dan Desmulyati (2019) adalah sebagai berikut:

1. *Codeigniter* adalah *framework* yang bersifat *free* dan *open-source*.
2. *Codeigniter* memiliki ukuran kecil dibandingkan dengan *framework* lain.
3. Aplikasi yang dibuat menggunakan *Codeigniter* bisa berjalan cepat.
4. *Codeigniter* menggunakan pola design *Model-View-Controller* (MVC) sehingga satu file tidak terlalu berisi banyak kode.
5. *Codeigniter* dapat diperluas sesuai dengan kebutuhan .
6. *Codeigniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *Codeigniter* dapat diproses melalui dokumentasi yang disertai di dalam paket distribusinya.

2.13 Pengujian ISO 25010

Standar *ISO/IEC 25010* pertama kali diperkenalkan pada tahun 1991 melalui pertanyaan tentang definisi kualitas perangkat lunak. Dokumen standard *ISO/IEC 25010* sangat panjang. Hal ini dikarenakan orang memiliki motivasi berbeda yang memungkinkan untuk tertarik pada kualitas perangkat lunak. Secara keseluruhan, *ISO/IEC 25010* memiliki 6 karakteristik untuk mengukur kualitas perangkat lunak secara menyeluruh, antara lain *functional suitability*, *reliability*, *performance efficiency*, *usability*, *maintainability*, dan *portability* (Estdale dan Georgiadou, 2018).

1. *Functional Suitability*

Pengujian *functional suitability* merupakan tingkat dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi yang spesifik.

2. *Reliability*

Pengujian *reliability* ini merupakan kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.

3. *Performance efficiency*

Pengujian *performance* ini dilakukan untuk mengukur karakteristik performa dari komponen aplikasi. Aspek ini merupakan aspek untuk mengukur keandalan sistem informasi yang digunakan pengembangan aplikasi sistem informasi. yang sama.

4. *Usability*

Pengujian *usability* ini merupakan kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.

5. *Maintainability*

Pengujian *maintainability* dilakukan untuk menguji efektifitas dan efisiensi perangkat lunak untuk dimodifikasi atau dikembangkan.

6. *Portability*

Pengujian *portability* ini merupakan kemampuan perangkat lunak untuk di transfer dari satu lingkungan ke lingkungan lain.



Gambar 2.2 Model Pengujian Perangkat Lunak ISO 25010 (Estdale dan Georgiadou, 2018).

Masing-masing karakteristik model *ISO 25010* dibagi menjadi beberapa sub-karakteristik kualitas. Tabel karakteristik model *ISO 25010* dapat dilihat pada tabel 2.5:

Tabel 2.5 Karakteristik *ISO 25010* (Estdale dan Georgiadou, 2018).

Karakteristik	Indikator	Deskripsi
<i>Functional Suitability</i>	<i>Functional Completeness</i>	Kemampuan perangkat lunak dalam menyediakan fungsi dapat mencakup semua tugas dan tujuan pengguna secara spesifik.
	<i>Functional Correctness</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem mampu menyediakan hasil yang tepat sesuai dengan kebutuhan.
	<i>Functional Appropriateness</i>	Kemampuan perangkat lunak dalam fungsi yang disediakan mampu menyelesaikan tugas dan tujuan secara spesifik.
<i>Performance Efficiency</i>	<i>Resource Utilization</i>	Kemampuan perangkat lunak sejauh mana jumlah dan jenis sumber daya yang digunakan produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria

Karakteristik	Indikator	Deskripsi
	<i>Capacity</i>	Kemampuan perangkat lunak sejauh mana batas maksimal sebuah produk atau sistem mampu memenuhi kriteria.
	<i>Time Behaviour</i>	Kemampuan perangkat lunak sejauh mana respon dan lama proses sebuah produk atau sistem saat menjalankan fungsinya sesuai dengan kriteria.
<i>Usability</i>	<i>Appropriateness</i> <i>Recognizability</i>	Kemampuan perangkat lunak sejauh mana pengguna mengetahui sebuah produk atau sistem sesuai dengan kebutuhan mereka.
	<i>Learnability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan pengguna tertentu untuk mencapai tujuan mempelajari sebuah produk atau sistem secara efektif, efisien dan bebas dari resiko serta memenuhi kepuasan dalam konteks penggunaan.
	<i>Operability</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem

Karakteristik	Indikator	Deskripsi
		sistem mampu memenuhi kriteria dioperasikan dan dikontrol.
	<i>User Error Protection</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem melindungi pengguna dalam melakukan kesalahan.
	<i>User Interface Aesthetics</i>	Kemampuan perangkat lunak sejauh mana tampilan antarmuka memenuhi kesenangan dan kepuasan pengguna.
	<i>Accessibility</i>	Kemampuan perangkat lunak sejauh mana produk atau sistem dapat digunakan oleh pengguna secara luas untuk mencapai tujuan tertentu dalam konteks penggunaan
<i>Reliability</i>	<i>Maturity</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen memenuhi kriteria reliabilitas dibawah kondisi normal.
	<i>Availability</i>	Kemampuan perangkat lunak sejauh mana sistem, produk, atau komponen dapat beroperasi ketika diperlukan untuk digunakan.

Karakteristik	Indikator	Deskripsi
	<i>Fault Tolerance</i>	Kemampuan perangkat lunak sejauh mana sistem, produk atau komponen.
	<i>Recoverability</i>	Kemampuan perangkat lunak ketika produk atau sistem mengalami kejadian atau kegagalan dapat mengembalikan data dan menjalankan kondisi sistem yang diharapkan.
<i>Maintainability</i>	<i>Modularity</i>	Kemampuan perangkat lunak dimana sistem atau program terdiri dari komponen yang berlainan sehingga perubahan terhadap satu komponen minimal memiliki pengaruh terhadap komponen lain
	<i>Reusability</i>	Kemampuan perangkat lunak dimana sebuah asset dapat digunakan pada lebih dari satu sistem perangkat lunak atau pada pembangunan asset lainnya.
	<i>Analyzability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat dianalisis untuk mengetahui apa

Karakteristik	Indikator	Deskripsi
		yang menyebabkan kegagalan pada perangkat lunak atau untuk mengidentifikasi bagian yang dapat dimodifikasi.
	<i>Modifiability</i>	Kemampuan perangkat lunak dimana perangkat lunak dapat menghindari efek yang tidak diharapkan dari modifikasi yang dilakukan terhadap perangkat lunak.
	<i>Testability</i>	Kemampuan perangkat lunak dimana perangkat lunak memungkinkan modifikasi perangkat lunak untuk dilakukan validasi.
<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak dapat beradaptasi dengan perubahan lingkungan atau sistem yang berbeda.
	<i>Installability</i>	Kemampuan perangkat lunak dapat digunakan dalam lingkungan atau sistem tertentu.

Karakteristik	Indikator	Deskripsi
	<i>Replaceability</i>	Kemampuan perangkat lunak dapat menggantikan perangkat lunak lain apakah ada kebergantungan kepada perangkat lunak lain digunakan.