

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Beberapa penelitian yang pernah dilakukan sehubungan dengan sistem pemantauan bunker minyak adalah sebagai berikut :

Tabel 2.1 Tinjauan Pustaka

No	Judul Penelitian	Nama Tahun	Metode	Permasalahan	Solusi
1	Perancangan Model Dashboard Untuk Monitoring Evaluasi Mahasiswa	(Rahmayudha, 2017)	model wireframe	Masih perlu pengembangan dalam menentukan metode dashboard operasional untuk menghasilkan performance KPI yang baik, sehingga pengambilan data dapat dilakukan secara real times.	pengolahan data dan analisis berupa dashboard monitoring mahasiswa dan indeks performance KPI sistem informasi dan tingkat kualitas mahasiswa serta jurusan sistem informasi FMIPA Universitas Tanjungpura Pontianak, disajikan dalam bentuk grafik, tabel dan dashboard
2	Model Sistem Monitoring Tangki Bahan Bakar Minyak Spbu Dengan Menggunakan Web Aplikasi Dan Sms Gateway	(Siregar & Raymond, 2015)	Metode kepustakaan Dan metode whitebox.	Pengukuran tangki bahan bakar minyak pada SPBU saat ini masih menggunakan metode mekanis dipstick atau pengukuran secara langsung secara manual, apalagi jika keadaan malam hari dengan kondisi penerangan sekitar kurang memadai yang	sistem yang dapat digunakan untuk memonitor volume bahan bakar minyak pada SPBU berbasis web aplikasi, dan terintegrasi dengan sms gateway untuk

				kadang meyebabkan kesulitan dan ketidakakuratan pengukuran.	memberikan informasi kondisi bahan bakar minyak pada tangki BBM apabila BBM hampir mencapai level terendah
3	Rancang Bangun Sistem Monitoring Penggunaan Bahan Bakar Minyak (Bbm) Dan Temperatur Pada Generator Menggunakan Sms Berbasis Pengendali Mikro	(Kurniawan et al., 2015)	Kualitatif	Masih dilakukan secara manual	rancang bangun sistem ini dapat membantu operator dalam memonitor kerja dari generator tanpa harus berada langsung ditempat generator berada.
4	Sistem Informasi Monitoring Kereta Api Berbasis Web Server Menggunakan Layanan Gprs	(Novianta & Setyaningsih, 2015)	Metode development.	Permasalahan perkeretaapian Indonesia saat ini memang diakui sangat rumit, sulit, dinamis dengan tantangan yang terus berkembang. Sejak dulu hingga kini kereta api selalu berhadapan dengan tantangan kompetisi yang tinggi dari moda lain.	rancangan web base server sebagai sistem pemantauan secara on-line yang memanfaatkan jaringan GSM menggunakan layanan GPRS.

5	Sistem Informasi Monitoring Pengembangan Software Pada Tahap Development Berbasis Web	(Herliana & Rasyid, 2016)	metode pemodelan terstruktur	kurang efektifnya cara pemberian tugas dan permasalahan yang tidak tersimpan dengan baik dapat menghambat lancarnya pengembangan perangkat lunak.	membantu sistem analis akan kebutuhan yang mendetail tentang proyek yang sedang dikerjakan dan meningkatkan kinerja programmer dalam mengembangkan perangkat lunak
---	---	---------------------------	------------------------------	---	--

Berdasarkan penelitian yang telah dilakukan sebelumnya terdapat beberapa perbedaan peneliti sebagai berikut :

1. Membangun sistem bunker minyak kapal.
2. Sistem yang dibangun dapat menghitung secara otomatis sisa minyak dan pengisian minyak.
3. Metode pengembangan sistem menggunakan *prototype*
4. Dapat mencetak laporan bunker minyak secara periode

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 & 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. Informasi

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi (Kristanto, 2018).

Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang (Al-Fatta, 2012).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Sutabri, 2018)

Kualitas dari suatu informasi tergantung dari 3 (tiga) hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timelines*), dan relevan (*relevance*) (Sutabri, 2018).

1) Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan.

2) Tepat pada waktunya (*timelines*)

Informasi yang datang pada penerima tidak boleh terlambat

3) Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya.

2.4. Sistem Informasi

Menurut Kristanto (2018), Sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.

Menurut Sutabri (2012), Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Berdasarkan definisi diatas disimpulkan bahwa sistem informasi adalah kumpulan prosedur yang dibuat manusia yang terdiri dari komponen-komponen yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dan kegiatan strategi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.5. Monitoring

Menurut (Taufeni & Pemerintah, 2013), Peran *monitoring* adalah untuk mengenali dan mengevaluasi perkembangan yang terjadi akibat tindakan yaitu mengenali apakah pelaksanaan tindakan sesuai dengan rencana tindakan dan apakah telah terjadi peningkatan dengan adanya tindakan.

Monitoring dapat diartikan yaitu mengumpulkan secara teratur, mengkaji dan bertindak atas informasi tentang pelaksanaan suatu proyek atau kegiatan, yang pada umumnya digunakan untuk memeriksa kinerja terhadap target serta memastikan kepatuhan terhadap peraturan (*Universitas Hasanuddin* 2011).

Berdasarkan definisi diatas disimpulkan bahwa *monitoring* adalah kegiatan yang dilakukan dengan mengecek kegiatan dan menghasilkan informasi tentang kinerja pelaksanaan suatu proyek atau kegiatan dengan melihat apakah telah terjadi peningkatan dengan adanya tindakan serta memastikan kepatuhan terhadap peraturan.

2.6. Bunker Minyak

Pengisian bahan bakar merupakan salah satu proses di kapal yang telah menjadi alasan untuk beberapa kecelakaan di masa lalu. Pengisian bahan bakar di kapal dapat berupa minyak diesel, kargo dll. Pengisian bahan bakar atau minyak diesel membutuhkan kehati-hatian dan kewaspadaan untuk mencegah segala jenis kecelakaan kebakaran dan tumpahan minyak yang mungkin terjadi di atas kapal yang dilakukan pengisian oleh pihak KKM (Sholihah et al., 2020).

Berikut ini adalah contoh kegiatan bunker minyak, dapat dilihat pada Gambar 2.1

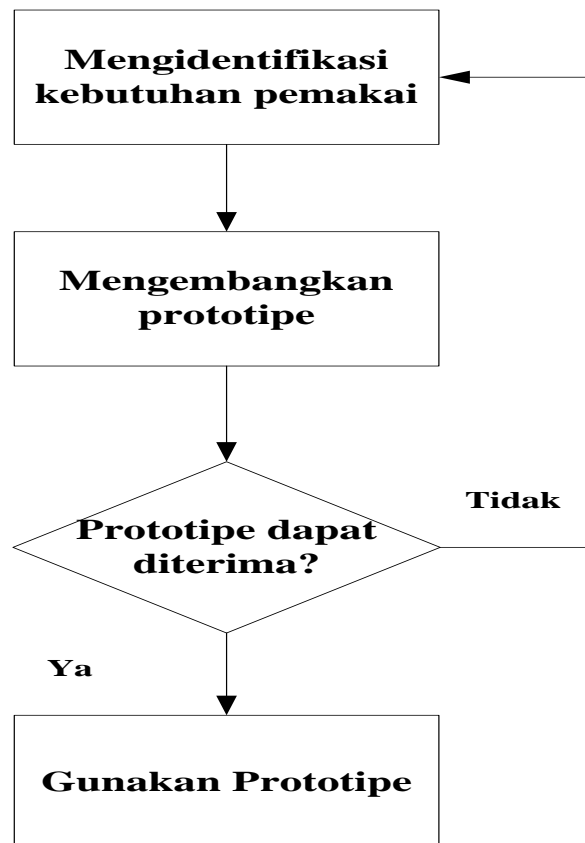


Gambar 2.1 Kegiatan Bunker Minyak

2.7. Pengembangan Sistem *Prototype*

Menurut (McLeod, 2014), *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*.

Tujuan 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: McLeod (2014)

Menurut McLeod (2014) Ada 4 tahapan dalam pengembangan sistem model

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. Menggunakan sistem operasional

Tahapan ini dilakukan setelah semua tahapan selesai dan tidak ada kendala apapun dalam pembuatan sistem.

2.7.1 UML (*Unified Modeling Language*)


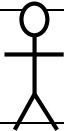
Menurut (Rosa & Shalahudin, 2018) *Unified Modeling Language (UML)* adalah bahasa standar untuk menulis perangkat lunak dalam bentuk gambar. *UML* dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sebuah sistem perangkat lunak. Beberapa jenis diagram *UML* antara lain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





2.7.2 Use Case Diagram

Menurut (Rosa & 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. simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 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

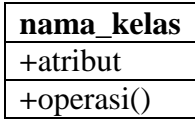


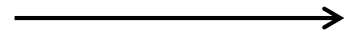
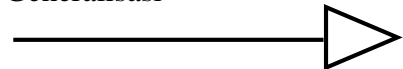

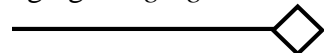
	walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
Ekstensi/ <i>extend</i>	Relasi <i>use case</i> 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

Sumber: (Rosa & Shalahudin, 2018)

2.7.3 Class Diagram

Menurut (Rosa & 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 horizontal. Bagian atas berisi nama kelas. Bagian tengah berisi daftar atribut kelas. Dan bagian bawah merupakan *operation* dari kelas tersebut. simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.3.

Tabel 2.3 Simbol *Class Diagram*





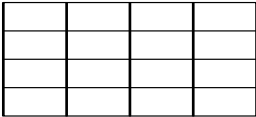


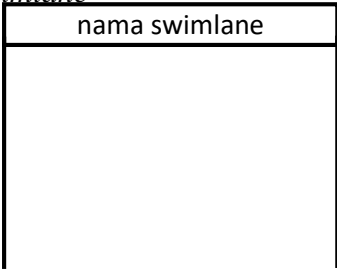
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed 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>)

Sumber: (Rosa & Shalahudin, 2018)

2.7.4 *Activity Diagram*

Menurut (Rosa & 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. Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2.4 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.
Percabangan/ <i>decision</i> 	Asosiasi percabangan 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

2.9 Pengertian SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada *Relation DBMS (Database Management System)* (Rosa & Shalahudin, 2018).

Singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan kinerja basis data (Nugroho, 2015)

SQL adalah perangkat lunak *relation database management system (RDBMS)* yang didesain untuk melakukan proses manipulasi database berukuran besar dengan berbagai fasilitas.

Jadi *Structure Query Language* adalah perangkat lunak *relation database management system (RDBMS)* mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan kinerja basis data.

2.10 Pengujian ISO 9126

Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software. Standar ISO 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. Menurut (Abran et al., 2018) ISO 9126 adalah standar internasional yang diterbitkan oleh ISO untuk evaluasi kualitas

perangkat lunak dan merupakan pengembangan dari ISO 9001. Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut.

- 1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
- 2) *Reliability* (Kehandalan). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
- 3) *Usability* (Kebergunaan). Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
- 4) *Efficiency* (Efisiensi). Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
- 5) *Maintainability* (Pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.
- 6) *Portability* (Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.

Masing-masing karakteristik kualitas perangkat lunak model ISO 9126 dibagi menjadi beberapa sub-karakteristik kualitas. Berikut adalah tabel karakteristik Kualitas Perangkat Lunak Model ISO 9126 :

Tabel 2.6 Karakteristik ISO 9126

Karakteristik	Sub Karakteristik	Deskripsi
<i>Functionality</i>	<i>Suitability</i>	Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.
	<i>Accuracy</i>	Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.
	<i>Security</i>	Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup (<i>hacker</i>) maupun otorisasi dalam modifikasi data.
	<i>Interoperability</i>	Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.
	<i>Compliance</i>	Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.
<i>Reliability</i>	<i>Maturity</i>	Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak.

	<i>Fault tolerance</i>	Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak
	<i>Recoverability</i>	Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan.
<i>Usability</i>	<i>Understandibility</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipahami.
	<i>Learnability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipelajari.
	<i>Operability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan.
	<i>Attractiveness</i>	Kemampuan perangkat lunak dalam menarik pengguna.
<i>Efficiency</i>	<i>Time behavior</i>	Kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya.
	<i>Resource behavior</i>	Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan.

<i>Maintainability</i>	<i>Analyzability</i>	Kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan.
	<i>Changeability</i>	Kemampuan perangkat lunak untuk dimodifikasi tertentu.
	<i>Stability</i>	Kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak.
	<i>Testability</i>	Kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain.
<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda.
	<i>Instalability</i>	Kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda.
	<i>Coexistence</i>	Kemampuan perangkat lunak untuk berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya.
	<i>Replaceability</i>	Kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya.

Sumber: (Al-Qutaish 2010, 172-173)

Adapun alasan penggunaan ISO 9126 karena ISO sudah berstandar *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. Kualitas produk perangkat lunak ISO 9126 memiliki enam karakteristik pendukung yang dapat digunakan sebagai acuan dalam menilai maupun memberikan masukan terhadap kualitas perangkat lunak yang akan dibangun yang akan menghasilkan nilai uji yang terukur. Indikator yang digunakan dalam pengujian ISO 9126 dilihat dari sisi *Functionality*, *Usability*, dan *Efficiency*.