

BAB II TINJAUAN PUSTAKA

1.1 Tinjauan Studi

Tinjauan pustaka digunakan sebagai acuan terhadap penelitian yang akan dikembangkan dan dapat dilihat pada Tabel 2.1:

Tabel 1.1 Tinjauan Studi

1.	Judul	Rancang Bangun Sistem Monitoring Kegiatan Kelompok Tani Berbasis <i>WEB</i> Di Kecamatan PATTALLASSANG Kabupaten GOWA
	Penulis	ST.Hajar
	Tahun	2017
	Permasalahan	Masalah bagi kelompok – kelompok tani yang ada di pedesaan khususnya di kecamatan Pattallassang adalah mengenai Transparasi pengelolaan dana atau bantuan dan pembagian kepada kelompok tani
	Metode Penelitian	<i>Waterfall</i>
	Hasil Penelitian	Tujuan peneliti sudah tercapapai, yaitu membangun system Monitoring kegiatan Kelompok Tani Berbasi <i>Web</i> di Kecamatan Pattallassang Kabupaten Gowa
	Saran	Sistem penjadwalan masih di input manual jadi diharapkan sistem penjadwalanya bisa otomatis
2.	Judul	Rancang Bangun Sistem Informasi Manajemen Pemberdayaan Kelompok Tani Pada Dinas Pertanian Kabupaten BARRU
	Penulis	Nurzakina Sopyan
	Tahun	2017
	Permasalahan	Masalah pemberdayaan kelompok tani di Dinas Pertanian Kabupaten BARRU masih kurang efektif dan efisien dimana kurangnya kepatuhan kelompok tani.
	Metode Penelitian	<i>Waterfall</i>
	Hasil Penelitian	Dengan adanya sistem pemberdayaan kelompok tani ini, membantu dan memudahkan kelompok tani dalam permintaan bantuan dan keluhan
	Saran	Sebaiknya system ini ditambahkan fitur laporan penjadwalan kegiatan

3.	Judul	Rancang Bangun Sistem Informasi Potensi Pertanian Tanam Pangan Di Kabupaten JOMBANG
	Penulis	Mayska Fidya Santoso, Bambang Hariadi
	Tahun	2012
	Permasalahan	Masalah yang terjadi dalam Sub Terminal Agrabisnis adalah tidak adanya kepastian tentang ketersediaan komoditas pangan, sehingga diperlukan suatu sistem informasi potensi pertanian tanaman pangan di kabupaten Jombang yang dapat memberikan informasi secara terperinci
	Metode Penelitian	Metode Diagram alir data
	Hasil Penelitian	User dapat mengetahui jumlah stok komoditas pertanian di Kecamatan Megaluh dengan cepat
	Saran	Dapat dikembangkan lebih lanjut dengan melakukan pengembangan system.
4.	Judul	Perancangan <i>Website</i> Pertanian Di Ploso Gede Ngluwar Magelang sebagai Media Informasi dan Pemasaran (Studi Kasus pada Kelompok Tani Hidayah)
	Penulis	Pudji Widodo, Imam Ikhsanudin
	Tahun	2018
	Permasalahan	Sulitnya mendapatkan pupuk, serta kurang informasi mengenai harga – harga hasil pertanian dipasaran.
	Metode Penelitian	<i>Waterfall</i>
	Hasil Penelitian	<i>Website</i> ini dapat menjadi media informasi resmi Kelompok Tani Hidayah untuk mempermudah memberikan informasi kepada publik
5.	Judul	Sistem Pengolahan Data Kelompok Tani Kecamatan Sungai Tabuk Kabupaten Banjar
	Penulis	Mayang Sari, M.Kom.
	Tahun	2017
	Permasalahan	Sulitnya memberikan informasi ketersediaan stok pupuk kepada kelompok tani serta informasi penyaluran pupuk kepada anggota kelompok tani dengan cepat
	Metode Penelitian	<i>Action Research</i>
	Hasil Penelitian	Kios warga tani dapat memajemen data secara lebih praktis, lebih terorganisir, dan terkomputerisasi.

Saran	Hendaknya aplikasi manajemen data kelompok tani ini dapat dilengkapi dengan system penjualanya.
-------	-------------------------------------------------------------------------------------------------

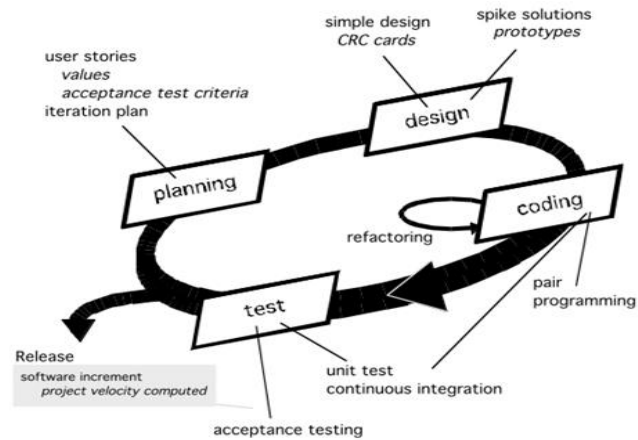
Berdasarkan kelima jurnal tersebut peneliti mengacu pada jurnal dengan Penerapan metode *extreme programming* sebagai arah dalam penelitian yang akan dibangun, adapun perbedaan antara kelima jurnal dengan peneliti yaitu:

1. Metode pengembangan yang diterapkan berupa *Extreme Programming*
2. Memiliki dua sisi perbedaan antara pengguna perangkat *mobile android* dan *admin* menggunakan *web admin dashboard* untuk memanajemen data informasi.
3. Penambahan fitur informasi edukasi mengenai cara penggunaan produk secara detail dan penggunaan yang tepat.
4. Memanajemen pengolahan data serta dokumen seperti koperasi, bantuan peralatan pertanian dan informasi mengenai kelompok tani secara sistematis

1.2 *Extreme Programming*

Menurut Pressman (2012) *extreme programming* adalah pendekatan *extreme programming* merupakan suatu pendekatan berorientasi objek dan sebagai pengembangan perangkat lunak cepat sedikit lebih rinci dengan tujuan memberikan ulasan secara ringkas. Pengembangan *extreme programming* dapat dilihat menggunakan suatu alur tahapan pengembangan yang dapat dilihat pada Gambar 2.1

:



Gambar 1.1 *Extreme Programming*

Sumber: (Pressman, 2012)

Berdasarkan tahapan tersebut merupakan suatu paradigma yang diinginkan mencakup didalam seperangkat aturan dan praktik-praktik dalam empat konteks kegiatan kerangka kerja yaitu :

1. Perencanaan

Perencanaan yang bertujuan awal untuk mendengarkan kebutuhan secara fungsional dan non fungsional serta membuat penggambaran sistem secara *interface* sebagai keperluan *outputnya*.

2. Perancangan

Penggunaan rancangan bertujuan untuk membangun dan menggambarkan sistem yang akan dibuat guna mempermudah proses pengkodean.

3. Pengkodean

Pengkodean bertujuan untuk membuat hasil perancangan dengan melakukan konstruksi atau pembuatan sistem secara seluruh menggunakan kode.

4. Pengujian

Pengujian dilakukan untuk menguji hasil pengkodean dan selanjutnya dilakukan evaluasi dan implementasi.

Kelebihan Model Extreme Programming :

Komunikasi dalam XP dibangun dengan melakukan pemrograman berpasangan (*pair programming*). *Developer* didampingi oleh pihak klien dalam melakukan *coding* dan *unit testing* sehingga klien bisa terlibat langsung dalam pemrograman sambil berkomunikasi dengan *developer*. Selain itu perkiraan beban tugas juga diperhitungkan.

1. Menekankan pada kesederhanaan dalam pengkodean: “*What is the simplest thing that could possibly work?*” Lebih baik melakukan hal yang sederhana dan mengembangkannya besok jika diperlukan. Komunikasi yang lebih banyak mempermudah, dan rancangan yang sederhana mengurangi penjelasan.
2. Setiap *feed back* ditanggapi dengan melakukan tes, *unit test* atau *system integration* dan jangan menunda karena biaya akan membengkak (uang, tenaga, waktu).
3. Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki.

Kelemahan Model Extreme Programming :

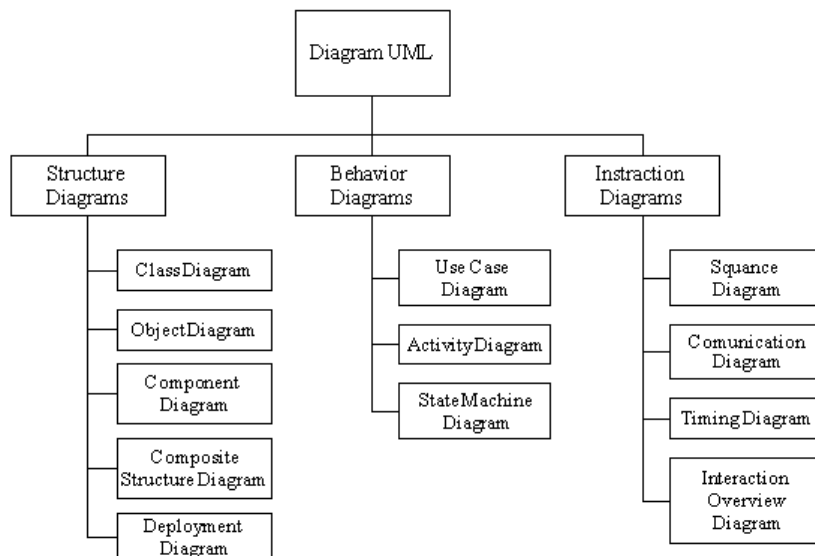
Pengembangan *exreme programming* yang digunakan tentunya masih memiliki beberapa kelemahan atau kekurangan seperti berikut.

1. *Developer* harus selalu siap dengan perubahan karena perubahan akan selalu diterima.

2. Tidak bisa membuat kode yang detail di awal (*prinsip simplicity* dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga).

1.3 *Unified Modelling Language (UML)*

Menurut Rosa dan Salahuddin (2013), UML (*Unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut adalah bagan UML pada Gambar 2.2.

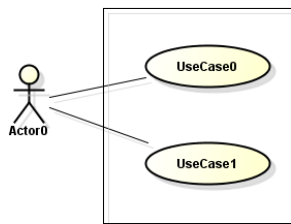


Gambar 1.2 Bagan UML
(Sumber : Rosa dan Salahuddin, 2013)

Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*).

1.3.1 Use Case Diagram

Menurut Rosa dan Salahuddin (2013) *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut adalah konsep *Use Case Diagram* pada Gambar 2.3.

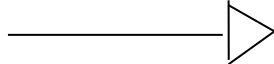
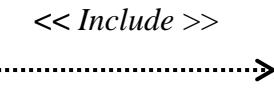
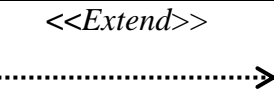


Gambar 1.3 *Use Case Diagram*
(Sumber : Rosa dan Salahuddin, 2013)

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.2 berikut ini:

Tabel 1.2 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<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 <i>frase</i> nama <i>use case</i> .
2.		Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi (<i>association</i>): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.



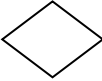
4.		Generalisasi (<i>generalization</i>): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.		<i>Include</i> berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.


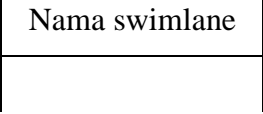

Sumber: (Rosa dan Salahuddin, 2013)

1.3.2 Activity Diagram

Menurut Rosa dan Salahuddin (2013) *activity Diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.3 berikut ini:

Tabel 1.3 Simbol *Activity Diagram*

No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

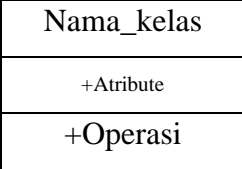

4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber: (Rosa dan Salahuddin, 2013)

1.3.3 *Class Diagram*

Menurut Rosa dan Salahuddin (2013) *class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.4 berikut ini:

Tabel 1.4 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	Antar Muka/ <i>Interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

	Nama_Interface	
3.	Asosiasi / <i>Asociation</i> —————	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	Asosiasi Berarah / <i>Directed Association</i> —————>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi —————▷	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Ketergantungan / <i>dependency</i> —————>	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> —————◊	Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>)

Sumber: (Rosa dan Salahuddin, 2013)

1.4 Mobile

Menurut Maressa (2014) aplikasi *mobile* yaitu program siap pakai yang direkap untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ketempat yang lain. Maka aplikasi *mobile* dapat di artikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari satu tempat ketempat yang lain serta mempunyai ukuran yang kecil. HTML 5 adalah HTML5 merupakan bahasa markah untuk

menstrukturkan dan menampilkan isi dari jejagat jembat, sebuah teknologi yang dipakai sebagai standar dari internet, HTML5 adalah revisi ke lima dari HTML dengan tujuan untuk mempermudah penerapan pada media teknologi terbaru

1.4.1 *CodeIgniter*

CodeIgniter menurut (Asroni, 2018) merupakan *framework* untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. *CodeIgniter* memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat mengembangkan dalam perangkat *web*, dekstop maupun *mobile*. *CodeIgniter* memiliki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan.

1.4.2 *JQuery Mobile*

Menurut Zammetti (2013) *JavaScript* menjadi bahasa pemrograman yang paling banyak digunakan dan salah satu yang paling sering digunakan adalah *jQuery*, *library JavaScript* dirancang untuk menyederhanakan scripting HTML. Kalau dilansir dari situs *jQuery* sendiri, *jQuery* disebut memiliki karakteristik yang kecil, cepat, dan punya banyak fitur.

Menurut Zammetti (2013) *jQuery* yaitu memberikan kemampuan pada *jQuery* untuk bisa melakukan lintas *platform*. Sehingga *jQuery* tetap akan memiliki fungsi yang sama sekaligus memperbaiki *error* yang terjadi ketika dijalankan di berbagai jenis *browser* seperti *Safari*, *Google Chrome*, *Firefox*, *Android*, dan *IOS*.

1.4.3 *MySql*

MySQL Sabar, Heryanto dan Lestari (2019) 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.

Menurut (Kadir, 2016), 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.

1.5 Metode Pengujian Sistem

Metode pengujian sistem merupakan metode yang digunakan untuk melakukan testing pada sistem yang dibangun sehingga di peroleh hasil berupa sistem yang sesuai fungsinya dengan menerapkan metode ISO 25010

1.5.1 ISO 25010

ISO/IEC 25010 atau (International Organisation for Standardisation, 2011) merupakan model kualitas sistem dan perangkat lunak. Model kualitas produk terdiri dari delapan karakteristik yang berhubungan dengan sifat statis perangkat lunak dan sifat dinamis dari sistem komputer. Karakteristik yang didefinisikan oleh kedua model tersebut relevan untuk semua produk perangkat lunak dan sistem komputer.

Karakteristik dan subkarakteristik memberikan terminologi yang konsisten untuk menentukan, mengukur dan mengevaluasi kualitas sistem dan perangkat lunak.

1.5.2 Tahapan Pengujian Sistem

Tahapan pengujian sistem digunakan untuk mengetahui proses pengujian yang akan dilakukan, berikut adalah tahapan pengujian ISO:

a. *Functional Suitability*

Sejauh mana perangkat lunak mampu menyediakan fungsi yang memenuhi kebutuhan yang dapat digunakan dalam kondisi tertentu. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu.

1. *Functional completeness*, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna secara spesifik.
2. *Functional correctness*, sejauh mana produk atau sistem menyediakan hasil yang benar sesuai kebutuhan.
3. *Functional appropriateness*, sejauh mana fungsi yang disediakan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.

b. *Compatibility*

Sejauh mana sebuah produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen dan/atau menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat keras dan environment perangkat lunak yang sama. Karakteristik ini dibagi menjadi 2 karakteristik yaitu.

1. *Co-existence*, sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien sementara berbagi sumber daya dengan produk atau sistem yang lain tanpa merugikan produk atau sistem tersebut.
2. *Interoperability*, sejauh mana dua atau lebih produk, sistem atau komponendapat bertukar informasi dan menggunakan informasi tersebut.

c. *Usability*

Sejauh mana sebuah produk atau sistem dapat digunakan oleh user tertentu untuk mencapai tujuan dengan efektif, efisien, dan kepuasan tertentu dalam konteks penggunaan. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu.

1. *Appropriateness recognizability*, sejauh mana pengguna dapat mengetahui apakah sistem atau produk sesuai kebutuhan mereka.
2. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu yang belajar menggunakan sistem atau produk dengan efisien, efektif, kebebasan dari resiko dan kepuasan dalam konteks tertentu.
3. *Operability*, sejauh mana produk atau sistem mudah dioperasikan dan dikontrol.
4. *User error protection*, sejauh mana produk atau sistem melindungi pengguna terhadap membuat kesalahan.
5. *User interface aesthetics*, sejauh mana antarmuka pengguna dari produk atau sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.

6. *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai tujuan tertentu sesuai konteks penggunaan.

d. *Reliability*

Sejauh mana sebuah sistem, produk atau komponen dapat menjalankan fungsi tertentu dalam kondisi tertentu selama jangka waktu yang ditentukan. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

1. *Maturity*, sejauh mana produk atau sistem mampu memenuhi kebutuhan secara handal di bawah keadaan normal.
2. *Availability*, sejauh mana produk atau sistem siap beroperasi dan dapat diakses saat perlu digunakan.
3. *Fault tolerance*, sejauh mana produk atau sistem tetap berjalan sebagaimana yang dimaksud meskipun terjadi kesalahan pada perangkat keras atau perangkat lunak.
4. *Recoverability*, sejauh mana produk atau sistem mampu dapat memulihkan data yang terkena dampak secara langsung dan menata ulang kondisi system seperti yang diinginkan ketika terjadi gangguan.

e. *Security*

Sejauh mana sebuah produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan level otorisasi yang dimiliki. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu.

1. *Confidentiality*, sejauh mana produk atau perangkat lunak memastikan data hanya bisa diakses oleh mereka yang berwenang untuk memiliki akses.

2. *Integrity*, sejauh mana produk atau perangkat lunak mampu mencegah akses yang tidak sah untuk memodifikasi data.
3. *Non-repudiation*, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan terhadap peristiwa atau tindakan tersebut.
4. *Accountability*, sejauh mana tindakan dari suatu entitas dapat ditelusuri secara unik untuk entitas.
5. *Authenticity*, sejauh mana identitas subjek atau sumber daya dapat terbukti menjadi salah satu yang diklaim.

f. *Portability*

Sejauh mana keefektifan dan efisiensi sebuah sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan pada lingkungan yang berbeda. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu.

1. *Adaptability*, sejauh mana produk atau sistem dapat secara efektif dan efisien disesuaikan pada perangkat lunak, perangkat keras dan lingkungan yang berbeda.
2. *Installability*, sejauh mana produk atau sistem dapat berhasil dipasang atau dihapus dalam lingkungan tertentu.
3. *Replaceability*, sejauh mana produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama pada lingkungan yang sama.

g. *Performance Efficiency*

Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu.

Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

1. *Time behaviour*, sejauh mana respon dan pengolahan waktu produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
2. *Resource utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
3. *Capacity*, sejauh mana batas maksimum parameter produk atau sistem dapat memenuhi persyaratan.

h. *Maintainability*

Sejauh mana keefektifan dan efisiensi dari sebuah produk atau sistem dapat dirawat. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

1. *Modularity*, sejauh mana sistem terdiri dari komponen terpisah sehingga perubahan atau modifikasi pada salah satu komponen tersebut memiliki dampak yang kecil terhadap komponen yang lain.
2. *Reusability*, sejauh mana aset dapat digunakan lebih oleh satu sistem atau digunakan untuk membangun aset lain.
3. *Analyzability*, tingkat efektivitas dan efisiensi untuk mengkaji dampak perubahan pada satu atau lebih bagian-bagian produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.

4. *Modifiability*, sejauh mana produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menurunkan kualitas produk yang ada.
5. *Testability*, tingkat efektivitas dan efisiensi untuk membentuk kriteria uji dari produk, sistem atau komponen dan uji dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.

