

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Tinjauan pustaka merupakan sumber referensi yang diambil dari jurnal penelitian yang berkaitan dengan penelitian, berikut penjelasan tinjauan pustaka :

Tabel 2.1 Tinjauan Pustaka

1.	Judul	Analisis dan perancangan sistem informasi simpan pinjam berbasis website pada koperasi karya abadi
	Penulis	Ainin Najmi, Mohammad Nadjib
	Tanggal/Tahun	2020
	Tujuan Penelitian	Tujuan penelitian adalah merancang sistem informasi yang disusun untuk proses simpan pinjam dan membantu melakukan pekerjaan dengan cepat yaitu memberikan alternatif solusi untuk memperbaiki permasalahan pada koperasi, serta untuk menguji sistem informasi simpan pinjam berbasis web pada Koperasi Karya Abadi.
	Permasalahan	Pelayanan informasi transaksi yang sekarang berjalan masih dilakukan secara manual, contohnya pada pinjaman ketua harus menyertakan tanda tangannya untuk bukti kuat pinjaman itu diterima dan untuk proses pencairan dana pinjaman. Hal tersebut dapat dilihat proses transaksi pengajuan pinjaman. Dimana pada proses manual tersebut memakan waktu.
	Subjek Penelitian	Sistem Informasi Simpan Pinjam
	Metode Penelitian	Waterfall

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

	HasilPenelitian	Hasil dari penelitian ini adalah sistem informasi simpan pinjam dapat mengurangi permasalahan seperti mempermudah setiap proses pengajuan maupun Transaksinya.
2.	Judul	Rancang Bangun Aplikasi Pelaporan Otomatis Untuk Koperasi Simpan Pinjam Berbasis Web
	Penulis	Hamdan Sidik, Yosep Septiana, Dini Destiana, Siti Fatimah
	Tanggal/Tahun	2020
	TujuanPenelitian	Tujuan penelitian ini adalah memajukan bagi seluruh anggota dan khususnya untuk orang banyak
	Permasalahan	Belum adanya notifikasi yang memperlihatkan masa tenggang untuk pembayaran pinjaman
	SubjekPenelitian	Rancang Bangun Aplikasi Pelaporan
	MetodePenelitian	Rational Unified Process(RUP)
	HasilPenelitian	Hasil dari penelitian ini adalah aplikasi koperasi simpan pinjam pelaporan otomatis berbasis <i>web</i>
3.	Judul	Implementasi Sistem Informasi Simpan Pinjam Berbasis Multi User
	Penulis	Nurdin,N & Putra G.A.W.E
	Tanggal/Tahun	2020
	TujuanPenelitian	Tujuan penelitian ini adalah membangun sistem Informasi yang dapat membantu pengelola koperasi dalam mengolah data-data transaksi simpan pinjam dan menyajikan informasi berupa laporan yang dapat menjadi dasar dalam pengambilan kebijakan untuk mengembangkan KUD Setia Darma.

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

	Permasalahan	Perhitungan dan penyusunan laporan dapat menimbulkan permasalahan, karena transaksi simpan pinjam dicatat terlebih dahulu dalam sebuah buku kemudian di input kedalam komputer. Selain itu penggunaan aplikasi <i>Microsoft Excel</i> sangat mengandalkan keterampilan operatornya karena harus memiliki kemampuan dalam menyusun kolom dan baris yang ada didalam aplikasi <i>Microsoft Excel</i> .
	SubjekPenelitian	Sistem Informasi Simpan Pinjam
	MetodePenelitian	Waterfall
	HasilPenelitian	Hasil penelitian ini adalah sistem informasi simpan pinjam yang dibangun dapat mempercepat proses pengolahan data serta menghasilkan laporan dengan tepat waktu, akurat dan relevan
4.	Judul	Rancang Bangun Aplikasi Simpan Pinjam Koperasi Berbasis Web Studi Kasus PT.Damai Indah Golf
	Penulis	Hadi Zakaria, Fifih Fatmawati
	Tanggal/Tahun	2021
	TujuanPenelitian	Tujuan dari penelitian ini adalah untuk mempermudah dalam menyajikan laporan simpan pinjam.
	Permasalahan	Sistem informasi koperasi simpan pinjam yang digunakan di koperasi PT. Damai Indah Golf masih secara manual sehingga mengakibatkan terjadinya keterlambatan dalam proses simpan pinjam.
	Subjek Penelitian	Sistem Informasi Simpan Pinjam
	MetodePenelitian	Waterfall

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

	Hasil Penelitian	Hasil penelitian ini adalah sistem informasi simpan pinjam yang dibangun dapat mempercepat proses pengolahan data serta menghasilkan laporan dengan tepat waktu, akurat dan relevan
5.	Judul	Aplikasi Koperasi Simpan Pinjam Berbasis Web Pada Torus Jaya Mandiri
	Penulis	Joko S Dwi Raharjo, Achmad Sidik, Nurhayati Marbun
	Tanggal/Tahun	2021
	Tujuan Penelitian	Tujuan dari penelitian ini adalah merancang sistem informasi agar dalam pengolahan data dan pembuatan laporan menjadi lebih cepat, tepat dan akurat. Tanpa harus melakukan proses penarikan angsuran yang dilakukan oleh kolektor
	Permasalahan	Pengelolaan data simpan pinjam masih dilakukan secara manual yaitu proses penarikan angsuran dilakukan oleh kolektor dengan cara menagih ke rumah nasabah langsung. sehingga menyebabkan kesalahan pencatatan data dan keterlambatan pembuatan laporan
	Subjek Penelitian	Sistem Informasi Simpan Pinjam
	Metode Penelitian	Waterfall
	Hasil Penelitian	Hasil penelitian ini adalah sistem informasi simpan pinjam yang dibangun dapat mempercepat proses pengolahan data serta menghasilkan laporan dengan tepat waktu, akurat dan relevan

Berdasarkan penelitian terdahulu seperti diatas menunjukkan bahwa dengan adanya sistem akan mempermudah proses integrasi data antar pengguna, serta dapat dilihat perbedaannya yaitu :

1. UML (*Unit Modelling Language*) sebagai alat pengembangan sistem serta metode pengujian sistem menggunakan *iso25010*.
2. Sistem mampu mengintegrasikan data antar pengurus koperasi, ketua, manager keuangan dan anggota koperasi.
3. Menggunakan metode *extreme programming*.

2.2 Pengertian Koperasi

Menurut [10] yang menyatakan bahwa koperasi adalah suatu perserikatan dengan tujuan berusaha selalu dengan semangat tidak memikirkan diri sendiri sedemikian rupa, sehingga masing-masing sanggup menjalankan kewajibannya sebagai anggota dan mendapat imbalan sebanding dengan pemanfaatan mereka terhadap organisasi.

2.3 Simpan Pinjam

Menurut [10] menyatakan bahwa simpan pinjam merupakan sebuah kegiatan menyimpan dan meminjam uang yang modalnya diperoleh dari simpanan pokok dan simpanan wajib para anggota koperasi. Kemudian modal yang telah terkumpul tersebut dipinjamkan ke para anggota koperasi dan terkadang dipinjamkan kepada orang lain yang bukan anggota koperasi yang memerlukan pinjaman uang, baik untuk keperluan konsumtif maupun modal usaha.

2.3.1 Jenis-jenis simpanan

Menurut [10] Jenis-jenis Simpan Pinjam :

1. Simpanan Pokok

Simpanan pokok adalah sejumlah uang yang sama banyaknya dan sama nilainya yang wajib dibayarkan oleh anggota kepada koperasi pada saat masuk menjadi anggota.

2. Simpanan Wajib

Simpanan wajib adalah sejumlah simpanan tertentu yang tidak harus sama, wajib dibayar oleh anggota kepada koperasi dalam waktu dan kesempatan tertentu. Simpanan wajib tidak dapat diambil selama yang bersangkutan menjadi anggota.

3. Tabungan Koperasi

Tabungan koperasi adalah simpanan pada koperasi yang penyetornya dilakukan berangsur-angsur atau kuasanya dengan menggunakan buku tabungan koperasi, setiap saat pada hari kerja koperasi.

2.3.2 Jenis-jenis pinjaman berdasarkan jangka waktunya :

1. Pinjaman jangka pendek, yaitu pinjaman yang jangka waktu pengembaliannya kurang dari 1 tahun. Misalnya pinjaman untuk membiayai kelancaran koperasi usaha termasuk pinjaman modal kerja
2. Pinjaman jangka menengah, yaitu pinjaman yang jangka waktu pengembaliannya 1 sampai 3 tahun. Biasanya nya pinjaman ini untuk menambah modal kerja, misalnya untuk membiayai pengadaan bahan buku. Pinjaman jangka menengah dapat pula dalam bentuk pinjaman investasi.
3. Pinjaman jangka panjang, yaitu pinjaman yang jangka waktu pengembaliannya atau jatuh temponya melebihi 3 tahun. Misalnya pinjaman investasi yaitu pinjaman untuk pengadaan sarana/alat produksi.

2.4 Sistem Informasi

Definisi menurut [11] Sistem informasi adalah sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi mengenai saldo persediaan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegritasi dan saling melengkapi dengan

menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

2.5 *Web Based*

Menurut [12], *Web Based* adalah aplikasi yang dibuat berbasis *web* yang membutuhkan *web server* dan *browser* untuk menjalankannya. Sistem berbasis *website* memudahkan terhadap proses pencatatan, pencarian data, serta memberikan data laporan yang baik [13].

Menurut [13] Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya :

1. Tidak membutuhkan *hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.
2. Server yang dibutuhkan cukup diinstallkan *tools* pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).
4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

2.6 **PHP**

Menurut [14], PHP adalah bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis.

Sehingga menurut [15] PHP merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa HTML dan *script* lainnya.

2.7 *MySql*

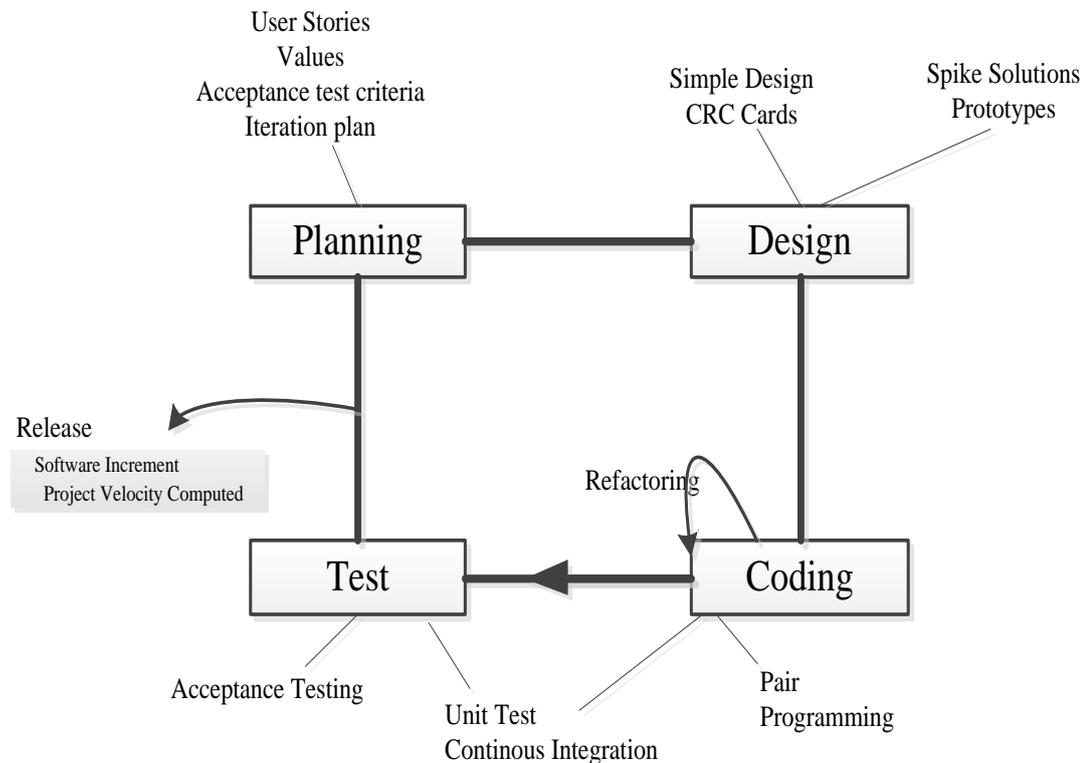
Menurut [16], *MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data.

Menurut [17] mendefinisikan *mysql* adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

MySQL merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

2.8 *Metode Extreme Programming*

Menurut [18] *extreme programming* berdasarkan sejarah singkat bahwa pengembangan perangkat lunak banyak digunakan untuk pengembangan yang lebih cepat dengan meliputi tahapan *planning*, *design*, *coding* dan *testing*. Berikut merupakan konsep *Extreme programming*.



Gambar 2.1 *Extreme Programming*

Sumber: [18]

2.8.1 Kerangka Kerja Extreme Programming

Pengembangan yang dilakukan menggunakan XP dengan proses yang lebih cepat dengan tahapan seperti *planning*, *design*, *coding* dan *testing* :

1. *Planning*/Perencanaan
Tahap ini dimulainya dengan pemahaman konteks bisnis dari aplikasi dengan mendefinisikan keluaran seperti fitur, fungsi, penentuan waktu dan biaya serta alur pengembangan.
2. *Design*/Perancangan
Tahap perencanaan secara sederhana dengan alat mendesain kartu CRC (*Class Responsibility Collaborator*) yang digunakan untuk pemetaan kelas-kelas yang akan diguanakna pada diagram UML.
3. *Coding*/Pengkodean
Hal utama dalam pengembangan menggunakan XP yaitu *pari programming* (Proses pembuatan program melibatkan 2 atau lebih programmer).
4. *Testing*/Pengujian
Tahap ini fokus pada pengujian fitur pada aplikasi sehingga tidak ada kesalahan dan sesuai dengan proses bisnisnya.

2.8.2 Keuntungan dan Kerugian *Extreme Programming*

Keuntungan pada penerapan metode XP yaitu :

- a. Dalam hal XP menjalin komunikasi yang baik dengan klien pada pengembangan aplikasi.
- b. Saling menghargai antar developer dan meningkatkan komunikasi.
- c. Dapat menjadi pembelajaran bagi orang lain.
- d. Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat
- e. Dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat.
- f. Developer tidak berkerja secara berlebihan.
- g. Dengan XP dapat membuat keputusan yang bersifat teknis.

2.8.3 Class Responsibility Collaboration (CRC) Card

Menurut [19] *Class Responsibility Collaboration (CRC) Card* merupakan tools yang digunakan untuk mendefinisikan behavior dan responsibility dari masing-masing class dan hubungan kolaborasi antara *class-class* tersebut. *CRC Card* dikembangkan dengan interaksi antara analis dan pengguna. Masing-masing orang diminta untuk mendeskripsikan logika yang diperlukan untuk memenuhi suatu *responsibility*, dan informasi apa saja yang dibutuhkan tetapi tidak dimilikinya. *Class* lain yang memiliki informasi yang dibutuhkan akan menjadi *collaborator* untuk *behavior* tersebut.

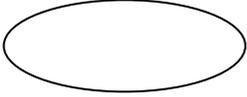
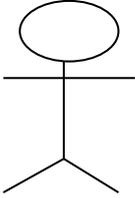
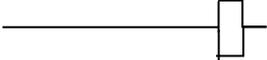
2.9 Alat Pengembang Sistem

Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram [20]. Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut [21], *Unified Modelling Language* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language*.

2.9.1 Use Case Diagram

Menurut [21], *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 simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.2 :

Tabel 2.2 Simbol Use Case Diagram

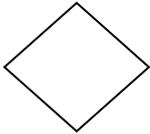
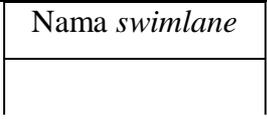
No	Simbol	Deskripsi
1.		<p><i>Use case</i> : Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i>.</p>
2.		<p>Aktor : seseorang/sesuatu yang berinteraksi dengan yang akan dibuat diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda.</p>
3.		<p>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.</p>
4.		<p>Generalisasi (<i>generalization</i>) : merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum.</p>
5.	<p><<Include>></p>	<p><i>Include</i> berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.</p>
6.	<p><<Extend>></p> 	<p>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.</p>

Sumber:[21]

2.9.2 Activity Diagram

Menurut [21], *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:

Tabel 2.3 Simbol *Activity Diagram*

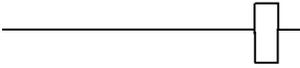
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		<i>Swimlane</i> yaitu merupakan sebuah aktivitas Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber:[21]

2.9.3 Class Diagram

Menurut [21], *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 :

Tabel 2.4 Simbol *Class Diagram*

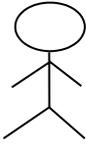
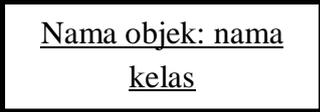
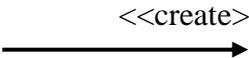
No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/<i>Interface</i></p>  <p>Nama_<i>Interface</i></p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / <i>Asociation</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / <i>Directed Association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Ketergantungan / <i>dependency</i></p> 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>)

Sumber:[21]

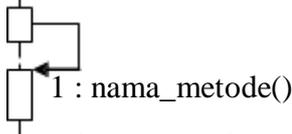
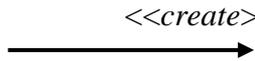
2.9.4 Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada *sequence diagram* pada Tabel 2.5 :

Tabel 2.5 Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	Aktor  Atau <u>Nama</u> <u>aktor</u> Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda di awal frase nama aktor.
2.	Garis hidup /lifeline 	Menyatakan kehidupan suatu objek.
3.	Objek 	Menyatakan objek yang berinteraksi peran.
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Tabel 2.5 Simbol *Sequence Diagram* (Lanjutan)

6.	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/ metode, maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe <i>send</i> 1 : masukan 	Meyatakan bahwa suatu objek mengirimkan data/masukan/informasi/ ke objek lainnya, arah panah mengarah pada objek yang menerima kembalian.
8.	Pesan tipe <i>return</i> 1: keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

Sumber : [21]

2.10 Pengujian ISO 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan *ISO/IEC 9126* tentang *software engineering* [22]. *Product quality* ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

1. Kualitas dalam model penggunaan,
2. Model kualitas produk, dan
3. Data model kualitas

Model kualitas produk terdiri dari delapan karakteristik yang berhubungan dengan sifat statis perangkat lunak dan sifat dinamis dari sistem komputer. Model ini berlaku untuk sistem komputer dan produk perangkat lunak. 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.

2.10.1 *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.

2.10.2 *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.

2.10.3 Usability

Sejauh mana sebuah produk atau sistem dapat digunakan oleh user tertentu untuk mencapai tujuan dengan efektif, *eficiency*, 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.

2.10.4 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.

2.10.5 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.

2.10.6 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.

2.10.7 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 maksimumparameter produk atau sistem dapat memenuhi persyaratan.

2.10.8 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.