

BAB II

LANDASAN TEORI

2.1 Studi Literatur

Berdasarkan hasil yang didapatkan melalui kajian literatur terhadap penelitian terdahulu yang telah dilakukan. Maka penulis merujuk pada beberapa literatur yang cukup berkaitan dengan penelitian ini. Berikut ialah tabel kajian literatur pustaka yang menjadi referensi penulis, dapat dilihat pada tabel 2.1

Tabel 2. 1 Penelitian Pengembangan Sistem

No	Detail Referensi	Keterangan
1	Judul	Pemodelan SCRUM dalam Pengembangan Sistem Informasi Kesehatan pada Klinik Ar-Rokhim Sragen Kabupaten Sragen SCRUM
	Tahun	2019
	Metode	<i>SCRUM</i>
	Permasalahan	Proses bisnis pada klinik Ar-Rokhim pada bagian pendaftaran, pencatatan rekam medis pasien, penjualan obat, pembuatan laporan dan penghitungan biaya rawat belum memaksimalkan aplikasi sistem yang ada. Hal ini membuat proses bisnis yang dilakukan oleh klinik kurang efektif. Salah satu metode pengembangan sistem yang sering digunakan untuk mengembangkan sistem yaitu waterfall, metode ini memiliki kekurangan karena tidak diperkenankan adanya perubahan pada siklus pengembangan, membutuhkan tim yang banyak dan waktu pengembangan yang cukup lama.(Kurniawan & Sani, 2019)
Hasil	Tujuan dari penelitian ini ialah untuk menghasilkan pengembangan sistem informasi dengan cepat dalam penyesuaian perubahan dan sesuai dengan target yang diharapkan.(Kurniawan & Sani, 2019)	

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

No	Detail Referensi	Keterangan
2	Judul	Perancangan Website Sistem Informasi Pelayanan Rawat Jalan Pada Puskesmas Benda Baru Pamulang.
	Tahun	2021
	Metode	<i>Agile Scrum Development</i>
	Permasalahan	Pengolahan data secara manual memiliki kelemahan yaitu membutuhkan waktu yang cukup lama, sering terjadi kesalahan dalam memasukan data pasien, dan juga saat laporan menjadi tidak efektif karena berkas pasien terkadang terdapat kerusakan atau hilang. (Dewi et al., 2021)
Hasil	Hasil penelitian yang dapat disimpulkan bahwa Website rawat jalan Puskesmas Benda Baru Pamulang ini telah dirancang dan diuji coba dengan menggunakan blackbox testing. Dengan aplikasi ini dapat membantu Puskesmas Benda Baru Pamulang dalam sistem rawat jalan. (Dewi et al., 2021)	
3	Judul	Sirame: Sistem Informasi Rekam Medis Klinik Bulan Sabit Merah Indonesia (BSMI) Kota Sukabumi.
	Tahun	2022
	Metode	<i>Scrum Agille</i>
	Permasalahan	Klinik Bulan Sabit Merah Indonesia (BSMI) Kota Sukabumi masih terdapat kendala dalam pengolahan data rekam medis pasien karena masih menggunakan tulisan tangan di kartu rekam medis, sehingga data rawan hilang atau rusak. Sistem kearsipan belum terkomputerisasi menghambat proses pencarian dan penyimpanan data pasien. (Septiani et al., 2022)
Hasil	Hasilnya sistem rekam medis Klinik Bulan Sabit Merah Indonesia (BSMI) Kota Sukabumi dapat memudahkan dalam pengelolaan data rekam medis, pencarian data dan lebih efektif dalam penyimpanan data. (Septiani et al., 2022)	

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

No	Detail Referensi	Keterangan
4	Judul	Perancangan Sistem Informasi Manajemen Data Warga Untuk Monitoring Kesehatan Masyarakat Dalam Tingkat RW Dengan Metode SCRUM (Studi Kasus: Dusun Wonokambang)
	Tahun	2021
	Metode	<i>Scrum</i>
	Permasalahan	Covid-19 dapat dengan mudah menyebar dan menginfeksi tubuh manusia. Pada kondisi seperti ini tidak dapat dilakukan pendataan warga secara langsung dan perlu adanya alat bantu untuk memantau kasus Covid-19.(Putro et al., 2021)
	Hasil	Dihasilkan sebuah sistem informasi manajemen berbasis web yang digunakan sebagai pengelolaan data penduduk dan melakukan monitoring kesehatan warga yang diterapkan pada tingkat RT/RW yang dapat diakses secara realtime.(Putro et al., 2021)
5	Judul	Perancangan Website Sistem Informasi Pelayanan Rawat Jalan Pada Puskesmas Benda Baru Pamulang.
	Tahun	2020
	Metode	<i>Waterfall</i>
	Permasalahan	sistem yang digunakan pada puskesmas sukahaji masih menggunakan sistem manual. Hal ini menimbulkan beberapa kendala, terutama ketidakvalidan dan ketidakcocokan data. Sering pula terjadi kendala pada saat pencarian informasi data pasien, begitu pula dengan sulitnya mencari informasi stok obat maupun proses pembuatan laporan untuk kepala puskesmas.(Dewi, 2020)
	Hasil	Sistem informasi puskesmas berbasis web dirancang bertujuan untuk membangun sistem informasi yang terkomputerisasi, sehingga memudahkan pihak puskesmas mengolah data pasien dan rekam medis pasien hingga menjadi laporan.(Dewi, 2020)

Berdasarkan pembahasan literatur diatas menunjukkan terdapat persamaan dan perbedaan antara penelitian terdahulu dengan penelitian sekarang. Berikut adalah perbedaan dengan penelitian sekarang yaitu :

1. Beberapa penelitian masih menggunakan metode lama seperti *waterfall* untuk penelitian sekarang akan menggunakan Metode *Scrum*.
2. Penelitian ini akan menyajikan informasi Pelayanan Poliklinik Umum, Pelayanan Kehamilan, Pelayanan Tindakan Medik Umum, Pemeriksaan Laboratorium, Pelayanan Vaksin , dan Informasi Penyakit.
3. Sistem ini akan dibuat dengan menggunakan *framework* Laravel.

2.2 Sistem Informasi

Secara umum, sistem dapat diartikan sebagai sekumpulan elemen yang saling berhubungan atau terintegrasi yang dirancang untuk mencapai suatu tujuan tertentu. Misalnya, jika sistem memiliki elemen yang tidak membantu mencapai tujuan yang sama, elemen tersebut tentu saja bukan bagian dari sistem yang akan dibuat. Informasi adalah hasil pengolahan data yang masuk akal dan berguna (Pradana, 2016). Sistem informasi memiliki beberapa tujuan utama, yaitu : Peningkatan produktivitas, Pengurangan biaya, Peningkatan pengambilan keputusan, Peningkatan layanan ke pelanggan, Peningkatan strategi-strategi baru. Berikut merupakan beberapa karakteristik sistem informasi, yakni antara lain:

1. Komponen Sistem Informasi

Suatu sistem informasi terdiri dari beberapa komponen yang saling berkaitan, yang artinya saling bekerja sama membentuk suatu kesatuan.

2. Batasan

Batasan merupakan daerah yang membatasi suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Keluaran

Keluaran adalah hasil dari pemrosesan dan pengelompokan menjadi keluaran yang berguna.

4. Lingkungan Luar Sistem Informasi

Lingkungan suatu sistem informasi adalah batas dari sistem yang mempengaruhi operasi sistem informasi. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem informasi tersebut.

5. Penghubung

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Penghubung ini merupakan sumber-sumber daya yang mengalir dari satu subsistem ke subsistem lainnya.

6. Masukan

Masukan adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk di proses. Masukan dapat berupa hal-hal berwujud maupun tidak berwujud.

7. Pengolahan

Pengolahan merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna.

8. Sasaran/Tujuan

Suatu sistem informasi pasti mempunyai tujuan atau sasaran. Sasaran dari sistem informasi sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.3 Pelayanan Kesehatan

Pelayanan Kesehatan adalah segala upaya yang dilakukan untuk memberikan tindakan medis yang diselenggarakan sendiri atau bersama-sama dalam suatu organisasi untuk memelihara dan meningkatkan kesehatan, mencegah dan mengobati penyakit, serta memulihkan kesehatan individu, keluarga, kelompok masyarakat (Handayani, 2016). Pelayanan kesehatan bisa dilakukan di beberapa tempat pelayanan Kesehatan seperti klinik, rumah sakit atau puskesmas.

2.4 Web

Web adalah kumpulan halaman yang menampilkan informasi tentang data teks, data gambar, data animasi, audio, video, dan semua kombinasi baik statis maupun dinamis yang terhubung ke jaringan internet (Destiningrum & Adrian, 2017). Secara garis besar web adalah perangkat lunak yang menampilkan dokumen di web yang memungkinkan pengguna untuk mengakses internet melalui perangkat lunak dengan koneksi internet.

2.5 PHP

PHP adalah singkatan dari *PHP Hypertext Preprocessor*. Bahasa yang ditafsirkan memiliki kesamaan dengan bahasa C dan Perl, yang memiliki perintah sederhana yang digunakan untuk membangun aplikasi web. PHP/F1 adalah nama awal untuk PHP (*Personal Home Page/Form Interface*). Pertama kali dibuat oleh Rasmus Lerdoff. PHP pada awalnya merupakan program CGI yang dirancang

untuk menerima input melalui formulir yang ditampilkan di *browser web*. PHP memudahkan pemeliharaan situs web Anda. Proses *update* dapat dijalankan menggunakan aplikasi yang dibangun dengan *script*. PHP adalah *script* untuk *script web* sisi *server*, *script* yang membuat dokumen HTML dengan cepat, dokumen HTML yang dihasilkan oleh aplikasi, bukan dokumen HTML yang dibuat dengan editor teks atau editor HTML (Andrianto & Nursikuwagus, 2017).

2.6 Laravel

Laravel adalah kerangka kerja PHP yang dirilis di bawah lisensi MIT dan dibangun di atas konsep MVC (*Model View Controller*) (Nugroho & Rohimi, 2020). Laravel juga merupakan pengembangan situs web berbasis MVP yang ditulis dalam PHP yang meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan pemeliharaan awal dan menyediakan sintaks yang ekspresif, jelas, dan hemat waktu.

2.7 MySQL

MySQL adalah *Relational Database Management System* (RDBMS) di distribusikan secara gratis di bawah GPL (*General Public License*) (Andrianto & Nursikuwagus, 2017). Dimana setiap orang dapat menggunakan MySQL secara gratis, tetapi tidak boleh digunakan kepentingan komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.8 Metode Scrum

2.8.1 Pengertian Scrum

Scrum merupakan salah satu model dari metodologi *Agile* pada manajemen pengembangan proyek. Scrum bisa digunakan untuk pengembangan system secara keseluruhan, pengembangan sistem sebagian maupun proyek internal/pelanggan (Destiningrum & Adrian, 2017). Sedangkan *Scrum* merupakan suatu kerangka kerja di mana orang-orang bisa menyelesaikan permasalahan yang kompleks yang selalu terdapat perubahan disetiap tahapan, yang di mana saling menghasilkan produk dengan nilai tinggi secara kreatif dan produktif (Schwaber & Sutherland, 2013).

Scrum bersifat ringan, mudah dipahami dan sulit dikuasai. *Scrum* tidaklah suatu proses maupun metode untuk meningkatkan produk tetapi metode ini lebih sebagai kerangka kerja di mana di dalamnya dapat memasukkan bermacam-macam proses serta metode yang akan dipergunakan. *Scrum* terdiri dari *event*, *roles* dan *artifacts* (Schwaber & Sutherland, 2013). Scrum menggunakan pendekatan *transparency*, *inspection* dan *adaptation*. Tujuan utama *Scrum* adalah untuk melihat permasalahan yang ada, dan melakukan adaptasi terhadap masalah tersebut. Pengembangan perangkat lunak menggunakan *Scrum* menekankan untuk mengambil setiap langkah pada pengembangan perangkat lunak secara singkat.

2.8.2 Tim Scrum

Dalam kerangka kerja Scrum memiliki beberapa bagian di dalam tim yang memegang kendali terhadap proses pengembangan serta menentukan solusi terbaik untuk menyelesaikan tugas. Tim Scrum memproses produk secara berkala dan bertahap, diantaranya adalah :

1. *Scrum Master*

Scrum Master bertugas untuk menjamin proses *Scrum* sudah dipahami dan dikerjakan sesuai dengan teori, praktik dan aturan main *Scrum*. *Scrum Master* merupakan seorang *leader* yang melayani tim *Scrum* serta membantu merubah interaksi-interaksi yang tidak bermanfaat untuk bisa memaksimalkan nilai yang dihasilkan oleh tim *Scrum* (Schwaber & Sutherland, 2013).

2. *Product Owner*

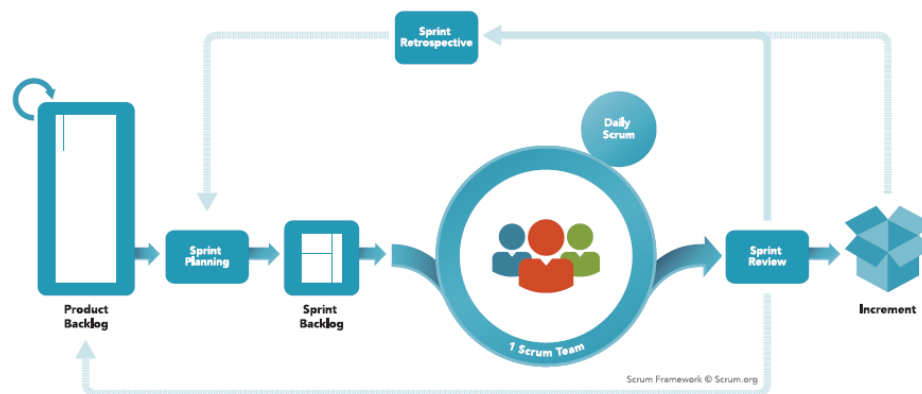
Product Owner bertanggung-jawab untuk memaksimalkan value suatu produk dan hasil kerja tim *development*. *Product Owner* merupakan satu-satunya orang yang bertanggung jawab untuk mengelola *Product Backlog* (Schwaber & Sutherland, 2013).

3. *Team Development*

Team Development terdiri dari para professional yang bekerja untuk menghasilkan tambahan potongan produk atau yang disebut dengan *inkremen*. *Team Development* dibentuk dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaan secara mandiri. Pekerjaan yang ada di dalam akan meningkatkan efisiensi dan efektif (Schwaber & Sutherland, 2013).

2.8.3 Tahapan *Scrum*

Menurut Organisasi SCRUM Internasional berikur adalah gambar tahapan SCRUM, dapat dilihat pada gambar 2.1 :



Gambar 2. 1 Tahapan SCRUM

Penjelasan tahapan pada SCRUM :

1. *Product Backlog*

Product Backlog adalah daftar terurut, dari setiap hal yang berkemungkinan dibutuhkan di dalam produk, dan juga merupakan sumber utama, dari daftar kebutuhan mengenai semua hal yang perlu dilakukan terhadap produk. *Product Owner* bertanggung-jawab terhadap *Product Backlog*, termasuk isinya, ketersediaannya, dan urutannya. *Product Backlog* menjabarkan semua fitur, fungsi, kebutuhan, penyempurnaan dan perbaikan terhadap produk di rilis mendatang. Item *Product Backlog* memiliki atribut deskripsi, urutan, estimasi dan nilai bisnis (Schwaber & Sutherland, 2013).

2. *Sprint Planning*

Proses pekerjaan yang akan dilaksanakan di dalam *Sprint* akan direncanakan pada saat *Sprint Planning*. Perencanaan ini dibuat secara kolaboratif oleh seluruh anggota tim *Scrum*. *Sprint Planning* dibatasi maksimum delapan jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang pendek Batasan waktunya biasanya lebih singkat. *Scrum Master*

wajib memastikan bahwa *event* atau pekerjaan dilaksanakan sesuai dengan tujuan dan waktu yang sudah ditentukan (Schwaber & Sutherland, 2013).

3. *Sprint Backlog*

Sprint Backlog adalah sekumpulan item *Product Backlog* yang telah dipilih untuk dikerjakan di *Sprint*, juga di dalamnya rencana untuk mengembangkan potongan tambahan produk dan merealisasikan *Sprint Goal*. *Sprint Backlog* adalah perkiraan mengenai fungsionalitas apa yang akan tersedia di Inkremen selanjutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitas tersebut menjadi potongan tambahan produk yang “Selesai” (Schwaber & Sutherland, 2013).

4. *Daily Scrum*

Daily Scrum biasanya memiliki durasi kurang lebih 15 menit pada setiap pertemuannya selama sprint berlangsung. Pada penerapannya, durasi *Daily Scrum* dapat mengelola jumlah anggota dan jenis. Fungsi *Daily Scrum* untuk mengoptimalkan kolaborasi dan kinerja tim dengan mengecek tugas yang telah dilakukan sejak *Daily Scrum* sebelumnya (Schwaber & Sutherland, 2013).

5. *Sprint Review*

Sprint Review dilakukan pada akhir Sprint untuk meninjau inkremen dan merubah *Product Backlog* bila diperlukan. Pada saat *Sprint Review*, tim *Scrum* dan *stakeholder* berkolaborasi untuk membahas apa yang telah dikerjakan dalam *Sprint* yang baru selesai. Berdasarkan hasil yang sudah dilakukan maka semua perubahan *Product Backlog* pada saat *Sprint* untuk

menentukan apa yang dapat dikerjakan berikutnya (Schwaber & Sutherland, 2013).

6. *Sprint Retrospective*

Sprint Retrospective adalah sebuah kesempatan bagi Tim *Scrum* untuk meninjau pekerjaan yang sudah dilakukan dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya. *Sprint Retrospective* dilangsungkan setelah *Sprint Review* selesai dan sebelum *Sprint Planning* berikutnya. Ketika sebuah item *Product Backlog* atau Inkremen dikatakan “Selesai”, setiap pihak harus mengerti dengan apa yang dimaksud dengan “Selesai”. Walaupun definisi ini berbeda-beda antar tim *Scrum*, sesama anggota tim harus memiliki pemahaman yang sama mengenai pekerjaan yang harus mereka selesaikan guna memastikan adanya transparansi. Ini adalah definisi selesai untuk Tim *Scrum* dan ini digunakan untuk memeriksa apakah pekerjaan untuk mengembangkan Inkremen dianggap selesai (Schwaber & Sutherland, 2013).

2.8.4 Artefak *Scrum*

Artefak *Scrum* mempresentasikan pekerjaan atau nilai, yang bertujuan untuk menyediakan transparansi dan kesempatan-kesempatan untuk peninjauan dan adaptasi. Artefak yang didefinisikan oleh *Scrum* secara khusus dirancang untuk meningkatkan transparansi dari informasi kunci, dengan begitu semua pihak dapat memiliki pemahaman yang sama terhadap artefak (Schwaber & Sutherland, 2013).

1. *Product Backlog*

Product Backlog adalah daftar terurut, dari setiap hal yang berkemungkinan dibutuhkan di dalam produk, dan juga merupakan sumber utama, dari daftar

kebutuhan mengenai semua hal yang perlu dilakukan terhadap produk. *Product Owner* bertanggung-jawab terhadap *Product Backlog*, termasuk isinya, ketersediaannya, dan urutannya. *Product Backlog* menjabarkan semua fitur, fungsi, kebutuhan, penyempurnaan dan perbaikan terhadap produk di rilis mendatang. Item *Product Backlog* memiliki atribut deskripsi, urutan, estimasi dan nilai bisnis (Schwaber & Sutherland, 2013).

2. *Sprint Backlog*

Sprint Backlog adalah sekumpulan item *Product Backlog* yang telah dipilih untuk dikerjakan di *Sprint*, juga di dalamnya rencana untuk mengembangkan potongan tambahan produk dan merealisasikan *Sprint Goal*. *Sprint Backlog* adalah perkiraan mengenai fungsionalitas apa yang akan tersedia di Inkremen selanjutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitas tersebut menjadi potongan tambahan produk yang “Selesai” (Schwaber & Sutherland, 2013).

3. *Increment*

Inkrement (tambahan potongan produk) adalah gabungan dari semua item *Product Backlog* yang diselesaikan pada *Sprint* berjalan dan nilai-nilai dari *Inkrement sprint-sprint* sebelumnya. Pada akhir *Sprint*, inkremen terbaru harus “Selesai”, yang artinya berada dalam kondisi yang berfungsi penuh dan memenuhi definisi “Selesai” yang dibuat oleh Tim Scrum. Terlepas apakah *Product Owner* akan merilis produknya, produk harus selalu berada dalam kondisi yang berfungsi penuh (Schwaber & Sutherland, 2013).

4. *Definition of Done*

Ketika sebuah item *Product Backlog* atau Inkremen dikatakan “Selesai”, setiap pihak harus mengerti dengan apa yang dimaksud dengan “Selesai”. Walaupun definisi ini berbeda-beda antar tim Scrum, sesama anggota tim harus memiliki pemahaman yang sama mengenai pekerjaan yang harus mereka selesaikan guna memastikan adanya transparansi. Ini adalah definisi selesai untuk Tim Scrum dan ini digunakan untuk memeriksa apakah pekerjaan untuk mengembangkan Inkremen dianggap selesai (Schwaber & Sutherland, 2013).

2.9 Puskesmas

Puskesmas adalah satuan kerja teknis dinas kesehatan kabupaten/kota yang bertanggung jawab atas terselenggaranya pembangunan kesehatan di satu atau sebagian wilayah. Fungsi Puskesmas, sebagai pusat pengembangan kesehatan masyarakat, pusat peningkatan taraf kesehatan masyarakat, dan pusat pelayanan kesehatan masyarakat pada serta sebagai ujung tombak pelayanan di bidang kesehatan masyarakat (Sucipto & Hermawan, 2017). Dalam penelitian ini menggunakan Puskesmas Kisam Ilir yang berlokasi di Jl. Lintas Desa Tanjung Jati, Desa Tanjung Jati, Kecamatan Kisam Ilir, Kabupaten Ogan Komering Ulu Selatan, Provinsi Sumatera Selatan.

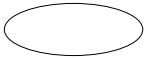





2.10 UML

UML merupakan mekanisme standar industri untuk memvisualisasikan, menentukan, membangun dan mendokumentasikan sistem perangkat lunak. UML telah menjadi standar resmi untuk kebutuhan seperti pemodelan data, pemodelan bisnis, dan pengembangan *real time* (Rosa & Salahuddin, 2018).

2.10.1 Use Case Diagram

Usecase merupakan permodelan untuk menggambarkan *behavior* sistem informasi yang akan dibuat (Rosa & Salahuddin, 2018). *Usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut symbol-simbol yang akan digunakan dalam menggambarkan *Usecase Diagram*, dapat dilihat pada tabel dibawah ini :


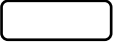



Tabel 2. 2 *Usecase Diagram*

No	Simbol	Deskripsi
1		<i>Usecase</i> Fungsional 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>usecase</i> .
2		<i>Actor</i> Actor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. Diluar sistem informasi, biasanya dinyatakan menggunakan kata benda.
3		Asosiasi/ <i>association</i> Asosiasi/ <i>association</i> merupakan komunikasi antara <i>actor</i> dan <i>usecase</i> yang berpartisipasi pada <i>usecase</i> atau <i>usecase</i> memiliki interaksi dengan <i>actor</i> .
4		Generalisasi (<i>Generalization</i>) Merupakan hubungan (umum – khusus) antara dua buah <i>usecase</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5		<i>Include</i> <i>Include</i> berarti <i>usecase</i> yang ditambahkan akan dipanggil saat <i>usecase</i> tambahan dijalankan.
6		<i>Extend</i> Extend merupakan <i>usecase</i> tambahan ke sebuah <i>usecase</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>usecase</i> tambahan itu.

2.10.2 Activity diagram

Activity diagram adalah diagram yang menggambarkan *workflow* atau aliran kerja dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa & Salahuddin, 2018). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Activity Diagram*, dapat dilihat pada tabel dibawah ini :

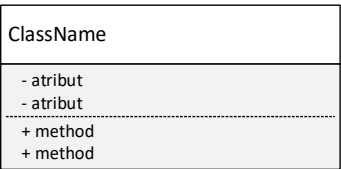




Tabel 2. 3 *Activity Diagram*

No	Simbol	Deskripsi
1		Status awal aktivitas sistem, sebuah awal proses dari sebuah sistem.
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		Pengabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.10.3 Class Diagram

Class Diagram adalah sebuah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa & Salahuddin, 2018). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram*, dapat dilihat pada tabel dibawah ini :

Tabel 2. 4 *Class Diagram*

No	Simbol	Deskripsi
1		<i>Class</i> adalah Blok-blok pembangun pada pemrograman berbasis objek yang terdiri dari tiga bagian yaitu class, atribut dan method.
2		<i>Interface</i> atau yang disebut dengan antar muka.
3		Asosiasi (<i>Association</i>) Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan simbol.
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 simbol.
5		Generalisasi adalah relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus)

2.11 ISO 25010

ISO/IEC 25010 adalah standar model kualitas dari sebuah sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang *software engineering* (Lamada et al., 2020). Dikutip dari (P. Miguel et al., 2014) Menurut IEE *Standard Glossary of Software Enggunnering Terminology* kualitas produk software didefinisikan sebagai :

1. Sejauh mana suatu sistem, komponen atau proses memenuhi persyaratan yang ditentukan.
2. Sejauh mana suatu sistem, komponen atau proses memenuhi kebutuhan atau harapan pengguna.

Karakteristik ISO/IEC 25010 ini dibagi menjadi beberapa karakteristik antara lain yaitu :

2.11.1 *Functional Suitability*

Karakteristik sejauh mana perangkat lunak mampu menyediakan fungsi yang memenuhi kebutuhan yang dapat digunakan dalam kondisi dan keadaan tertentu (P. Miguel et al., 2014). *Functional Suitability* memiliki sub karakteristik yaitu :

1. *Functional completeness*, sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan.
2. *Functional Correctness*, sejauh mana produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang dibutuhkan.
3. *Functional Appropriateness*, sejauh mana fungsi memfasilitasi pencapaian tugas dan tujuan tertentu.

2.11.2 *Reliability*

Sejauh mana suatu sistem, produk atau komponen melakukan fungsi tertentu dalam kondisi tertentu untuk jangka waktu tertentu (P. Miguel et al., 2014). Karakteristik ini terdiri dari sub karakteristik berikut :

1. *Maturity*, sejauh mana sistem, produk, atau komponen beroperasi dan dapat diakses saat diperlukan untuk digunakan.
2. *Availability*, sejauh mana sistem, produk atau komponen beroperasi dan dapat diakses saat diperlukan untuk digunakan.
3. *Fault Tolerance*, sejauh mana sistem, produk atau komponen beroperasi sebagaimana dimaksud meskipun ada kesalahan perangkat keras atau perangkat lunak.

4. *Recoverability*, sejauh mana dalam hal gangguan atau kegagalan, produk atau sistem dapat memulihkan data yang terpengaruh secara langsung dan membangun kembali keadaan sistem yang diinginkan.

2.11.3 *Performance Efficiency*

Karakteristik ini mewakili kinerja relative terhadap jumlah sumber daya yang akan digunakan dalam kondisi yang dinyatakan (P. Miguel et al., 2014).

Karakteristik ini terdiri dari sub-karakteristik berikut :

1. *Time Behaviour*, sejauh mana respon dan waktu pemrosesan serta tingkat input suatu produk atau sistem saat menjalankan fungsinya untuk memenuhi persyaratan.
2. *Resource Utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem ketika menjalankan fungsinya.
3. *Capability*, sejauh mana batas maksimum parameter produk atau sistem memenuhi persyaratan.

2.11.4 *Usability*

Sejauh mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi dan kepuasan dalam penggunaannya (P. Miguel et al., 2014). Karakteristik ini terdiri dari sub-karakteristik berikut :

1. *Appropriateness recognizability*, sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai untuk kebutuhan mereka.
2. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan pembelajaran menggunakan produk atau sistem dengan efektivitas, efisiensi, kebebasan

dari risiko yang didapat dan tingkat kepuasan dalam ranah penggunaan tertentu.

3. *Operability*, sejauh mana produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikendalikan.
4. *User Error Protection*, sejauh mana sistem melindungi pengguna dari membuat kesalahan.
5. *User Interface Aesthetics*, sejauh mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.
6. *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh orang-orang dengan jangkauan karakteristik dan kemampuan terluas untuk mencapai tujuan tertentu dalam ranah penggunaan tertentu.

2.11.5 *Security*

Sejauh mana suatu produk atau sistem melindungi informasi dan data sehingga orang atau produk atau sistem lain memiliki tingkat akses data yang sesuai dengan jenis dan tingkat otorisasi mereka (P. Miguel et al., 2014). Karakteristik ini terdiri dari sub-karakteristik berikut :

1. *Confidentiality*, sejauh mana produk atau sistem memastikan bahwa data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
2. *Integrity*, sejauh mana sistem, produk atau komponen mencegah akses tidak sah masuk ke dalam atau memodifikasi program atau *database*.
3. *Non-repudation*, sejauh mana tindakan dapat dibuktikan telah terjadi sehingga peristiwa atau Tindakan tidak dapat dihindari.

4. *Accountability*, sejauh mana Tindakan suatu entitas dapat dilacak secara unik ke entitas tersebut.
5. *Authenticity*, sejauh mana identitas subjek atau sumber daya dapat dibuktikan sebagai yang diklaim.

2.11.6 *Compatibility*

Sejauh mana suatu produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen lain dan menjalankan fungsi yang diperlukan sambil berbagai lingkungan perangkat keras atau perangkat lunak yang sama (P. Miguel et al., 2014). Berikut karakteristik ini terdiri dari berikut :

1. *Co-existence*, sejauh mana suatu produk dapat melakukan fungsi yang diperlukan secara efisien sambil berbagai lingkungan dan sumber daya yang sama dengan produk lain, tanpa dampak yang merugikan pada produk lain.
2. *Interoperability*, sejauh mana dua atau lebih sistem produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah dipertukarkan.

2.11.7 *Maintainability*

Karakteristik ini mewakili tingkat efektivitas dan efisiensi dimana produk atau sistem dapat dimodifikasi untuk memperbaikinya atau menyesuaikannya dengan perubahan lingkungan dan persyaratan (P. Miguel et al., 2014).

Karakteristik ini terdiri dari sub karakteristik yaitu :

1. *Modularity*, sejauh mana sistem atau program komputer terdiri dari komponen-komponen sedemikian rupa sehingga perubahan pada satu komponen memiliki dampak minimal pada komponen lainnya.

2. *Reusability*, sejauh mana asset dapat digunakan di lebih satu sistem.
3. *Analysability*, tingkat efektivitas dan efisiensi yang memungkinkan untuk menilai dampak pada produk atau sistem dari perubahan yang dimaksudkan untuk satu atau lebih bagiannya atau untuk mendeteksi suatu produk untuk kekurangan atau penyebab kegagalan untuk mengidentifikasi bagian untuk dimodifikasi.
4. *Modifiability*, sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menimbulkan kekurangan atau menurunkan kualitas produk yang ada.
5. *Testability*, tingkat efektivitas dan efisiensi dengan kriteria pengujian yang dapat ditetapkan untuk sistem produk atau komponen dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.

2.11.8 *Portability*

Tingkat efektifitas dan efisiensi dimana sistem produk atau komponen dapat ditransfer dari satu perangkat keras, lunak atau operasional dan penggunaan lainnya (P. Miguel et al., 2014). Karakteristik ini terdiri dari sub yaitu :

1. *Adaptability*, sejauh mana produk atau sistem dapat secara efektif dan efisien diadaptasi untuk perangkat.
2. *Installability*, tingkat efisiensi dimana produk atau sistem dapat berhasil dapat berhasil dipasang dan dilepas di lingkungan tersebut.
3. *Replaceability*, sejauh mana suatu produk dapat menggantikan sistem yang sudah ada.

(P. Miguel et al., 2014)