

BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Sistem Informasi



Gambar 2.1 Sistem Informasi

Ada 2 kelompok pendekatan dalam pendefinisian sistem. Ada yang menekankan pada prosedurnya dan ada juga yang menekankan pada komponen atau elemennya, diantaranya Pendapat pertama menekankan sistem pada komponennya. “Sistem adalah sebuah kumpulan elemen-elemen yang berinteraksi dalam mencapai suatu tujuan tertentu”. Pendapat kedua yaitu menekankan sistem pada prosedurnya. “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama dalam melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu” (Andalia et al., 2015).

Sistem informasi adalah sebuah kumpulan metode normal dimana data dikumpulkan dan diproses menjadi informasi dan dialokasikan pada pengguna. Perencanaan sistem informasi merupakan bagian penting dalam sebuah organisasi dalam menentukan kebutuhan sistem informasi dalam kurun 3 - 5 tahun mendatang dan dimasukkan ke dalam rencana pengembangan sistem informasi (Akbar, Andriansyah, & Utomo, 2016).

2.1.2. Sistem Informasi *Monitoring*



Gambar 2.2 Sistem Informasi *Monitoring*

Monitoring adalah kegiatan pengamatan yang dapat memberikan informasi tentang status, pengukuran dan penilaian yang diselesaikan secara berulang-ulang dari waktu ke waktu, aktifitas pengamatan umumnya dilakukan untuk tujuan tertentu, yaitu untuk memeriksa proses objek atau untuk mengevaluasi kondisi atau kemajuan tujuan yang akan terjadi manajemen atas dampak tindakan berasal dari beberapa jenis objek antara lain yaitu tindakan untuk mempertahankan

manajemen yang sedang berjalan (Putra & Febriani, 2013). *Monitoring* adalah proses pengumpulan dan analisis informasi berdasarkan indikator yang ditetapkan secara sistematis tentang kegiatan program sehingga dapat dilakukan tindakan koreksi untuk penyempurnaan program kegiatan tersebut. Pemantauan yang dapat dijelaskan sebagai kesadaran (*awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu (Aprisa & Monalisa, 2015).

Tujuan dari monitoring adalah mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan dan mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan (Herliana & Rasyid, 2016). Kegiatan monitoring dimaksudkan untuk mengetahui kecocokan dan ketepatan kegiatan yang dilaksanakan dengan rencana yang telah disusun (Dan, Sistem, Dan, Vol, & Tahun, 2016).

Monitoring merupakan salah satu kegiatan vital dalam pengerjaan suatu proyek. Boleh dikatakan, jika proses monitoring tidak dilakukan dengan perhitungan dan kesungguhan, dapat dipastikan proyek yang dikerjakan akan mengalami banyak kerugian, mulai dari kerugian waktu, finansial bahkan bisa jadi akan membuat proyek yang sedang dijalani dianggap gagal oleh konsumen (Herliana & Rasyid, 2016).

2.1.3. Pemrograman Berbasis *Website*

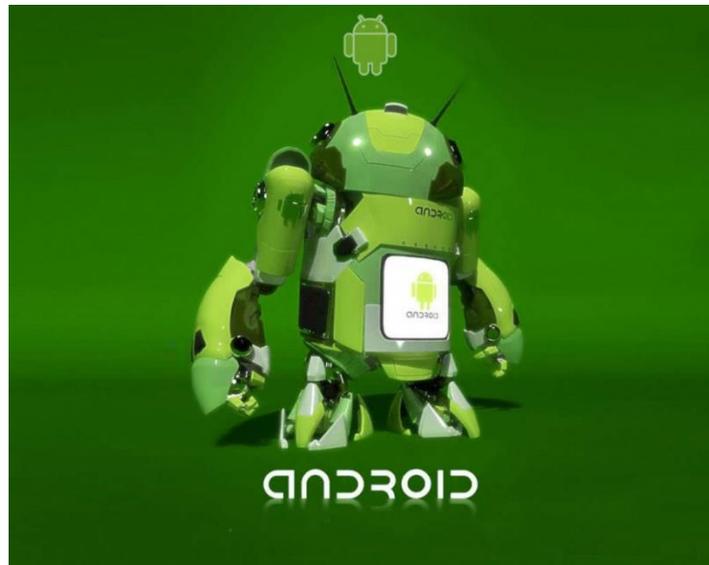


Gambar 2.3 *Website*

Website merupakan sebuah situs pada *internet* yang menyediakan kumpulan informasi yang berhubungan dengan riwayat hidup pemilik situs. *Website* ialah suatu halaman yang memuat situs-situs halaman *web* yang berada di *internet* yang berperan sebagai media penyampaian informasi, komunikasi, ataupun transaksi (Puji Hastanti, Eka Purnama, & Uly Wardati, 2015).

Sejarah *website* dimulai pada bulan maret 1989 ketika Tim *berner lee* yang bekerja di Laboratorium Fisika Partikel di Eropa atau yang dikenal dengan nama *CERN (Consei European Pour la Recherche Newclaire)* yang berada di *Genewa, Swiss*, mengajukan sebuah protokol (suatu tata cara untuk berkomunikasi) sistem distribusi informasi *internet* yang digunakan untuk berbagai informasi diantara para fisikawan. Protokol inilah yang selanjutnya dikenal sebagai protokol *World Wide Web* dan dikembangkan oleh *World Wide Web Consortium* dari sejumlah organisasi yang berkepentingan (Puji Hastanti et al., 2015).

2.1.4. Pemrograman Berbasis *Android*



Gambar 2.4 *Android*

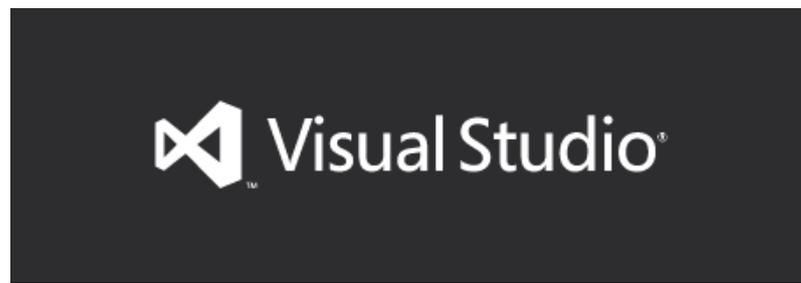
Android adalah sebuah aplikasi lunak untuk perangkat *mobile* yang melingkupi suatu sistem operasi, *middleware* serta aplikasi dasar yang dipublikasikan oleh *Google*. *Android SDK* (*Software Development Kit*) yang menyediakan *API* serta *Tools* yang diperlukan untuk mengembangkan aplikasi perangkat lunak pada *platform android* dengan menggunakan Bahasa *Java*. Sejarah *Android* diawali tahun 2005, dimana ditahun tersebut *Google* memulai membentuk *platform Android* secara intensif. Pada 12 November 2007 *Google* beserta *Open Handset Alliance (OHA)* yaitu persetujuan perangkat *mobile* terbuka, merilis *Google Android SDK*, setelah diumumkan seminggu sebelumnya (Sifauttjani et al., 2017).

Android adalah sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan

platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau *smartphone*. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia* (Juansyah, 2015).

2.2. Teori Khusus

2.2.1. *Visual Studio*



Gambar 2.5 *Visual Studio*

Menurut (Teknik Informatika, Negeri Tanah Laut Jl AYani Km, & Tanah Laut Kalimantan Selatan, 2016) *Visual Studio* adalah sebuah perangkat lunak yang bisa dipergunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi usaha, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi seperti aplikasi *Console*, aplikasi *Windows*, aplikasi *Web* dan aplikasi *Android*. *Visual Studio* meliputi kompiler, *Software Development Kit (SDK)*, *Integrated Development Environment (IDE)*, dan dokumentasinya.

Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*.

2.2.2. *Android Studio*



Gambar 2.6 *Android Studio*

Menurut (Juansyah, 2015) *Android Studio* merupakan *Integrated Development Environment (IDE)* resmi untuk mengembangkan aplikasi *Android* dan gratis (*open source*). *Android Studio* diluncurkan oleh *Google* pada tanggal 16 mei 2013 pada event *Google I/O Conference* ditahun 2013. Sejak saat itu, *Android Studio* menggantikan *Eclipse* sebagai *IDE* resmi untuk mengembangkan aplikasi *Android*.

Android Studio sendiri dikembangkan berdasarkan *IntelliJ IDEA* yang mirip dengan *Eclipse* disertai dengan *ADT plugin (Android Development Tools)*. *Android Studio* memiliki fitur:

1. Projek berbasis pada *Gradle Build*
2. *Refactory* dan pembenahan bug yang cepat
3. *Tools* baru yang bernama “*Lint*” dapat memonitor kecepatan, kegunaan, serta kompetibilitas aplikasi dengan cepat.
4. Mendukung *Proguard* dan *App-signing* untuk keamanan.
5. Memiliki *GUI* aplikasi *android* lebih mudah
6. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan.

2.2.3. *SQL Server*



Gambar 2.7 *SQL Server*

Structured Query Language (SQL) merupakan sebuah bahasa yang digunakan untuk mengakses data pada sebuah *database*. Bahasa ini adalah bahasa baku yang digunakan pada manajemen basis data relasional. Saat ini hampir

seluruh server *database* yang ada telah mendukung bahasa tersebut untuk melakukan manajemen datanya (Wibawa, 2013).

SQL merupakan salah satu jenis bahasa pemrograman yang digunakan untuk menyusun *query database*, melakukan pembaharuan, dan tugas-tugas pengelolaan lain yang berkaitan dengan *database*. *SQL* bukan sebuah bahasa pemrograman kompleks yang bisa membuat aplikasi sendiri, namun cukup andal untuk menciptakan fungsi interaktif di dalam program database lain (Mustakim & Krisnawati, 2013).

2.2.4. Database Management System



Gambar 2.8 *Database Management System*

Database Management System (DBMS) adalah sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut *DBMS* jika memenuhi persyaratan minimal sebagai berikut:

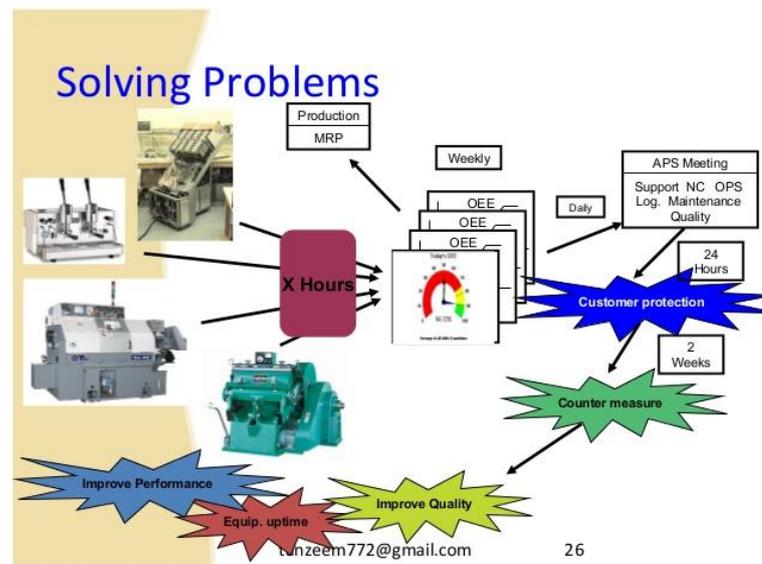
1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data

3. Mampu menangani akses data
4. Mampu menangani backup data

(Habibie, Isnanto, & Kridalukmana, 2016).

Menurut Kristanto, *Database Management System (DBMS)* ialah kumpulan data yang bekerjasama satu dengan yang lainnya bersama dengan sebuah program untuk dikelola. *DBMS* terdiri dari basis data dan perangkat lunak pengelola data yang dipergunakan untuk melihat, menambah, membarui dan menghapus data. Sedangkan menurut Connolly dan Begg, *Database Management System (DBMS)* merupakan suatu sistem aplikasi yang memungkinkan pengguna untuk menghasilkan, mengatur serta memelihara akses ke dalam sebuah *database* (Habibie et al., 2016).

2.2.5. Overall Equipment Effectiveness



Gambar 2.9 Overall Equipment Effectiveness

Overall Equipment Effectiveness (OEE) merupakan produk dari kegiatan operasi dengan *six big losses* pada mesin/peralatan. Keenam faktor dalam *six big losses* dapat dikelompokkan menjadi tiga komponen utama dalam *OEE* untuk dapat digunakan. Dalam mengukur kinerja mesin/peralatan yaitu *downtime losses*, *speed losses* dan *defect losses*. Menurut Nakajima (1988), *OEE* merupakan ukuran menyeluruh yang mengidentifikasi tingkat produktivitas mesin/peralatan dari kinerja secara teori. Pengukuran ini sangat penting untuk mengetahui area mana yang perlu ditingkatkan produktivitasnya ataupun efisiensi mesin/peralatan dan juga dapat menunjukkan area *bottleneck* yang terdapat pada proses produksi. *OEE* juga merupakan alat ukur untuk mengevaluasi dan memperbaiki cara yang tepat untuk menjamin peningkatan produktivitas penggunaan mesin/peralatan (Alvira, Helianty, & Prasetyo, 2015).

OEE adalah hasil yang dapat dinyatakan sebagai rasio *output* aktual dari peralatan dibagi dengan output maksimum peralatan di bawah kondisi performa terbaik. *OEE* didasarkan pada pengukuran tiga rasio utama, yaitu *Availability (A)*, *Performance (P)*, dan *Quality (Q)*. *Availability* merupakan suatu rasio yang menggambarkan pemanfaatan waktu yang tersedia dalam kegiatan operasi mesin dan peralatan. *Availability* merupakan rasio dari *operation time*, dengan mengeliminasi *downtime* peralatan terhadap *loading time* (Suliantoro, Susanto, Prastawa, Sihombing, & Anita, 2017).

Formula yang digunakan untuk mengukur *availability*:

$$A = \frac{(\text{Loading time} - \text{Downtime})}{\text{Loading time}} \times 100\%$$

Rumus 2.1 Perhitungan *Availability*

Performance merupakan suatu rasio yang menggambarkan kemampuan dari peralatan dalam menghasilkan barang. Rasio ini merupakan hasil dari *operating speed rate* dan *net operating rate*. Formula pengukuran rasio:

$$P = \frac{(\text{Ideal cycle time} \times \text{Processed amount})}{\text{Operating time}} \times 100\%$$

Rumus 2.2 Perhitungan *Performance*

Quality merupakan suatu rasio yang menggambarkan kemampuan peralatan dalam menghasilkan produk yang sesuai dengan standar. Formula yang digunakan untuk pengukuran rasio:

$$Q = \frac{(\text{Processed amount} - \text{Defect amount})}{\text{Processed amount}} \times 100\%$$

Rumus 2.3 Perhitungan *Quality*

Nilai *OEE* dapat diperoleh dengan mengalikan ketiga rasio utama tersebut. Secara matematis formula pengukuran nilai *OEE*:

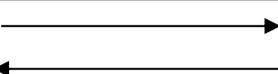
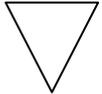
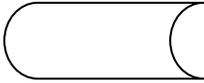
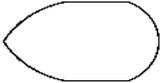
$$OEE = (A \times P \times Q) / 10000$$

Rumus 2.4 Perhitungan *Overall Equipment Effectiveness*.

2.2.6. Aliran Sistem Informasi

Menurut (Ismael, 2017), Aliran Sistem Informasi adalah skema arus pekerjaan dan keseluruhan dari sebuah sistem. Skema ini menjelaskan urutan dari prosedur yang berada pada sebuah sistem. Simbol yang dapat digunakan pada Aliran Sistem Informasi ini seperti pada tabel 2.1.

Tabel 2.1 Simbol Aliran Sistem Informasi

Simbol	Deskripsi
	Proses Komputer
	Dokumen
	Garis Alur
	Manual Proses
	File Storage
	Penghubung antar halaman
	<i>Input menggunakan keyboard</i>
	<i>Input output menggunakan disket</i>
	<i>Output di monitor</i>
	Penghubung

2.2.7. UML

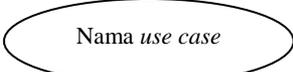
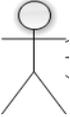
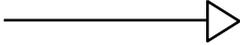
2.2.7.1. Use Case Diagram

Menurut Rosa A.S.,(2015), Diagram *use case* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* (Rosa A.S, 2016);

1. Aktor merupakan 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 itu orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukaran pesan antara unit atau aktor

Tabel 2.2 Penjelasan simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
<p data-bbox="288 412 399 445"><i>Use case</i></p> 	<p data-bbox="810 412 1375 535">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>.</p>
<p data-bbox="288 535 446 568">Aktor / <i>actor</i></p>  <p data-bbox="608 600 738 633">Nama aktor</p>	<p data-bbox="810 535 1375 633">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.</p>
<p data-bbox="288 672 539 705"><i>Asosiasi / association</i></p> 	<p data-bbox="810 672 1375 770">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>
<p data-bbox="288 815 443 848"><<<i>extend</i>>></p> 	<p data-bbox="810 781 1375 938">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek</p>
<p data-bbox="288 938 612 972"><i>Generalisasi / generalization</i></p> 	<p data-bbox="810 938 1375 1061">Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p data-bbox="288 1061 624 1095"><i>Menggunakan / include / uses</i></p> <p data-bbox="421 1095 564 1128"><<<i>include</i>>></p>  <p data-bbox="443 1162 564 1196"><<<i>uses</i>>></p> 	<p data-bbox="810 1061 1375 1182">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p data-bbox="810 1182 1375 1249">Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ol data-bbox="826 1249 1375 1464" style="list-style-type: none"> <li data-bbox="826 1249 1375 1339">1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan <li data-bbox="826 1339 1375 1464">2. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan. <p data-bbox="810 1464 1375 1554">Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan</p>

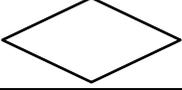
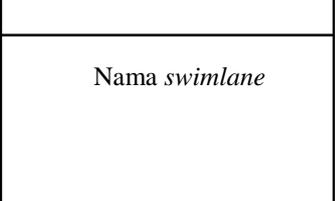
2.2.7.2. Diagram Aktifitas (*Activity Diagram*)

Menurut Nama et al., (2015), Diagram aktivitas digunakan untuk menggambarkan perilaku aliran kontrol atau arus objek yang mungkin terjadi pada sistem. Semua kegiatan dikoordinasikan oleh beberapa model itu bisa dimulai karena tindakan lain sudah selesai dijalankan, objek dan data tersedia, atau beberapa pemicu eksternal lainnya yang menyebabkan aliran terjadi.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa A.S, 2016):

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada perangkat lunak.

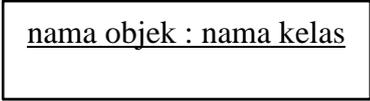
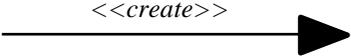
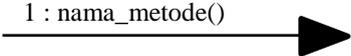
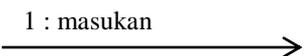
Tabel 2.3 Penjelasan simbol-simbol *Activity Diagram*

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.
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

2.2.7.3. *Sequence Diagram*

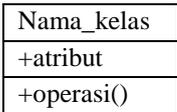
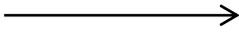
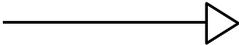
Menurut Djafar (2015), *Sequence diagram* yang dirancang menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*, interaksi yang terjadi antar kelas, operasi apa saja yang terlibat, urutan antara operasi dan informasi yang perlu dilakukan.

Tabel 2.4 Penjelasan *Sequence Diagram*

Simbol	Deskripsi
Aktor  Nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek operasi memanggil / metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe <i>send</i> 1 : masukan 	Menyatakan bahwa suatu objek mengirim data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
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
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

2.2.7.4. Diagram Kelas (*Class Diagram*)

Tabel 2.5 Penjelasan simbol-simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> nama_ <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 beralih / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum – khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Class diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Djafar, 2015).

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

2.3. Penelitian Terdahulu

Pada tinjauan penelitian terdahulu akan dibahas secara lengkap jurnal dan artikel yang mendukung sebagai dasar pembahasan interpretasi penelitian pada bahan sebelumnya.

1. Usulan Peningkatan *Overall Equipment Effectiveness (OEE)* Pada Mesin *Tapping Manual* Meminimumkan *Six Big Losses* Oleh Dianra Alvira, Yanti Helianty, Hendro Prasetyo, Fakultas Teknik, Institut Teknologi Nasional (Itenas) Bandung, 2015, PT X ingin meningkatkan nilai efektivitas penggunaan mesin *tapping* agar nantinya PT X dapat memenuhi semua permintaan konsumen dengan memperbaiki laju produksinya agar tidak terhambat atau terhenti. Berdasarkan masalah yang ada untuk melakukan peningkatan nilai efektivitas mesin maka dapat dilakukan pengukuran nilai *overall equipment effectiveness (OEE)* (Alvira et al., 2015).
2. Penerapan Metode *Overall Equipment Effectiveness (OEE)* Dan *Fault Tree Analysis (FTA)* Untuk Mengukur Efektifitas Mesin Reng Oleh Hery Suliantoro, Novie Susanto, Heru Prastawa, Iyain Sihombing, Anita M., Fakultas Teknik, Universitas Diponegoro Semarang, 2017, penelitian untuk mengukur tingkat efektivitas mesin reng dengan menggunakan metode *Overall Equipment Effectiveness (OEE)*, menganalisa penyebab *six big losses* mesin reng dengan menggunakan *Fault Tree Analysis*

(FTA), dan memberikan rekomendasi perbaikan untuk meningkatkan efektivitas mesin reng (Suliantoro et al., 2017).

3. Pengukuran Nilai *Overall Equipment Effectiveness (OEE)* Sebagai Pedoman Perbaikan Efektivitas Mesin *CNC Cutting* Oleh Agil Septiyan Habib dan H. Hari Supriyanto, Ir., MSIE, Fakultas Teknik, Institut Teknologi Sepuluh Nopember Surabaya, 2012, penelitian ini memiliki fokus utama terhadap faktor *availability* dan *performance* dalam kaitannya untuk memperbaiki nilai *OEE* mesin *CNC Cutting* (Septiyan & Supriyanto, 2012).
4. *Understanding of business performance from the perspective of manufacturing strategies: fit manufacturing and overall equipment effectiveness* Oleh Raghed Ibrahim Esmaeel, Norhayati Zakuan, Noriza Mohd Jamal, Hamed Taherdoost, Fakultas Ekonomi & Administrasi, *The University of Mosul Iraq*, 2018, Mengelola OEE dalam industri manufaktur merupakan strategi penting untuk perbaikan berkelanjutan tepat waktu kualitas pengiriman dan layanan untuk memenuhi kepuasan pelanggan serta harapan mereka. Mencapai kepuasan pelanggan sangat bergantung pada kinerja vendor, keandalan, menanggapi kebutuhan pelanggan, dan perbaikan terus-menerus (Esmaeel, Zakuan, Jamal, & Taherdoost, 2018).
5. *Equivalence assessment method for the resource efficiency of equipment, technologies and production systems* Oleh Kuznetsov A.P., Koriath H.-J., Kalyashina A.V., Langer T., Fakultas Teknik, *Moscow State University*

of Technology Russia, 2018, Para penulis menjelaskan bahwa *OEE* tidak mendukung pengukuran efisiensi yang tepat mengenai penyesuaian, konfigurasi ulang dan waktu pengaturan. Mereka mengusulkan penggunaan indikator untuk produktivitas peralatan secara keseluruhan, yang terhubung ke *OEE* melalui waktu penggunaan kapasitas peralatan (Kuznetsov, Koriath, Kalyashina, & Langer, 2018).