

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

2.1.1 Pengertian Sistem

Menurut pendapat (Suryadharma & Budyastuti, 2019) sistem adalah suatu kumpulan dari elemen-elemen yang saling berhubungan satu dengan yang lain demi menggapai suatu tujuan tertentu. Sedangkan menurut Soeherman dan Pinontoan dalam jurnal (Burrahman, 2018) sistem merupakan serangkaian item yang saling berinteraksi dan memiliki hubungan sama demi mencapai suatu tujuan. Dan menurut pendapat (Fredy & Harman, 2021) kumpulan jejaring aktifitas dan langkah yang sama-sama terhubung sehingga menghasilkan tujuan tertentu disebut dengan sistem. Dengan demikian, dapat disimpulkan bahwa sistem adalah kumpulan perangkat elektronik atau langkah yang memiliki hubungan terintegrasi satu sama lain untuk memproses data demi mencapai suatu tujuan.

Sistem yang kadang-kadang disebut sebagai sistem yang dinamis memiliki tiga komponen dasar yang melakukan intraksi ataupun fungsi menurut (Trisianto, 2018) adalah sebagai berikut :

1. *Input*, menangkap dan perakitan elemen yang memasuki sistem untuk melakukan proses.
2. *Processing*, yang melibatkan proses transformasi yang melakukan konversi dari proses input hingga output.
3. *Output*, hasil yang telah sukses diproses atau hasil transformasi yang dilakukan oleh *processing*.

2.1.2 Pengertian informasi

Menurut (Anggraini et al., 2020) informasi adalah data yang diolah menjadi lebih berarti dan berguna bagi yang menerima. Dan (Suryadharma and Budyastuti, 2019) juga menambahkan tentang pengertian informasi yaitu data dikerjakan menjadi lebih berarti dan memiliki arti bagi setiap penerima informasi tersebut.

Berdasarkan uraian teori yang sudah dipaparkan diatas maka penulis menyimpulkan bahwa informasi adalah pengolahan data yang dilakukan sedemikian rupa agar menjadi informasi yang layak diterima.



Gambar 2. 1 Siklus pengolahan data menjadi informasi

2.1.3 Sistem Informasi

Sistem informasi menurut (Yuliandra & Wulan, 2018) adalah hasil dari rancangan manusia dan tersusun dari komponen-komponen pada suatu organisasi untuk mencapai tujuan tertentu, yaitu memberikan informasi-informasi yang berguna dan bermanfaat. Dalam (Taufiq et al., 2020) menyatakan sistem informasi merupakan sebuah perpaduan atau kerja sama antara manusia, perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, sumber data, kebijakan, serta prosedur yang mempunyai kemampuan untuk mengumpulkan,

menyimpan, mengubah, dan menyebarluaskan informasi dari organisasi. Sedangkan menurut (Taufiq et al., 2019) sistem informasi adalah data yang telah dikumpulakn serta diolah kemudian menghasilkan berbagai informasi yang dapat berguna bagi setiap penerimanya.

Menurut (Elisa et al., 2020) menyatakan bahwa ada beberapa komponen sistem informasi yaitu:

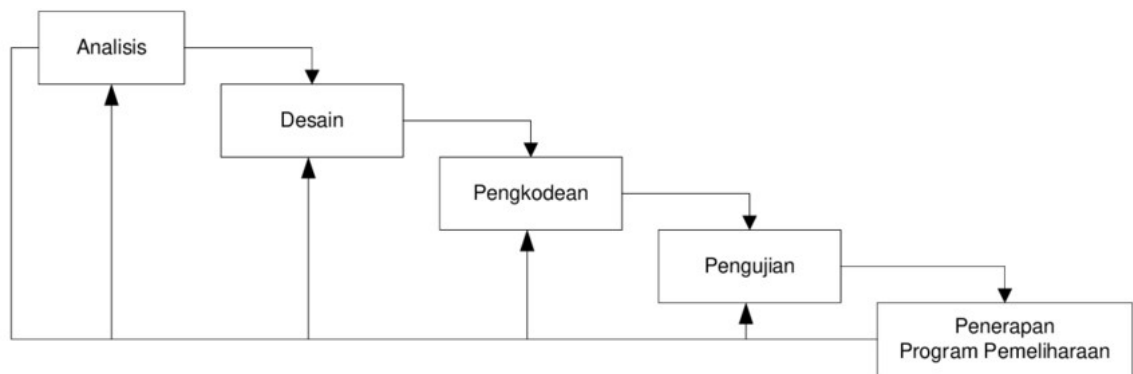
1. Komponen masukan (*input*), data yang masuk kedalam sistem informasi
2. Komponen model, merupakan himpunan langkah, logika dan model matematika yang mengolah data yang tersimpan di dalam database dengan cara yang telah ditentukan untuk menghasilkan keluaran (*output*) sesuai keinginan.
3. Komponen keluaran (*output*), merupakan hasil informasi yang berkualitas dan dokumentasi yang berguna bagi setiap tingkatan manajemen serta yang menggunakan sistem.
4. Komponen teknologi, adalah alat yang digunakan sistem infomasi, teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output* dan memantau pengendalian sistem.
5. Komponen basis data, adalah kumpulan data yang saling berhubungan yang tersimpan di dalam komputer dengan menggunakan *software database*.
6. Komponen kontrol, ialah komponen yang melakukan pengendalian masalah yang ada pada sistem informasi.

2.1.4 Siklus Hidup Pengembangan Sistem (SDLC)

SDLC (*System Development Life Cycle*) ialah cara pengembangan atau Sistem perangkat lunak dengan model dan metodologi yang membantu orang mengembangkan sistem perangkat lunak dengan cara yang didasarkan pada praktik terbaik atau metode yang telah teruji dengan baik (Sitinjak et al., 2020). Pendapat lain mengenai SDLC menurut (Burrahman, 2018) merupakan metode klasik yang digunakan untuk mengembangkan, memelihara, dan menggunakan sistem informasi yang mencakup sejumlah fase dan tahapan. Sedangkan menurut (Hasanah & Sri Untari, 2020) SDLC merupakan proses mengembangkan atau mengubah sistem perangkat lunak menggunakan model dan metodologi yang digunakan untuk mengembangkan sistem perangkat lunak dikenal sebagai pengembangan perangkat lunak. Model-model dalam SDLC yang ditulis dalam buku (Hasanah & Sri Untari, 2020) memiliki beberapa bagian yaitu ;

1. *Waterfall Model* (Model Air Terjun)

Model air terjun merupakan pendekatan klasik dalam pengembangan perangkat lunak yang menggambarkan metode pengembangan linier dan berurutan (Hasanah & Sri Untari, 2020).



Gambar 2. 2 Diagram *waterfall*

Aktifitas-aktifitas model *Waterfall*

1. *Requirement Gathering and Analysis*, mengumpulkan segala kebutuhan-kebutuhan, kemudian menganalisis dan didefinisikan sesuai dengan kebutuhan program yang akan dibangun. Fase ini harus dikerjakan dengan lengkap dan akurat agar dapat memperoleh desain yang tepat.
2. *Desain*, pada tahap ini perancang menghasilkan sistem secara menyeluruh dan menentukan alur perangkat lunak hingga algoritma yang detail.
3. Implementasi, tahapan mengubah desain menjadi kode-kode program. Kode program yang dihasilkan berupa modul-modul yang akan diintegrasikan menjadi sistem yang lengkap.
4. *Integration and Testing*, pada tahap ini melakukan integrasi terhadap modul-modul yang sudah di siapkan dan menguji sistem apakah sudah sesuai dengan desain dan fungsi pada *software* berfungsi dengan baik atau tidak.
5. Verifikasi merupakan klien atau pengguna melakukan pengujian apakah sistem tersebut telah sesuai dengan yang disetujui.

6. *Operation and maintenance*, merupakan proses instalasi dan perbaikan sesuai dengan dengan yang disetujui.

Model *waterfall* memiliki keunggulan yaitu :

1. Pencerminan kepraktisan rekayasa, yang dapat membuat kualitas *software* dapat terjaga.
2. Jenis model yang bersifat lengkap sehingga proses *maintenance* gampang.
3. Dapat menghindari kesalahan konseptual, akurasi biaya dapat dipantau dengan relatif.

Kelemahan model *waterfall* :

1. Pengembangan perangkat lunak lambat, disebabkan proses tidak bisa lompat-lompat sehingga membutuhkan waktu yang lama dalam pengembangannya.
2. Kinerja tidak optimal dan efisien
3. Konflik, bug, dan kesalahan program suatu saat menyebabkan kenaikan biaya dan waktu yang lama.
4. Spesifikasi di awal sulit dimengerti oleh pengguna disebabkan lebih abstrak daripada apa yang seharusnya dilakukan aplikasi (Hasanah & Sri Untari, 2020).

2. Model *Prototype*

Prototyping adalah langkah membuat sebuah *prototype* di mana *prototype* sendiri adalah sebuah model dari sebuah model produk yang mungkin belum memiliki semua fitur produk sesungguhnya namun sudah memiliki fitur – fitur utama dari produk sesungguhnya dan biasa digunakan untuk keperluan *testing/uji coba* untuk bahan uji coba sebelum berlanjut ke fase pembuatan produk

sesungguhnya. Dengan metode *prototyping* ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan suatu produk (Hasanah & Sri Untari, 2020).



Gambar 2. 3 Ilustrasi Model *Prototype*

Tahapan-tahapan pemngembangan model *prototype*, yaitu :

1. Mendengarkan Pelanggan, dalam tahap ini melakukan pengumpulan kebutuhan dari sistem dengan cara mendengarkan keluhan dari pelanggan.
2. Merancang dan membuat *prototype*, pada tahap ini melakukan perancangan dan pembuatan *prototype* sistem.
3. Uji coba, dalam tahap ini melakukan uji coba *prototype* sistem oleh pengguna atau pelanggan.

Kelebihan dan kekurangan model *prototype* adalah sebagai berikut :

Kelebihan model *prototype* memiliki komunikasi yang baik antara pelanggan dan pengembang. Pengembang dapat melakukan pengerjaan lebih baik untuk menentukan apa saja kebutuhan pelanggan. Dapat menghemat waktu. Penerapan

lebih mudah karena pengguna mengetahui apa yang diharapkannya. Pelanggan ikut serta dalam proses pengembangan sehingga memudahkan pengembang mengetahui produk yang dibutuhkan oleh pengguna (Hasanah & Sri Untari, 2020). Sedangkan kelemahan dari model *prototype* adalah memiliki resiko tinggi terhadap masalah-masalah yang tidak terstruktur dengan baik, adanya perubahan besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu.

3. Model *Rapid Application Development* (RAD)

Model *rapid application development* (RAD) adalah model pengembangan perangkat lunak yang pengembangannya tergolong dalam Teknik incremental (bertingkat) (Hasanah & Sri Untari, 2020). Dan juga merupakan strategi siklus hidup yang ditujukan untuk menyediakan pengembangan yang jauh lebih cepat dengan mendapat dan kualitas lebih baik dibanding dengan hasil yang dicapai melalui siklus tradisional.



Gambar 2. 4 Ilustrasi model RAD

RAD memiliki 3 fase sesuai dengan gambar diatas, tiga fase tersebut adalah :

1. *Requirement planning*, dalam fase ini diketahui apa saja yang menjadi kebutuhan sistem, dengan melakukan identifikasi kebutuhan informasi dan masalah yang dihadapi guna menentukan tujuan, batasan-batasan sistem, kendala dan juga apa yang menjadi alternatif pemecahan masalah.
2. *Design workshop*, yaitu melakukan identifikasi solusi alternatif dan memilih solusi yang terbaik. Kemudian membuat desain proses bisnis dan desain pemrograman untuk data-data yang telah didapatkan dan dimodelkan dalam arsitektur sistem informasi. Alat yang digunakan umumnya *unified modelling language* (UML).
3. *Implementation*, setelah melakukan desain *workshop*, selanjutnya sistem diimplementasikan (*coding*) ke dalam bentuk yang dimengerti oleh mesin yang diwujudkan dalam bentuk program atau unit program.

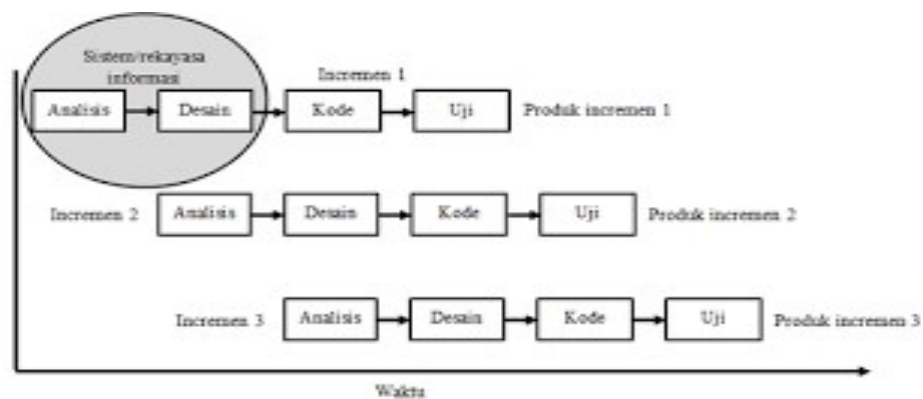
Kelebihan model rapid application development (RAD) adalah sangat berguna dilakukan terhadap user yang tidak memahami kebutuhan-kebutuhan apa saja yang digunakan pada proses pengembangan perangkat lunak. Dapat dilakukan dengan waktu yang singkat, dapat mengurangi biaya, proses pengiriman lebih mudah karena menggunakan potongan-potongan *script*. Lebih mudah diamati, fleksibel, dan sebagainya.

Sedangkan kekurangan dari *rapid application development* (RAD) adalah membutuhkan sumber daya manusia (SDM) yang ahli untuk proyek berskala besar, menuntut pengembang dan pelanggan untuk memiliki komitmen dalam aktifitas *rapid fire* untuk melengkapi sebuah sistem dalam waktu yang singkat, jika tidak adanya komitmen tersebut dapat mengakibatkan kegagalan. Dapat terjadi

lebih banyak kesalahan jika hanya mengutamakan kecepatan dibandingkan dengan biaya dan kualitas.

4. Model Iteratif

Metode ini merupakan pengembangan dari *prototyping model* dan digunakan ketika *requirement* dari *software* akan terus berkembang dalam tahap-tahap pengembangan aplikasi tersebut. Pada setiap iterasi, modifikasi desain yang dibuat dan kemampuan fungsional baru ditambahkan. Ide di balik metode ini adalah untuk mengembangkan sistem melalui siklus berulang (*iterative*) dan dalam porsi yang lebih kecil terhadap waktu (*incremental*) (Hasanah & Sri Untari, 2020).



Gambar 2. 5 Ilustrasi model iterasi

Pengembangan berulang dan incremental merupakan kombinasi dari kedua desain *iterative* atau metode iteratif dan incremental menciptakan model untuk pembangunan.

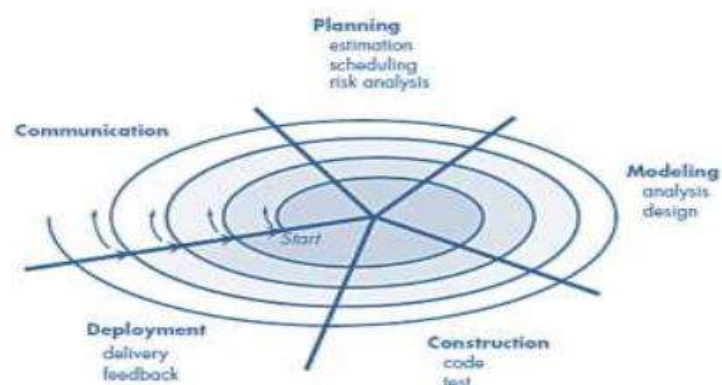
Keuntungan dari *iterative* model adalah pengguna dapat mencoba sistem yang telah dikembangkan kemudian dapat memberi masukan. Lebih mudah

dibangun dan tidak membutuhkan waktu yang lama, kesalahan dan kelalaian dalam penembangan dapat dengan mudah diketahui.

Kelemahan dari model iteratif adalah setiap iterasi bergantung pada *prototype* sebelumnya, formal *end-offphase* kemungkinan tidak terjadi. Dokumentasi sering tidak lengkap, lebih berfokus pada pembuatan *prototype*, isu-isu seperti sistem *backup & recovery*, sistem *performance* dan *security*, kurang atau tidak diperhatikan dan sering terlupakan.

5. Model Spiral

Model *spiral* ialah model pengembangan sistem yang diilustrasikan berupa *spiral*, tidak merepresentasikan rangkaian tahapan dengan penelusuran balik (*tracking back*), tidak ada fase-fase tahapan yang tetap seperti spesifikasi atau rancangan (Hasanah & Sri Untari, 2020). Juga merupakan model *software* yang revolusioner yang merangkai sifat *iterative* dari *prototype* dengan cara kontrol dan aspek sistematis dari model sekuensi linier.



Gambar 2. 6 Ilustrasi model spiral

Putaran-putaran (*loop*) pada model spiral dibagi menjadi empat bagian, yaitu:








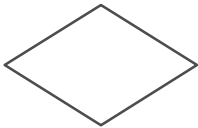


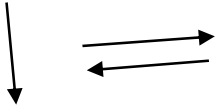
1. *Objective setting* (penetapan tujuan), pada tahap ini melakukan identifikasi kendala pada proses dan produk, tujuan spesifik untuk proyek, membuat rencana pengelolaan yang rinci, merencanakan resiko dalam proyek, strategi alternatif.
2. *Risk assessment and reduction* (penilaian dan pengurangan resiko), dalam tahap ini melihat hasil identifikasi pada detail analisis pada resiko proyek di tahap sebelumnya, sebagai contoh jika ada resiko bahwa requirement tidak pantas atau kurang, maka dibuat prototype sistem.
3. *Development and validation* (pengembangan dan validasi), dalam tahap ini dilakukan pembuatan *prototype* lembaran (*throwaway*) akan menjadi pendekatan pengembangan yang paling baik jika resiko pengguna lebih besar atau dominan.
4. *Planning* (perencanaan), melakukan *review* pada proyek dan pengambilan keputusan terkait kelanjutan tahapan pada putaran spiral selanjutnya.

2.1.5 Aliran Sistem Informasi (ASI)

Aliran sistem informasi (ASI) menurut Iswandy dikutip dari (Masrizal et al., 2021) adalah bagan alir yang menunjukkan aliran dari laporan dan formulir termasuk tembusan-tembusan. Maka dari itu dibutuhkan pedoman-pedoman dalam membuat aliran sistem informasi (ASI), dan menurut Wadisman dalam jurnal (Masrizal et al., 2021) aliran sistem informasi (ASI) ialah dapat digunakan sebagai alat bantu untuk menunjukkan alur pekerjaan secara menyeluruh dari sistem. Berikut adalah tabel simbol-simbol yang digunakan dalam aliran sistem informasi

:

Tabel 2. 1 Simbol aliran sistem informasi

Simbol	Keterangan
	Simbol dokumen, menunjukkan dokumen input dan <i>output</i> baik secara manual, mekanik, dan komputer.
	Simbol pengurutan offline, menunjukkan pekerjaan secara manual
	Simbol pita magnetik, menunjukkan input atau output yang menggunakan kartu plong
	Simbol diskette, menunjukkan proses operasi komputer
	Simbol dari operasi luar, menunjukkan proses dari luar komputer.
	Simbol diskette, menamoilkan <i>input</i> dan <i>output</i> menggunakan diskette
	Simbol keyboard, menampilkan input menggunakan keyboard.
	Simbol keputusan, ditampilkan untuk menyelesaikan kondisi di dalam program.
	Simbol proses terdefenisi, ditampilkan untuk operasi yang rinciannya ada ditempat lain.
	Simbol penghubung, untuk menampilkan sambungan dari bagan alir yang terputus dihalaman lainnya atau halaman yang lama.
	Simbol garis alir, digunakan untuk proses arus



2.1.6 UML (Unified Modelling Language)

Menurut (Akil, 2018) UML merupakan bahasa pemodelan visual yang digunakan untuk spesifikasi, visualisasi, membangun, dan melakukan dokumentasi rancangan dari sistem perangkat lunak. Kemudian (Suendri, 2018) juga memberikan pendapat bahwa UML adalah suatu bahasa berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis *object-oriented* (OO).


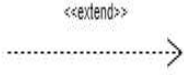
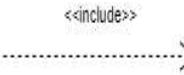

Menurut (Sains & Febrina, 2020) UML memiliki beberapa macam diagram yang dapat digunakan yaitu:

1. *Use Case Diagram*, adalah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Menurut Prabowo Pudjo Widodo dalam jurnal (Suendri, 2018) *use case* menggambarkan *external view* dari sistem yang akan dibuat modelnya, modelnya dapat dijabarkan di dalam diagram *use case*, akan tetapi perlu diketahui bahwa diagram tidak identik dengan model disebabkan model lebih luas daripada diagram.

Tabel 2. 2 Jenis-jenis *Relationship* dalam *use case*

<i>Relationship</i>	<i>Function</i>	<i>Notation</i>
<i>Actor</i>	Orang atau pengguna yang dapat melakukan interaksi dengan system.	
<i>Use case</i>	Fungsionalitas yang dibuat oleh system untuk berinteraksi dan saling bertukar pesan dengan actor.	




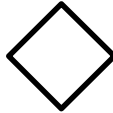

Tabel 2.2 lanjutan

<i>Association</i>	Jalur komunikasi antar aktor dengan <i>use case</i> dimana aktor tersebut berpartisipasi.	
<i>Extend</i>	Penyisipan sebuah fungsionalitas tambahan ke dalam <i>use case</i> dasar yang bersifat opsional.	
<i>Include</i>	Penyisipan sebuah fungsionalitas tambahan ke dalam <i>use case</i> dasar dimana fungsionalitas tersebut bersifat mandatory (wajib)	
<i>Use case generalization</i>	Sebuah hubungan antara <i>use case</i> umum dengan <i>use case</i> yang lebih spesifik dimana <i>use case</i> yang lebih spesifik mewarisi <i>use case</i> umum tersebut	

2. *Activity Diagram*, Menggambarkan aktivitas aliran kerja dari sistem atau proses bisnis atau fitur yang ada pada perangkat lunak. Menurut (Suendri, 2018) *activity diagram* merupakan cara menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi, juga menggambarkan lebih dari

satu proses aksi dalam waktu yang bersamaan. Simbol-simbol dalam *activity diagram* menurut (Hasanah & Sri Untari, 2020) adalah sebagai berikut :



Tabel 2. 3 simbol-simbol *activity* diagram

Simbol	Nama	keterangan
	Initial state	Menggambarkan awal mula suatu aliran aktivitas
	Final state	Menggambarkan berakhirnya aktivitas
	Activity	Menggambarkan aktivitas yang dilakukan
	Decision	Menggambarkan pilhaan kondisi atau cabang-cabang aktivirtas tertentu
	Transition	Untuk menghubungkan satu komponen ke komponen yang lain


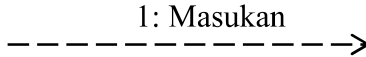
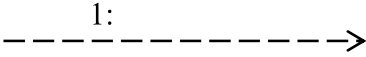
3. *Sequences Diagram*, menggambarkan sifat-sifat objek pada *use case* dengan menjelaskan waktu hidup objek dan pesan yang dikirimkan diterima antar

objek. *Sequence diagram* menggambarkan tingkah laku dari satu scenario tunggal, diagram menunjukkan objek-objek terlibat dalam proses tersebut dan bagaimana urutan penyampaian pesan-pesan antara objek tersebut, sebuah *sequence diagram* menggambarkan interaksi dengan grafik dua dimensi (Akil, 2018). Simbol-simbol dalam *sequence diagram* adalah sebagai berikut :

Tabel 2. 4 simbol-simbol dalam *sequence diagram*

Simbol	Keterangan
 <p>Aktor</p>	Orang, proses, atau sistem lain yang melakukan interaksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat
<p><i>Life line</i></p> 	Menyatakan garis kehidupan dari objek dan diakhiri dengan tanda x pada titik di mana kelas tidak lagi berinteraksi.
<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>Nama objek: nama kelas</p> </div>	Berpartisipasi secara berurutan dengan mengirimkan dan/atau menerima pesan, penempatannya dibagian atas diagram

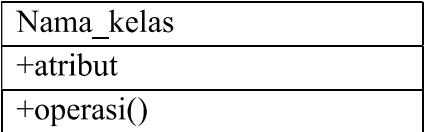


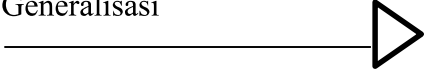
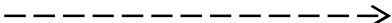
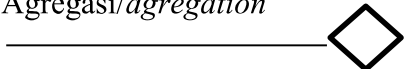
Tabel 2.4 lanjutan

<p>Objek sedang aktif atau berinteraksi</p> 	<p>Menandakan suatu objek mengirim atau menerima pesan, ditempatkan diatas sebuah garis hidup.</p>
<p>Pesan</p>	<p>Objek mengirim suatu pesan ke objek lain</p>
<p><i>Create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objhek yang dibuat.</p>
<p>Masukan</p> 	<p>Suatu objek mengirimksn massukan ke objek lain, arah panah mengarah pada objek yang dikirimi.</p>
<p>Keluaran</p> 	<p>Metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Destroy</p> 	<p>Suatu objek mengakhiri hidup objek lain, mengarah pada objek diakhiri.</p>

4. *Class diagram*, merupakan spesifikasi, jika dipakai, a objek, dan merupakan inti dari pengembangan dan desain berorientasi objek (Elisa et al., 2020).

Sedangkan menurut (Suendri, 2018) kelas diagram merupakan sebagai suatu set objek yang memiliki set atribut dan perilaku yang sama, kelas juga sering disebut sebagai kelas objek, kelas juga memiliki tiga area utama, yaitu nama, atribut, dan operasi. Simbol-simbol dalam kelas diagram adalah sebagai berikut :

Tabel 2. 5 simbol-simbol *class diagram*

Simbol	Keterangan Simbol
	Struktur Penulisan Class pada UML
Antarmuka/ <i>interface</i> Nama_ <i>interface</i>	Pada UML dan OOP pemrograman berorientasi objek <i>interface</i> juga memiliki pengertian yang sama .
Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah 	Satu kelas dapat dikaitkan dengan kelas lain dalam arti karena bagaimana mereka terkait dalam definisinya.
Generalisasi 	Relasi general atau umum
Kebergantungan/ <i>dependency</i> 	Relasi yang menunjukkan adanya nilai saling dependent (ketergantungan)
Agregasi/ <i>agregation</i> 	Relasi yang bersifat agregasi

2.2. Tinjauan Teori Khusus

2.2. 1. *Spray Painting*

Menurut (Kristanto et al., 2017) *Spray* (penyemprotan) merupakan salah proses dalam pengecatan dengan cara mencampurkan atau mengkabutkan bahan cat menggunakan udara, pengkabutan dilakukan dengan menggunakan *spraygun*. Sedangkan *painting* yang berasal dari kata *paint* (cat) adalah cairan yang digunakan untuk melapisi permukaan suatu bahan bertujuan untuk memperindah (*decorative*), memperkuat (*reinforcing*), dan melindungi (*protective*) bahan tersebut. Sedangkan menurut (Tyagita et al., 2020) *spray paint* (pengecatan) merupakan sebuah proses pengaplikasian cat berbentuk cair terhadap sebuah objek, untuk membuat lapisan tipis terhadap objek dan kemudian menjadi lapisan keras atau lapisan cat bertujuan untuk melindungi objek tersebut.

Dari pendapat para ahli diatas dapat disimpulkan *spray paint* (pengecatan) adalah proses pewarnaan suatu produk dengan cara mengkabutkan udara dengan bahan cat dengan tujuan memprindah ataupun melindungi produk tersebut dengan lapisan cat.

2.2. 2. Material

Material (Iqbal et al., 2021) merupakan sebagai gabungan dari bahan-bahan yang diolah untuk menghasilkan suatu produk atau barang jadi supaya lebih bermanfaat. Dikutip dari jurnal (Martha & Setiawan, 2018) material atau bahan baku adalah bahan yang sangat dibutuhkan oleh suatu perusahaan untuk diolah menjadi barang jadi atau produk yang akan memiliki nilai tambah dengan kualitas

yang lebih baik. Berdasarkan pengertian tersebut material merupakan bahan-bahan yang dibutuhkan oleh perusahaan untuk diolah untuk menghasilkan suatu produk yang memiliki nilai dan kualitas terbaik.

2.2. 3. Pengelolaan Material

Pengelolaan atau pengendalian menurut (Rahayu et al., 2019) suatu cara atau mekanisme yang diterapkan dengan baik dengan tujuan melindungi perusahaan dari risiko atau meminimalisir dampak risiko tersebut pada perusahaan jika risiko tersebut terjadi suatu waktu. Pengelolaan/pengendalian material menurut (Maulana & Prasetya, 2021) adalah suatu prosedur dalam manajemen untuk penerapan kebijakan-kebijakan penggunaan material, pengontrolan akuntabilitas dilakukan untuk mengukur material dilokasi tertentu pada saat terjadi penambahan atau pengurangan material.

Dapat disimpulkan bahwa pengelolaan/pengendalian material adalah sebuah cara dalam menjalankan kebijakan-kebijakan untuk melindungi perusahaan dari resiko tertentu jika terjadi suatu waktu, juga pengontrolan terhadap material jika ada penambahan ataupun pengurangan material.

2.2. 4. Sistem pengelolaan material

Sistem pengelolaan material atau manajemen material menurut Bell dan Stukhart yang dikutip dari (Ardi et al., 2017) merupakan suatu manajemen material yang dibutuhkan untuk perencanaan dan pengendalian mutu material, jumlah material, dan penempatan peralatan tepat waktu, sesuai dengan harga dan kebutuhan. Selanjutnya (Siswanto et al., 2021) mendefinisikan sistem manajemen

material merupakan sebuah sistem yang mengkoordinasikan seluruh aktivitas bertujuan merencanakan, mengawasi volume dan waktu terhadap pengadaan material melalui penerimaan atau perolehan dari segi perubahan bentuk perpindahan bahan mentah dan bahan yang sedang dalam proses ataupun bahan jadi.

Dari pendapat para ahli di atas dapat disimpulkan bahwa sistem manajemen material adalah sebuah sistem manajemen yang bertujuan untuk melakukan pengelolaan atau pengkoordinasian perencanaan, penggunaan material, dan pengawasan material dari proses awal penerimaan hingga akhir proses pengolahan material.

2.2. 5. Website

Menurut Juliansyah dikutip dari jurnal (Hutasoit & Silalahi, 2022) *website* atau sistem adalah kumpulan halaman yang dapat menampilkan teks, gambar, video, atau audio. Halaman-halaman ini saling berhubungan dan dapat ditautkan ke situs web lain.. Pendapat lain mengenai *website* dikemukakan dalam jurnal (Elisa et al., 2020) adalah sekumpulan halaman yang berguna untuk menampilkan informasi, teks, gambar diam atau bergerak, animasi, suara dan/atau gabungan semuanya. Sedangkan menurut (Zahara & Harman, 2021) *website* merupakan aplikasi atau perangkat lunak yang digunakan untuk memproses penelitian dengan tujuan mendapatkan informasi dari situs.

Dari hasil penjelasan di atas dapat disimpulkan pengertian website merupakan suatu situs halaman web browser yang di desain sedemikian rupa yang bertujuan

untuk mendapatkan informasi baik berupa teks, audio, suara gambar diam atau bergerak.

2.2. 6. *Database*

Database dalam sistem informasi menurut (Yuliandra & Wulan, 2018) digunakan sebagai tempat penyimpanan data dan informasi, sebagai alat pemrosesan data dan menyediakan laporan mengenai data atau informasi yang disimpan di *database*. Pendapat lain (Sitinjak et al., 2020) *database* merupakan sebuah sistem penyimpanan yang menyimpan kumpulan informasi yang disusun secara terstruktur dengan tujuan memudahkan dalam pengaksesan.

Dari pendapat para ahli diatas dapat disimpulkan pengertian dari *database* merupakan sebuah tempat yang dirancang untuk menyimpan data-data dan informasi yang dirancang khusus agar memudahkan pengguna untuk mendapatkan informasi.

Sebuah *database* terbentuk dari beberapa gabungan *file*, *file* dalam *database* dapat dikategorikan dalam beberapa tipe (Yuliandra & Wulan, 2018) yaitu ;

1. *Master file*, *file* ini memiliki tipe *file* induk acuan yang penyimpanannya relatif statis (nilai tidak berubah) dan *file* induk dinamik (dapat berubah).
2. *Transaction file*, digunakan untuk mencatat data hasil transaksi yang memuat tentang tanggal dilakukan transaksi.
3. *Report file*, berisi informasi yang akan ditampilkan.
4. *History file*, berisi riwayat data yang sudah tidak aktif, tetapi perlu disimpan apabila diperlukan diwaktu yang akan datang.

5. *Backup file*, Salinan *file* yang masih aktif di *database* pada saat tertentu.
6. *Working file*, dibuat oleh suatu proses program sementara untuk menghemat pemakaian memori selama proses apabila memori komputer tidak memadai.

2.2. 7. PHP (*Hypertext Preprocessing*)

PHP merupakan bagian terpenting dalam membuat *website* yang dinamis, hal tersebut karena dalam PHP terhadap *script* yang berisi kode-kode untuk membuat web (Putra & Nita, 2019). Selanjutnya (Nirsal et al., 2020) berpendapat bahwa PHP ialah bahasa pemrograman yang bekerja dalam *web server*, yang artinya *script* PHP harus disimpan di dalam *server* dan dieksekusi atau proses dalam *server* tersebut, penggunaan PHP dapat membuat *website* yang dibangun lebih dinamis dan interaktif.

Maka berdasarkan pendapat para ahli di atas PHP merupakan berupa *script* yang berisi kode-kode dalam pembuatan *website* dan juga salah satu bagian terpenting dalam pembuatan *website* yang dinamis dan interaktif. PHP juga berjalan atau beroperasi menggunakan *webserver*.

2.2. 8. MySQL

MySQL merupakan aplikasi yang digunakan untuk mengelola basis data yang banyak digunakan untuk membuat aplikasi yang membutuhkan *database* (Sitinjak et al., 2020). Dikutip dari jurnal (Putra & Nita, 2019) *MySQL* merupakan bahasa komputer atau bahasa pemrograman yang berfokus untuk pembuatan *database* atau penyimpanan data. Sedangkan menurut (Putra & Nita, 2019) *MySQL* merupakan bahasa terstruktur yang digunakan secara khusus mengelola *database*, juga

merupakan sistem manajemen bersifat *relational*, berarti data yang dikelola di *database* diletakkan di dalam beberapa tabel terpisah sehingga manipulasi data dapat dilakukan dengan cepat.

Dapat disimpulkan bahwa *MySQL* merupakan sebuah aplikasi yang digunakan sebagai alat dalam membuat *database* yang diletakkan dalam susunan beberapa tabel untuk mempermudah dalam melakukan pengerjaan *database*.

Dikutip dari (Trisianto, 2018) *MySQL* memiliki beberapa kelebihan yaitu

1. *Portability MySQL* dapat berjalan dengan stabil diberbagai sistem operasi seperti *windows, linux, FreeBSD, solaris*, dan sebagainya.
2. *Open source*, didistribusikan secara gratis dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. *Multi user*, dapat digunakan banyak pengguna dalam waktu yang bersamaan tanpa ada masalah atau konflik.
4. *Performance turning*, memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain memproses lebih banyak SQL dalam satuan waktu.
5. *Coloumn types*, memiliki tipe kolom yang kompleks, seperti *integer, double, char, text, date* dan lainnya.
6. *Command and function*, memiliki operator dan fungsi secara penuh dan mendukung.

7. *Security*, memiliki banyak lapisan keamanan seperti *level subnetmask*, nama *host*, dan ijin akses pengguna dengan sistem perijinan yang mendetail dan *password* yang terenkripsi.
8. *Scability and limits*, mampu menangani *database* dalam skala besar dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris, juga batas indeks mencapai 32 indeks pada tiap tabel.
9. *Connectivity*, dapat dilakukan koneksi dengan *clients* menggunakan *protocol* TCP/IP, *unix socket* (UNIX), atau *named pipes* (NT).
10. *Interface*, memiliki antar muka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*application programming interface*).
11. *Clients and tools*, dilengkapi dengan beberapa tools yang dapat digunakan untuk administrasi *database* pada setiap *tool* yang ada disertsakan petunjuk *online*.
12. Struktur tabel, lebih fleksibel dalam menangani *ALTER TABLE*, dibanding *database* lainnya.

Sedangkan kelemahan MySQL adalah *feature-creep* MySQL selalu berusaha kompatibel dengan beberapa standar serta berusaha memenuhinya namun jika diungkapkan kenyataannya bahwa fitur-fitur tersebut belum lengkap dan belum sesuai standar.

2.2.9. XAMPP

XAMPP menurut (Nirsal et al., 2020) merupakan singkatan dari X (empat sistem operasi), *Apache*, *MySQL*, *PHP*, dan *Perl*. XAMPP adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket XAMPP sudah terdapat *Apache (web server)*, *MySQL (database)*, *PHP (server side scripting)*, *Perl*, *FTP server*, *PhpMyAdmin* dan berbagai pustaka bantu lainnya. Dengan menginstal XAMPP maka tidak perlu lagi melakukan instalasi dan melakukan konfigurasi *web server Apache*, *PHP*, dan *MySQL* secara manual. XAMPP akan otomatis menginstalasi dan mengonfirmasi.