

2.1. Sistem Informasi

Secara umum sistem informasi adalah sekelompok elemen-elemen dalam suatu organisasi yang saling berintegrasi dengan menggunakan masukan, proses dan keluaran dengan maksud yang sama untuk mencapai suatu tujuan dan dapat digunakan untuk mengambil keputusan yang tepat (Whitten, 2007 dalam Jurnal Rancang Bangun Sistem Informasi Penggajian Karyawan).

Menurut Alter Kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan suatu organisasi. Jika suatu sistem tertentu didefinisikan, maka sering terdapat sejumlah sistem yang lebih kecil, yaitu yang dinamakan subsistem. Bila terus dianalisis, akan sampai pada elemen-elemen dasarnya. Seperti yang dijelaskan oleh Burchdan Strater (1974:9) menyatakan suatu sistem dapat dirumuskan sebagai setiap kumpulan bagian-bagian atau subsistem-subsistem yang disatukan, yang dirancang untuk mencapai suatu tujuan. Setiap bagian dalam organisasi selalu membutuhkan keputusan yang cepat dan tepat, juga membutuhkan bagian-bagian yang lain untuk pembuatan keputusan.

2.2. Sistem Informasi Manajemen (SIM)

Menurut Reymond Mcleod .JR (2008) didefinisikan sebagai suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa. Para pemakai biasanya membentuk suatu *entitas* organisasi formal perusahaan atau subunit di bawahnya. SIM mencerminkan suatu sikap para eksekutif yang menginginkan agar komputer tersedia untuk semua pemecah masalah perusahaan. Ketika SIM berada pada tempatnya dan berfungsi seperti yang diinginkan, SIM dapat membantu manajer dan pemakai lain dan di luar perusahaan mengidentifikasi dan memahami masalah. Kelemahan utama SIM adalah tidak terarah pada kebutuhan khusus dari pemecah masalah perorangan.

SIM tidak menyediakan secara tepat informasi yang diperlukan untuk memecahkan masalah setelah masalah itu diidentifikasi dan diketahui (Jr, 1998).

Menurut James O'Brien (2005,h.443) Jenis awal dari sistem informasi yang dikembangkan untuk mendukung pengambilan keputusan manajerial. SIM menghasilkan produk informasi yang mendukung banyak kebutuhan pengambilan keputusan harian dari para manajer dan praktisi bisnis.

2.3. Pelayanan Perizinan

Pelayanan administrasi pemerintahan atau pelayanan perizinan dapat didefinisikan sebagai segala bentuk jasa pelayanan yang ada pada perinsipnya menjadi tanggung jawab dan dilaksanakan oleh Instansi Pemerintah di Pusat, di Daerah, dan di lingkungan Badan Usaha Milik Negara atau Badan Usaha Milik Daerah, baik dalam rangkalah upaya pemenuhan kebutuhan masyarakat maupun dalam rangka pelaksanaan ketentuan peraturan perundang-undangan, yang bentuk produk pelayanannya adalah izin atau warkat (Ratminto Winarsih, 2005).

2.4. *Unified Modeling Language* (UML)

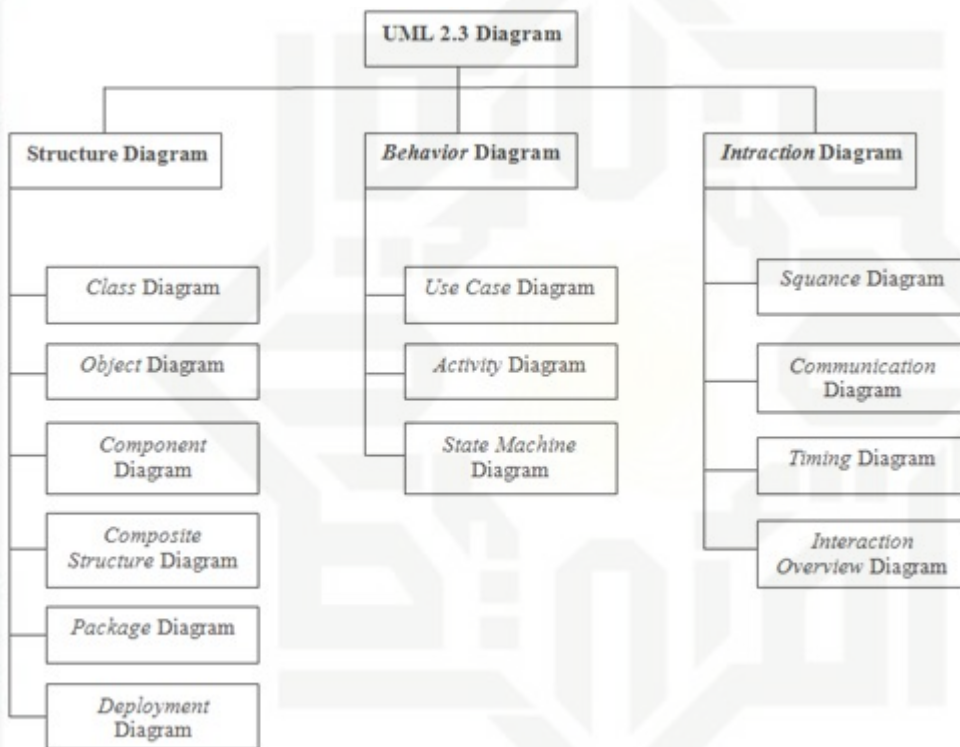
Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasan yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karna itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang (Rosa dan Shalahuddin, 2013).

Pada perkembangan perangkat teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan

menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan permodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa dan Shalahuddin, 2013).

2.4.1 Diagram UML

Menurut (Rosa & Shalahuddin, 2013)UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini.



Gambar 2.1 Diagram UML(Rosa dan Shalahuddin, 2013)

Berikut penjelasan singkat dari pembagian kategori tersebut.

1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.4.2 *Class Diagram*

Menurut (Rosa dan Shalahuddin, 2013) diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dinuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau programmer dapat membuat kelas-kelas didalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas *Main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *usecase* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *usecase*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat dari basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multivalued pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggungjawabkan atau tetap ada di dalam perancangan kelas.



Hal terpenting adalah kolom telpon tetap tetap dapat diakses dari perancangan kelas. Kelas data biasanya adalah kelas yang terkait dengan

pengaksesan tabel pada basis data sesuai dengan nama kelasnya, misalnya kelas Anggota maka akan digunakan untuk mengakses tabel yang menyimpan data anggota.



Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas (*utility class*) seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Berikut adalah simbol-simbol yang ada pada *class diagram*:

Tabel 2.1 Simbol-simbol *class diagram*

Simbol	Deskripsi
	Kelas pada struktur sistem
	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 berarah/ <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)

Tabel 2.1 Simbol-simbol *class diagram* (Lanjutan)

Simbol	Deskripsi
	Relasi antar kelas dengan makna keberuntungan antar kelas
	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

(Sumber : Rosa A. S, 2013)

2.4.3 Use Case Diagram

Menurut (Rosa dan Shalahudin, 2013) *Usecase* atau diagram *usecase* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penanaman pada *usecase* adalah nama didefinisikan menjadi sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *usecase* yaitu pendefinisian apa yang disebut aktor dan *usecase*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Usecase* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *usecase*:

Tabel 2.2 Simbol-simbol *usecase diagram*

Simbol	Deskripsi
<i>Usecase</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya

Tabel 2.3 Simbol-simbol *Usecase Diagram* (Lanjutan)

- Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Simbol	Deskripsi
<i>Usecase</i>	dinyatakan dengan menggunakan kata kerja di awal frase nama <i>usecase</i> .
Aktor/ <i>actor</i> Nama <i>actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Sehingga jadi simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi/ <i>association</i>	Komunikasi antara faktor dan <i>usecase</i> yang berpartisipasi pada <i>usecase</i> memiliki interaksi dengan aktor.
Ekstensi/ <i>extend</i> <<extend>>	Relasi <i>usecase</i> tambahan ke sebuah <i>usecase</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>usecase</i> tambahan itu, mirip <i>inheritance</i> pada pemograman beorientasi objek, biasanya <i>usecase</i> tambahan memiliki nama depan yang sama dengan <i>usecase</i> yang ditambahkan.
	<<extend>> <<extend>

Tabel 2.3 Simbol-Simbol *Usecase Diagram* (Lanjutan)

Simbol	Deskripsi
	<<extend>> Arah panah mengarah pada <i>usecase</i> yang ditambahkan; biasanya <i>usecase</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>usecase</i> yang menjadi induknya.
Generalisasi/ <i>generalisation</i>	Hubungan generalisasi dan spsialisasi (umum-khusus) antara dua buah

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

	<p><i>usecase</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p> <p>Arah panah mengarah pada <i>usecase</i> yang menjadi generalisasinya (umum)</p>
Menggunakan/Include/uses	Relasi <i>usecase</i> tambahan ke sebuah

Tabel 2.2 Simbol-simbol *Usecase Diagram* (Lanjutan)

Simbol	Deskripsi
Menggunakan/Include/uses	<p><i>Usecase</i> yang ditambahkan memerlukan <i>usecase</i> untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>usecase</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>Include</i> di <i>usecase</i>:</p> <ol style="list-style-type: none"> 1. <i>Include</i> berarti <i>usecase</i> yang ditambahkan dan akan selalu di panggil saat <i>usecase</i> tambahan dijalankan, misalkan pada kasus berikut: <p><<Include>></p> <ol style="list-style-type: none"> 2. <i>Include</i> berarti <i>usecase</i> yang tambahan akan selalu melakukan pengecekan apakah <i>usecase</i> yang ditambahkan telah dijalankan sebelum <i>usecase</i> tambahan dijalankan, misal pada kasus berikut :

© Hak Cipta Dilindungi Undang-Undang	<<Include>>
--------------------------------------	-------------

Tabel 2.2 Simbol-simbol *Usecase Diagram* (Lanjutan)

Simbol	Deskripsi
Menggunakan/ <i>Include/uses</i>	Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

2.4.4 *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

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.

Dalam pembuatan *activity diagram*, penggunaan dan penempatan simbol-simbol harus sesuai dengan deskripsi dari simbol itu sendiri, agar perancangan

sistem tergambar dengan jelas. Deskripsi Simbol-simbol *activity diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 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 yang biasanya diawali dengan kata kerja.
Percabangan/ <i>decicion</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabunga/ <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
Atau	

(Sumber: Rosa A. S, 2013)

2.4.5 *Sequance Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima

antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *usecase* beserta metode metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *usecase* (Rosa A. S, 2013).

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *usecase* yang memiliki proses sendiri atau yang penting semua *usecase* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *usecase* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Penggunaan simbol-simbol *sequence diagram* memiliki deskripsi yang berbeda-beda. Penjelasan simbol pada *sequence diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor Nama aktor Atau Tanpa waktu aktif	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Garis hidup/ <i>lifeline</i> -----	Menyatakan kehidupan suatu objek
Objek	Menyatakan objek yang beriteraksi pesan

Tabel 2.4 Simbol-simbol *Sequence Diagram* (Lanjutan)

Simbol	Deskripsi
Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya:

<p>© Hak cipta dimiliki UIN Suska Riau</p>	<p>1. <i>Login()</i>, 2. Cek status <i>login()</i>, 3. <i>open()</i></p> <p>Maka cek status <i>login()</i> dan <i>open()</i> dilakukan di dalam metode <i>login()</i>, Aktor tidak memiliki waktu aktif.</p>
<p>Pesan tipe <i>create</i> <<create>></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i> 1: nama_metode()</p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada apada objek lain atau dirinya sendiri</p> <p>1. Nama_metode()</p> <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada</p>

Tabel 2.4 Simbol-simbol *Sequence Diagram* (Lanjutan)

Simbol	Deskripsi
<p>Pesan tipe <i>call</i> 1 : nama_metode()</p>	<p>Pada diagram kelas sesuai dengan kelas objek berinteraksi.</p>
<p>Pesan tipe <i>send</i> 1 : masukan</p>	<p>Menyatakan bahwa suatu objek megirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim .</p>
<p>Pesan tipe <i>return</i> 1 : keluaran</p>	<p>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.</p>

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

<p>Pesan <i>tipe destroy</i></p> <pre><<destroy>></pre>	<p>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>.</p>
---	---

(Sumber: Rosa A.S, 2013)

2.5 Model Spiral

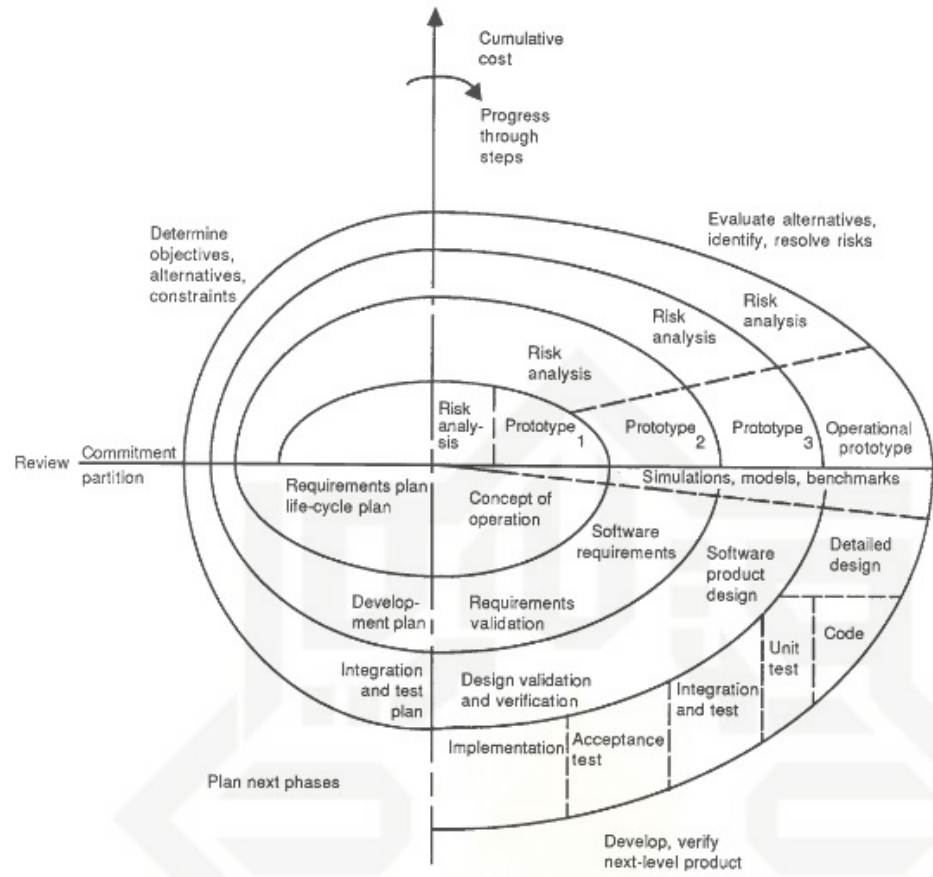
Model spiral merupakan suatu model proses perangkat lunak evolusioner yang menggabungkan pendekatan *prototyping* yang bersifat iteratif dengan aspek sistematis dan terkendali yang kita jumpai pada model air terjun (*waterfall*). Model spiral menyediakan kemungkinan untuk pengembangan perangkat lunak yang cepat, yang bergerak dari versi perangkat lunak yang kurang lengkap menjadi versi yang semakin lengkap (Pressman, 2010).

Menggunakan model spiral, perangkat lunak dikembangkan mengikuti sejumlah peluncuran produk yang bersifat evolusioner. Selama tahap yang lebih awal, produk atau perangkat lunak yang diluncurkan mungkin berupa sebuah model atau suatu *prototipe*. Pada langkah- langkah iterasi berikutnya versi- versi perangkat lunak yang semakin lengkap akan dihasilkan.

Hak cipta milik UIN Suska Riau

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 2.2 Tahapan Pada Model Proses Spiral (Boehm, 1988)

Pada Gambar 2.2 adalah proses spiral, yang setiap *loop* mewakili satu *fase* dari *software process*. *Loop* paling dalam berfokus pada kelayakan dari sistem, *loop* selanjutnya tentang definisi dari kebutuhan, *loop* berikutnya berkaitan dengan desain sistem dan seterusnya. Setiap *loop* dibagi menjadi beberapa sektor:

1. *Objective settings* (menentukan tujuan): menentukan tujuan dari fase yang ditentukan. Batasan-batasan pada proses dan produk sudah diketahui. Perencanaan sudah disiapkan, resiko dari proyek sudah diketahui, alternatif strategi sudah disiapkan berdasarkan resiko-resiko yang diketahui dan sudah direncanakan.
2. *Risk assessment and reduction* (penanganan dan pengurangan resiko): setiap resiko dianalisis secara detil pada sektor ini. Langkah-langkah penanganan dilakukan, misalnya membuat *prototype* untuk mengetahui ketidakcocokan kebutuhan.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. *Development and Validation* (pembangunan dan pengujian): Setelah evaluasi resiko, maka model pengembangan sistem dipilih. Misalnya jika resiko *user interface* dominan, maka membuat *prototype user interface*. Jika bagian keamanan yang bermasalah, maka menggunakan model formal dengan perhitungan matematis, dan jika masalahnya adalah integrasi sistem model *waterfall* lebih cocok.

4. *Planning*: Proyek dievaluasi atau ditinjau-ulang dan diputuskan untuk terus ke fase *loop* selanjutnya atau tidak. Jika melanjutkan ke fase berikutnya rencana untuk *loop* selanjutnya.

Model spiral dibagi menjadi sejumlah aktivitas kerangka kerja yang didefinisikan oleh tim pengembang perangkat lunak. Saat proses yang bersifat evolusioner di atas dimulai, tim perangkat lunak memulai melakukan aktivitas-aktivitas yang diimplikasikan oleh jalur melingkar di sekitar bagian dalam spiral, sesuai dengan arah putaran jam, dimulai dari bagian tengah. Titik-titik pengukuran (*milestone*) suatu kombinasi dari produk - produk kerja dan kondisi - kondisi yang dipelihara di sepanjang lintasan spiral dicatat setiap saat langkah - langkah evolusioner dilewati.

2.6 *Website*

Menurut Nur Marlina dan Dimas Sasongko (2010), bahwa *website* merupakan sebuah halaman berisi informasi yang dapat dilihat jika komputer terkoneksi dengan internet. Dengan adanya *website*, semua orang di dunia bisa mendapatkan dan mengelola informasi dengan berbagai sumber yang tersedia di internet. *Website* sendiri saat ini bisa memuat berbagai macam media, mulai dari teks, gambar, suara bahkan video.

2.7 *Hyper Text Markup Language (HTML)*

HTML ini merupakan bahasa standar yang digunakan oleh protokol *Hyper Text Transfer Protocol* (HTTP) (Sano, 2005). HTML digunakan untuk membuat

halaman *web*. Sebuah file dokumen yang ditulis dalam format HTML akan dibaca dan diterjemahkan oleh *web browser* untuk kemudian disajikan dalam bentuk *web*.

2.8 PHP Hypertext Preprocessor (PHP)

Menurut Ramadhani Dkk. (2013), PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs *web* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Contoh terkenal dari aplikasi PHP adalah forum (phpBB) dan MediaWiki (*software* di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla, Postnuke, Xaraya, dan lain-lain.

2.9 Dinas Perindustrian dan Perdagangan Kota Pekanbaru

Awal terbentuknya Dinas Perindustrian dan Perdagangan Kota Pekanbaru merupakan perpaduan antara Departemen Perdagangan Kota Madya Pekanbaru dan Departemen Perindustrian Kota Madya Pekanbaru. Secara kronologis perkembangannya dapat diuraikan antara lain; Tahun 1981: Departemen Perdagangan Kota Madya Pekanbaru, Tahun 1981: Departemen Perindustrian Kota Madya Pekanbaru, Tahun 1996: Bergabunglah Departemen Perindustrian dengan Departemen Perdagangan Departemen Perindustrian dan Perdagangan Kota Madya Pekanbaru.

Pada Tahun 2001 Pemerintah Kota Pekanbaru berdasarkan [Peraturan Daerah Kota Pekanbaru Nomor 7 Tahun 2001](#) Departemen Perindustrian dan Perdagangan Kota Madya Pekanbaru berubah menjadi Dinas Perindustrian dan Perdagangan Kota Pekanbaru. Perubahan ini terjadi bukan dari namanya saja tetapi juga mengalami perubahan terhadap susunan organisasi dan tugas-tugas pokok pada dinas tersebut.

Visi dari Dinas Perdagangan dan Perindustrian Kota Pekanbaru adalah Sesuai dengan fungsinya Visi Dinas Perindustrian dan Perdagangan Kota Pekanbaru adalah “Terwujudnya Pusat Perdagangan Dan Jasa Yang Didukung Oleh Industri Yang Mapan, Guna Menunjang Ekonomi Kerakyatan”.

Tabel 2.5 Misi Dinas Perdagangan dan Perindustrian Kota Pekanbaru

No.	Misi
1.	Menciptakan industri yang kondusif dan distribusi barang yang merata
2.	Menumbuhkembangkan industri dan perdagangan yang berwawasan lingkungan.
3.	Meningkatkan mutu produk industri yang mempunyai daya saing dan bertanggung jawab.
4.	Memanfaatkan sumber daya yang ada dan meningkatkan kualitas dan profesionalisme dibidang industri dan perdagangan.
5.	Menyediakan informasi industri dan perdagangan yang akurat.
6.	Meningkatkan pembinaan dan kerjasama dalam pengembangan pasar, distribusi, promosi peningkatan penggunaan produk dalam negeri dan pengawasan barang beredar/jasa serta perlindungan konsumen.

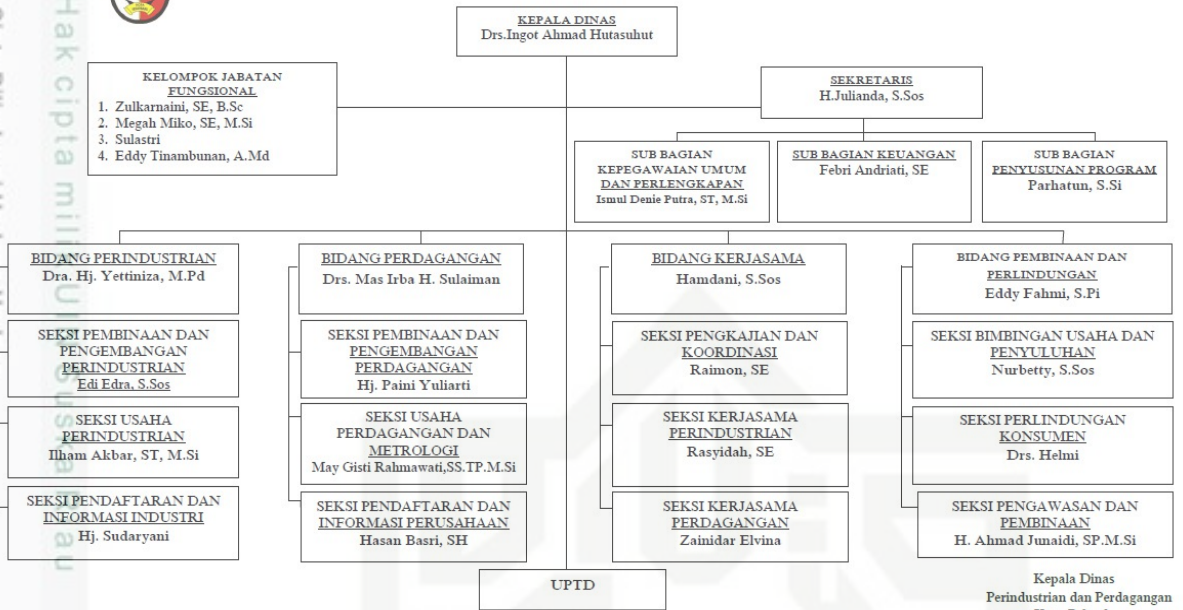
2.10 Struktur Organisasi

Struktur organisasi dari Dinas Perindustrian dan Perdagangan Kota Pekanbaru dapat dilihat pada Gambar 2.3.

- Hak Cipta Dilindungi Undang-Undang**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



STRUKTUR ORGANISASI DINAS PERINDUSTRIAN DAN PERDAGANGAN KOTA PEKANBARU



Dasar :

- PP Nomor 41 tahun 2007 tentang Organisasi Perangkat Daerah
- Perda nomor 8 tahun 2008 tanggal 31 Juli 2008 tentang Pembentukan Susunan Organisasi, Kedudukan & Tugas Pokok Dinas – Dinas di Lingkungan Kota Pekanbaru

Drs. Ingot Ahmad Hutasuht
Pembina Tk.I
NIP. 19710926 199101 1 001

Gambar 2.3: Struktur Organisasi

(Sumber: Dinas Perdagangan dan Perindustrian Kota Pekanbaru)