

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Pengertian sistem menurut beberapa ahli yaitu, Menurut (Tata Sutabri, 2012) pada buku Analisis Sistem Informasi, pada dasarnya sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Menurut Gelinas dan (Dull, 2012), Sistem merupakan seperangkat elemen yang saling bergantung yang bersama-sama mencapai tujuan tertentu. Dimana sistem harus memiliki organisasi, hubungan timbal balik, integrasi dan tujuan pokok. Dengan demikian dapat disimpulkan bahwa sistem merupakan seperangkat elemen yang saling berhubungan yang bersama-sama mencapai suatu tujuan tertentu dalam proses yang teratur yang dapat mendukung sistem yang lebih besar dan saling memiliki ketergantungan untuk mencapai tujuan tertentu.

2.2 Informasi

Menurut (McLeod dikutip oleh Yakub, 2012) pada buku Pengertian Sistem Informasi, Informasi adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Sedangkan Menurut (Tata Sutabri, 2012) pada buku Analisis Sistem Informasi, Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

2.3 Sistem Informasi

Menurut (Satzinger, Jackson, dan Burd, 2012), Sistem informasi merupakan kumpulan dari komponen-komponen yang mengumpulkan, memproses, menyimpan, dan menyediakan output dari setiap informasi yang dibutuhkan dalam proses bisnis serta aplikasi yang digunakan melalui perangkat lunak, database dan bahkan proses manual yang terkait.

Menurut (Stair and Reynolds, 2012), Sistem Informasi adalah suatu sekumpulan elemen atau komponen berupa orang, prosedur, database dan alat yang saling terkait untuk memproses, menyimpan serta menghasilkan informasi untuk mencapai suatu tujuan (goal). Menurut Gelinas dan (Dull, 2012) Sistem Informasi adalah sistem yang di buat secara umum berdasarkan seperangkat komputer dan komponen manual yang dapat dikumpulkan, disimpan dan diolah untuk menyediakan output kepada user. Dengan demikian dapat disimpulkan bahwa sistem informasi adalah suatu kombinasi modul yang terorganisir yang berasal dari komponen-komponen yang terkait dengan hardware, software, people dan network berdasarkan seperangkat komputer dan menghasilkan informasi untuk mencapai tujuan.

2.4 Pemograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai komponen objek yang berisi data dan operasi yang diberlakukan terhadapnya (Rosa, 2016). Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metodologi berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas, yang meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemograman berorientasi objek dan pengujian berorientasi objek. Keuntungan menggunakan metodologi pemograman berorientasi objek adalah sebagai berikut :

- a. Meningkatkan produktivitas, karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).
- b. Kecepatan pengembangan, karena sistem yang dibangun baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
- c. Kemudahan pemeliharaan, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
- d. Adanya konsistensi, karena sifat pewarisan dan pengurangan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

- e. Meningkatkan kualitas perangkat lunak, karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.5 Konsep Dasar Berorientasi Objek

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman dan pengujian perangkat lunak (Rosa, 2016). Beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek adalah sebagai berikut :

- a. Kelas (*class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama dan memiliki sifat (atribut). Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

- b. Objek (*object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur dan hal-hal lainnya yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat berpengaruh pada status objeknya.

- c. Metode (*method*)

Operasi atau metode sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi.

- d. Atribut (*attribute*)

Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas.

- e. Abstraksi (*abstraction*)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

- f. Enkapulasi (*encapsulation*)

Pembungkusan atribut dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

g. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.

h. Antarmuka (*interface*)

Antarmuka sangat mirip dengan kelas, akan tetapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

i. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

j. Generalisasi dan Spealisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

k. Komunikasi Antar Objek

Komunikasi antar objek dilakukan lewat pesan yang dikirim dari satu objek ke objek lainnya.

l. Poliformisme (*polymorphism*)

Kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

m. *Package*

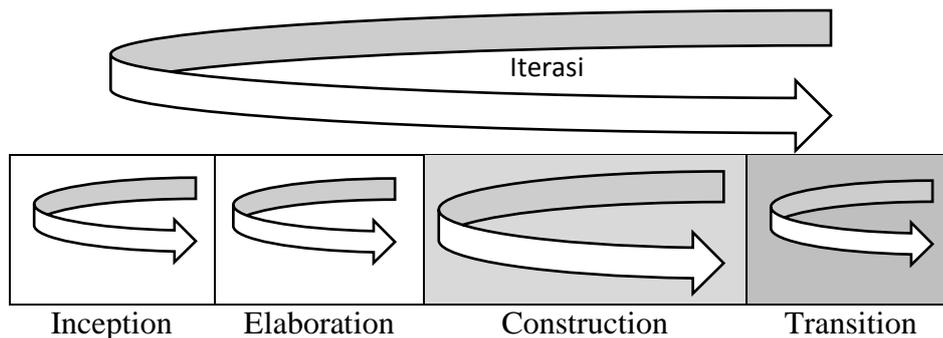
Merupakan sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.6 Metode Pengembangan Perangkat Lunak

Unified Process atau dikenal juga dengan proses iteratif dan inkremental merupakan sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan inkremental (bertahap dengan proses menaik). Iteratif bisa dilakukan di dalam setiap tahap atau iteratif tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang inkremental, dimana

setiap iterasi akan memperbaiki iterasi berikutnya (Rosa, 2016). Salah satu *Unified Process* yang terkenal adalah RUP (*Rational Unified Process*).

RUP adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang, fokus pada arsitektur, lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik dan penstrukturan yang baik. RUP memiliki empat buah tahap fase, yaitu seperti pada Gambar 2.1.



Gambar 2.1 Alur Hidup RUP (Sumber : Rosa, 2016)

a. *Inception* (permulaan)

Tahap ini lebih pada memodelkan bisnis yang dibutuhkan dan mendefinisikan kebutuhan akan sistem yang akan dibuat. Tahap yang dibutuhkan pada permulaan ini adalah :

1. Memahami ruang lingkup dari proyek (termasuk biaya, waktu, kebutuhan, resiko dan lainnya).
2. Membangun kasus bisnis yang dibutuhkan.

Hasil yang diharapkan pada tahap ini adalah memenuhi *lifecycle objective milestone* (batas/tonggak objektif dari siklus) dengan kriteria berikut :

1. Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya dan perkiraan jadwal.
2. Kebutuhan dimengerti dengan pasti dan sejalan dengan kasus primer yang dibutuhkan.
3. Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, risiko dan proses pengembangan.
4. Ruang lingkup purwarupa (*prototype*) yang akan dikembangkan.

5. Membangun garis dasar dengan membandingkan perencanaan aktual dengan perencanaan yang direncanakan.

Jika pada akhir tahap ini target yang diinginkan tidak dicapai maka dapat dibatalkan atau diulang kembali setelah dirancang ulang agar kriteria yang diinginkan dapat dicapai.

b. *Elaboration* (perluasan atau perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*). Hasil yang diharapkan pada tahap ini adalah memenuhi *lifecycle objective milestone* (batas/tonggak objektif dari siklus) dengan kriteria berikut :

1. Model kasus yang digunakan (*use case*) dimana kasus dan aktor yang terlihat telah didefinisikan dan sebagian besar kasus harus dikembangkan.
2. Deskripsi dari arsitektur perangkat lunak telah dibuat.
3. Rancangan arsitektur yang dapat diimplementasikan dan mengimplementasikan *use case*.
4. Kasus bisnis atau proses bisnis dan daftar resiko yang sudah mengalami perbaikan.
5. Rencana pengembangan untuk seluruh proyek telah dibuat.
6. Purwarupa (*prototype*) yang dapat didemonstrasikan untuk mengurangi setiap resiko teknis yang diidentifikasi.

Jika pada akhir tahap ini target yang diinginkan tidak dicapai, maka dapat dibatalkan atau diulang kembali.

c. *Construction* (konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak atau kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

d. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau inisialisasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem.

2.7 Alat Bantu Perancangan Sistem

2.7.1 UML

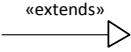
Menurut Rosa (2016), banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak yang sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

2.7.2 Usecase

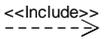
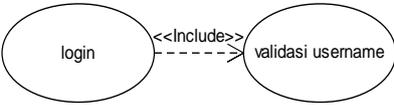
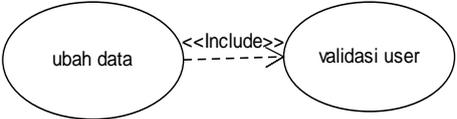
Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami (Rosa, 2016). Ada dua hal utama pada *use case* yaitu pendefinisian apa yang dibuat aktor dan *use case*.

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

Tabel 2.1 Simbol *Use Case* Diagram

Keterangan	Simbol	Deskripsi
Use Case		<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>use case</i></p>
Aktor		<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
Asosiasi		<p>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>
Ekstensi		<p>Relasi use case 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 pemograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan use case yang ditambahkan,</p> <div style="text-align: center;">  <pre> graph LR A((validasi username)) -- «extends» --> B((validasi user)) C((validasi sidik jari)) -- «extends» --> B </pre> </div> <p>misal</p> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>

Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)

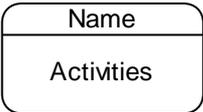
Keterangan	Simbol	Deskripsi
Generalisasi		<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
Menggunakan <i>include/use</i>	 	<p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <p>a. Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut :</p>  <p>b. Include 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, misal pada kasus berikut :</p>  <p>Ke dua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

2.7.1.1 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa, 2016) :

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Tabel 2.2 Simbol Diagram Aktivitas

Keterangan	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		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.7.1.2 Sequence Diagram

Sequence Diagram adalah salah satu dari diagram - diagram yang ada pada UML, *sequence diagram* ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem (Rosa, 2016). Berikut komponen - komponen yang ada pada *sequence diagram* :

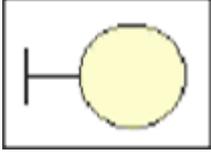
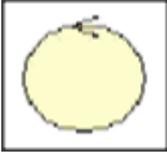
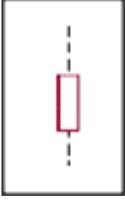
- a. Object - adalah komponen berbentuk kotak yang mewakili sebuah class atau object. Mereka mendemonstrasikan bagaimana sebuah object berperilaku pada sebuah system.
- b. Activation boxes - adalah komponen yang berbentuk persegi panjang yang menggambarkan waktu yang diperlukan sebuah object untuk menyelesaikan tugas. Lebih lama waktu yang diperlukan, maka activation boxes akan lebih panjang.
- c. Actors - adalah komponen yang berbentuk *stick figure*. Komponen yang mewakili seorang pengguna yang berinteraksi dengan system.
- d. Lifeline - adalah komponen yang berbentuk garis putus - putus. Lifeline biasanya memuat kotak yang berisi nama dari sebuah object. Berfungsi menggambarkan aktifitas dari object.

Simbol-simbol yang ada pada diagram kelas adalah seperti pada Tabel 2.3

Tabel 2.3 Simbol Diagram *Sequence*

Simbol	Deskripsi
<p>Actor</p> 	Menggambarkan orang yang berinteraksi dengan sistem

Tabel 2.3 Simbol Diagram *Sequence* (Lanjutan)

Simbol	Deskripsi
<p data-bbox="284 344 472 376">Boundary Class</p> 	<p data-bbox="852 479 1150 544">Menggambarkan sebuah penggambaran dari form.</p>
<p data-bbox="284 714 445 745">Control Class</p> 	<p data-bbox="852 714 1315 779">Menggambarkan penghubung antara boundary dengan tabel.</p>
<p data-bbox="284 936 379 967">Lifeline</p> 	<p data-bbox="852 1070 1315 1135">Menggambarkan tempat mulai dan berakhirnya sebuah pesan.</p>
<p data-bbox="284 1236 448 1267">Line Message</p> 	<p data-bbox="852 1337 1275 1368">Menggambarkan pengiriman pesan.</p>

2.7.1.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa, 2016).

Kelas-kelas yang ada pada struktur sistem, harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

a. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

b. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

c. Kelas yang diambil dari pendefinisian *use case*

Kelas yang menangani fungsi-fungsi yang baru ada diambil dari pendefinisian *use case*.

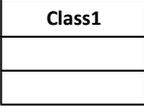
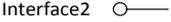
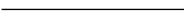
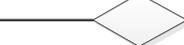
d. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas tersebut 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 seperti koneksi ke basis data, membaca *file* teks dan lainnya.

Dalam mengidentifikasi metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain, sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum, maka sebuah metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah. Simbol-simbol yang ada pada diagram kelas adalah seperti pada Tabel 2.4

Tabel 2.4 Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Natarmuka/<i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
<p>Asosiasi</p> 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Kebergantungan</p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
<p>Agregasi</p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.8 Website

Menurut (Arief, 2011) website adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, animasi, video) didalamnya yang menggunakan protokol HTTP (*Hypertext Transfer Protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*. Menurut (Sibero, 2011) Website adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan *internet*. Website adalah sering juga disebut Web, dapat diartikan suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam

informasi teks, data, gambar diam ataupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, yang dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau hyperlink. Atau definisi website adalah kumpulan dari berbagai macam halaman situs, yang terangkum didalam sebuah domain atau juga subdomain, yang lebih tempatnya berada di dalam WWW (World Wide Web) yang tentunya terdapat di dalam Internet. Halaman website biasanya berupa dokumen yang ditulis dalam format Hyper Text Markup Language (HTML), yang bisa diakses melalui HTTP, HTTP adalah suatu protokol yang menyampaikan berbagai informasi dari server website untuk ditampilkan kepada para user atau pemakai melalui web browser berada pada layar sebuah smartphone dan tentu juga di pengaruhi oleh keterbasan ukuran layar dan jenis layar sentuhnya. Di dalam web mempunyai prinsip dalam sistemnya. yakni *develop once run everywhere* yang dapat berartikan bahwa web ini beroperasi dalam lintas platform untuk sekali pengembangan. jadi kesimpulanya web lebih relatif tidak membutuhkan banyak biaya dalam proses pengembangan jika di bandingkan dengan mobile aplikasi di sebabkan dapat berjalannya dengan baik di semua browser dan platform. web juga dapat di distribusikan tanpa harus mendapatkan approve dari pihak tertentu serta pengguna tidak perlu susah payah mencari diweb. namun dengan demikian bukan berarti web ini tidak memiliki kekurangan. tetap saja web ini mempunyai kekurangan namanya juga sistem pasti tidak lepas dengan kekurangan meskipun tidak membuat rugi pengguna secara financial atau material. dan kekurangan web ini terdapat pada domain yang dimana pengguna web harus memiliki 2 buah domain dan bila ada gangguan atau masalah maka akan rumit dalam memperbaikinya.

2.9 PHP (*Pear Hypertext Preprocessor*)

PHP singkatan dari *Pear Hypertext Preprocessor* yaitu bahasa pemrograman web server-side yang bersifat open source. Menurut Betha Sidik, dalam bukunya yang berjudul *Pemrograman Web Dengan PHP* (Betha Sidik, 2012), menyebutkan bahwa PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman script – script yang membuat dokumen HTML secara on

the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman server side". PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

Keunggulan PHP :

- a. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. Dalam sisi pengembangannya lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
- c. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
- d. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.
- e. PHP memiliki tingkat akses yang lebih cepat.
- f. PHP memiliki tingkat keamanan yang tinggi.
- g. PHP mampu berjalan di beberapa *server* seperti *Apache, Microsoft IIS, PWS* dan *Xitami*.
- h. PHP mendukung akses ke beberapa *database* baik yang bersifat *free/gratis* atau *komersial*, seperti *MySQL, mSQL* dan *Microsoft SQL Server*. PHP bersifat *free/gratis*.

2.10 MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user. Menurut (Betha Sidik, 2012), dalam bukunya yang berjudul Pemrograman Web Dengan PHP, menyebutkan bahwa MySQL merupakan software database yang termasuk paling populer di

lingkungan Linux, kepopuleran ini karena ditunjang karena performansi query dari databasenya yang saat itu bisa dikatakan paling cepat dan jarang bermasalah”.MySQL adalah implementasi dari system manajemen basisdata relasional (RDBMS). MySQL dibuat oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael “Monty”. Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL (*General Public License*) namun tidak boleh menjadikan produk turunan yang bersifat komersial. Pada saat ini MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database yaitu SQL. SQL (*Structured Query Language*) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangkan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan database lebih user-friendly dibandingkan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni. SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemrograman seperti C, dan Delphi.

2.11 Database

Menurut (Sutarman, 2012), *Database* sekumpulan file yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan diantaranya. Menurut (Ladjamudin, 2013), *Database* adalah sekumpulan data store (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya. Dari pengertian diatas penulis menyimpulkan *Database* adalah sekumpulan file yang saling berhubungan yang menyimpan data dan tersimpan dalam sebuah media penyimpanan.

2.12 Customer Relationship Management (CRM)

Menurut (Turban, 2015) *Customer Relationship Management (CRM)* adalah suatu pendekatan pelayanan kepada konsumen yang berfokus pada pembangunan jangka panjang dan hubungan konsumen yang berkelanjutan yang dapat memberikan nilai tambah bagi pelanggan maupun perusahaan. Pada penelitian lain, merekomendasikan untuk meningkatkan layanan alumni dengan aktifitas CRM pada dasarnya bertujuan agar perusahaan dapat mengenali pelanggan secara detail dan melayani mereka sesuai kebutuhannya. Tujuannya adalah menggunakan data dengan lebih baik untuk pengelolaan hubungan dengan pelanggan. Teknologi model CRM menjadi landasan untuk mengetahui perubahan dan penentu keberhasilan organisasi, yaitu melakukan komunikasi yang baik untuk memperoleh pelanggan, pengiriman barang sesuai pesanan pelanggan, dan mempertahankan pelanggan dengan memberikan pelayanan yang baik. Integrasi itu semua tidak cukup hanya dengan mengembangkan konseptual tetapi perlu diubah dengan mode simulasi dinamis. Satu elemen kunci CRM adalah customer touch points merupakan hal penting bagi perusahaan yang berorientasi dan memfokuskan diri pada kebutuhan pasar atau pelanggan saat ini dan dimasa yang akan datang. Karenanya perlu dibuat antar muka antara organisasi dan pelanggan. Seperti perangkat : web, telepon, dan email. Applications adalah perangkat lunak (Software) yang mendukung proses-proses tersebut.

2.13 Penelitian Terkait

Penelitian yang terkait dengan penelitian yang sudah dilakukan sebelumnya adalah seperti yang tertera pada Tabel 2.5.

Tabel 2.5 Penelitian Terkait

No	Judul	Penulis	Masalah	Metode	Hasil
1.	Analisis Perancangan Customer Relationship Management Berbasis Web pada PT ASP JAKARTA	Idris Gautama So, Shella	Proses pelayanan di PT ASP belum memanfaatkan teknologi dan sistem informasi secara maksimal, seperti proses pemasaran kapal, penyampain informasi ketersediaan kapal, informasi pembayaran, update informasi ke pelanggan, serta kritik dan saran dari pelanggan semuanya masih dilakukan dengan via telepon atau fax.	Metode penelitian yang digunakan adalah dengan menganalisis kondisi persaingan dalam industri yang dijalankan perusahaan dengan model lima kekuatan potter san kerangka perumusan strategi, tahap masukan, tahap pencocokkan dan tahap keputusan untuk memilih strategi yang tepat bagi perusahaan.	Berdasarkan hasil analisis dengan menggunakan kerangka perumusan strategi, yaitu dibagi dalam tiga tahap. Tahap masukan dengan matriks EFE, IFE, CPM selanjutnya tahap pencocokkan dengan matriks SWOT, IE dan tahap keputusan dengan matriks QSPM, dapat disimpulkan bahwa strategi penetrasi pasar merupakan strategi yang paling cocok untuk diterapkan oleh PT ASP. Strategi

					penetrasi pasar yang dimaksud adalah dengan menerapkan CRM berbasis web, yang berguna untuk peningkatan pelayanan kepada pelanggan dan pengenalan perusahaan kepada masyarakat luas yang dapat diakses kapan pun dan dimana pun.
2.	Penerapan Customer Relationship Management Berbasis Web (studi kasus pada Sistem Informasi Pemasaran di toko YEN-YEN)	Ovi Dyantina, Mira Afrina, Ali Ibrahim	Proses pemasaran produk masih dilakukan secara konvensional dengan cara pelanggan memberikan informasi mengenai tempat, keistimewaan dan produk ke calon pelanggan lain.	Metode yang digunakan dalam pengembangan sistem menggunakan metode FAST (Framework for the Application of System Thinking) dengan pendekatan Model Driven Development Strategy.	1. Pada sistem informasi pemasaran yang sedang berjalan masih terdapat kelemahan-kelemahan terutama pada proses penyampaian informasi promosi. 2. Dengan menggunakan sistem terkompute

					risasi yang berbasis web maka proses penyimpanan data dapat lebih akurat dan tersimpan rapih, serta proses penyampaian informasi promosi dapat lebih cepat dan respon terhadap penjualan pun akan lebih cepat.
3.	Pemodelan Customer Relationship Management Berbasis Web dan Sms dalam Pengelolaan Lowongan Pkerjaan Bagi Alumni Perguruan Tinggi dengan Framework Zachman	Sri Karnila	Banyaknya sumber daya manusia di usia produktif dengan macam-macam latar belakang pendidikan. Menurut badan pusat statistik sarjana menganggur mengalami peningkatan dari tahun 2009-2011.	Metode penelitian menggunakan Framework Zachman. Zachman mengidentifikasi tingkatan arsitektur yang dimulai dengan tingkat konseptual hingga detail rancangan dan kontruksi sebuah system.	Penelitian ini menghasilkan sebuah model CRM berbasis WEB dan SMS yang mampu menjadi media komunikasi antara alumni dan perusahaan dalam penyediaan informasi tentang pengelolaan lowongan pekerjaan. Dan data yang terintegrasi

					dalam dokumen, pengelolaan lowongan pekerjaan, data sesuai kebutuhan informasi masing-masing bagian, yang menyediakan layanan informasi data alumni.
4.	E-Customer Relationship Management (E-CRM) untuk pemesanan dan penjualan makanan ringan (studi kasus : CV.Dwi Putra Tulang Bawang Barat)	Farid Ali Hanafi, Sri Karnila	CV. Dwi Putra Tulang Bawang Barat merupakan salah satu perusahaan yang bergerak dibidang penjualan dan pemesanan makanan ringan di tulang bawang barat. Proses penjualan dan pemesanan makanan ringan pada CV. Dwi Putra Tulang Bawang Barat sejauh ini masih dilakukan secara konvensional yaitu pembeli datang langsung ke tempat penjualan untuk melakukan pembelian.	Metode yang digunakan dalam penelitian ini adalah metode waterfall.	<ol style="list-style-type: none"> 1. sistem E-CRM lebih mempermudah customer dalam melakukan pemesanan karena dapat dilakukan secara online. 2. sistem yang mempunyai diskon produk sehingga dapat menarik pelanggan baru dan mempertahankan pelanggan lama. 3. mempermudah pihak

					CV. Dwi Putra Tulang Bawang Barat karena mempunyai data stok barang yang akurat dan terkomputerisasi.
5.	Perancangan Aplikasi Berbasis Customer Relationship Management (CRM) untuk IKM Tas Yanri di Bogor	Yan Andriariza	Industri kecil menengah (IKM) terbukti dapat bertahan menghadapi krisis ekonomi yang melanda Indonesia pada tahun 1998, bahkan IKM memberikan kontribusi cukup besar dalam menggerakkan roda ekonomi pasca krisis ekonomi tersebut. IKM Tas Yanria dalam hal pemasaran produk-produk mereka, kalah bersaing dengan produk-produk industri China yang terkenal dengan produk-produknya yang murah.	Metode yang digunakan dalam penelitian ini menggunakan pendekatan kualitatif. Sedangkan jenis penelitian ini adalah penelitian deskriptif dan penelitian tindakan (action research).	Perancangan proses sistem informasi berbasis CRM di IKM Tas Yanri ini dibangun berdasarkan pada proses bisnis yang berjalan di IKM Tas Yanri, dengan dilakukan beberapa pengembangan terhadap proses bisnis tersebut, berdasarkan pada konsep-konsep CRM, sehingga diperoleh model dasar sistem

					informasi CRM tersebut merupakan hasil analisa dari proses bisnis yang berjalan di IKM Tas Yunri, dengan melakukan perbaikan terhadap proses-proses yang ada, dengan melakukan penambahan , pengurangan ataupun modifikasi terhadap proses yang ada.
--	--	--	--	--	--