



BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Pengertian Komputer Berbasis Internet

Menurut Kasemin (2016:8), *Komputer berbasis internet* merupakan hasil dari perkembangan komputer yang dapat menghubungkan banyak orang sekaligus secara serentak dalam waktu bersamaan di dunia maya, terutama bagi orang yang ingin saling menyampaikan pesan kepada siapa saja yang membutuhkannya. Dalam dunia pendidikan sangat di perlukan kecanggihan teknologi dan internet yang ada untuk memanfaatkannya perlu keahlian khusus yang ada..

Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa komputer berbasis website adalah sebuah kemajuan perkembangan komputer dimana komputer saat ini mampu menghubungkan setiap pengguna komputer di seluruh dunia yang gunannya untuk menciptakan sebuah pesan komunikasi terutama dibidang pendidikan.

2.1.2 Pengertian Komputer

Menurut Kadir (2017:2), komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang di lakukan oleh manusia. Meskipun komputer berasal dari kata “komputasi”, komputasi yang memang dilaksanakannya mungkin tidak terlihat secara eksplisit. Ketika orang menggunakan komputer untuk membuat dokumen, berbagai perhitungan yang di lakukan tidak terlihat.

Menurut Kadir (2019:2), Komputer merupakan peralatan elektronik yang umum di jumpai di mana saja dan bahkan dalam bentuk apa saja. Komputer tidak selalu berbentuk *PC* maupun *notebook*, tetapi bisa melekat dalam perangkat mikro gelombang, pembuat kopi, maupun mesin cuci.



Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa komputer adalah elektornik yang mampu membuat pekerjaan manusia lebih mudah atau lebih efisien.

2.1.3 Pengertian Internet

Menurut Sofana (2015:5), *Internet* adalah interkoneksi jaringan komputer skala besar (mirip *WAN*), yang dihubungkan menggunakan protokol khusus. Jadi sebenarnya internet merupakan bagian dari *WAN*. Cakupan internet adalah satu dunia bahkan tidak menutup kemungkinan antarplanet. Koneksi antar jaringan komputer dapat dilakukan berkat dukungan protokol yang khas, yaitu *TCP/IP* (*Transmission Control Protocol/Internet Protocol*).

Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa internet adalah berguna untuk jaringan koneksi untuk dunia bahkan antar planet sekalipun yang di dukung protokol yang khas, yaitu *TCP/IP*.

2.2 Teori Judul

2.2.1 Pengertian Sistem Informasi

Menurut Anggraeni dan Irviani (2017:2), *Sistem Informasi* merupakan suatu kombinasi teratur dari orang-orang, software, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.”

Menurut Kristanto (2018:12), Sebuah *sistem informasi* merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan persngkat keras dan perangkat lunak tersebut.”



Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa sistem informasi adalah sebuah kombinasi dari perangkat-perangkat komputer dengan pengguna, serta jaringan komunikasi yang nantinya akan menghasilkan sebuah data informasi yang dibutuhkan setiap masing-masing pengguna sistem informasi.

2.2.2 Pengertian Website

Menurut Abdulloh (2018:1), *website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang di sediakan melalui jalur koneksi internet sehingga dapat di akses dan dilihat oleh semua orang di seluruh dunia. Halaman website dibuat menggunakan bahasa standar yaitu HTML. Skript HTML ini akan diterjemahkan oleh web browser sehingga dapat ditampilkan dalam bentuk informasi yang dapat di baca oleh semua orang.

Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa website adalah informasi berupa teks, gambar, animasi, suara dan video yang akan di koneksikan ke internet agar semua orang bisa melihat atau mengakses website tersebut.

2.2.3 Pengertian Skripsi

Menurut Zaenal (2015:3), *Skripsi* merupakan bukti kemampuan akademik seorang mahasiswa dalam mengekspresikan karya ilmiahnya, menyelesaikan dan menjawab masalah secara ilmiah, serta dalam menyusun laporan secara sistematis.”.

Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa skripsi adalah sebuah kombinasi dari perangkat-perangkat komputer dengan pengguna, serta jaringan komunikasi yang nantinya akan menghasilkan sebuah data informasi yang dibutuhkan setiap masing-masing pengguna sistem informasi.



2.2.4 Pengertian Pengajuan Judul Skripsi

Menurut penulis *pengajuan judul skripsi* merupakan suatu langkah awal serta konsep pemikiran awal yang dilakukan mahasiswa guna menyelesaikan suatu karya ilmiah tersebut agar sesuai dengan keinginan yang dikehendaki tentunya disesuaikan pada masing-masing teknik pengumpulan data dan para sumber sumber observasi serta berdasarkan penerapan mata kuliah yang terjadi selama masa perkuliahan berlangsung.

2.2.5 Pengertian Fakultas Hukum Universitas Tamansiswa Palembang

Menurut penulis Fakultas Hukum *Universitas Tamansiswa Palembang* merupakan salah satu Fakultas Hukum Universitas swasta terkemuka yang berada di kota Palembang dengan akreditasi dan akademisi yang telah disesuaikan dengan kemajuan teknologi saat ini sehingga mampu memberikan pendidikan yang baik kepada semua peserta didik yang mereka ajarkan.

2.2.6 Pengertian Algoritma Cosine Similarity

Menurut Pratama, Faisal, dan Hanani (2019:23), *Algoritma cosine similarity* yaitu metode perhitungan antara dua buah dokumen yang bertujuan untuk mengetahui tingkat kemiripan antar dokumen tersebut. Perhitungan metode cosine similarity didasarkan pada dua buah vektor yang memiliki jumlah kata pada dua dokumen yang dibandingkan.

Algoritma Cosine Similarity dapat diimplementasikan untuk menghitung nilai kemiripan antar kalimat dan menjadi salah satu teknik untuk mengukur kemiripan teks yang populer. Dalam menghitung cosine similarity, pertama yang dilakukan yaitu melakukan perkalian skalar antara query dengan dokumen kemudian dijumlahkan, setelah itu melakukan perkalian antara panjang dokumen dengan panjang query yang telah dikuadratkan, setelah itu di hitung akar pangkat dua. Selanjutnya hasil perkalian skalar tersebut di bagi dengan hasil perkalian panjang dokumen dan query.



Rumus dapat dilihat sebagai berikut :

$$\text{cosSim}(d_j, q_k) = \frac{\sum_{i=1}^n (td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^n td_{ij}^2 \times \sum_{i=1}^n tq_{ik}^2}}$$

Keterangan:

$\text{cosSim}(d_j, q_k)$: tingkat kesamaan dokumen dengan query tertentu

td_{ij} : term ke-i dalam vektor untuk dokumen ke-j

tq_{ik} : term ke-i dalam vektor untuk query ke-k

n : jumlah *term* yang unik dalam data set

Gambar 2.1 Rumus Algoritma Cosine Similarity

Langkah-langkah perhitungan manual algoritma Cosine Similarity.

1. Ditentukan terlebih dahulu masing-masing query, yaitu query dari jawaban (D), query dari key jawaban (Q) dan gabungan keduanya (Queries)
2. Ketiga query tersebut dihilangkan stoplist atau simbol-simbol yang tidak mempengaruhi penilaian, seperti tanda titik, tanda koma, tanda seru, dan sebagainya
3. Ketiga query tersebut dihilangkan stopwords atau kata-kata umum yang lazim digunakan dalam suatu query, seperti "dan", "jika", "di", "namun", "tetapi", dan sebagainya

2.2.7 Pengertian Metode Rapid Application Development (RAD)

Menurut Utami dan Asnawati (2015:21), *RAD* model proses pengembangan perangkat lunak tambahan yang menekankan siklus perkembangan yang sangat pendek.

Menurut Sukamto dan Shalahuddin (2018:133), *RAD* adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek.



Terdapat tiga fase dalam RAD yang melibatkan penganalisis dan pengguna dalam tahap penilaian, perancangan, dan penerapan. Adapun ketiga fase tersebut adalah requirements planning (perencanaan syarat-syarat), RAD design workshop (workshop desain RAD), dan implementation (implementasi). Sesuai dengan metodologi RAD, berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi.

1. Requirements Planning (Perencanaan Syarat-Syarat)

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan.

2. RAD Design Workshop (Workshop Desain RAD)

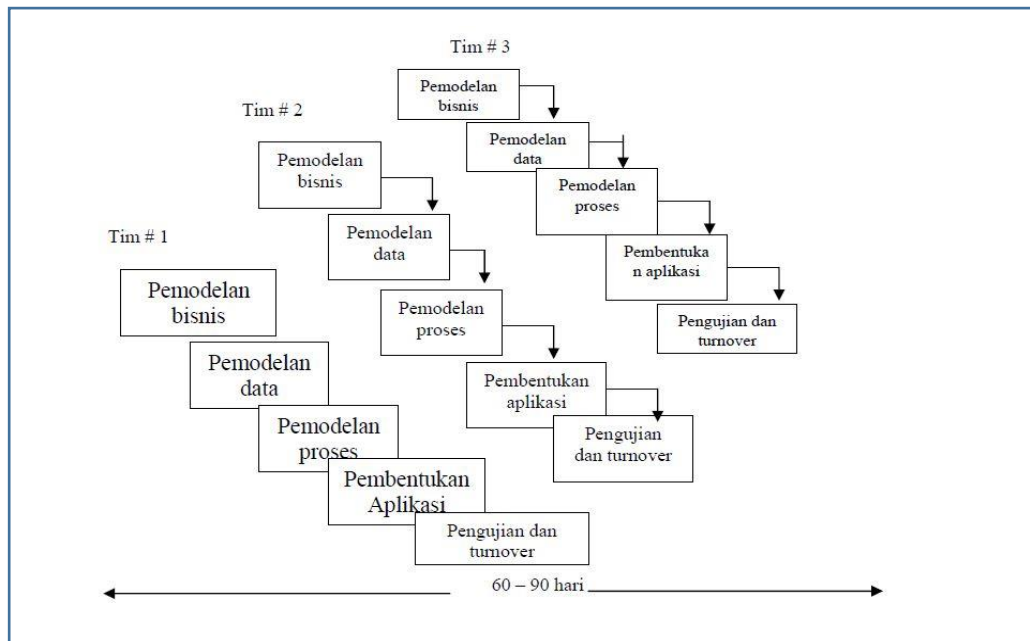
Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. Workshop desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna.

3. Implementation (Implementasi)

Pada fase *implementasi* ini, penganalisis bekerja dengan para pengguna secara intens selama workshop dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi.



Berikut adalah gambar pemodelan metode *RAD* :



Gambar 2.2 Pemodelan Metode *RAD*

Tahapan-tahapan pada metode *RAD*:

1. Pemodelan Bisnis

Pemodelan yang dilakukan untuk memodelkan fungsi bisnis untuk mengetahui informasi apa yang terkait proses bisnis, informasi apa saja yang harus dibuat, siapa yang harus membuat informasi itu, bagaimana alur informasi itu, proses apa saja yang terkait informasi itu.

2. Pemodelan Data

Memodelkan data apa saja yang dibutuhkan berdasarkan pemodelan bisnis dan mendefinisikan atribut-atributnya beserta relasinya dengan data-data yang lain.

3. Pemodelan Proses

Mengimplementasikan fungsi bisnis yang sudah didefinisikan terkait dengan pendefinisian data.



4. Pembentukan Aplikasi

Mengimplementasikan pemodelan proses dan data menjadi program. Model *RAD* sangat menganjurkan pemakaian komponen yang sudah ada jika dimungkinkan.

5. Pengujian dan Turnover

Menguji komponen-komponen yang dibuat. Jika sudah teruji maka tim pengembang komponen dapat beranjak mengembangkan komponen berikutnya.

Berikut ini adalah beberapa kelebihan *Rapid Application Development* menurut (Yurindra, 2017) :

1. Mampu meminimalkan kesalahn-kesalahan dengan menggunakan alat-alat bantuan (CASE tools).
2. Membeli sistem yang baru memungkinkan untuk lebih menghemat biaya ketimbang mengembangkan sendiri.
3. Proses pengiriman menjadi lebih mudah, hal ini dikarenakan proses pembuatan lebih banyak menggunakan potongan-potongan script
4. Mudah diamati karena menggunakan model prototype, sehingga user lebih mengerti akan sistem yang dikembangkan.
5. Lebih fleksibel karena pengembang dapat melakukan proses desain ulang pada saat bersamaan.
6. Bisa mengurangi penulisan kode yang kompleks karena menggunakan wizard.



2.2.8 Pengertian Sistem Informasi Pengajuan Judul Skripsi Pada Fakultas Hukum Universitas Tamansiswa Palembang Berbasis Website Dengan Menggunakan Algoritma Cosine Similarity

Suatu sistem informasi yang digunakan untuk mempermudah antara mahasiswa, kaprodi dan pihak yang terlibat dalam memproses aktivitas pengajuan judul skripsi mahasiswa pada fakultas hukum universitas tamansiswa Palembang secara online.

2.3 Teori Khusus

2.3.1 Pengertian Unified Modelling Language (UML)

Menurut Sukamto dan Shalahuddin (2018:133), *Unified Modelling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

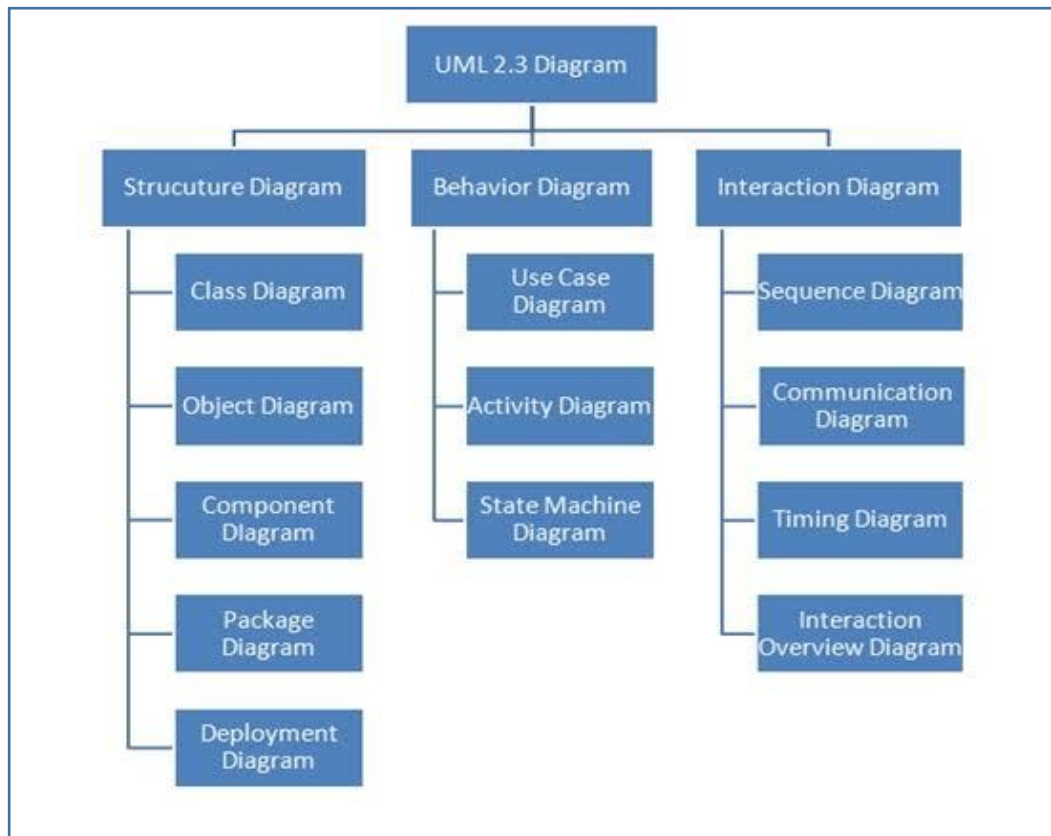
Menurut Mulyani (2016:42), *Unified Modelling Language (UML)* adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk mendokumentasikan dan melakukan spesifikasi pada sistem.

Dari pernyataan yang telah ditemukan oleh pengarang di atas maka penulis menarik kesimpulan bahwa Unified Modelling Language adalah sebuah bahasa pemodelan pembangunan perangkat lunak yang disesuaikan dengan teknologi yang berkembang saat ini. didalam pemodelan ini tentunya dalam proses kerja terdapat bergabai macam diagram yang nantinya menggambarkan struktur sistem yang akan dibangun. pada penelitian ini penulis menggunakan 3 macam model diagram uml yang nantinya digunakan untuk menggambarkan sistem yang ada yaitu: Usecase Diagram, Class Diagram, serta Activity Diagram.



2.3.1.1 Macam-macam Diagram Unified Modelling Language (UML)

Menurut Sukamto dan Shalahuddin (2018:140), *Unified Modelling Language (UML)* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 ksteogri. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini.



Gambar 2.3 Pemodelan Unified Modelling Language

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

- a. Structure Diagram
Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- b. Behavior Diagram
Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.



c. Interaction Diagram

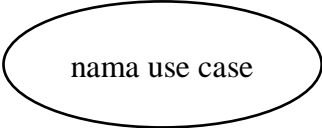
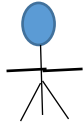
Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.3.2 Pengertian Use Case Diagram

Menurut Mulyani (2018:42), *Use Case Diagram* yaitu diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor. Diagram ini hanya menggambarkan secara global.


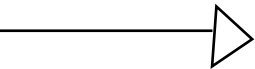


Menurut Sukamto dan Shalahuddin (2018:155), *Use Case Diagram* merupakan pemodelan untuk melakukan (behavior) sistem informasi yang akan dibuat. Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Tabel 2.1 Simbol-simbol Use Case Diagram

Simbol	Deskripsi
<p><i>Use case</i></p>  <p>nama use case</p>	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama Use Case.</p>
<p>aktor / <i>actor</i></p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informaasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi actor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda diawal frase nama actor.</p>



Lanjutan Tabel 2.1 Simbol-simbol Use Case Diagram

Simbol	Deskripsi
asosiasi / <i>association</i> 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
ekstensi / <i>extend</i> <<extend>>	Relasi use case tambahan ke sebuah use case dimana use case tambahan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan.
Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
menggunakan / <i>include / uses</i> <<include>>  <<uses>> 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan.

(Sumber: Rosa dan Salahuddin (2018:156))


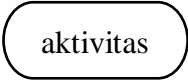
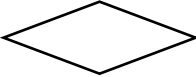



2.3.3 Pengertian Activity Diagram

Menurut Mulyani (2018:55), *Activity Diagram* yaitu diagram yang digunakan untuk menggambarkan alur kerja (aktivitas) pada use case (proses), logika, proses bisnis dan hubungan antara aktor dengan alur – alur kerja use case.

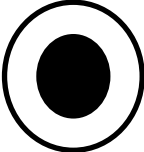
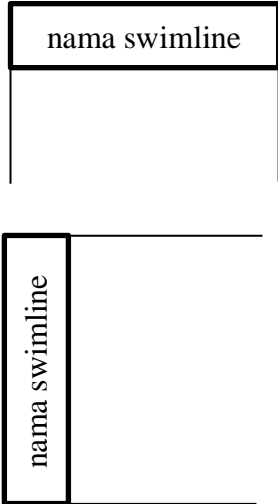
Menurut Sukanto dan Shalahuddin (2018:155), *Activity Diagram* merupakan diagram aktivitas yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari suatu sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Tabel 2.2 Simbol-simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.



Lanjutan Tabel 2.2 Simbol-simbol Activity Diagram

Simbol	Deskripsi
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber: Rosa dan Salahuddin (2018:162))

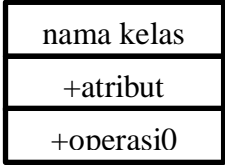
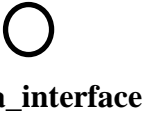

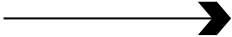
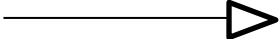
2.3.4 Pengertian Class Diagram

Menurut Sulianta dan Umbara (2015:154), *Class Diagram* merupakan diagram yang harus menjelaskan hubungan apa saja yang terjadi antara suatu objek dengan objek lainnya sehingga terbentuk suatu sistem aplikasi.

Menurut Sukamto dan Shalahuddin (2018:141), *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas yang memiliki apa yang disebut atribut, metode atau operasi.

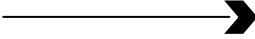
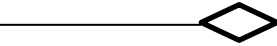


Tabel 2.3 Simbol-simbol Class Diagram

Simbol	Deskripsi
	Kelas pada struktur sistem
	Sama seperti konsep interface dalam pemrograman berorientasi objek
	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
	Relasi antarkelas dengan makna kelas yang satu akan digunakan dengan kelas lain, asosiasi biasanya juga disertai multiplicity.
	Relasi antarkelas dengan makna generalisasi - spesialisasi (umum khusus).



Lanjutan Tabel 2.3 Simbol-simbol Class Diagram

Simbol	Deskripsi
	Relasi antarkelas dengan makna kebergantungan antarkelas.
	Relasi antarkelas dengan makna seua-bagian (whole-part)

(Sumber: Rosa dan Salahuddin (2018:146))


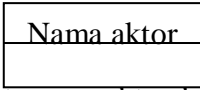

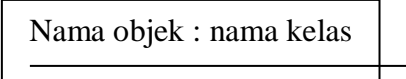

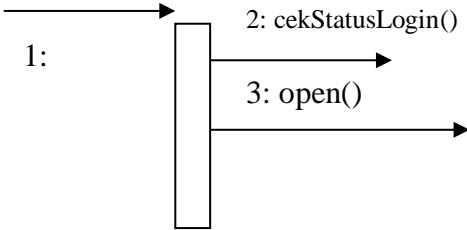
2.3.5 Pengertian *Sequence Diagram*

Rosa dan Shalahuddin (2018:141), menjelaskan bahwa diagram *sequence* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada use case.

Banyaknya diagram *sequence* yang harus digambarkan adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. Berikut simbol-simbol pada *Sequence Diagram* :

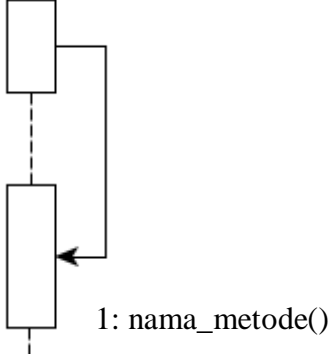


Tabel 2.4 Simbol-simbol pada Sequence Diagram

Simbol	Deskripsi
<p data-bbox="539 465 614 495">Actor</p>  <p data-bbox="523 607 625 627">nama aktor</p> <p data-bbox="549 651 600 680">atau</p>  <p data-bbox="464 763 691 788">tanpa waktu aktif</p>	<p data-bbox="858 432 1369 719">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>
<p data-bbox="443 792 708 824">Garis hidup / <i>lifeline</i></p> 	<p data-bbox="858 831 1318 862">Menyatakan kehidupan suatu objek</p>
<p data-bbox="533 958 616 990">Objek</p> 	<p data-bbox="858 965 1366 1032">Menyatakan objek yang berinteraksi pesan</p>
<p data-bbox="497 1111 654 1142">Waktu aktif</p> 	<p data-bbox="858 1111 1369 1323">Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p data-bbox="858 1585 1366 1688">Maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>



Lanjutan Tabel 2.4 Simbol-simbol pada Sequence Diagram

<p>Pesan tipe call</p> <p style="text-align: center;"><<create>></p> <p style="text-align: center;">—————></p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe send</p> <p style="text-align: center;">1:</p> <p style="text-align: center;">—————></p>	<p>menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p> <p>]</p>
<p>Pesan tipe return</p> <p style="text-align: center;">1: keluaran</p> <p style="text-align: center;">-----></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>

(Sumber: Rosa dan Salahuddin (2018:146))

2.3.6 Black Box Testing

Rosa dan Shalahuddin (2016:275), “Menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.



2.4 Teori Program

2.4.1 Pengertian Codeigniter



Gambar 2.4 Logo Codeigniter

(Sumber: Supono dan Virdiandry Putratama (2019))

Menurut Supono dan Putratama (2016:109), *Codeigniter* adalah aplikasi open source berupa framework dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP.

Menurut Rahmawati (2017:5), *Codeigniter* merupakan framework PHP dengan konsep MVC (*Model, View, Controller*) yang dapat memudahkan developer (pengembang) untuk membuat aplikasi web dengan cepat.

2.4.2 Pengertian XAMPP



Gambar 2.5 Logo XAMPP

(Sumber: Yudhanto dan Prasetyo (2019))

Menurut Yudhanto dan Prasetyo (2019:17), *XAMPP* adalah kompilasi program aplikasi gratis terfavorit di kalangan developer/programmer yang berguna untuk pengembangan website berbasis *PHP* dan *MySQL*.



Menurut Dadan dan Kerendi (2015:28), *XAMPP* adalah salah satu aplikasi web server apache yang terintegrasi dengan mysql dan phpmyadmin. *XAMPP* adalah singkatan dari *X*, *Apache Server*, *MySQL*, dan *phpmyadmin*.

2.4.3 Pengertian PHPMYADMIN



Gambar 2.6 Logo PhpMyAdmin

(Sumber: Nugroho (2019))

Menurut Nugroho (2019:8), phpMyadmin merupakan sebuah program bebas yang berbasis web yang dibuat menggunakan aplikasi *PHP*, tujuan dibuatnya program ini adalah untuk mengakses database *MySQL*, intinya adalah di gunakan untuk menjadi administrator dari server *MySQL*. Dengan adanya program ini akan mempermudah dan mempersingkat kinerja kita, dengan kelebihan-kelebihan yang ada mengakibatkan para pengguna awam tidak harus mampu untuk mengetahui sintaks-sintaks *SQL* dalam pembuatan database dan table.

Menurut Menurut Yudhanto dan Prasetyo (2019:19), *phpMyadmin* adalah aplikasi web untuk mengelola database *MYSQL* dan database *MARIADB* dengan mudah melalui antarmuka (interface) grafis. Aplikasi web ini ditulis menggunakan bahasa pemrograman *PHP*.



2.4.4 Pengertian PHP



Gambar 2.7 Logo *PHP*

(Sumber: Enterprise (2019))

Menurut Enterprise (2019:8), *PHP* merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *website*. Oleh karena itu, *PHP* dapat dijalankan menggunakan *browser*.

Menurut Sidik (2019:9), *PHP* merupakan secara umum dikenal 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*.

2.4.5 Pengertian Sublime Text



Gambar 2.8 Logo *PHP*

(Sumber: Muhammad Ibnu Sa'ad (2020))

Menurut Sa'ad (dalam Ardhana, 2020:39), *Sublime text* adalah salah satu program yang dilakukan untuk melakukan editor, seperti *HTML*, *PHP*, *CSS*. Program editor mudah digunakan dan dapat di-download secara free alias gratis.



Menurut Sa'ad (dalam Mifta Faridl, 2020:39), *Sublime* text adalah teks editor berbasis *python* yang cukup terkenal dikalangan pengembang, penulis, dan desainer.

2.4.6 Pengertian JavaScript



Gambar 2.9 Logo JavaScript

(Sumber: Rusmawan (2019))

Menurut Rusmawan (2019:7), *JavaScript* atau sering disingkat dengan *JS*, digunakan untuk membuat *website* kita lebih interaktif kepada *user* yang menggunakan *website* tersebut.

Menurut Setiawan (2019:194), *JavaScript* adalah bahasa scripting yang populer di sebagian besar browser. *JavaScript* disisipkan pada halaman web dengan menggunakan tag `<script>`.

2.4.7 Pengertian Basis Data

Menurut Abdulloh (2018:103), basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi.

Menurut Rozaq (2019:17), basis data adalah sekumpulan data yang sudah disusun sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi sehingga memudahkan pengguna dalam mengelola data dan juga mempermudah dalam memperoleh informasi.



2.4.8 Pengertian *Bootstrap*



Gambar 2.10 Logo *Bootstrap*

(Sumber: Sulistiono (2018))

Menurut Subagia (2018:45), *Bootstrap* diciptakan untuk mempermudah proses desain web bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Paket bootstrap berisi sekumpulan file CSS, font dan javascript yang siap diintegrasikan ke sebuah dokumen HTML menggunakan kaidah – kaidah tertentu.

Menurut Sulistiono (2018:17), *Bootstrap* adalah sebuah pustaka *open source* yang merupakan *framework CSS* dan *Javascript* untuk membuat *website* yang responsif.

2.4.9 Pengertian *CSS*



Gambar 2.11 Logo *CSS*

(Sumber: Rerung (2018))

Menurut Rerung (2018:133), *CSS* adalah bahasa yang dapat digunakan untuk mendefinisikan bagaimana suatu bahasa markup ditampilkan pada suatu media dimana bahasa markup salah satunya adalah HTML.



2.5 Referensi Terdahulu

Penelitian ini menggunakan 5 (lima) jurnal dari penelitian terdahulu yang dilakukan oleh beberapa orang yang membangun sistem informasi berbasis web dengan menggunakan *Metode Algoritma Cosine Similarity*, yaitu:

Menurut penelitian yang dilakukan oleh Pratama, Faisal, dan Hanani (2019) menerapkan *Metode Algoritma Cosine Similarity* untuk menghasilkan sistem deteksi plagiarisme pada artikel jurnal serta mampu terdeteksi apakah ada unsur kesamaan terhadap artikel lainnya.

Menurut penelitian yang dilakukan oleh Maskur dan Andriansyah (2015) menerapkan *Metode Algoritma Cosine Similarity* untuk digunakan dalam suatu sistem pencarian tugas akhir menggunakan web semantik. fungsinya membantu pencarian informasi yang relevan berdasarkan kata kunci.

Menurut penelitian yang dilakukan oleh Fataruba (2018) menerapkan *Algoritma Cosine Similarity* pada sistem pengecekan kemiripan jawaban ujian siswa pada SMA Negeri 11 Ambon. Dimana metode ini untuk menghitung kesamaan dan kemiripan antara kunci jawaban pengajar dengan kunci jawaban siswa.

Menurut penelitian yang dilakukan oleh Aprianto, dan Aribowo (2018) menerapkan Metode Algoritma Cosine Similarity dalam rancang bangun aplikasi pengecekan kemiripan judul skripsi pada teknik informatika UAD. Dimana dengan metode ini mampu menyelesaikan pihak jurusan untuk menentukan apakah ada kesamaan judul skripsi diantara mahasiswa.

Menurut penelitian yang dilakukan oleh Kurniadi, Haviana, dan Novianto (2020) dimana penelitian ini menggunakan Metode Algoritma Cosine similarity yang diterapkan pada sistem arsip dokumen pada Universitas Islam Sultan Agung Semarang. Tujuannya adalah agar pencarian data yang dilakukan bisa relevan dengan menghitung bobot teks judul dokumen dan isi dokume yang kemudian akan dibobotkan dan dicocokkan pada database.