

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama seperti judul penelitian penulis. Namun penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya bahan kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa skripsi dan jurnal terkait dengan penelitian yang dilakukan penulis.

Rujukan penelitian yang pertama yaitu jurnal Mangore Anirudh K dan M Roberts Masillamani *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* tahun 2019 dengan judul *Efficient Cryptographic Encryption Techniques for Data Privacy Preservation*. Dalam penelitiannya peneliti menganalisa algoritma kriptografi sebagai alat untuk mengamankan data agar tidak diketahui oleh orang lain (*privacy*), peneliti hanya melakukan analisis terhadap algoritma kriptografi untuk pengamanan data-data.

Rujukan penelitian yang kedua yaitu jurnal Ahmed Baita Garko, Kabir Musa Muhammad dan AbdulKarim Salih Aish mahasiswa program studi *Computer Science, Federal University* tahun 2016 dengan judul *Security Of Email In Cloud Computing*. Dalam penelitiannya peneliti menghubungkan antar email server dan client agar dapat saling terhubung dengan memasang keamanan pada email dengan memanfaatkan penyimpanan pada *cloud*.

Rujukan penelitian yang ketiga yaitu jurnal Mufti Baihaqi dan Ahmad Hanafi Mahasiswa STMIK Jendral Achmad Yani Yogyakarta Program Studi Teknik Informatika Tahun 2015 dengan judul *Perancangan dan Pembuatan E-Journal Menggunakan Open Journal System Di STMIK Jendral Achmad Yani Yogyakarta*. Dalam penelitiannya peneliti membuat aplikasi e-journal dengan

menggunakan Open Journal System(OJS) yang bersifat multi-platform. Dan diimplementasikan pada kampusnya sendiri.

Sedangkan penelitian yang akan dilakukan oleh penulis sedikit berbeda dengan penelitian sebelumnya yaitu Membuat suatu aplikasi kemananan data pencarian jurnal berbasis web yang dapat terhubung ke email pengguna dengan mengeknripsi *password user*. Dengan menggunakan aplikasi dengan bahasa pemograman PHP dan MYSQL. Dimana admin akan mengelola aplikasi tersebut. Dan *user* bisa melakukan pencarian jurnal dengan melakukan register terlebih dahulu.

2.2 Algoritma

Algoritma adalah sistem kerja komputer memiliki *brainware*, *hardware*, dan *software*. Tanpa salah satu dari ketiga sistim tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada *software* komputer. *Software* terbangun atas susunan program) dan *syntax* (cara penulisan/pembuatan program). Untuk menyusun program atau *syntax*, diperlukannya langkahlangkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu *software*. Maka, algoritma berperan penting dalam penyusunan program atau *syntax* tersebut.

Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu *software*. Dalam dunia sehari-hari, mungkin tanpa kita sadari algoritma telah masuk dalam kehidupan kita. Algoritma berbeda dengan logaritma. Logaritma merupakan operasi matematika yang merupakan kebalikan dari eksponen atau pemangkatan (Maulana, 2017).

2.3 Kriptografi

Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan (*Cryptography is the art and science of keeping messages secure*) selain itu ada pengertian tentang kriptografi yaitu kriptografi merupakan ilmu yang mempelajari

teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata “seni” di dalam definisi di atas maksudnya adalah mempunyai cara yang unik untuk merahasiakan pesan. Kata “*graphy*” di dalam “*cryptography*” itu sendiri sudah menyiratkan sebuah seni.

Pada awal mulanya kriptografi disamakan dengan enkripsi, dimana kriptografi membuat informasi atau data yang dirubah menjadi kalimat yang tidak dapat dibaca dengan mata telanjang. Atau dipahami sebagai ilmu tentang menyembunyikan pesan. Namun, seiring dengan berkembangnya teknologi ilmu pengetahuan seperti sekarang ini. Maka, pengertian *cryptography* berkembang menjadi ilmu tentang teknik yang digunakan dalam menyelesaikan persoalan keamanan data (Hafiz, 2020).

2.4 MD5

MD5 atau Message-Digest algoritim 5 adalah fungsi hash kriptografik. Algoritma ini terutama digunakan untuk melakukan pemeriksaan integritas file dalam berbagai situasi. Dalam ilmu kriptografi, MD5 adalah salah satu algoritma hash yang paling populer. Hash atau hashing sendiri adalah proses perubahan suatu data menjadi data lain dengan panjang tertentu, sedemikian sehingga data itu tidak dapat dipulihkan kembali. Teknik ini biasa digunakan dalam enkripsi data, misalnya untuk menyimpan password agar tidak ada yang dapat mengetahuinya meskipun dia dapat melihat hash dari password itu (Saputro, 2015).

2.4.1 Cara Kerja Algoritma MD5

Setiap pesan yang akan dienkrpsi, terlebih dahulu dicari berapa banyak bit yang terdapat pada pesan. Kita anggap sebanyak b bit. Di sini b adalah bit non negatif integer, b bisa saja nol dan tidak harus selalu kelipatan delapan.

Langkah-langkah pembuatan message digest secara garis besar:

1. Penambahan bit-bit pengganjal (padding bits).
2. Penambahan nilai panjang pesan semula.

3. Inisialisasi penyangga (buffer) MD.

4. Pengolahan pesan dalam blok berukuran 512 bit.

Berikut adalah penjelesannya:

1. Penambahan Bit-bit Pengganjal

Pesan ditambah dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 modulo 512.

Jika panjang pesan 448 bit, maka pesan tersebut ditambah dengan 512 bit menjadi 960 bit. Jadi, panjang bit-bit pengganjal adalah antara 1 sampai 512.

Bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

2. Penambahan Nilai Panjang Pesan

Pesan yang telah diberi bit-bit pengganjal selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula.

Jika panjang pesan > 264 maka yang diambil adalah panjangnya dalam modulo 264. Dengan kata lain, jika panjang pesan semula adalah K bit, maka 64 bit yang ditambahkan menyatakan K modulo 264.

Setelah ditambah dengan 64 bit, panjang pesan sekarang menjadi kelipatan 512 bit.

3. Inisialisai Penyangga MD

- MD5 membutuhkan 4 buah penyangga (buffer) yang masing-masing panjangnya 32 bit. Total panjangpenyangga adalah $4 \times 32 = 128$ bit. Keempat penyangga ini menampung hasil antara dan hasil akhir.
- Keempat penyangga ini diberi nama A, B, C, dan D. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) sebagai berikut:

A = 01234567

B = 89ABCDEF

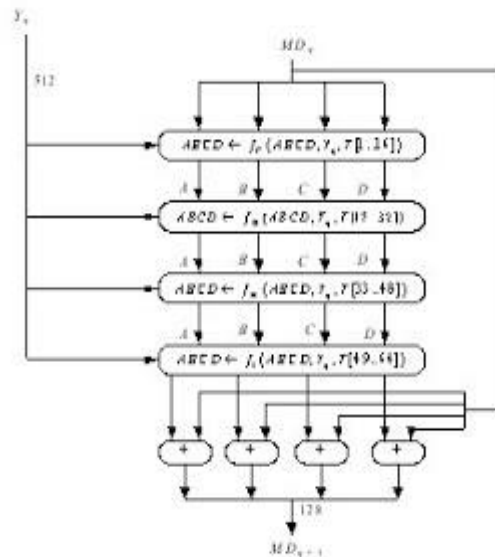
C = FEDCBA98

D = 76543210

4. Pengolahan Pesan dalam Blok Berukuran 512 bit.

- Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y_0 sampai Y_{L-1}). *

- Setiap blok 512-bit diproses bersama dengan penyangga MD menjadi keluaran 128-bit, dan ini disebut proses MD5. Gambaran proses MD5 diperlihatkan pada Gambar berikut ini.



Gambar 2.1 Proses Algoritma MD5

2.4.2 Inisialisasi MD5

Pada MD-5 terdapat empat buah word 32 bit register yang berguna untuk menginisialisasi message digest pertama kali. Register-register ini di inisialisasikan dengan bilangan hexadesimal.

word A: 01 23 45 67

word B: 89 AB CD EF

word C: FE DC BA 98

word D: 76 54 32 10

Register-register ini biasa disebut dengan nama Chain variabel atau variabel rantai.

2.4.3 Proses Pesan di Dalam Blok 16 word

Pada MD-5 juga terdapat 4 (empat) buah fungsi nonlinear yang masing-masing digunakan pada tiap operasinya (satu fungsi untuk satu blok), yaitu:

$$F(X,Y,Z) = (X \dot{\cup} Y) \dot{\cup} ((\emptyset X) \dot{\cup} Z)$$

$$G(X,Y,Z) = (X \dot{\cup} Z) \dot{\cup} (Y \dot{\cup} (\emptyset Z))$$

$$H(X,Y,Z) = X \dot{\wedge} Y \dot{\wedge} Z$$

$$I(X,Y,Z) = Y \dot{\wedge} (X \dot{\cup} (\emptyset Z))$$

($\dot{\wedge}$ untuk XOR, $\dot{\cup}$ untuk AND, $\dot{\cup}$ untuk OR dan $\emptyset$ untuk NOT).

Pada Gambar 3.2 dapat dilihat satu buah operasi dari MD-5 dengan operasi yang dipakai sebagai contoh adalah $FF(a,b,c,d,M_j,s,t_i)$ menunjukkan $a = b + ((a + F(b,c,d) + M_j + t_i) \lll \lll)$

$FF(a,b,c,d,M_j,s,t_i)$ menunjukkan $a = b + ((a + F(b,c,d) + M_j + t_i) \lll \lll)$

$GG(a,b,c,d,M_j,s,t_i)$ menunjukkan $a = b + ((a + G(b,c,d) + M_j + t_i) \lll \lll)$

$HH(a,b,c,d,M_j,s,t_i)$ menunjukkan $a = b + ((a + H(b,c,d) + M_j + t_i) \lll \lll)$

$II(a,b,c,d,M_j,s,t_i)$ menunjukkan $a = b + ((a + I(b,c,d) + M_j + t_i) \lll \lll)$

2.5 Internet

Internet adalah suatu jaringan komunikasi yang menghubungkan satu media elektronik dengan media yang lainnya. Standar teknologi pendukung yang dipakai secara global adalah *Transmission Control Protocol* atau *Internet Protocol Suite* (disingkat sebagai istilah TCP/IP). TCP/IP ini merupakan protokol pertukaran paket (dalam istilah asingnya *Switching Communication Protocol*) yang bisa digunakan untuk miliaran lebih pengguna yang ada di dunia. Sementara itu, istilah “*internetworking*” berarti cara/prosesnya dalam menghubungkan rangkaian internet beserta penerapan aturannya yang telah disebutkan sebelumnya (Nancy, 2019).

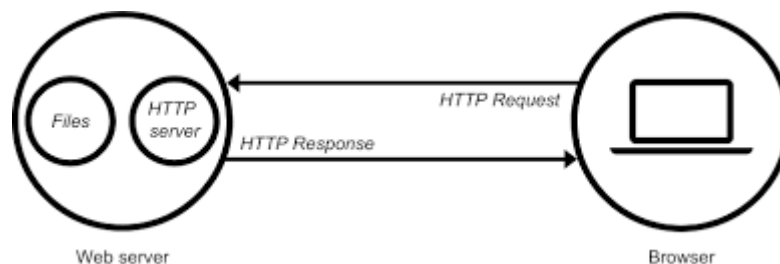
2.6 Server

Server adalah suatu tempat penyimpanan data atau informasi yang berfungsi untuk memberikan layanan (*service*) kepada client yang terhubung dengannya dan juga merupakan suatu sistem komputer yang berfungsi untuk memberikan layanan dan mengontrol akses setiap klien yang terhubung dengan sebuah jaringan komputer. Suatu server didukung oleh prosesor, RAM, dan sistem operasi khusus yang disebut dengan sistem operasi jaringan (*network operating system*).

Tugas utama server adalah melayani komputer *client*, dan di bagi menjadi beberapa fungsi sesuai dengan jenis server, berikut penjelasannya. Sebuah server dapat menyimpan berbagai jenis data, dokumen, dan informasi lainnya yang dapat diakses oleh klien. Dalam hal ini, klien memerlukan server agar dapat terhubung dengan sebuah jaringan atau untuk meminta data/file yang ada di server (Efendi, 2018).

2.7 Web Server

Web server adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui browser kemudian memberikan tanggapan permintaan dalam bentuk halaman situs web atau lebih umumnya dalam dokumen HTML. Jika merujuk pada hardware, web server digunakan untuk menyimpan semua data seperti HTML dokumen, gambar, file CSS *stylesheets*, dan *file JavaScript*. Sedangkan pada sisi software, fungsi web server adalah sebagai pusat kontrol untuk memproses permintaan yang diterima dari browser. Jadi sebenarnya semua yang berhubungan dengan *website* biasanya juga berhubungan dengan web server, karena tugas web server adalah mengatur semua komunikasi yang terjadi antara *browser* dengan server untuk memproses sebuah *website*.



Gambar 2.2 Cara kerja web server

Saat mengambil halaman website, *browser* mengirimkan permintaan ke server yang kemudian diproses oleh web server. HTTP request dikirimkan ke web server. Sebelum memproses HTTP request, web server juga melakukan pengecekan terhadap keamanan. Pada web server, HTTP request diproses dengan bantuan HTTP server. HTTP server merupakan perangkat lunak yang bertugas

menerjemahkan URL (alamat situs web) serta HTTP (protokol yang digunakan browser untuk menampilkan halaman *website*). Kemudian web server mengirimkan HTTP response ke *browser* dan memprosesnya menjadi halaman situs web.

https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server

Pada saat web server menerima HTTP *request* dari *browser*, jika diperlukan web server akan mengirimkan *query* ke *database* untuk memenuhi permintaan HTTP *request* yang dikirimkan oleh *browser* (Yasin, 2018).

2.8 Web browser

Web Browser adalah suatu program atau software yang digunakan untuk menjelajahi internet atau untuk mencari informasi dari suatu web yang tersimpan didalam komputer. Dengan menggunakan *web browser*, para pengguna internet dapat mengakses berbagai informasi yang terdapat di internet dengan mudah.

Web Browser disebut juga sebagai perambah atau peramban, adalah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh server web. Di internet sebenarnya banyak betebaran macam-macam *Web Browser*. Beberapa contoh *Web Browser* diantaranya Internet Explorer, Mozilla Firefox, Safari, Opera, dll. Fungsi *Web Browser* adalah untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh web server (Suryani, 2016).

2.9 Aplikasi Web

Aplikasi berbasis web adalah sebuah aplikasi yang dapat diakses menggunakan web browser atau penjelajah web melalui jaringan internet atau intranet. Meskipun hingga saat ini ternyata lebih banyak, lebih luas, dan lebih komersil dalam pemakaiannya. Banyak dari perusahaan-perusahaan berkembang yang menggunakan aplikasi berbasis web dalam merencanakan sumber daya mereka dan untuk mengelola perusahaan mereka. Beberapa yang lain

mendefinisikan bahwa pengertian aplikasi web adalah program yang tersimpan pada server kemudian dikirim melalui internet dan diakses melalui antar muka atau interface berupa *web browser* (Hatmoko, 2019).

2.10 PHP

PHP (*Hypertext Processor*) disebut bahasa pemrograman server *side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada *web browser (client)*.

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat website hanya menggunakan HTML saja. Web yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan website statis, dimana konten dan halaman web bersifat tetap (Andre, 2015).



Gambar 2.3 Tampilan Logo PHP

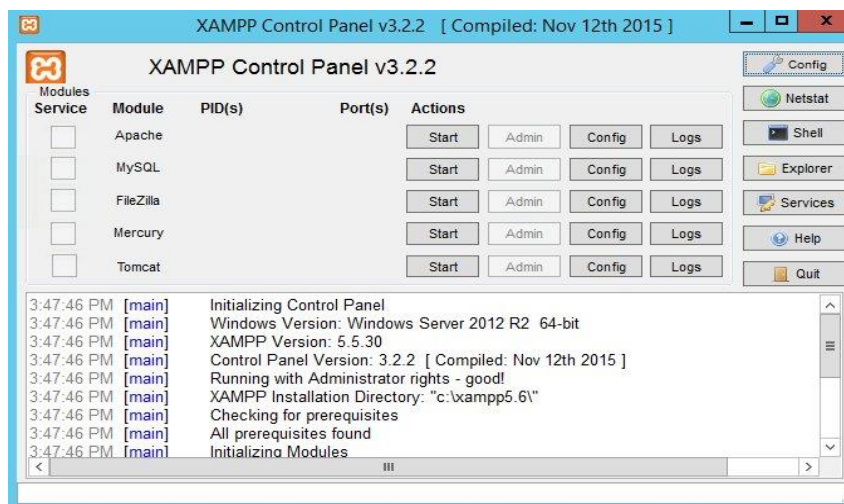
(Sumber : www.duniailkom.com)

2.11 XAMPP

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer local. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *Panel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

Software XAMPP dapat diperoleh secara gratis dari situs www.apachefriends.org atau dapat juga dicari di *google* untuk beberapa versi tertentu. XAMPP adalah perangkat lunak yang bebas, yang mendukung banyak

sistem operasi dan merupakan komplikasi dari beberapa program. Fungsi adalah sebagai *server* yang berdiri sendiri yang terdiri dari atas program *apache* HTTP *server*, *MySQL database*, dan penerjemah bahasa yang ditulis dalam dengan pemrograman PHP dan Perl (Kadir, 2016).



Gambar 2.4 Tampilan Awal XAMPP

(Sumber :Kadir, 2016.)

2.12 MySQL

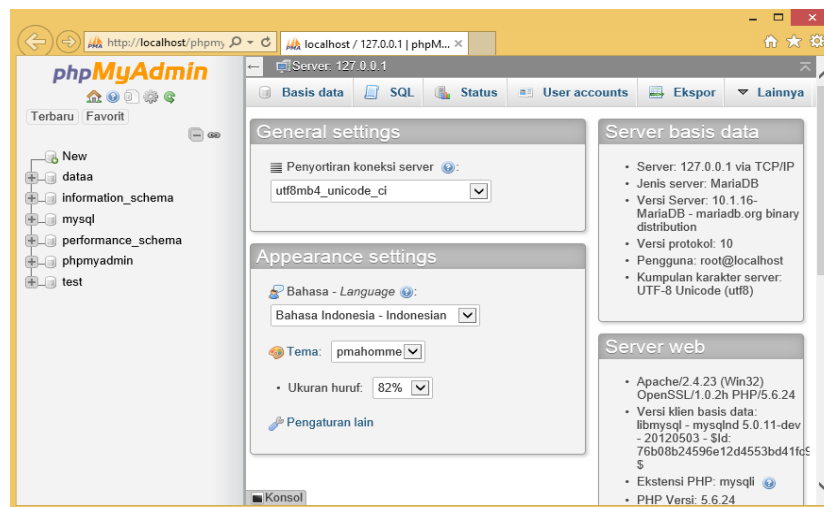
MySQL (*My Structure Query Language*) adalah sebuah program pembuat database yang bersifat open source. MySQL sebenarnya produk yang berjalan pada *platform* Linux. Karena sifatnya *open source*, maka MySQL dapat dijalankan pada semua platform baik Windows maupun Linux. Selain itu, MySQL juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi Multi User (Banyak Pengguna). Saat ini database MySQL telah digunakan hampir oleh semua *programer database*, apalagi dalam pemograman web.

Kelebihan lain dari MySQL adalah mengguakan bahasa *query* standar yang dimiliki SQL (*Structure Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur yang telah di standarkan untuk semua program pengakses *database* seperti Oracle, Posgres SQL, SQL Server, dan lain-lain (Kadir, 2016).

2.13 PhpMyAdmin

PhpMyAdmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan PhpMyAdmin, dapat memudahkan dalam membuat *database*, membuat tabel, *menginsert*, menghapus dan *update* data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual.

Software ini dirilis secara perdana pada tahun 1998 silam dan telah berlisensi *General Public* atau GNU. Ditambah lagi, *software* ini juga support dengan *multi lingual* bahasa atau berbagai pilihan bahasa yang tersedia dalam *user interface* yang ada didalamnya. Jadi tidak ada yang perlu dicemaskan dan khawatir lagi tentang penggunaan dari *software* yang satu ini. Karena memang manfaat yang akan bisa ditemukan dan maksimalkan didalamnya ada begitu banyak (Kadir, 2016).



Gambar 2.5 Tampilan PhpMyAdmin

(Sumber :Kadir, 2016.)

2.14 CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman.

Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, subbab, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda (Azzam, 2015).

2.15 Permodelan Sistem Dengan UML(*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah sekumpulan diagram yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML dapat digunakan untuk mempermudah pengembangan aplikasi yang berkelanjutan. UML dapat dikatakan juga sebagai perkembangan, bahasa pemodelan di bidang rekayasa perangkat lunak yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sebuah sistem. UML terdiri dari banyak elemen-elemen grafis yang digabungkan dalam bentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model (Alexandra, 2019).

2.15.1 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan alur kerja dari berbagai aktivitas *user* atau sistem, orang yang melakukan aktivitas, dan aliran berurutan dari aktivitas ini.

Tabel 2.1 Simbol *Activity Diagram*

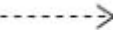
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2.15.2 Use Case Diagram

Use case Diagram adalah diagram yang menunjukkan peran *user* dan bagaimana peran tersebut ketika menggunakan sistem. *Use case* diagram juga dapat digunakan untuk memrepresentasikan interaksi *user* dengan sistem dan menggambarkan spesifikasi kasus penggunaan.

Tabel 2.2 Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.15.3 Sequence Diagram

System Sequence Diagram adalah diagram yang menggambarkan interaksi *user* dengan sistem secara sekuensial (berurutan).

Tabel 2.3 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi