

BAB II

LANDASAN TEORI

2.1 Rekam Medis

Berdasarkan PERATURAN MENTERI KESEHATAN REPUBLIK INDONESIA NOMOR 269/MENKES/PER/III/2008 TENTANG REKAM MEDIS disebutkan bahwa :

Rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien. (Pasal 1).

Dan pada Pasal 2 disebutka : (1) Rekam medis harus dibuat secara tertulis, lengkap dan jelas atau secara elektronik., (2)Penyelenggaraan rekam medis dengan menggunakan teknologi informasi elektronik diatur lebih lanjut dengan peraturan tersendiri.

Pada Pasal 3 Ayat 1 disebutkan : Isi rekam medis untuk pasien rawat jalan pada sarana pelayanan kesehatan sekurang-kurangnya memuat :

- a. Identitas pasien;
- b. Tanggal dan waktu;
- c. Hasil anamnesis, mencakup sekurang-kurangnya keluhan dan riwayat penyakit;
- d. Hasil pemeriksaan fisik dan penunjang medik;
- e. Diagnosis;
- f. Rencana penatalaksanaan;

- g. Pengobatan dan/atau tindakan;
- h. Pelayanan lain yang telah diberikan kepada pasien;
- i. Untuk pasien kasus gigi dilengkapi dengan odontogram klinik;
dan
- j. Persetujuan tindakan bila diperlukan.

2.2 Kriptografi

Menurut (Sentot Kromodimoeljo, Desember 2009)Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi untuk mendapatkan kembali data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Rahasia terletak di beberapa parameter yang digunakan, jadi kunci ditentukan oleh parameter. Parameter yang menentukan kunci dekripsi itulah yang harus dirahasiakan (parameter menjadi ekuivalen dengan kunci).

Dalam kriptografi klasik, teknik enkripsi yang digunakan adalah enkripsi simetris dimana kunci dekripsi sama dengan kunci enkripsi. Untuk public key cryptography, diperlukan teknik enkripsi asimetris dimana kunci dekripsi tidak sama dengan kunci enkripsi. Enkripsi, dekripsi dan pembuatan kunci untuk teknik enkripsi asimetris memerlukan komputasi yang lebih intensif dibandingkan enkripsi simetris, karena enkripsi asimetris menggunakan bilanganbilangan yang sangat besar. Namun, walaupun enkripsi asimetris lebih “mahal” dibandingkan

enkripsi simetris, public key cryptography sangat berguna untuk key management dan digital signature.

Secara garis besar, proses enkripsi adalah proses pengacakan “naskah asli” (plaintext) menjadi “naskah acak” (ciphertext) yang “sulit untuk dibaca” oleh seseorang yang tidak mempunyai kunci dekripsi. Yang dimaksud dengan “sulit untuk dibaca” disini adalah probabilitas mendapat kembali naskah asli oleh seseorang yang tidak mempunyai kunci dekripsi dalam waktu yang tidak terlalu lama adalah sangat kecil. Jadi suatu proses enkripsi yang baik menghasilkan naskah acak yang memerlukan waktu yang lama (contohnya satu juta tahun)¹ untuk didekripsi oleh seseorang yang tidak mempunyai kunci dekripsi. Satu cara untuk mendapatkan kembali naskah asli tentunya dengan menerka kunci dekripsi, jadi proses menerka kunci dekripsi harus menjadi sesuatu yang sulit. Tentunya naskah acak harus dapat didekripsi oleh seseorang yang mempunyai kunci dekripsi untuk mendapatkan kembali naskah asli.

Kriptografi memiliki beberapa aspek keamanan antara lain :

- Kerahasiaan (confidentiality), menjamin bahwa data-data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja. Kerahasiaan bertujuan untuk melindungi suatu informasi dari semua pihak yang tidak berhak atas informasi tersebut.
- Otentikasi (authentication), merupakan identifikasi yang dilakukan oleh masing – masing pihak yang saling berkomunikasi, maksudnya beberapa pihak yang berkomunikasi harus mengidentifikasi satu sama

lainnya. Informasi yang didapat oleh suatu pihak dari pihak lain harus diidentifikasi untuk memastikan keaslian dari informasi yang diterima.

- Integritas (integrity), menjamin setiap pesan yang dikirim pasti sampai pada penerimanya tanpa ada bagian dari pesan tersebut yang diganti, diduplikasi, dirusak, diubah urutannya, dan ditambahkan. Integritas data bertujuan untuk mencegah terjadinya perubahan informasi oleh pihak-pihak yang tidak berhak atas informasi tersebut. Untuk menjamin integritas data ini pengguna harus mempunyai kemampuan untuk mendeteksi terjadinya manipulasi data oleh pihak-pihak yang tidak berkepentingan. Manipulasi data yang dimaksud di sini meliputi penyisipan, penghapusan, maupun penggantian data.
- Nirpenyangkalan (Nonrepudiation), mencegah pengirim maupun penerima mengingkari bahwa mereka telah mengirimkan atau menerima suatu pesan. Jika sebuah pesan dikirim, penerima dapat membuktikan bahwa pesan tersebut memang dikirim oleh pengirim yang tertera. Sebaliknya, jika sebuah pesan diterima, pengirim dapat membuktikan bahwa pesannya telah diterima oleh pihak yang ditujunya. (Ariyus, 2008).

2.3 Vigenère cipher

Menurut (Rinaldi Munir,2019) Vigenère cipher adalah salah satu algoritma kriptografi klasik yang diperkenalkan pada abad 16 atau kira-kira pada tahun 1986. Algoritma kriptografi ini dipublikasikan oleh seorang diplomat dan juga kriptologis yang berasal dari Prancis, yaitu Blaise de Vigenère, namun sebenarnya algoritma ini telah digambarkan sebelumnya pada buku *La Cifra del Sig.* Giovan Batista Belaso, sebuah buku yang ditulis oleh GiovanBatista Belaso, pada tahun

1553 .Cara kerja dari Vigenère cipher ini mirip dengan Caesar cipher, yaitu mengenkripsi plaintext pada pesan dengan cara menggeser huruf pada pesan tersebut sejauh nilai kunci pada deret alphabet. Vigenère cipher adalah salah satu algoritma kriptografi klasik yang menggunakan metode substitusi abjad majemuk. Substitusi abjad-majemuk mengenkripsi setiap huruf yang ada menggunakan kunci yang berbeda, tidak seperti Caesar cipher yang menerapkan metode substitusi abjad-tunggal yang semua huruf disuatu pesan dienkripsi menggunakan kunci yang sama. Sebagai contoh Caesar cipher jika terdapat

plaintext: KRIPTOGRAFI

Maka jika dienkripsi dengan dengan nilai kunci 3 akan didapat

ciphertext: NULSWRJUDIL

Dari ciphertext yang didapat dapat kita lihat bahwa huruf K dienkripsi menjadi N, huruf R dienkripsi menjadi huruf I, dan seterusnya dimana huruf pada pesan digeser sejauh nilai kunci. Algoritma Caesar cipher sangat sederhana sehingga sangat berisiko untuk dipecahkan karena hanya dibutuhkan pengetahuan satu huruf dari plaintext untuk mengetahui kunci yang digunakan. Vigenère cipher yang menerapkan metode substitusi abjad-majemuk tidak memiliki permasalahan tersebut karena setiap huruf pada pesan yang dienkripsi dengan Vigenère cipher ini akan digeser dengan nilai yang berbeda tergantung dengan kunci yang diberikan. Kunci yang digunakan pada Vigenère cipher berbeda dengan yang digunakan pada Caesar cipher. Jika pada Caesar cipher kuncinya hanya satu nilai saja, maka pada Vigenère cipher kunci yang digunakan berbentuk deretan huruf. Kunci yang berbentuk deretan kata tersebut akan memungkinkan

setiap huruf plainteks untuk dienkripsi dengan kunci yang berbeda. Jika panjang kunci yang digunakan lebih pendek dari panjang plainteks maka kunci akan diulang sampai panjang kunci sama dengan panjang plainteks. Algoritma ini akan meminimalkan kemungkinan dipecahkannya cipherteks jika satu huruf plainteks diketahui. Model matematika dari enkripsi pada algoritma Vigenère cipher ini adalah seperti berikut:

$$C_i = E_k(M_i) = (M_i + K_i) \bmod 26$$

Dan model matematika untuk deskripsinya adalah:

$$M_i = D_k(C_i) = (C_i - K_i) \bmod 26$$

Dengan C memodelkan cipherteks, M memodelkan Plainteks, dan K memodelkan kunci. Contoh dari penerapan algoritma Vigenère cipher adalah jika kita memiliki sebuah plainteks yang ingin dienkripsi: *

KRIPTOGRAFI

Dan kita menggunakan kunci:

SANDI

Maka plainteks akan dienkripsi dengan cara:

Plaintext : KRIPTOGRAFI

Kunci : SANDISANDIS

Ciphertext : CRVSBGGEDNA

Huruf pada kunci akan dikonversi menjadi sebuah nilai, misalnya $A = 0$, $B = 1$, sampai dengan $Z = 25$. Setelah itu prosesnya sama seperti pada Caesar cipher dimana setiap huruf pada plainteks akan digeser sejauh nilai kunci yang posisinya bersesuaian. Pergeseran huruf-huruf ini bisa dipetakan dalam bentuk tabel 26x26 yang memetakan antara huruf pada plainteks dengan huruf pada kunci seperti yang diperlihatkan pada Gambar 2.1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 2. 1 Tabel 26x26

2.4 Internet dan Website

Internet adalah komputer yang terhubung dengan jaringan dan saling berkomunikasi dengan waktu dan wilayah yang tidak terbatas.(Darma, Jarot S., Shenia A,” Buku Pintar Menguasai Internet”)

Aplikasi Web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis Web. Fitur-fitur aplikasi Web biasanya berupa data persistence, mendukung transaksi dan komposisi halaman Web dinamis yang dapat dipertimbangkan sebagai hibridisasi, antara hipermedia dan sistem informasi. Aplikasi Web adalah bagian dari client-side yang dapat dijalankan oleh browser Web. Client-side mempunyai tanggung jawab untuk pengeksekusian proses bisnis. (Janner Simarmata,”Rekayasa Web”)

PHP adalah akronim dari Hypertext Preprocessor, yaitu suatu bahasa pemrograman berbasis kode-kode (script) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke web browser menjadi kode HTML. Kode PHP mempunyai ciri-ciri khusus, yaitu:

- a. Hanya dapat dijalankan menggunakan web server, misal: Apache.
- b. Kode PHP diletakkan dan dijalankan di web server.
- c. Kode PHP dapat digunakan untuk mengakses database, seperti:
- d. MySQL, PostgreSQL, Oracle, dan lain-lain.
- e. Merupakan software yang bersifat open source.
- f. Gratis untuk di-download dan digunakan.
- g. Memiliki sifat multipaltform, artinya dapat dijalankan menggunakan sistem operasi apapun, seperti: Linux, Unix, Windows, dan lain-lain.

Dengan menggunakan PHP, selain memberikan keuntungan seperti pada beberapa poin di atas, juga didukung oleh banyak komunitas. Hal ini yang membuat PHP terus berkembang. Selain itu, Anda dapat belajar lebih banyak lagi tentang tips dan trik penggunaannya dari berbagai komunitas, lembaga pendidikan, ataupun melalui media internet. (Diar Puji Oktavian, "Menjadi Programmer Jempolan Menggunakan PHP")

Pada dasarnya, PHP memungkinkan halaman web statis menjadi dinamis. "PHP" adalah singkatan dari "PHP: Hypertext Preprocessor". Kata "Preprocessor" berarti bahwa PHP membuat perubahan sebelum halaman HTML dibuat. Ini memungkinkan pengembang untuk membuat aplikasi yang kuat yang dapat menerbitkan blog, mengontrol perangkat keras dari jarak jauh, atau menjalankan situs web yang kuat seperti Wikipedia atau Wikibooks. Tentu saja, untuk mencapai sesuatu seperti ini, Anda memerlukan aplikasi database seperti MySQL.

Menurut (Achmad Solichin "Pemrograman Web dengan PHP dan Mysql") MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya

masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

Beberapa kelebihan MySQL antara lain :

- Free (bebas didownload)
- Stabil dan tangguh
- Fleksibel dengan berbagai pemrograman
- Security yang baik
- Dukungan dari banyak komunitas
- Kemudahan management database.
- Mendukung transaksi
- Perkembangan software yang cukup cepat.

2.5 UML

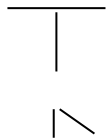
1. Use Case Diagram




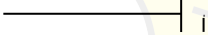

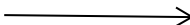
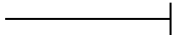
Menurut (Rosa A S dan M. Shalahudin, 2016) dalam buku “Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)”, Use Case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Dua hal utama pada Use Case yaitu:

1. Aktor : merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri
2. Use Case : merupakan fungsionalisasi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Tabel 2. 1 Simbol – simbol yang terdapat pada *use case*


Komponen <i>Use case</i>	Penjelasan
	Merupakan sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem.
Aktor	




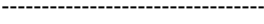
 Use Case	Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
 Asosiasi	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
<extends>  Extend	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu.
 Generalisasi	Hubungan generalisasi dan spesialisasi (umum khusus) antara dua buah use case dimana fungsinya yang satu adalah fungsi yang lebih umum dari lainnya.
 Dependency	Dependency sebuah elemen bergantung dalam Beberapa cara ke elemen lainnya.
<<include>>  <<use>>  Include / Use	Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

2. Sequence Diagram

Menurut (Rosa A S dan M. Shalahudin 2014:165), diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dengan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Tabel 2. 2 Simbol – simbol yang terdapat pada *sequence diagram*


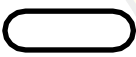



Komponen <i>sequence diagram</i>	<i>sequence</i>	Penjelasan
		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.

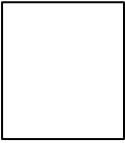
<p>Garis hidup</p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.</p>
<p><<create>></p>  <p>Pesan tipe <i>create</i></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>1 : nama _metode</p>  <p>Pesan tipe <i>call</i></p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p>
<p>1 : masukan</p>  <p>Pesan tipe <i>send</i></p>	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>1 : keluaran</p>  <p>Pesan tipe <i>retrun</i></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>

3. Activity Diagram

Menurut (Rosa A S dan M. Shalahuddin, 2016) dalam buku “Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)”, Activity Diagram atau diagram aktivitas menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dilakukan oleh sistem.

Tabel 2. 3 Simbol – simbol yang terdapat pada *activity diagram*

Komponen <i>Activity Diagram</i>	Penjelasan
 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 / decision	Asosiasi percabangan dimana jika ada pilihan aktivitas digabungkan menjadi satu.
 Penggabungan / join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
 Status akhir	Status akhir yang dilakukan sistem Diagram aktivitas memiliki suatu

	akhir.
 Swimlane	Memisahkan organisasi yang Bertanggung jawab terhadap aktivitas Yang terjadi.

2.6 Tools

2.6.1 Sublime Text

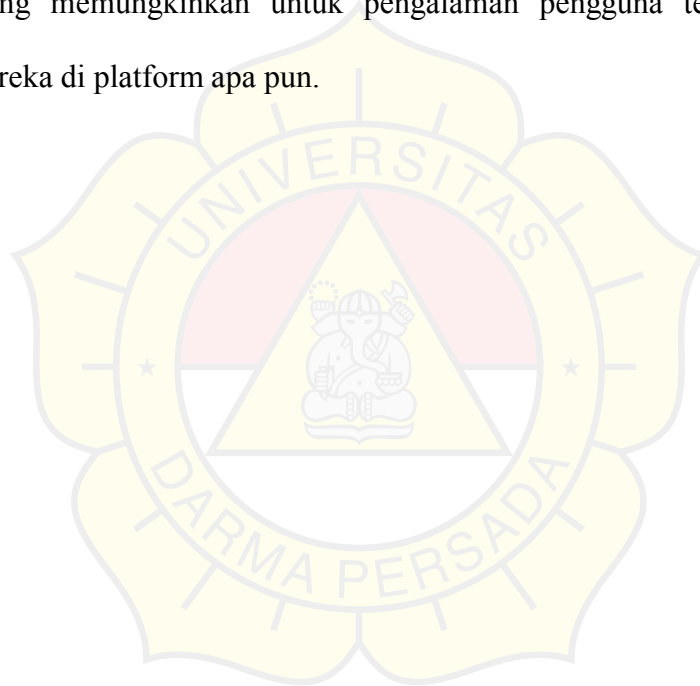
Sublime Text merupakan teks editor yang canggih untuk kode, *markup*, dan prosa. Sublime Text memiliki antarmuka yang polos, dengan fitur yang luar biasa, dan memiliki kinerja yang luar biasa. Fitur dan kinerja yang luar biasa tersebut terletak pada plugin yang bisa kita install dengan mudah. Plugin Sublime tersedia sangat banyak sekali dan sebagian besar dari antaranya gratis.

2.6.2 Xampp

Xampp banyak orang tahu dari pengalaman mereka sendiri bahwa tidak mudah untuk menginstal server web Apache dan semakin sulit jika Anda ingin menambahkan MariaDB, PHP dan Perl. Tujuan XAMPP adalah membangun distribusi instalasi yang mudah bagi para pengembang untuk masuk ke dunia Apache. Agar nyaman bagi pengembang, XAMPP dikonfigurasi dengan semua fitur dihidupkan.

2.6.3 Matereliazze

Materialize adalah CSS framework responsif yang modern berdasarkan Desain Bahan oleh Google. Framework ini berisi template desain berbasis HTML dan CSS untuk tipografi, formulir, tombol, navigasi dan komponen antarmuka lainnya, serta juga ekstensi opsional JavaScript. Dibuat dan dirancang oleh Google, Material Design adalah bahasa desain yang menggabungkan prinsip-prinsip klasik dari desain yang sukses bersama dengan inovasi dan teknologi. Tujuan Google adalah untuk mengembangkan sistem desain yang memungkinkan untuk pengalaman pengguna terpadu di semua produk mereka di platform apa pun.



2.7 SDLC

System Development Life Cycle (SDLC) merupakan gambaran dari suatu usaha dalam merancang sistem yang akan selalu bergerak seperti roda, yang melewati beberapa langkah atau tahapan antara lain tahap investigate, analisa, desain, implementasi dan perawatan. Setiap langkah dalam SDLC mempunyai tujuan-tujuan yang mendukung tujuan-tujuan penyusunan sistem, yaitu menyusun sistem informasi secara efisien dan efektif.

Langkah – langkah SDLC (Reymond Mc Leod, 2010:187)

- a) Tahap Perencanaan / Investigation Tahap Perencanaan berkenaan dengan studi awal untuk membangun sistem baru.
- b) Tahap Analisis Analisis sistem dapat diartikan sebagai suatu proses untuk memahami sistem yang ada dengan tujuan untuk merancang sistem baru atau diperbaharui.
- c) Tahap Perancangan / Desain sistem. Pada tahap selanjutnya adalah mendesain sistem baru agar dapat berjalan lebih baik, dan diharapkan dapat mengatasi masalah-masalah yang ada serta sedapat mungkin dapat mengantisipasi kemungkinan-kemungkinan dari masa akan datang.
- d) Tahap Penerapan Penerapan merupakan kegiatan memperoleh dan mengidentifikasi sumber daya fisik dan konseptual untuk menghasilkan suatu sistem yang bekerja, dengan melakukan beberapa hal yaitu : merencanakan penerapan, mengumumkan penerapan, mendapatkan sumber daya perangkat keras, mendapatkan sumber daya perangkat lunak, menyiapkan database, menyiapkan fasilitas fisik, training pengguna, dan masuk sistem baru.
- e) Tahap pemeliharaan Dalam penggunaan sistem dipandang perlu diadakan pemeliharaan sistem. Hal tersebut di ketahui atas beberapa alasan, antara lain : bermaksud untuk memperbaiki kesalahan, menjaga kemutakhiran sistem, dan meningkatkan system.