

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penulis melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu:

Penelitian yang dilakukan oleh Siti Nurhaya Nugraha (2018) yang berjudul "Implementasi Algoritma Kriptografi ElGamal Untuk Enkripsi Dan Dekripsi Pesan". Penelitian ini menghasilkan sebuah aplikasi untuk proses pengiriman atau pun menghasilkan informasi yang bersifat rahasia. Metode ElGamal digunakan untuk menyelesaikan masalah pengamanan informasi.

Penelitian lain yang dilakukan oleh Aisyatul Karima, L. Budi Handoko, Ari Saputro Dkk (2017) dengan judul "Pemfaktoran Bilangan Prima pada Algoritma ElGamal untuk Keamanan Dokumen PDF" Menghasilkan aplikasi keamanan data dokumen pdf dimana algoritma ElGamal dapat digunakan untuk meningkatkan keamanan dokumen dengan beberapa ekstensi file.

Penelitian lain yang dilakukan oleh Relina Hutagalung (2018) yang berjudul "Implementasi Algoritma Prediction By Partial Matching (Ppm) Pada

Kompresi File Teks Terenkripsi Elgamal” Menghasilkan aplikasi menggunakan algoritma Prediction by Partial Matching, sehingga didapatkan sebuah file yang sudah terenkripsi dan memiliki ukuran yang lebih kecil dari ukuran aslinya.

Penelitian lain yang dilakukan oleh Relina Hutagalung (2018) dengan judul “Implementasi Algoritma Prediction By Partial Matching (Ppm) Pada Kompresi File Teks Terenkripsi Elgamal”. Penelitian ini menghasilkan suatu teknik adaptive statistical data compression yang berdasarkan pada context modeling dan teknik prediksi. Model PPM menggunakan seperangkat simbol pada deretan simbol yang belum dikompresi untuk memprediksi deretan simbol selanjutnya yang akan dikompresi. Algoritma PPM ditemukan dan dikembangkan oleh John Cleary dan Ian Witten, selanjutnya algoritma ini diimplementasikan menjadi suatu program kompresi oleh Alistair Moffat pada bulan Juli tahun 1987. Pada proses, PPM menjalankan rangkaian model deret konteks, dari 0 hingga nilai maksimum k yang ditetapkan sebelumnya, untuk memprediksi karakter-karakter berikutnya. Untuk tiap statistik model konteks order i yang memuat semua simbol yang telah tersusun, setiap panjang i berikutnya dalam input dan berapa kali terjadiannya. Peluang kemungkinan dihitung dari statistik-statistik dengan cara setiap model.

II.2. Landasan Teori

II.2.2. Data Konsumen

Data konsumen adalah kumpulan informasi setiap konsumen yang pernah mencicipi atau membeli produk kopi kamu. Data konsumen meliputi nama, nomor telepon, alamat rumah, dan alamat email.

Selain itu, data konsumen juga dapat berupa riwayat pembelian, jadwal janji

temu, riwayat percakapan, dan beberapa hal yang menjadi kebutuhan konsumen. kamu bisa memanfaatkan data konsumen untuk kepentingan bisnis kopi kamu.

Umumnya dalam mengelola data konsumen, kamu membutuhkan software atau aplikasi database pelanggan seperti CRM. Salah satu contoh software CRM adalah kontak yang mampu menyimpan, melihat, dan mengelola data konsumen secara lebih mudah.

CRM akan menampilkan data pelanggan atau konsumen secara 360 derajat yang berisikan semua informasi kontak, aktivitas terakhir kontak, dan deals.

II.2.1. Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk membantu meayani kebutuhan seseorang untuk melakukan aktivitas. Dengan adanya aplikasi maka kebutuhan akan pelayanan sebuah aktivitas menjadi lebih baik. Dimana setiap pekerjaan dapat dilakukan dengan mudah kalau menggunakan sebuah aplikasi (Agusdi Syafrizal: 2018).

II.2.2. Kriptografi

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. *Kriptografi* adalah ilmu pengetahuan dan seni menjaga *message* agar tetap aman. Tujuan penerapan *kriptografi* adalah untuk membuat sesuatu yang tersembunyi, dapat suatu pesan rahasia berupa teks, suara, gambar dan video. Di dalam sistem *kriptografi* terdapat 5 bagian yaitu

1. *Plaintext* adalah pesan atau data dalam bentuk aslinya teks yang dapat terbaca.

Plaintext adalah masukan bagi algoritma enkripsi.

2. *Secret Key* adalah masukan bagi algoritma enkripsi merupakan nilai yang

bebas terhadap teks asli dan menentukan hasil keluaran algoritma enkripsi.

3. *Chiphertext* adalah keluaran algoritma enkripsi. *Ciphertext* dapat dianggap sebagai pesan tersembunyi yang akan terlihat acak.
4. *Algoritma Enkripsi* memiliki 2 masukan teks asli dan kunci rahasia. *Algoritma enkripsi* melakukan transformasi terhadap teks asli sehingga menghasilkan teks sandi.
5. *Algoritma Dekripsi* memiliki 2 masukan yaitu teks sandi dan kunci rahasia. *Algoritma dekripsi* memulihkan kembali teks sandi menjadi teks asli bila kunci rahasia *algoritma enkripsi* sama dengan *algoritma dekripsi* (Guntur Tri Wibowo, dkk: 2015).

II.2.3. Metode ElGamal

Algoritma ElGamal merupakan algoritma dalam kriptografi yang termasuk dalam kategori algoritma asimetris. Keamanan algoritma ElGamal terletak pada kesulitan penghitungan logaritma diskret pada bilangan modulo prima yang besar sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sukar. Algoritma ElGamal mempunyai kunci publik berupa tiga pasang bilangan dan kunci rahasia berupa satu bilangan. Algoritma ini mempunyai kerugian pada cipherteksnya yang mempunyai panjang dua kali lipat dari plainteksnya. Akan tetapi, algoritma ini mempunyai kelebihan pada enkripsi. Untuk plainteks yang sama, algoritma ini memberikan cipherteks yang berbeda (dengan kepastian yang dekat) setiap kali plainteks di enkripsi. Algoritma ElGamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Algoritma ini merupakan cipher blok, yaitu melakukan proses enkripsi pada blok-blok plainteks dan menghasilkan blok-blok cipherteks yang kemudian dilakukan proses dekripsi dan hasilnya digabungkan

II.2.4. PHP (*Hypertext Preprocessor*)

Menurut Jubille Enterprise (2017:1) PHP adalah bahasa pemrograman yang digunakan untuk membuat sistem berbasis *website*. Sebagai sebuah sistem, *website* tersebut hendaknya memiliki sifat dinamis dan interaktif. Memiliki sifat dinamis, artinya *website* tersebut bisa berubah tampilan kontennya sesuai kondisi tertentu (misalnya, menampilkan produk yang berbeda-beda untuk setiap pengunjung). Artinya interaktif *website* tersebut dapat memberi *feedback* bagi *user* (misalnya, menampilkan hasil pencarian produk).

Menurut Jubille Enterprise, (2017:1) PHP merupakan bahasa pemrograman berjenis *service-side*. Dengan demikian, PHP akan diproses oleh *server* yang hasil olahannya akan dikirim kembali ke *browser*. Oleh karena itu, salah satu *tool* yang harus tersedia sebelum memulai pemrograman PHP adalah *server*.

II.2.5. MySQL

Menurut Solichin (2018:138), MySQL merupakan salah satu perangkat lunak basis data yang sangat populer. Saat ini tersedia versi MySQL yang berbayar (MySQL *Enterprise Edition*), namun tetap tersedia versi MySQL yang gratis (MySQL *Community Edition*). MySQL *Community Edition* dapat diunduh secara gratis, dan bebas digunakan dalam berbagai keperluan. Walaupun demikian, secara fitur dan kemampuan, MySQL versi tidak terlalu jauh berbeda dengan versi berbayar.

Kelebihan MySQL

1. Bersifat *open source*, yang memiliki kemampuan untuk dapat dikembangkan lagi
2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dunia dalam pengolahan data
3. *Super performance* dan *realible*, tidak bisa diragukan, pemrosesan *database*-

nya sangat cepat dan stabil

4. Sangat muda dipelajari (*easy of use*)
5. Memiliki dukungan support (group) penggunaan MySQL
6. *Multiuser*, dimana MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami konflik

II.2.6. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) digunakan sebagai suatu cara untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna sistem/perangkat lunak. Dengan adanya bahasa yang bersifat standar, komunikasi perancang dengan pemrogram (komunikasi antar anggota kelompok pengembang) serta calon pengguna diharapkan menjadi mulus, adapun pengertian UML menurut para ahli dapat dipaparkan sebagai berikut :

Menurut Shofwan Hanief & Dian Pramana (2018 : 166) menyatakan bahwa: “*Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisas, merancang dan mendokumentasikan sistem piranti lunak”

Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML adalah sebagai berikut:

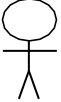
1. *Use case* Diagram

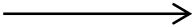
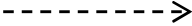
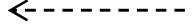
Use Case Diagram merupakan pemodelan sistem informasi perilaku yang akan dilakukan. Kasus penggunaan menggambarkan interaksi antara satu atau lebih aktor

dengan sistem informasi yang akan dibuat. Dapat dikatakan bahwa use case memungkinkan untuk mengetahui apa saja fungsi dari sistem informasi dan siapa yang berhak menggunakan fungsi tersebut.

Jadi, dapat disimpulkan *Use case* adalah langkah – langkah atau urutan kegiatan yang dilakukan actor dan sistem informasi yang akan dibuat. Secara singkat, *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan.

Tabel II. 2.1 Simbol *Use Case Diagram*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.

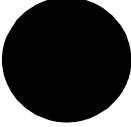
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

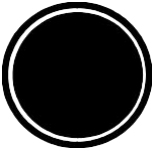
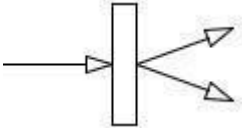
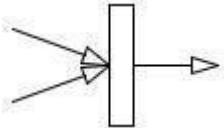
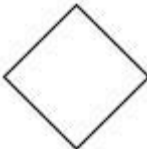
(Ismail et al., 2021)

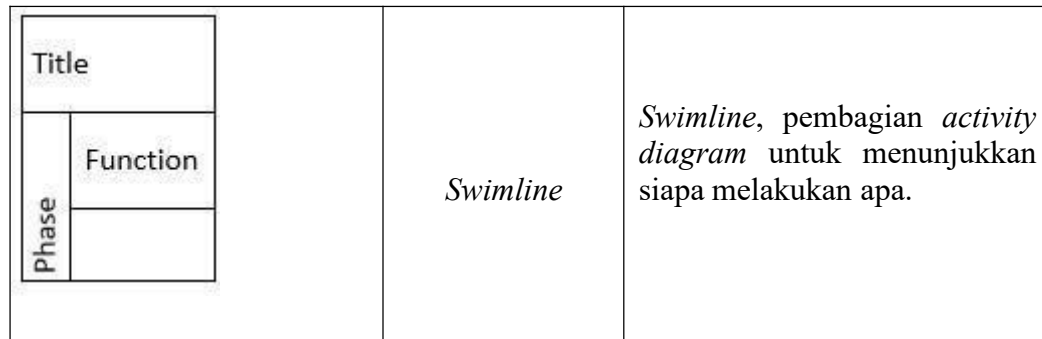
2. Activity Diagram (Diagram Aktivitas)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau properti bisnis. Simbol-simbol yang digunakan dalam *Activity Diagram* yaitu:

Tabel II.2.2 Simbol Activity Diagram

Gambar	Nama	Keterangan
	<i>Start Point</i>	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.

	<i>End Point</i>	<i>End Point</i> , akhir dari aktivitas.
	<i>Action/Activities</i>	<i>Action/Activities</i> , menggambarkan suatu proses kegiatan/bisnis.
	<i>Fork Node</i>	<i>Fork Node</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join Node</i>	<i>Join Node</i> (Penggabungan), digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan (<i>true/false</i>).



(Meiliana et al., 2017)

3. Class Diagram (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan rinci dari setiap kelas dalam model desain suatu sistem, juga menunjukkan aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class Diagram Ini juga menunjukkan atribut dan operasi kelas, serta batasan yang terkait dengan objek yang terkait. *Class Diagram* Biasanya mencakup: kelas (*class*), hubungan, *asosiasi*, *Generalisasi* dan *Agregasi*, Atribut (*Attributes*), Operasi (*Operation / Method*) visibilitas, tingkat akses objek eksternal ke operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.2.3 Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau satu atau lebih

1..*	Satu atau lebih
0..1	Boleh tidak ada, maksimal satu
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Thomas et al., 2021)