

BAB I. PENDAHULUAN

1.1 Latar Belakang

Warna memegang peran sentral dalam ekspresi nilai-nilai dan karakteristik *brand*. Dalam konteks palet warna *brand*, warna berkolaborasi untuk mengkomunikasikan pesan atau citra tertentu. Warna berperan sebagai identitas yang memainkan peran kunci dalam menyampaikan pesan, membangkitkan emosi, dan menciptakan daya tarik visual. Beberapa pelaku bisnis belum memberikan perhatian yang cukup terhadap desain web dalam strategi pemasaran mereka. Dalam sebuah survei dengan 12 direktur perusahaan, sebagian besar dari mereka (11 dari 12) ternyata memiliki pemahaman yang terbatas tentang teori warna, dan keputusan mereka mengenai warna didasarkan pada perasaan semata. Keselarasan warna dalam desain web memiliki potensi untuk memengaruhi respons pengunjung dan persepsi merek. Peran serta *brand* manager menjadi sangat krusial dalam memahami dan mengelola kepribadian merek yang diinginkan oleh konsumen. Kelebihan pengetahuan tentang teori warna sangat penting, dan pelaku bisnis perlu memastikan bahwa desain web mereka selaras dengan karakter merek. Saat ini, penelitian akademis tentang bagaimana warna memengaruhi persepsi konsumen, termasuk kepribadian merek, tingkat familiaritas, daya tarik, dan niat untuk membeli masih terbatas (Setyaningrum, 2015).

Ketika merancang sebuah *website*, penting untuk mempertimbangkan dengan cermat kombinasi warna yang digunakan untuk menciptakan tampilan yang menarik. Pemilihan kombinasi warna harus memperhatikan keseimbangan dan kontras yang tepat (Sasongko dkk., 2020).

Penggunaan warna dalam desain web memiliki kemampuan untuk mempengaruhi emosi dan respon pengunjung dalam membuat keputusan. Beberapa organisasi bisnis memiliki hubungan khusus dengan warna tertentu dalam identitas visual merek mereka. Misalnya, perusahaan layanan komunikasi sering menggunakan warna merah (seperti Telkomsel), biru (seperti XL), atau kuning (seperti Indosat) (Swasty & Utama, 2017).

Seorang desainer kemasan makanan harus memiliki pemahaman mendalam tentang aspek seni, tata letak, dan komposisi, sambil memperhitungkan unsur-unsur filosofis, bentuk, simbol, serta karakteristik fisik, biologis, dan kimia dari produk dan kemasannya. Ini adalah aspek penting yang perlu dikuasai oleh para pengusaha muda yang tertarik untuk memasuki bisnis kuliner (Julyaningsih dkk., 2023).

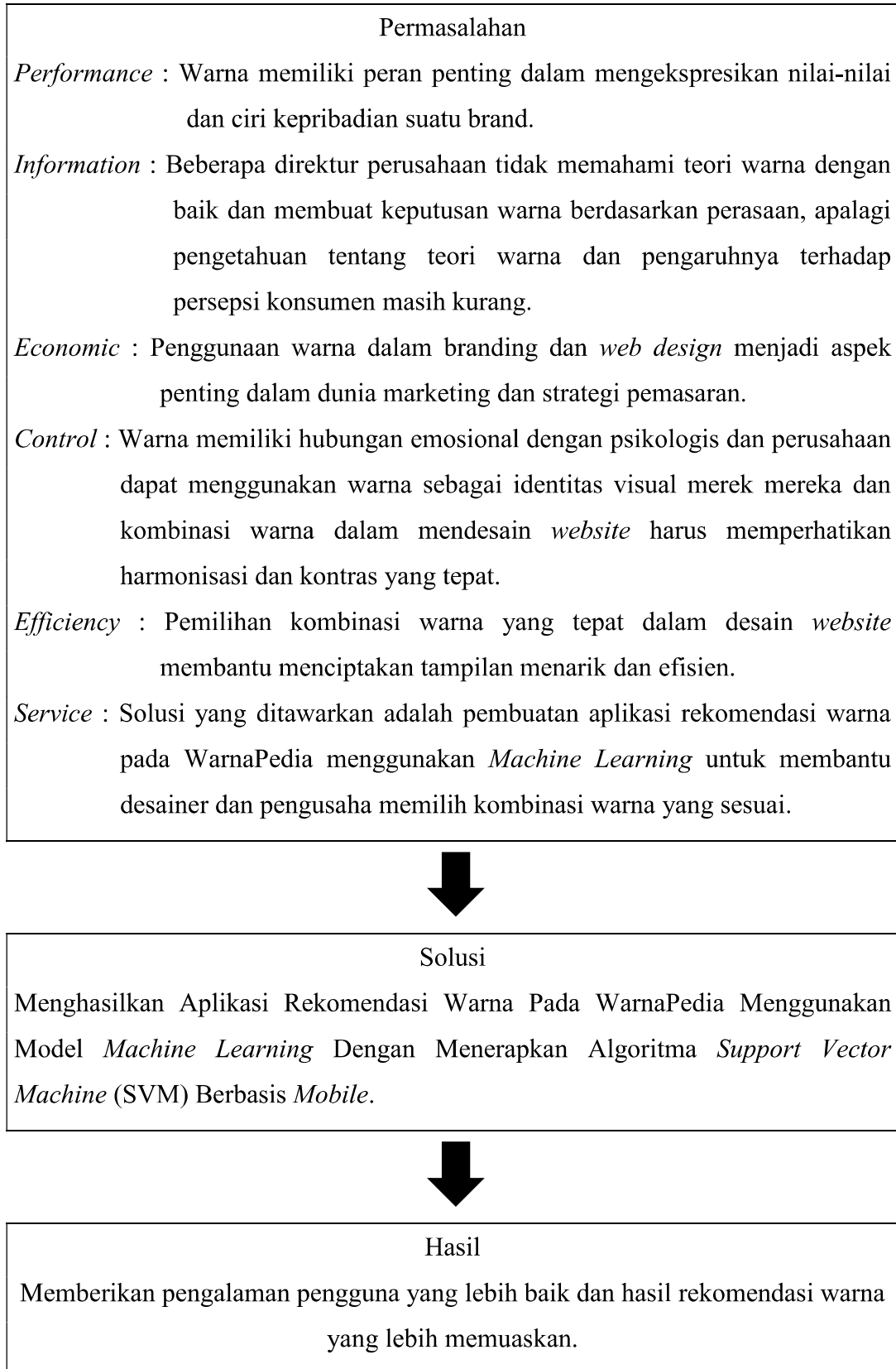
Untuk mengatasi tantangan yang dijelaskan di atas, sebuah solusi diusulkan dalam bentuk pengembangan aplikasi rekomendasi warna dengan menggunakan model *Machine Learning* dengan implementasi algoritma *Support Vector Machine* (SVM) yang dapat diakses melalui perangkat *mobile*.

1.2 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah menghasilkan Aplikasi Rekomendasi Warna Pada WarnaPedia Menggunakan Model *Machine Learning* Dengan menerapkan Algoritma *Support Vector Machine* (SVM) Berbasis *Mobile*.

1.3 Kerangka Pemikiran

Masih ada sebagian orang yang menghadapi kesulitan dalam memilih warna, yang dapat menghambat proses desain dan berpotensi memengaruhi pengalaman pengguna. Oleh karena itu, sebagai respons terhadap masalah ini, diperlukan solusi yang mencakup pembuatan sebuah aplikasi *mobile* berbasis model *Machine Learning* yang menggunakan algoritma *Support Vector Machine* (SVM) untuk memberikan rekomendasi warna dalam platform WarnaPedia, dan untuk kerangka pemikirannya dapat dilihat pada gambar 1.



Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Melalui pembuatan tugas akhir ini, diharapkan dapat memberikan manfaat berikut:

1. Memberikan bantuan kepada desainer *website* di Indonesia dalam memperbaiki proses pemilihan warna dan menghasilkan desain yang lebih efisien.
2. Memperkenalkan teknologi *Machine Learning* kepada mahasiswa di Polinela.

BAB II. TINJAUAN PUSTAKA

2.1 Aplikasi

Menurut definisi dari Kamus Besar Bahasa Indonesia (KBBI), aplikasi merujuk pada implementasi dari suatu sistem yang dirancang untuk memproses data dengan menggunakan aturan atau bahasa pemrograman tertentu (Aulya dkk., 2022). Aplikasi dapat diklasifikasikan ke dalam beberapa kategori berikut:

1. *Enterprise*: Aplikasi ini digunakan oleh organisasi besar dengan tujuan menghubungkan aliran informasi antar departemen. Contohnya adalah aplikasi seperti *IT Helpdesk* dan *Travel Management*.
2. *Enterprise-Support*: Merupakan aplikasi yang berfungsi sebagai dukungan bagi operasional perusahaan. Contoh-contoh meliputi *Database Management*, *Email Server*, dan Sistem Jaringan.
3. *Individual Worker*: Aplikasi ini ditujukan untuk pengguna individu yang dapat mengolah atau mengedit data sesuai kebutuhan masing-masing. Contoh termasuk *Microsoft Office*, *Photoshop*, dan *Acrobat Reader*.
4. Aplikasi Akses Konten: Jenis aplikasi ini digunakan oleh individu untuk mengakses konten, tanpa melakukan pemrosesan atau pengeditan data, dan memiliki kustomisasi yang terbatas. Contohnya adalah *Game*, *Media Player*, dan *Web Browser*.
5. Aplikasi Pendidikan: Aplikasi ini biasanya berbentuk simulasi dan berisi konten pembelajaran tertentu.
6. Aplikasi Simulasi: Digunakan untuk keperluan penelitian, pengembangan, dan simulasi lainnya. Sebagai contoh, simulasi pengaturan lalu lintas.
7. Pengembangan Media Aplikasi: Aplikasi ini dirancang untuk memproses atau mengembangkan media, seringkali untuk tujuan komersial, hiburan, dan pendidikan. Contohnya termasuk perangkat lunak animasi digital dan konverter audio video.

Semua aplikasi memiliki peran dan tujuan mereka masing-masing sesuai dengan kategori yang mereka miliki.

2.2 Warna

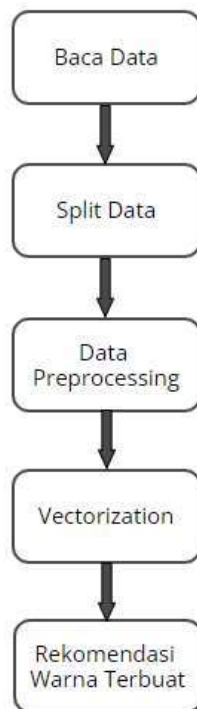
Warna adalah komponen yang tidak dapat dipisahkan dari kehidupan sehari-hari kita. Ini juga berkaitan dengan kemajuan teknologi dan memiliki dua bentuk dasar: warna cahaya (spektrum) sebagai unsur aditif dan warna bahan (pigmen) yang terkait dengan materi. Selain itu, warna juga membawa makna, simbol, dan emosi yang terkait dengan psikologi warna, dan ini berhubungan dengan berbagai bidang ilmu seperti filsafat, seni, agama, kepribadian, semiotika, dan hermeneutika. Dalam konteks seni visual, warna merupakan elemen dasar yang setara dengan garis, bentuk, tonalitas, pola, dan tekstur. Sementara dalam dunia film, warna berfungsi sebagai pendukung elemen visual yang meliputi pencahayaan, setting, gerakan karakter, ekspresi, kostum, dan tata rias karakter. (Paksi, 2021).

2.3 *Machine Learning*

Machine Learning adalah penerapan komputer dan algoritma matematika yang mengadopsi pendekatan pembelajaran dari data untuk menghasilkan prediksi di masa depan. Proses pembelajaran ini melibatkan dua tahap penting, yaitu pelatihan (*training*) dan pengujian (*testing*). *Machine Learning* merupakan subbidang dalam ilmu kecerdasan buatan (*Artificial Intelligence*) yang mendalam, digunakan secara luas untuk mengatasi berbagai masalah, dan dapat diklasifikasikan menjadi tiga kategori utama: *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Ulasan dari berbagai bidang mencerminkan beragam pemecahan masalah dan algoritma, dengan penekanan khusus pada perkembangan terbaru di bidang kedokteran dan medis, meskipun berbagai sektor seperti industri, teknologi, dan transportasi juga berperan penting (Roihan dkk., 2020).

Machine Learning membawa manfaat yang besar dalam menyelesaikan berbagai masalah dan memberikan kemudahan dalam berbagai tugas. Contohnya, dalam konteks rumah sakit dan kesehatan, *Machine Learning* memungkinkan dokter untuk dengan cepat mendiagnosa penyakit jantung tanpa memerlukan waktu yang lama. Semakin berkembangnya informasi tentang *Machine Learning*, semakin jelas bahwa teknologi ini memiliki kemampuan untuk belajar sendiri

tanpa perlu pengawasan yang ketat setiap kali digunakan (Telaumbanua dkk., 2020), dan untuk diagram alirnya dapat dilihat pada gambar 2.



Gambar 2. Diagram Alir *Machine Learning* untuk rekomendasi warna

1. Membaca Data

Data akan dibaca oleh *Machine Learning* dengan format CSV file, kemudian diproses agar menjadi data yang siap untuk dilatih.

2. *Split Data*

Split data merupakan proses untuk membagi sekumpulan data menjadi data latih dan data uji. Pembagian data tersebut berlaku untuk setiap teks.

3. *Data Preprocessing*

Data Preprocessing adalah proses penting sebelum data siap untuk dilatih, berikut adalah beberapa cara untuk melakukan *Data Preprocessing*.

a. *Cleaning data*

Cleaning data adalah proses untuk mengolah *dataset* agar tiap huruf menjadi non kapital dan menghilangkan tanda baca.

b. *Remove Stopwords*

Remove Stopwords merupakan proses untuk menghilangkan *stopwords* dari setiap kalimat. *Stopwords* berbahasa Indonesia dimuat oleh *sub-library stopwords* yang terdapat dalam *library nltk*, selanjutnya *stopwords* tersebut digunakan untuk menghilangkan kata hubung.

c. *Tokenizing*

Tokenizing adalah proses untuk memberikan sebuah identifier khusus berbentuk numerik bagi setiap kata, kemudian angka/token tersebut disusun sesuai susunan kalimat yang terdapat pada *dataset*.

d. Membuat *sequences* dan melakukan *padding*

Untuk mempersiapkan data urutan dalam *Machine Learning*, langkah pertama adalah membagi data, seperti teks, menjadi bagian-bagian kecil seperti kata-kata. Ini membantu model memahami struktur urutan informasi dalam data. Selanjutnya, dilakukan *padding* dengan menambahkan nilai khusus (biasanya 0) pada *sequences* yang lebih pendek untuk menyamakan panjang dengan *sequences* terpanjang. Hal ini penting karena model *Machine Learning* membutuhkan input dengan dimensi yang konsisten. Dengan melakukan *sequences* dan *padding*, data urutan siap digunakan oleh model untuk pelatihan dan prediksi.

4. *Vectorization*

Vektorisasi dengan metode TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah proses untuk mengubah teks atau dokumen menjadi representasi vektor berdasarkan bobot yang dihitung dari frekuensi kata dalam dokumen dan pentingnya kata dalam keseluruhan dokumen.

5. Rekomendasi warna terbuat

Setelah proses prediksi selesai maka hasil rekomendasi warna akan keluar sesuai masukan user.

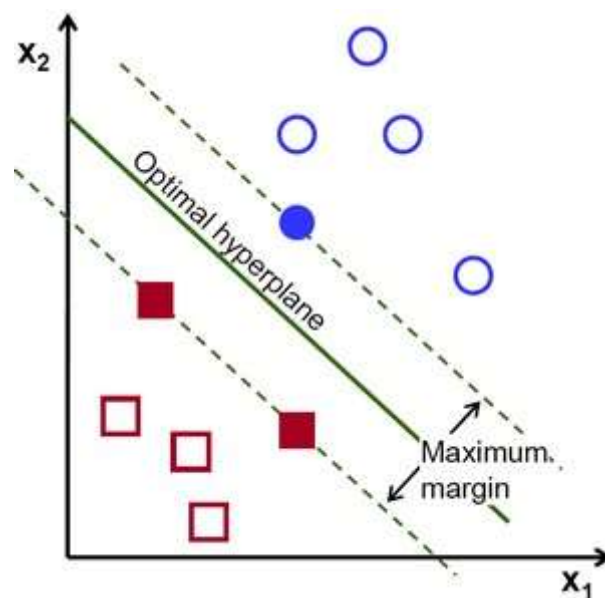
2.4 *Sentiment Analysis*

Sentiment analysis adalah metode yang digunakan untuk secara otomatis memahami, mengekstrak, dan menganalisis data teks dengan tujuan untuk

mengidentifikasi sentimen atau perasaan yang terkandung dalam kalimat opini. Teknik ini berguna untuk menggambarkan pandangan atau kecenderungan opini individu terhadap suatu topik atau objek, apakah itu bersifat positif atau negatif (Rahutomo dkk., 2018).

2.5 *Support Vector Machine*

Support Vector Machine (SVM) diperkenalkan untuk pertama kalinya oleh Vapnik pada tahun 1992 sebagai salah satu metode pembelajaran yang berlandaskan prinsip *Structural Risk Minimization* (SRM). Tujuan utama dari SVM adalah untuk menemukan *hyperplane* optimal yang dapat memisahkan dua kelas data dalam ruang input. Metode ini bekerja dengan menggunakan fungsi linier dalam ruang fitur yang memiliki dimensi yang lebih tinggi, dan menerapkan prinsip pembelajaran statistik. Tingkat akurasi dari model yang dihasilkan oleh SVM sangat tergantung pada jenis fungsi kernel yang digunakan serta parameter-parameter yang dikonfigurasi (Parapat & Furqon, 2018), dan untuk gambar *Support Vector Machine* (SVM) dapat dilihat pada gambar 3.



Gambar 3. Algoritma SVM (Sumber : Trivusi, 2022)

2.6 Aplikasi *Mobile*

Aplikasi *mobile* adalah perangkat lunak yang secara khusus dirancang untuk digunakan pada platform *mobile* seperti *iOS*, *Android*, atau *Windows Mobile*. Aplikasi *mobile* seringkali memiliki antarmuka pengguna yang dirancang khusus dengan cara berinteraksi yang sesuai dengan karakteristik platform *mobile*. Mereka juga dapat berinteraksi dengan sumber daya berbasis web untuk mengakses berbagai informasi yang relevan dengan aplikasi, dan memiliki kemampuan untuk memproses data secara lokal untuk pengumpulan, analisis, dan penampilan data sesuai dengan kebutuhan platform *mobile*. Selain itu, aplikasi *mobile* dapat menyimpan data secara persisten di dalam platform tempat mereka berjalan (Jainuri dkk., 2022).

2.7 *Android Studio*

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) yang menjadi standar resmi untuk pengembangan aplikasi *Android* dan tersedia secara gratis sebagai *open source*. *Google* mengumumkan peluncuran *Android Studio* pada tanggal 16 Mei 2013 dalam acara *Google I/O Conference* tahun 2013. Sejak saat itu, *Android Studio* menggantikan peran *Eclipse* sebagai IDE utama untuk membuat aplikasi *Android* (Fansuri, 2015).

2.8 *Python*

Python adalah salah satu bahasa pemrograman yang digunakan secara luas oleh perusahaan besar dan pengembang perangkat lunak untuk membuat berbagai jenis aplikasi, termasuk yang berbasis *desktop*, *web*, dan *mobile*. Bahasa pemrograman ini awalnya dikembangkan oleh Guido van Rossum di Belanda pada tahun 1990, dan namanya diambil dari salah satu acara televisi favorit Guido, yaitu *Monty Python's Flying Circus*. Awalnya, *Python* dikembangkan sebagai hobi pribadi oleh van Rossum, namun kemudian menjadi salah satu bahasa pemrograman yang sangat populer di dunia industri dan pendidikan karena kesederhanaannya, strukturnya yang ringkas, sintaksis yang mudah dipahami,

serta berbagai pustaka yang luas tersedia untuk penggunaannya (Romzi & Kurniawan, 2020).










2.9 Java Development Kit

Java Development Kit (JDK) diperlukan dalam pengembangan aplikasi Android karena bahasa pemrograman Android menggunakan bahasa *Java*. JDK berperan penting dalam proses penulisan kode program. Seperti *Java Runtime Environment* (JRE), JDK juga menyediakan *Java Virtual Machine* (JVM) untuk menjalankan program *Java*. JDK merupakan suatu lingkungan eksekusi program yang beroperasi di atas sistem operasi. Para pengembang memerlukan JDK untuk melakukan kompilasi kode, memperbaiki *bug*, dan menjalankan program yang ditulis dalam bahasa pemrograman *Java*. Teknologi *Java* memiliki tiga komponen utama, yaitu spesifikasi bahasa pemrograman, antarmuka pemrograman aplikasi, dan spesifikasi mesin virtual. JDK diperlukan khususnya dalam pengembangan fitur-fitur *Java*, tetapi tidak diperlukan untuk pengembangan dalam bahasa pemrograman lain. Versi JDK yang dapat digunakan untuk mengembangkan aplikasi Android meliputi JDK 5, JDK 6, atau versi yang lebih baru (Ramadhan, 2018).

2.10 Flowchart

Flowchart adalah representasi grafis dari langkah-langkah dan urutan prosedur dalam sebuah program atau proses tertentu. Biasanya, ini digunakan untuk membantu pemahaman dan evaluasi lebih lanjut dalam penyelesaian masalah tertentu. *Flowchart* digunakan untuk menggambarkan berbagai jenis aktivitas, termasuk kegiatan manual, proses pemrosesan data, atau kombinasi dari keduanya (Budiman dkk., 2021). Tabel 1 menjelaskan simbol-simbol yang digunakan dalam *flowchart* beserta penjelasannya.

Tabel 1. Simbol *Flowchart*

Simbol	Nama	Fungsi
	Terminator	Permulaan atau akhir program
	Garis Alir (<i>Flow Line</i>)	Arah aliran program
	<i>Preparation</i>	Proses inisialisasi/proses pengolahan data
	Proses	Proses perhitungan/pengolahan data
	<i>Input/Output Data</i>	Proses <i>input/output</i> data Parameter, informasi
	<i>Predefined Process (Sub Program)</i>	Permulaan subprogram/proses menjalankan subprogram
	<i>Decision</i>	Perbandingan pernyataan, penyelesaian data yang memberikan pilihan untuk langkah selanjutnya
	<i>Display</i>	Menunjukkan langkah yang menampilkan informasi
	<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda

Sumber : (Fauzi, 2020)

2.11 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah sebuah model atau teknik pendekatan yang digunakan untuk menggambarkan hubungan dalam suatu model. ERD fokus pada penggambaran objek data (*Entity*) dan hubungan (*Relationship*) yang ada di antara entitas tersebut. Proses ini membantu analisis dalam merancang struktur basis data yang efisien untuk penyimpanan dan pengambilan data. ERD menggunakan simbol-simbol tertentu untuk menggambarkan elemen-elemen dalam diagramnya, yang meliputi:

1. Entitas: Representasi objek yang dapat berupa entitas fisik atau konseptual yang memiliki karakteristik untuk menyimpan data.
2. Atribut: Ciri-ciri umum yang dimiliki oleh sebagian besar atau seluruh instansi pada entitas tertentu.
3. Relasi: Hubungan alamiah yang terjadi antara satu atau lebih entitas.
4. *Link*: Garis penghubung yang menghubungkan atribut dengan entitas dan entitas dengan relasi (Fridayanthie & Mahdiati, 2016).

2.12 Cloud Computing

Cloud Computing adalah teknologi yang mengandalkan internet dan server yang terpusat dan terletak jauh untuk mengelola dan menyimpan data pelanggan. Teknologi ini memungkinkan konsumen dan bisnis untuk menggunakan aplikasi tanpa perlu melakukan instalasi, serta mengakses file pribadi mereka dari berbagai lokasi melalui koneksi internet. Pendekatan ini menciptakan efisiensi dengan mengkonsolidasikan penyimpanan, pemrosesan, dan pengelolaan data. Selain itu, salah satu keunggulan utama dari *Cloud Computing* adalah kemampuannya untuk meningkatkan produktivitas bisnis pelanggan dengan menghilangkan kebutuhan untuk berinvestasi dan mengelola infrastruktur pusat data sendiri (Dewi, 2016).

2.13 Google Cloud Platform

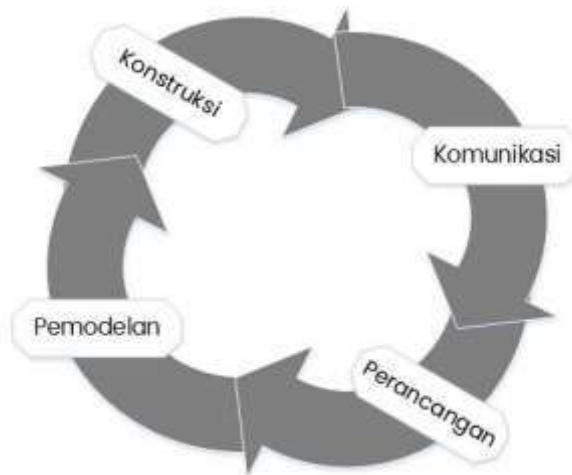
Google Cloud Platform (GCP) adalah platform komputasi awan publik yang disediakan oleh *Google* dan terdiri dari berbagai layanan platform. Pengguna GCP dapat membuat mesin virtual dengan cepat dan mudah (Falaq dkk., 2021).

2.14 Black-box Testing

Black-box Testing adalah metode pengujian perangkat lunak yang menekankan pada pengujian berdasarkan spesifikasi fungsional perangkat lunak. Dalam *Black-box Testing*, struktur internal atau kontrol perangkat lunak diabaikan sepenuhnya, dan perhatian utama tertuju pada informasi domain. Dengan menggunakan *Black-box Testing*, pengembang perangkat lunak dapat membuat serangkaian kondisi input yang akan mencakup semua syarat fungsional yang diperlukan oleh program tersebut (Jaya, 2018).

2.15 Metode *Prototype*

Prototyping adalah sebuah metode pengembangan perangkat lunak yang melibatkan pembuatan model kerja sistem yang berfungsi sebagai versi awal dari sistem yang akan dikembangkan. Dengan menggunakan metode *prototyping*, tujuan utamanya adalah menciptakan sebuah prototipe sistem yang akan menjadi perantara antara pengembang dan pengguna, sehingga keduanya dapat berinteraksi dalam proses pengembangan sistem informasi. Penting untuk mendefinisikan aturan-aturan di tahap awal agar pengembang dan pengguna memiliki pemahaman yang sama bahwa prototipe dibangun untuk menentukan kebutuhan awal (Aulya dkk., 2022), dan untuk gambar metode *prototype* dapat dilihat pada gambar 4.



Gambar 4. Metode *Prototype*