

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Penelitian Terdahulu**

Dalam penelitian ini penulis tidak lepas dari adanya penelitian terdahulu yang digunakan sebagai acuan dalam menyelesaikan permasalahan yang akan diteliti. Beberapa peneliti telah melaksanakan penelitian menggunakan objek masalah yang sejenis. Penelitian terdahulu yang dirujuk dijabarkan sebagai berikut

#### **Perancangan Sistem Informasi Pencarian dan Pemesanan Rumah Kos Berbasis WEB (Studi Kasus: Kota Bandar Lampung)**

Penelitian ini berjudul Perancangan Sistem Informasi Pencarian dan Pemesanan Rumah Kos Berbasis WEB (Studi Kasus: Kota Bandar Lampung). Pada saat ini di daerah bandar lampung, masih ada masyarakat yang belum memanfaatkan teknologi informasi untuk pencarian dan pemesanan rumah kos. Sistem yang sering dipakai, mereka masih harus bertanya kepada teman atau bertanya langsung kepemilik rumah kos untuk membandingkan letak, harga, fasilitas.

Penulis merancang menggunakan pengujian dengan metode ISO 9126 ditinjau dari aspek *functionality* dan *usability*. Tujuan pengujian ini adalah pembuatan sebuah WEB agar pencari rumah kos menemukan kos yang sesuai dengan kriteria mereka dan memudahkan pemilik mempromosikan kos (Budiman et al., 2019).

### **Analisis Sentimen untuk Komentar pada Sistem Pencarian Kos Menggunakan Metode *Support Vector Machine* (SVM).**

Penelitian ini berjudul Analisis Sentimen untuk Komentar pada Sistem Pencarian Kos Menggunakan Metode *Support Vectot Machine* (SVM). Penulis menggunakan metode SVM.

Kurangnya informasi berupa tempat penginapan berupa kos yang ada di daerah kampus UKSW. Tujuan penelitian ini adalah membantu mahasiswa maupun orang lain dalam mencari kos di daerah kampus UKSW berdasarkan kategori jenis, lokasi, dan harga serta memberikan informasi tentang kos yang diinginkan. Dan sistem ini akan membantu para pemilik kos untuk mempromosikan kos miliknya (Suryanputra et al., 2017).

### **Pengembangan Aplikasi Pencarian Tempat kos dan Rumah Kontrakan di Manado Berbasis Web**

Penelitian ini berjudul pengembangan aplikasi pencarian tempat kos dan rumah kontrakan di Manado berbasis web. Kurangnya informasi berupa tempat penginapan berupa kos atau kontrakan yang ada di Manado. Ada juga kriteria yang mereka pakai dalam memilih tempat kos atau kontrakan tersebut seperti mencari tempat yang murah, memiliki fasilitas yang memadai, atau dekat dengan tempat tujuan mereka.

Penelitian ini metode RAD dalam membuat siklus pengembangan sistem dan menggunakan XAMPP dan MySQL sebagai basis data dari pembuatan sistem, penulis juga memakai beberapa bahasa pemrograman yang mendukung pembuatan sistem ini yakni HTML,PHP,CSS, maupun Javascript. Tujuan penelitian adalah untuk digunakan oleh masyarakat sperti mahasiswa atau pekerja dalam



memilih tempat kos atau kontrakan yang sesuai dgn kebutuhan mereka (Suwito et al., 2020).

## **Perancangan Sistem Informasi Pencarian Rumah Kos Kabupaten Rembang Berbasis web**

Penelitian ini berjudul perancangan sistem pencarian rumah kos Kabupaten Rembang berbasis web. Letak geografis Kabupaten Rembang yang terletak di pesisir pantai yang sangat strategis serta dilalui Jalan Pantura sehingga banyak aktivitas pembukaan tempat wisata dan merupakan jalan raya utama yang dilalui oleh masyarakat yang sedang melakukan dinas kerja keluar kota. Melihat dari letak geografis Kabupaten Rembang tersebut maka banyak masyarakat dalam maupun luar kota yang ingin berwisata, melakukan bisnis, melanjutkan bersekolah atau hanya sekedar ingin mencari penginapan untuk beristirahat dalam melakukan dinas kerja keluar kota, maka terbentuklah aplikasi pencarian informasi rumah kos berbasis web yang dikembangkan menggunakan bahasa pemrograman PHP dan database MySQL. Penelitian ini menggunakan metode *System Development Life Cycle* (SDLC).

Tujuan penelitian ini pengguna dapat mudah menemukan rumah kos wilayah Kabupaten Rembang yang sesuai dengan kebutuhan (Basyah et al., 2022).

## **Sistem Pencarian Rumah Kos Berbasis Web**

Penelitian ini berjudul Sistem Pencarian Rumah Kos Berbasis Web. Metode penelitian yang digunakan adalah observasi secara langsung dan pengembangan sistem menggunakan metode *waterfall*. Tujuan penelitian ini adalah menghasilkan sistem dengan akses cepat dan akurat untuk pencarian maupun promosi rumah kost. Metode penelitian yang digunakan adalah observasi secara langsung dan pengembangan sistem menggunakan metode *waterfall*.



Sistem informasi pencarian rumah kost berbasis web dapat mempermudah user dalam melakukan pencarian dan penyewaan tanpa harus datang langsung ke lokasi sehingga menghemat waktu dan biaya serta membuat pemilik kost dapat lebih luas mempromosikan rumah kost miliknya (Yusma et al., 2021)

Dari penjelasan penelitian terdahulu diatas,maka diperoleh rincian dan yang dapat dilihat pada tabel 2.1



Tabel 2. 1 Penelitian terdahulu

No	Judul	Peneliti	Perbedaan	Persamaan	Hasil
1.	Perancangan Sistem Informasi Pencarian dan Pemesanan Rumah Kos Berbasis WEB (Studi Kasus: Kota Bandar Lampung)	Arief Budiman, Lara Sri Wahyuni, Suharsono Bantun (2019)	pada sistem pencarian menggunakan maps lokasi jarak terdekat	Peneliti merancang sistem informasi pencarian dan pemesanan rumah kos berbasis web.	Pencari kos dapat melakukan pencarian berdasarkan nama kota, daerah jenis kos, waktu sewa, fasilitas dan harga

2.	Analisis Sentimen untuk Komentar pada Sistem Pencarian Kos Menggunakan Metode <i>Support Vector Machine</i> (SVM).	Fandy Oktavianus Suryaputra, Banu W. Yohanes, Saptadi Nugroho (2018).	Perhitungan Pencarian rumah kos dengan perhitungan djikstra daerah kabupaten jombang	Peneliti menggunakan fitur informasi mengenai lokasi, harga, serta fasilitas yang disediakan.	Sistem yang berisi informasi mengenai lokasi, harga, serta fasilitas yang ditawarkan oleh tempat penyedia kos. terdapat sistem rating yang berasal dari hasil pengolahan SVM yang berfungsi untuk menentukan apakah komentar tentang tiap kos bersifat positif atau negatif.
----	--	---	--	---	--

3.	Pengembangan Aplikasi Pencarian Tempat kos dan Rumah Kontrakan di Manado Berbasis Web. Menggunakan metode RAD.	Angga Qurnen Suwito, Sartje Silimang, Alwin Melkie Sabmul (2020).	Metode yang digunakan dijikstra	Penelitian ini merancang sistem berbasis web	Hasil dari perancangan dan pembangunan dari pencarian tempat kos dan rumah kontrakan di Manado berbasis website adalah untuk digunakan oleh masyarakat seperti mahasiswa atau pekerja dalam memilih tempat kos atau kontrakan yang sesuai dengan kebutuhan mereka.
----	--	---	---------------------------------	--	--

4.	Perancangan sistem informasi pencarian rumah kos kabupaten rembang berbasis web. Metode yang digunakan System Development Life Cycle (SDLC)	Baby Lolita Basyah a, Elvina Djamahar dkk (2022)	Menggunakan metode djikstra untuk mencari rute kos terdekat	Sistem yang di rancang peneliti berbasis web	Pengguna dapat mudah menemukan rumah kos wilayah Kabupaten Rembang yang sesuai dengan kebutuhan dan dapat mengakses <a href="https://kosrembang.000webhostapp.com">https://kosrembang.000webhostapp.com</a> .
5.	Sistem Pencarian Rumah Kos Berbasis Web	Destiana Yusma, Nita Merlina,	Perhitungan Pencarian rumah kos dengan perhitungan	Peneliti merancang sistem informasi	Sistem informasi pencarian rumah kos berbasis web dapat mempermudah <i>user</i> dalam melakukan pencarian dan penyewaan tanpa harus datang



	Menggunakan metode Waterfall	Nurajijah (2021).	dijkstra daerah kabupaten jombang	pencarian dan pemesanan rumah kos berbasis web.	langsung ke lokasi sehingga menghemat waktu dan biaya serta membuat pemilik kos dapat lebih luas mempromosikan rumah kos miliknya
--	------------------------------	-------------------	-----------------------------------	---	---

Dengan memperhatikan beberapa penelitian terdahulu sebagai bahan acuan yang masih berkaitan dengan sistem informasi pencarian rumah kos, bahwa sebagian permasalahan yang dapat diangkat dari beberapa penelitian tersebut dan diakibatkan karena kurang optimalnya masyarakat dalam menemukan solusi untuk mencari kos dari destinasi terdekat. Oleh sebab itu, peneliti membangun sistem informasi pencarian rumah kos dengan menggunakan metode dijkstra dengan tujuan untuk memudahkan dan mengefisiensi waktu pencarian rumah kos

## 2.2 Kajian Pustaka

### 2.2.1 Sistem Informasi

Sistem informasi dapat diartikan sebagai sistem yang menghasilkan output berupa informasi yang berguna bagi tingkatan manajemen. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi suatu organisasi dan menyediakan pihak luar tertentu dengan laporan - laporan yang diperlukan (Jogiyanto, 2009).

Untuk menghasilkan informasi, suatu sistem informasi harus mempunyai enam komponen, yaitu:

- a. Komponen *input*, komponen ini merupakan bahan dasar pengolahan informasi karena *input* merupakan data yang masuk ke dalam sistem.
- b. Komponen *output*, merupakan produk sistem informasi. *Output* sistem informasi harus berupa informasi yang berguna bagi pemakainya.
- c. Komponen basis data, yaitu komponen data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.



- d. Komponen model, komponen yang menunjukkan pengolahan data lewat suatu model-model tertentu untuk menghasilkan informasi yang dibutuhkan.
- e. Komponen Teknologi, komponen yang berfungsi untuk mempercepat pengolahan data.
- f. Komponen Kontrol, komponen ini digunakan untuk menjamin bahwa informasi yang dihasilkan oleh sistem informasi merupakan sistem informasi yang akurat.

### 2.2.2 Rumah Kos

Rumah Kos merupakan kamar atau tempat tinggal yang disewakan dengan bayaran tertentu untuk setiap bulan atau tahunnya. Tempat kos lebih akrab digunakan sebagai domisili, karena kebanyakan tempat kos disewa dalam jangka waktu yang cukup lama. Biasanya alasan memilih tempat kos sebagai tempat tinggal adalah karena sekolah atau kuliah dalam jangka pendek ataupun lama, karena pekerjaan, dan karena biaya bangunan dan tanah yang mahal maka tempat kos menjadi pilihan masyarakat sementara (Abidin,2019).

### 2.2.3 Database Management System (DBMS)

DBMS merupakan kumpulan data yang terstruktur. Sistem manajemen basis data diperlukan untuk dapat memanipulasi atau mengolah data yang tersimpan dalam basis data komputer. MySQL merupakan sistem manajemen database relasional (RDBMS) open source berbasis SQL (*Structured Query Language*) artinya MySQL sudah menjadi bahasa standart dalam dunia database yang bekerja dengan model client-server **Invalid source specified..**

Terdapat 2 macam SQL yang terdapat dalam Mysql, yaitu DDL (*Data Definiton Language*) dan DML (*Data modeling Language*). *Data Definition Language* (DDL) adalah bahasa yang mengizinkan Database Administrator atau user untuk



mendeskripsikan dan memberi nama entitas, atribut, dan hubungan yang diperlukan aplikasi beserta integrity yang berhubungan dan batasan keamanan. Sedangkan *Data Modeling Language* (DML) adalah bahasa yang menyediakan seperangkat operasi untuk mendukung manipulasi data yang berada pada basis data. MySQL memiliki beberapa keunggulan diantaranya:

1. *Portability*  
Dapat berjalan dengan stabil di semua sistem operasi computer seperti windows, mac, linux, dll.
2. *Multi-user*  
Mysql dapat dijalankan oleh lebih dari satu user dalam waktu yang bersamaan tanpa terjadi error atau konflik.
3. *Performance Turning*  
Mysql memiliki kecepatan menakjubkan dalam melakukan pemrosesan query sederhana.
4. *Column Type*  
Memiliki tipe kolom yang sangat banyak seperti signed, unsigned, integer, float, double, char, varchar, timestamp, dll.
5. *Security*  
Mysql memiliki tingkat keamanan yang terdiri dari beberapa lapisan seperti level subnet mask, nama host dan izin akses user.

#### **2.2.4 Algoritma Dijkstra**

Algoritma dijkstra ditemukan oleh Edger Wybe Dijkstra pada tahun 1959. Algoritma ini merupakan algoritma yang dapat memecahkan masalah pencarian jalur terpendek dari suatu graf pada setiap simpul yang bernilai tidak negatif. Dijkstra merupakan algoritma yang termasuk dalam algoritma *greedy*, yaitu algoritma yang sering digunakan untuk mencegah masalah yang berhubungan dengan suatu optimasi. Dalam pencarian jalur terpendeknya algoritma *dijkstra* bekerja dengan mencari bobot yang paling minimal



dari suatu graf berbobot, jarak terpendek akan diperoleh dari dua atau lebih titik dari suatu graf dan nilai total yang didapat adalah yang bernilai paling kecil.

#### 1. Cara kerja algoritma Dijkstra

Algoritma Dijkstra memakai strategi greedy, di mana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter berupa tempat asal dan tempat tujuan. Berikut cara kerja algoritma dijkstra

- A. Inisialisasi
  - 1) Tentukan titik awal, yaitu lokasi pengguna saat ini.
  - 2) Setel jarak awal dari titik awal ke semua titik lain menjadi tak terbatas (kecuali titik awal sendiri yang diatur menjadi 0).
  - 3) Setel semua titik menjadi belum dikunjungi.
- B. Iterasi
  - 1) Pilih titik dengan jarak terpendek yang belum dikunjungi.
  - 2) Periksa semua tetangga dari titik tersebut.
  - 3) Jika jalur melalui titik saat ini lebih pendek dari jarak yang sudah dihitung sebelumnya, perbarui jarak tersebut.
  - 4) Tandai titik tersebut sebagai dikunjungi.
- C. Ulangi langkah 2 sampai semua titik telah dikunjungi atau sampai titik tujuan dicapai.
- D. Setelah semua titik dikunjungi, jarak terpendek antara titik awal dan titik tujuan akan tercatat dalam variabel jarak. Algoritma ini juga akan menghasilkan jalur terpendek dari titik awal ke titik tujuan.
- E. Untuk memperoleh jalur terpendek, langkah mundur dari titik tujuan ke titik awal dengan memilih tetangga dengan jarak terpendek hingga mencapai titik awal
  - a. Kelebihan dan Kekurangan



Kelebihan dari algoritma dijkstra antara lain :

1. Pengambilan keputusan yang cepat.
  2. Efektif dan tidak memakan waktu.
  3. Mudah diterapkan dalam banyak masalah yang ada.
- C. Kekurangan dari algoritma dijkstra antara lain :
1. Hasil akhirnya buruk, karena tidak memikirkan konsekuensinya.
  2. Tidak dapat memantau parameter.
  3. Tidak ada pilihan lain jika masalah tidak dapat diselesaikan.

### **2.2.5 Unified Modelling Language (UML)**

*Unified Modelling Language* ialah sebuah bahasa grafis yang menjadi standar untuk *memvisualisasi*, mendokumentasikan, mendeskripsikan dan merancang sistem perangkat lunak (Pratama, Yulia, & Gunawan, 2021). Dalam membuat model konsep *UML*, terdapat beberapa aturan yang harus diikuti. Bagaimana elemen pada model-model yang dibuat saling berhubungan antara satu dengan yang lain. *UML* menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai persepektif. Desain *UML* diantaranya adalah *use case diagram*, *class diagram*, *sequence diagram* dan *activity diagram*.



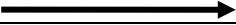

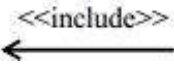
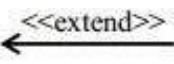
Desain *UML* diantaranya adalah *use case diagram*, *class diagram*, *sequence diagram* dan *activity diagram*

#### **1. Use Case Diagram**

*Use Case Diagram* merupakan permodelan untuk kelakua *Use Case Diagram* adalah deskripsi lengkap tipe interaksi antara si pengguna sistem dengan sistemnya. Fungsionalitas sistem digambarkan dengan sebuah *use case diagram*. Yang menjadi penekanan adalah “apa” yang diperbuat oleh sistem dan bukan “bagaimana”. *Use case* di presentasikan sebagai interaksi antara aktor dengan sistem. Notasi dari *Use Case Diagram* dapat dilihat pada table 2.3



Tabel 2. 2 Use Case Diagram



Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang dikembangkan.
	Use case : perangkat teringgi dari fungsionalitas yang dimiliki sistem.
	Association : adalah relasi antara aktor dan use case.
	Generalisasi : untuk memperhatikan struktur pewarisyang terjadi.
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi



## 2. Activity Diagram

*Activity Diagram* adalah Diagram yang menggambarkan aliran kerja (*workflow*) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Sukamto & Salahuddin, 2015).

Selain itu, diagram aktivitas juga dapat digunakan untuk mendefinisikan hal-hal yang lainnya seperti untuk rancangan proses bisnis, urutan atau pengelompokkan tampilan sistem, desain pengujian, dan desain interface yang akan ditampilkan pada perangkat lunak.

Tabel 2. 3 Activity Diagram

Simbol	Keterangan
	Simbol titik awal
	Simbol titik akhir

	<i>Activity</i>
	Pilihan untuk pengambilan keputusan



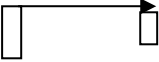


### 3. *Sequence Diagram*

*Sequence Diagram* merupakan sebuah kombinasi dari *diagram class* dan *object* yang mempunyai sebuah gambaran model statis. Walaupun demikian, ada juga *sequence diagram* yang sifatnya dinamis. Jenis diagram ini dapat menggambarkan bagaimana sebuah operasi dilakukan, pesan apa, yang akan dikirim dan kapan waktu pelaksanaannya.











Tabel 2. 4 Sequence Diagram

Simbol	Keterangan
	Aktor, menggambarkan pengguna sistem
	<i>Lifeline</i> , objek antarmuka yang saling berinteraksi.
	<i>Message</i> , spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi.
	<i>Message</i> , spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi.
	<i>SelfMessage</i> , menggambarkan aktifitas pesan pada aktor itu sendiri.

#### 4. Class Diagram

Diagram kelas (*class diagram*) merupakan diagram yang menggambarkan struktur sistem dalam hal mendefinisikan kelas-kelas yang akan dibuat untuk membangun sistem (Rosa AS, 2014).

Tabel 2. 5 Class Diagram

Simbol	Keterangan
	<i>Generalization:</i> untuk relasi antarkelas dengan makna generalisasi-spesialisasi.
	<i>Nary association:</i> Untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class:</i> kelas pada struktur sistem
	<i>Realization:</i> adalah operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency:</i> adalah relasi antarkelas dengan makna kebergantungan antar kelas.
	<i>Association:</i> adalah relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

### 2.2.6 Black box testing

*Black Box Testing* adalah metode testing yang berfokus pada pengujian dari masing-masing perangkat lunak. Seorang tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada fungsional perangkat lunak (Mustaqbal dkk., 2015). Dengan demikian, setiap input pada perangkat lunak dapat diperhatikan kinerjanya berdasarkan fungsi yang ingin dicapai. Berikut beberapa karakteristik dan aspek dari black box testing termasuk:

1. Tidak Memerlukan Pengetahuan Internal: Penguji tidak perlu mengetahui bagaimana kode sumber diimplementasikan atau detail teknis internal sistem.

2. Berdasarkan Spesifikasi: Pengujian dilakukan berdasarkan spesifikasi atau persyaratan fungsional yang telah ditentukan sebelumnya.
3. Fokus pada Fungsionalitas Eksternal: Black box testing menguji bagaimana sistem berfungsi dari luar, misalnya bagaimana sistem merespons input tertentu dan menghasilkan output yang diharapkan.
4. Tidak Mempedulikan Struktur Kode: Tidak ada perhatian terhadap struktur internal kode atau algoritma yang digunakan.
5. Input dan Output: Pengujian melibatkan memberikan input ke sistem dan memeriksa apakah output yang dihasilkan sesuai dengan harapan.
6. Tes Kasus Uji: Pengujian melibatkan pengembangan kasus uji berdasarkan skenario yang berbeda untuk menguji berbagai aspek fungsionalitas sistem.
7. Keuntungan: Black box testing dapat memberikan pandangan yang obyektif tentang bagaimana sistem beroperasi, mengidentifikasi masalah potensial dari perspektif pengguna akhir, dan mendorong perbaikan yang lebih umum.

### **2.2.7 Analisis Proses Bisnis**

Analisis proses bisnis merupakan suatu upaya mempelajari proses bisnis yang ada dalam satu atau beberapa organisasi, baik dalam operasi normal dan dalam situasi yang luar biasa (Djankov dkk, 2006). Tujuan analisis proses bisnis adalah untuk menganalisis dan mendokumentasikan kerja proses yang ada. Dalam kasus tertentu, mungkin akan membuat proses baru, dan tidak akan ada proses untuk dianalisis. Langkah-langkah umum dalam analisis proses bisnis termasuk:

- 1) Identifikasi Proses Utama: Identifikasi dan dokumentasikan proses utama dalam



organisasi(misalnya: pemasaran, produksi, penjualan, pengelolaan stok, pelayanan pelanggan, dst.).

- 2) Dokumentasi Alur Kerja: Gambarkan alur kerja dari awal hingga akhir untuk setiap proses. Dokumentasikan langkah-langkah, aktivitas, tugas, dan pihak yang terlibat.
- 3) Identifikasi Input dan Output: Kenali input dan output dalam proses bisnis. Ini membantu memahami pergerakan informasi dan produk di organisasi.
- 4) Kumpulkan Data: Kumpulkan data seperti waktu, biaya, dan sumber daya yang digunakan dalam setiap tahap proses.
- 5) Analisis Efisiensi: Temukan area di mana proses bisa lebih efisien, identifikasi bottleneck, dan masalah lainnya.
- 6) Pemetaan Teknologi: Kenali sistem dan teknologi yang digunakan dalam proses. Cari peluang untuk otomatisasi dan peningkatan efisiensi.
- 7) Identifikasi Peluang Perbaikan: Gunakan data dan informasi untuk temukan peluang perbaikan seperti pengurangan biaya, peningkatan kualitas, atau waktu respons yang lebih baik.
- 8) Desain Proses Baru: Rekomendasikan perubahan dalam proses bisnis untuk meningkatkan efisiensi dan kualitas. Rencanakan implementasi perubahan ini.
- 9) Implementasi dan Pemantauan: Terapkan perubahan yang direkomendasikan dan pantau hasilnya untuk memastikan pencapaian tujuan yang diharapkan.

### **2.2.8 Desain Basis Data**

Desain basisdata adalah organisasi data menurut model database. Perancang menentukan data apa yang harus disimpan dan bagaimana elemen data saling berhubungan. Dengan informasi ini, mereka dapat mulai menyesuaikan data dengan model database. Sistem manajemen basis data mengelola data yang sesuai. Langkah-langkah dalam



membangun basis data meliputi analisis kebutuhan bisnis, pemilihan model data, pemodelan skema, normalisasi, desain fisik, pemilihan DBMS, pengembangan aplikasi dan integrasi, pengujian dan optimalisasi, keamanan, serta pemeliharaan rutin.:

### **2.2.9 Desain User Interface**

Desain antarmuka pengguna (*User Interface Design*) adalah proses merancang tampilan grafis dan interaksi dari suatu aplikasi, situs web, atau perangkat lunak agar mudah digunakan, efisien, dan memberikan pengalaman pengguna yang positif. Tujuan utamanya adalah menciptakan antarmuka yang intuitif, menarik, dan efektif, sehingga pengguna dapat berinteraksi dengan aplikasi atau situs dengan lancar tanpa hambatan. Berikut adalah langkah-langkah umum dalam proses desain antarmuka pengguna:

1. **Pemahaman Kebutuhan Pengguna:** Pahami siapa target pengguna produk atau aplikasi Anda. Identifikasi kebutuhan, preferensi, dan harapan mereka terhadap antarmuka.
2. **Penelitian dan Analisis:** Lakukan penelitian tentang tren desain terbaru, persaingan, dan industri yang relevan. Analisis produk serupa atau kompetitor juga dapat memberikan wawasan berharga.
3. **Pengorganisasian Konten:** Tentukan bagaimana informasi dan konten akan diorganisir dalam antarmuka. Ini melibatkan perencanaan tata letak, struktur menu, dan hierarki informasi.
4. **Pemetaan Alur Penggunaan (User Flows):** Buat pemetaan visual tentang bagaimana pengguna akan berinteraksi dengan berbagai bagian antarmuka, mulai dari awal hingga tujuan akhir.
5. **Pembuatan Sketsa Kasar (Wireframing):** Buat sketsa kasar atau wireframe yang menunjukkan posisi elemen-elemen utama antarmuka, seperti tombol, teks, gambar,



dan ruang kosong. Wireframe membantu mengatur struktur dasar sebelum desain visual lebih detail.

6. **Desain Visual:** Buat desain visual yang lebih rinci dengan mengaplikasikan elemen-elemen grafis seperti warna, tipografi, ikon, dan gambar. Pastikan desain ini konsisten dengan identitas merek (branding) Anda.
7. **Interaksi dan Animasi:** Jika relevan, tentukan interaksi seperti animasi, efek transisi, atau respons terhadap tindakan pengguna. Ini dapat meningkatkan interaktivitas dan daya tarik antarmuka.
8. **Responsif dan Adaptable:** Pastikan desain antarmuka dapat beradaptasi dengan berbagai perangkat dan ukuran layar yang berbeda (responsif). Ini penting untuk memberikan pengalaman yang konsisten di berbagai platform.
9. **Pengujian Prototipe:** Buat prototipe interaktif dari desain Anda dan uji dengan pengguna potensial. Berdasarkan umpan balik, lakukan perubahan dan penyempurnaan jika diperlukan.
10. **Implementasi:** Setelah desain diuji dan disempurnakan, siapkan desain akhir untuk diimplementasikan oleh tim pengembangan. Pastikan elemen desain dieksekusi sesuai dengan rencana.
11. **Konsistensi Desain:** Pastikan desain antarmuka konsisten di seluruh produk atau aplikasi. Pengguna harus merasa familiar dengan tampilan dan pola interaksi.
12. **Pemantauan dan Perbaikan:** Setelah peluncuran, pantau interaksi pengguna dengan antarmuka dan reaksinya. Lakukan perbaikan berdasarkan umpan balik dan analisis data.

