

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

No.	Judul Jurnal	Penulis	Hasil Penelitian
1.	Rancang Bangun Aplikasi Pengelolaan Arsip Inaktif Dinas Kearsipan Dan Perpustakaan Daerah Kabupaten Grobogan Menggunakan Metode <i>Selection Sort</i> [5].	Susanto (2023)	Hasil penelitian ini menunjukkan bahwa Aplikasi Pengelolaan Arsip Inaktif Dinas Kearsipan dan Perpustakaan Daerah Kabupaten Grobogan dengan Metode <i>Selection Sort</i> telah memberikan kontribusi yang positif dalam meningkatkan efisiensi dan efektivitas pengelolaan arsip inaktif di institusi tersebut.
2.	Aplikasi Pengelolaan Surat Keluar Menggunakan <i>Sequential Search</i> Dan <i>Selection Sort</i> Pada Kpu Kota Blitar[6].	Haming, Nilal dkk., (2021)	Hasil dari penelitian ini menunjukkan bahwa penggunaan metode <i>sequential search</i> dan <i>Selection Sort</i> dalam aplikasi pengelolaan surat keluar KPU Kota Blitar telah berhasil. Dengan hasil pengujian yang sesuai dengan desain dan

algoritma yang direncanakan, aplikasi ini berhasil meningkatkan efisiensi dan efektivitas pengelolaan surat keluar di lingkungan KPU Kota Blitar.

3. Sistem Informasi Apotek Berbasis Web Menggunakan Algoritma *Sequential Search* Dan *Selection Sort*[7].. Lasriana & Gunaryati, Aris (2022) Hasil dari penelitian ini menunjukkan bahwa penerapan sistem informasi apotek dapat memberikan manfaat signifikan bagi pemilik dan karyawan apotek dalam mengelola laporan secara terkomputerisasi. Dengan menggunakan *Selection Sort* membantu mengurutkan data barang secara efisien.
4. Implementasi Algoritma *Selection Sort* Untuk Pengurutan Nilai Ipk Mahasiswa Universitas Potensi Utama[8]. Syahputra (2022) Hasil dari penelitian ini menunjukkan bahwa penerapan metode *Selection Sort* dalam pengurutan IPK mahasiswa kelas TIF B Universitas Potensi Utama efektif dalam menyusun laporan penilaian.

5. Implementasi Algoritma Harahap, Fandhepa Hasil dari penelitian ini *Selection Sort* dalam & Effendy, Irman menunjukkan bahwa Membangun Aplikasi (2023) terwujudnya sebuah Android Pemesanan Jasa aplikasi pemesanan jasa Make-up. make-up yang memanfaatkan algoritma *Selection Sort* untuk melakukan *sorting* pada pemesanan paket berdasarkan harga.

Berdasarkan tabel 2.1 diatas, dapat disimpulkan bahwa perbedaan utama pada penelitian ini yaitu objek penelitiannya yang mana penelitian ini berfokus mengimplementasikan *Selection Sort* Pada Surat Keluar Di Kelurahan Bulukerto Berbasis Web . Selanjutnya penelitian ini menggunakan pendekatan yang berbeda namun masih memiliki tujuan yang sama yaitu mengurutkan surat keluar di Kelurahan Bulukerto. Metode *Selection Sort* dipilih sebagai pendekatan khusus untuk mengurutkan surat keluar berdasarkan urutan keluar surat tersebut.

2.2 Sistem Pengelolaan Surat Keluar

Sistem pengelolaan surat keluar merupakan proses administratif yang mencakup pencatatan, pengarsipan, dan distribusi surat yang dikirim dari suatu organisasi. Proses ini bertujuan untuk memastikan surat keluar terdokumentasi dengan baik, dapat dilacak, dan mudah diakses saat diperlukan. Penggunaan teknologi informasi, seperti aplikasi berbasis web, dalam sistem pengelolaan surat keluar memungkinkan otomatisasi dalam pencatatan, penomoran, dan pengarsipan surat, yang meningkatkan efisiensi, akurasi, dan keamanan data. Selain itu, sistem ini memudahkan pencarian surat berdasarkan kriteria tertentu, seperti tanggal atau nomor surat, sehingga mempercepat proses administrasi dan mendukung pelayanan yang lebih baik[9].

2.3 Website

Website adalah sebuah halaman atau kumpulan halaman yang dapat diakses melalui internet. Halaman-halaman ini biasanya berisi informasi, konten multimedia, dan berbagai fitur interaktif yang dapat diakses oleh pengguna melalui perangkat seperti komputer, laptop, tablet, atau *smartphone*. *Website* dapat digunakan untuk berbagai tujuan, seperti memberikan informasi tentang suatu perusahaan, organisasi, produk, atau layanan. Selain itu, *website* juga dapat digunakan untuk berbagi konten seperti artikel, blog, foto, video, dan masih banyak lagi. Beberapa jenis *website* yang umum meliputi *website* bisnis, *website e-commerce*, blog pribadi, forum online, dan media sosial. *Website* biasanya terdiri dari berbagai elemen, seperti tata letak, teks, gambar, grafik, audio, video, dan *hyperlink* yang menghubungkan halaman-halaman tersebut. Desain visual dan fungsionalitas *website* dapat bervariasi tergantung pada kebutuhan dan preferensi pemilik *website*[10].

2.4 HTML

HTML singkatan dari *HyperText Markup Language*, merupakan sebuah bahasa pemrograman yang digunakan untuk membuat dan mengatur tampilan halaman web di *World Wide Web* (WWW). Semua dokumen HTML dapat dibaca dan dipublikasikan di web, dengan ekstensi file biasanya berakhir dengan *.html* atau *.htm*. Para ahli sepakat bahwa HTML adalah bahasa yang sangat tepat untuk menampilkan informasi pada halaman web karena menggunakan konsep *hypertext* dan menyediakan sejumlah perintah (markup) yang memungkinkan pengaturan tampilan informasi secara fleksibel sesuai kebutuhan. Dengan markup yang dimilikinya, HTML memungkinkan pengguna untuk menambahkan teks, gambar, link, dan elemen-elemen lainnya ke dalam halaman web, sehingga memungkinkan interaksi yang dinamis dan penyajian informasi yang menarik bagi pengguna internet[11].

2.5 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang kuat dan serbaguna yang digunakan untuk mengembangkan situs web dinamis.

Dikembangkan pertama kali pada tahun 1994 oleh Rasmus Lerdorf, PHP awalnya dirancang sebagai alat untuk mengelola situs web pribadinya, namun sejak itu telah berkembang menjadi salah satu bahasa pemrograman paling populer di dunia. Salah satu keunggulan utama PHP adalah kemampuannya untuk disematkan secara langsung ke dalam kode HTML, sehingga memungkinkan pengembang untuk membuat situs web yang dinamis dan interaktif dengan mudah. PHP juga mendukung berbagai basis data populer, seperti MySQL, PostgreSQL, dan SQLite, serta berbagai protokol jaringan seperti HTTP, FTP, dan IMAP. Dengan dukungan yang luas dari komunitas pengembang yang aktif, PHP terus berkembang dan menjadi pilihan utama dalam pengembangan aplikasi web modern[12].

2.6 MySQL

Seiring berkembangnya zaman, teknologi semakin berkembang pesat termasuk perangkat lunak. Salah satu contoh perangkat lunak adalah MySQL yang selalu di update oleh produsernya masing-masing. MySQL adalah pengembangan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TcX (perusahaan perangkat lunak asal Swedia). MySQL adalah DBMS yang open source dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database* server yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada. Seperti yang sudah disebutkan sebelumnya, MySQL masuk ke dalam jenis RDBMS (*Relational database Management System*). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah *database* terdapat satu atau beberapa tabel. MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database* SQL sebagai bahasa interaktif dalam mengelola data. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread, multi-user*[13].

2.7 Database

Basis data merupakan kumpulan data terorganisir yang saling terkait sehingga dapat disimpan, dimanipulasi, dan diakses dengan mudah oleh pengguna. Konsep relasi dalam hal ini mengacu pada cara data menggambarkan domain khusus, memfasilitasi penggunaan yang efektif dalam menjawab pertanyaan yang diajukan terhadap basis data tersebut. Sistem basis data, di sisi lain, merupakan koleksi data yang terstruktur sedemikian rupa sehingga memungkinkan penyimpanan dan manipulasi data (termasuk pembaruan, pencarian, pengolahan dengan perhitungan tertentu, dan penghapusan) dengan kemudahan[13].

2.8 Selection Sort

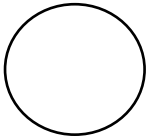
Selection Sort merupakan algoritma pengurutan sederhana yang bekerja dengan cara secara berulang memilih elemen terkecil dari sisa *Array* yang belum diurutkan, dan menukarnya dengan elemen pertama dari sisa *Array* tersebut. Proses ini diulangi untuk sisa *Array* yang semakin berkurang setiap iterasi, hingga seluruh *Array* diurutkan. Meskipun sederhana, *Selection Sort* memiliki kompleksitas waktu $O(n^2)$, yang membuatnya tidak efisien untuk jumlah data yang besar. Notasi $O(n^2)$ dalam konteks algoritma *Selection Sort* merujuk pada kompleksitas waktu algoritma tersebut. Dalam notasi Big O , $O(n^2)$ menunjukkan bahwa waktu eksekusi algoritma *Selection Sort* meningkat secara kuadrat seiring dengan peningkatan ukuran input (n), di mana n adalah jumlah elemen dalam *Array* yang diurutkan[2].

2.9 DFD (Data Flow Diagram)

Data Flow Diagram adalah representasi visual dari sistem yang menunjukkan hubungan antara sistem tersebut dengan entitas eksternal yang berinteraksi dengannya. Diagram ini sering digunakan untuk menggambarkan batasan sistem dan mengidentifikasi masukan (*input*) dan keluaran (*output*) yang terlibat dalam interaksi antara sistem dan entitas eksternal. Diagram konteks menggambarkan aliran informasi antara sistem dan entitas eksternal melalui *input* dan *output*. *Input* adalah masukan yang diberikan kepada sistem,

sedangkan output adalah hasil atau respons yang dihasilkan oleh sistem. Misalnya, *input* dapat berupa data yang dimasukkan oleh pengguna, sementara output dapat berupa tampilan hasil yang diberikan kepada pengguna[14].

Tabel 2.2 Komponen *Data Flow Diagram*

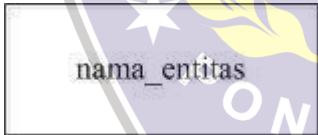
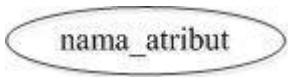
No	Simbol	Nama	Fungsi
1.		Entitas	Simbol ini mepresentasikan dari orang, organisasi, atau sistem yang berada di luar sistem yang sedang dianalisis, tetapi berinteraksi dengan sistem tersebut. Entitas ini bisa berupa pengguna, perangkat keras, atau sistem eksternal lainnya yang memberikan input ke sistem atau menerima output dari sistem.
2.		Arus Data	Simbol ini menggambarkan data tunggal atau kumpulan logis dari data yang mengalir dalam sistem. Arus data selalu bermula atau berakhir di suatu proses dalam DFD. Arus data menunjukkan bagaimana data diproses atau diolah oleh sistem.
3.		Proses	Proses adalah aktivitas atau fungsi dalam sistem yang melakukan operasi tertentu untuk tujuan bisnis yang spesifik. Proses bisa berupa aktivitas manual yang dilakukan oleh pengguna atau proses yang terkomputerisasi. Proses dalam DFD direpresentasikan dengan bentuk lingkaran atau bulatan.

Sumber : (Alhamda et al.,2023)

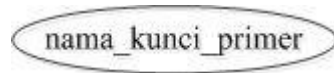
2.10 ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram adalah sebuah model konseptual yang digunakan untuk merancang basis data relasional. Dikembangkan berdasarkan teori himpunan dalam matematika, ERD memberikan representasi visual tentang entitas-entitas dan hubungan antar entitas dalam sebuah sistem basis data. Setiap entitas direpresentasikan sebagai tabel dengan atribut-atribut yang mencerminkan karakteristiknya. Hubungan antar entitas, seperti satu-ke-satu, satu-ke-banyak, atau banyak-ke-banyak, dijelaskan melalui garis yang menghubungkan tabel-tabel tersebut. ERD memiliki peran penting dalam proses perancangan dan pemahaman struktur basis data, membantu pengembang untuk mengidentifikasi dan memahami interaksi data serta menjaga integritas data melalui penetapan kunci primer dan kunci asing[15].

Tabel 2.3 Komponen ERD

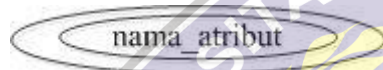
No.	Simbol	Deskripsi
1.	Entitas/ <i>Entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.

3. Atribut Kunci Primer



Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

4. Atribut Multinilai/Multi Value



Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

5. Relasi



Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.

6.

Assosiasi/association



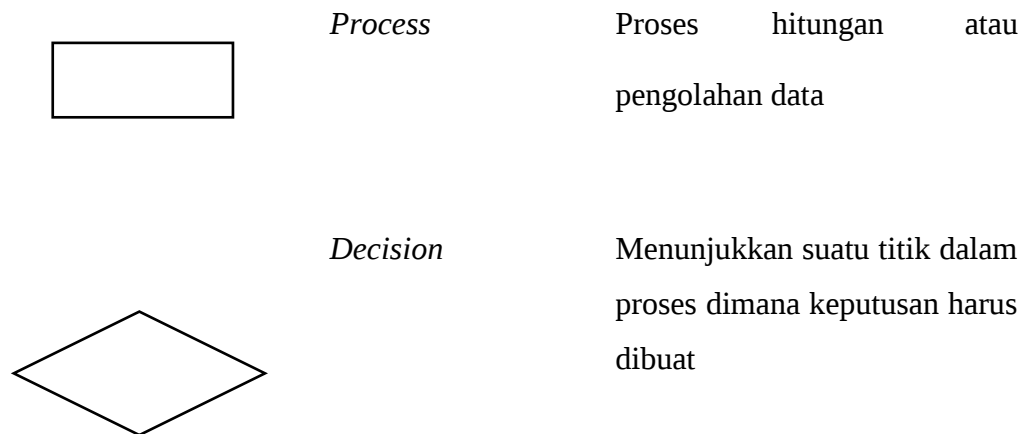
Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah Pemakaian kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B.

2.11 Flowchart

Flowchart atau diagram alir merupakan sebuah representasi visual yang digunakan untuk menunjukkan langkah-langkah instruksi atau algoritma secara berurutan dalam suatu sistem. Biasanya digunakan oleh analis sistem sebagai dokumentasi untuk menjelaskan logika dari sistem yang akan dibangun kepada para programmer. Dengan menggunakan *Flowchart*, solusi untuk berbagai masalah yang mungkin muncul dalam pembangunan sistem dapat diidentifikasi. *Flowchart* terdiri dari simbol-simbol yang mewakili berbagai proses, dan setiap simbol tersebut merepresentasikan suatu tindakan atau langkah. Garis penghubung digunakan untuk menghubungkan satu proses ke proses berikutnya, membentuk urutan logis dari instruksi yang diikuti dalam sistem tersebut[16].

Tabel 2.4 Komponen Flowchart

Simbol	Nama	Fungsi
	Terminator	Menandai awal dan akhir dari suatu alur kerja atau proses
	Garis Alir (Flow Line)	Arah aliran atau urutan dari suatu proses dalam <i>Flowchart</i>
	Input/Output Data	Merepresentasikan masukan atau keluaran dari suatu alur kerja atau proses



Sumber : (Khesya, 2021)

2.12 Metode *Waterfall*

Metode *Waterfall* dalam pengembangan perangkat lunak mengatur proses secara berurutan, dimana setiap langkah dilaksanakan berturut-turut dan tidak bisa dikembalikan ke tahap sebelumnya. Tahapan meliputi analisis kebutuhan perangkat lunak, desain, pembuatan kode program, dan pengujian. Analisis kebutuhan perangkat lunak mencakup pengumpulan informasi untuk menganalisis dan menetapkan kebutuhan perangkat lunak. Tahap desain melibatkan pembuatan desain program perangkat lunak, termasuk struktur data, arsitektur, dan representasi antarmuka, menggunakan notasi seperti *Activity Diagram*, *Use Case Diagram*, DFD, ERD, serta *Logical Record Structure*. Setelah itu, desain diubah menjadi program pada tahap pembuatan kode program, dan diikuti dengan tahap pengujian yang fokus pada pengujian logika dan fungsionalitas perangkat lunak menggunakan metode *blackbox*.

2.13 *White box testing*

White box testing adalah sebuah metode pengujian perangkat lunak di mana penguji memiliki akses ke dalam struktur internal, kode sumber, dan aliran program dari suatu aplikasi yang diuji. Dalam pengujian ini, penguji memahami struktur dan logika internal aplikasi untuk merancang

dan menjalankan serangkaian tes yang memeriksa apakah semua jalur logika program berjalan dengan benar.

Teknik-teknik yang umum digunakan dalam *white box testing* mencakup *Control Flow Testing*, yang memeriksa jalur-jalur yang diambil oleh program selama eksekusi, termasuk tes untuk semua keputusan logika yang ada dalam kode. *Data Flow Testing*, di sisi lain, memeriksa aliran data dalam program untuk menemukan kebocoran data, penggunaan data yang tidak valid, atau kondisi yang tidak terduga terkait dengan data. *Basic Path* atau *Path Testing* menguji semua jalur yang mungkin dalam program dari awal hingga akhir, melibatkan pengujian setiap jalur minimal satu kali. *Loop Testing*, pada gilirannya, memeriksa perilaku program saat melibatkan pengulangan atau *Looping*, termasuk pengujian untuk memastikan bahwa pengulangan berjalan dengan benar dan mengidentifikasi masalah seperti kondisi penghentian *Loop* yang tidak tepat[17].

