

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian yang dilakukan oleh Raafi dkk. (2024) bertujuan untuk merancang dan membangun sistem informasi inventaris penjualan berbasis *website* di toko konter handphone Reog Cell, Kabupaten Blitar. Permasalahan yang dihadapi oleh toko saat ini adalah kurangnya rincian dalam mencatat jumlah stok barang, terutama saat barang datang dalam jumlah besar, hal ini menyebabkan ketidak sesuaian antara catatan inventaris dan jumlah stok barang sebenarnya, terutama pada barang kecil. Pada penelitian ini menggunakan metode *waterfall*.

Pratama dkk. (2023) dalam penelitiannya yang bertujuan untuk menghasilkan sebuah sistem informasi pengelolaan inventaris aset yang mempunyai fitur untuk pencatatan aset dan pelaporan aset serta menguji penggunaannya. Untuk itu dengan proses pengolahan data inventaris tentunya harus dibantu dengan sebuah sistem yang mampu mengelolanya dengan lebih cepat dan efisien serta menghemat biaya dengan menggunakan metode *Rapid Application Development (RAD)*

Pada penelitian Gushelmi (2022) yang berjudul Penerapan Metode Rapid Application Development Dalam Membangun *Website* E-Commerce bertujuan untuk menghasilkan sistem penjualan *E-Commerce* yang akan membantu sistem penjualan berjalan, serta adanya pembukuan laporan yang jelas kepada pimpinan. Dengan adanya sistem baru ini diharapkan adanya perubahan tata kelola penjualan dan pembuatan laporan menjadi lebih baik menggunakan metode *Rapid Application Development (RAD)* menggunakan Bahasa pemrograman *PHP* dan *database MYSQL*.

Penelitian yang dilakukan oleh Butarbutar dkk. (2022) bertujuan untuk Tujuan dari sistem ini adalah untuk meningkatkan daya saing CV. Ramos Ponsel dengan mengimplementasikan Katalog Elektronik berbasis web yang

dapat diakses secara mudah dan efisien, menggantikan katalog cetak yang tidak efektif. Sistem ini bertujuan untuk memastikan pembaruan data katalog secara *real-time*, mencerminkan harga pasar terkini, serta menjangkau konsumen lebih luas. Dengan menggunakan metode *Waterfall* yang terstruktur, sistem diharapkan dapat memecahkan masalah yang ada, meningkatkan efisiensi operasional, dan memberikan pengalaman pengguna yang lebih baik melalui desain antarmuka yang intuitif dan pengujian sistem yang komprehensif.

Kemudian penelitian yang dilakukan oleh Profita dkk. (2022) bertujuan untuk melakukan transformasi sistem penjualan dari Risti Boutique yang sebelumnya dilakukan hanya secara *offline* menjadi *online* menggunakan *Content Management System (CMS) Prestashop* pada pembuatannya. Pengembangan sistem informasi menerapkan metode *Rapid Application Development (RAD)* yang memiliki tingkat kedinamisan yang tinggi dalam membangun sebuah sistem. Penelitian ini menghasilkan sistem informasi penjualan untuk manajemen pengelolaan toko Risti Boutique yang mampu mendokumentasikan segala hal terkait penjualan untuk membantu pihak toko dalam memasarkan produk dan mengembangkan usahanya.

Pada penelitian saat ini berjudul Rancang Bangun Sistem Manajemen Inventaris Dan Katalog Online Dengan Metode *Rapid Application Development* (Studi Kasus Butik Nela Busana) yang bertujuan untuk mengembangkan sistem yang dapat meningkatkan efisiensi operasional Butik Nela Busana dengan merancang sistem manajemen inventaris berbasis *website* untuk mempermudah pengelolaan stok barang secara otomatis dan akurat, serta menyediakan katalog online yang memudahkan pelanggan mengakses koleksi produk. Sistem yang peneliti kembangkan berbeda karena tidak hanya menyelesaikan masalah pencatatan inventaris dan katalog produk, tetapi juga memperkenalkan pembaruan stok otomatis, pengelolaan varian produk, rekomendasi produk yang disesuaikan dengan perilaku atau profil pelanggan dan memastikan data ukuran pelanggan tersimpan dengan aman.

Penelitian ini menggunakan metode Rapid Application Development (RAD) yang mengutamakan fleksibilitas dan kecepatan pengembangan.

Tabel 2. 1 Penelitian Terdahulu

Penulis	Judul Penelitian	Permasalahan	Metode Penelitian	Environment	Hasil
(Raafidkk., 2024)	Perancangan Sistem Informasi Inventaris Penjualan Berbasis Web Di Toko Konter Handphone Reog Cell	Kurangnya rincian dalam mencatat jumlah stok barang sehingga ketidaksesuaian antara catatan inventaris dan jumlah stok barang sebenarnya	Metode <i>waterfall</i>	Toko Konter Handphone Reog Cell	91% pengguna menganggap aplikasi ini berhasil dalam mengelola jumlah stok barang.
(Pratama dkk., 2023)	Penerapan Metode Rapid Application Development (RAD) pada Pengembangan Sistem Informasi	Pendataan serta pelaporan yang masih menggunakan sistem konvensional	Metode <i>Rapid Application Development (RAD)</i>	Desa Rempoh	Sistem ini mempermudah pemerintadesa dalam hal mengelola aset desa yang

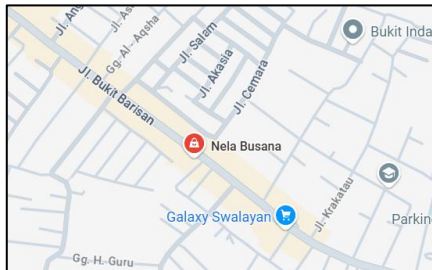
Penulis	Judul Penelitian	Permasalahan	Metode Penelitian	Environment	Hasil
	Inventaris Aset Desa Rempoah				mulanya menggunakan sistem konvensional
(Gushe lmi, 2022)	Penerapan Metode Rapid Application Development Dalam Membangun Website E-Commerce	Pemasaran dan promosi produk masih menggunakan banner dan media sosial Sehingga menyebabkan pemborosan biaya promosi dan keakuratan data laporan tidak sesuai	Metode <i>Rapid Application Development (RAD)</i>	Toko Christine Hakim Padang	Sistem dapat memudahkan pihak Toko Christine Hakim Padang dalam melakukan proses penjualan dan perluasan pasar secara efektif dan efisien.
(Butar butar dkk., 2022)	Perancangan Sistem Informasi	Banyaknya kompetitor-	Metode <i>waterfall</i>	Cv. Ramos Ponsel a	Memberikan kemudahan

Penulis	Judul Penelitian	Permasalahan	Metode Penelitian	Environment	Hasil
	E-Catalogue Berbasis Website Menggunakan Metode Waterfall	kompetitor sehingga memberikan dampak berkurangnya konsumen dalam memenuhi kebutuhan mereka di Cv.			pihak Cv. Ramos Ponsel dalam mengelola kategori data barang
(Profita dkk., 2022)	Penerapan Metode Rapid Application Development (RAD) untuk Digitalisasi UKM Industri Busana Muslim Abstrak Digitalisasi Usaha Kecil	Proses pembuatan laporan dan penyimpanan data belum terorganisir dengan baik, sehingga mengakibatkan terjadinya penumpukan atau hilangnya data transaksi penjualan	Metode <i>Rapid Application Development (RAD)</i>	Toko Risti Boutique	Sistem ini dapat melakukan penjualan secara online dan sebagai database dari toko tersebut

2.2 Landasan Teori

2.2.1 Butik Nela Busana

Butik Nela Busana yang berlokasi di Jl. Rukun No.29, Tengkerang Utara, Kec. Bukit Raya, Kota Pekanbaru, Riau 28289. Yang merupakan salah satu Usaha mikro, kecil, dan menengah (UMKM) yang bergerak dibidang *fashion* yang selalu berupaya meningkatkan produktivitas, kepuasan pelanggan, dan pada akhirnya meningkatkan pendapatan usaha.



Gambar 2. 1 Lokasi Butik Nela Busana

Butik Nela Busana didirikan pada tahun 2016 dan menawarkan berbagai macam pakaian untuk berbagai acara. Untuk wanita, tersedia kategori seperti baju kurung melayu, kebaya, termasuk gamis. Sementara untuk pria, butik ini menyediakan baju melayu, dan baju batik. Selain itu, butik ini juga menawarkan pakaian dengan variasi bordir atau payet, serta pakaian polos.

Pelanggan dapat memilih bahan yang disediakan oleh butik, atau mereka juga dapat membawa bahan sendiri. Untuk desain, pelanggan dapat melakukan permintaan khusus terkait model pakaian, meskipun tidak dapat dijamin akan dibuat persis sama, namun akan diusahakan semirip mungkin dengan yang diinginkan.

2.2.2 *Rapid Application Development (RAD)*

Menurut Ramadhan dkk. (2024) , *Rapid Application Development (RAD)* adalah teknik untuk mempercepat pengembangan sistem informasi dengan fokus pada

pengembangan sistem yang berfungsi. Berbeda dengan JAD yang menghasilkan model persyaratan, produk akhir RAD adalah sistem informasi baru. RAD menggabungkan teknik terstruktur dengan prototyping dan joint application development untuk mempercepat pengembangan sistem.

Adapun Teknik Teknik dari *rapid application development* meliputi:

1. *Prototyping*

System Prototyping merupakan metode pengembangan sistem yang menjalankan tahap analisis, desain, dan implementasi secara bersamaan untuk menciptakan versi awal dari sistem yang diusulkan. Versi sederhana ini kemudian diberikan kepada pengguna agar dapat diuji, dievaluasi, dan diberi masukan. Prototipe pertama ini hanya mencakup fitur minimal, sementara umpan balik dari pengguna digunakan untuk menyempurnakan sistem lebih lanjut (Agustina & Ratnasari, 2020).

2. *Iterative Development*

Iterative Development merupakan teknik pengembangan yang membagi proyek menjadi beberapa versi yang dikembangkan secara bertahap. Pada versi pertama, kebutuhan yang paling penting dan mendasar diintegrasikan ke dalam sistem. Versi awal ini dikembangkan dengan cepat menggunakan metode waterfall sederhana yang hanya melalui satu tahap implementasi. Pengguna dapat memberikan masukan yang kemudian diakomodasi dalam versi sistem berikutnya. Pendekatan ini memungkinkan pemenuhan kebutuhan bisnis secara lebih efektif. Namun, kelemahannya terletak pada penggunaan sistem versi awal yang masih terbatas serta proses adaptasi yang berulang setiap kali versi baru diperkenalkan (Agustina & Ratnasari, 2020).

3. *Reusability of Components*

Menggunakan kembali komponen perangkat lunak yang sudah ada untuk mempercepat pengembangan. sehingga tidak terjadi adanya perulangan

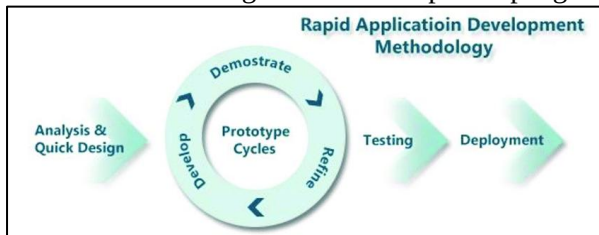
dalam penggunaan sebuah komponen yang mana dampaknya adalah pemborosan pada resources sistem tersebut(Purnamansyah & Zalsabil Pratama, 2024).

4. *User Involvement*

User involvement merupakan teknik dimana pengguna terlibat langsung dalam proses pengembangan untuk memberikan umpan balik secara berkala.

5. *Workshop dan Brainstorming*

Workshop dan brainstorming merupakan sesi diskusi antara tim pengembang dan pengguna yang bertujuan untuk merancang sistem secara cepat. Melalui sesi ini, kedua pihak dapat menyelaraskan ekspektasi dan visi terhadap aplikasi yang dikembangkan, sehingga memastikan bahwa kebutuhan pengguna dapat diakomodasi dengan baik dalam proses pengembangan.



Gambar 2. 2 *Metode Rapid Application Development*
(Sumber: Musfikir dkk., 2023)

Sebagaimana diilustrasikan dalam gambar 2.2 tahapan Metode *Rapid Application Development* dapat dijelaskan sebagai berikut:

1. *Analysis and Quick Design*

Fase ini langkah awal dalam pengembangan sistem pada metode *RAD*, di mana dilakukan identifikasi terhadap permasalahan yang ingin diselesaikan. Pada tahap ini juga dilakukan pengumpulan data dari pengguna untuk menentukan tujuan sistem dan kebutuhan informasi yang diperlukan. Keterlibatan seluruh pengguna sangat penting dalam proses identifikasi kebutuhan yang harus dipenuhi dalam pengembangan sistem aplikasi. Selain itu, tahap berikutnya adalah perancangan desain antarmuka yang

ramah pengguna, termasuk penentuan tampilan yang akan ditampilkan selanjutnya (Musfikir dkk., 2023).

2. *Prototype Cycles*

Tahap ini terus berputar selama masih ada kekurangan menurut pengguna. Fase ini terdiri dari 3 proses yaitu:

1) *Build*

Fase di mana pengembangan atau pembuatan aplikasi dilakukan secara intensif. Pada tahap ini, perangkat lunak mulai dikembangkan dalam bentuk website yang dirancang agar dapat berfungsi sesuai dengan kebutuhan pengguna. Proses ini mencakup pengkodean, pengujian awal, serta implementasi berbagai fitur yang telah dirancang pada tahap sebelumnya. Selain itu, dalam tahap ini juga dilakukan optimasi untuk memastikan bahwa sistem berjalan dengan baik, responsif, dan sesuai dengan spesifikasi yang telah ditentukan (Musfikir dkk., 2023).

2) *Demonstrate*

Pada tahap ini dilakukannya *testing*, dimana *developer* mendemonstrasikan sistem dihadapan klien (Musfikir dkk., 2023).

3) *Refine*

Tahap ini *developer* menyaring masukan masukan yang diberi klien yang kemudian di pertimbangkan untuk melakukan *rebuild* sistem (Musfikir dkk., 2023).

3. *Testing*

Pada tahap ini, proses pengujian mulai dilakukan sebelum sistemnya diimplementasikan. Pengujian dilakukan secara menyeluruh untuk memastikan bahwa sistem dapat berfungsi dengan baik sebelum digunakan secara produktif. Semua komponen dan antarmuka diuji secara detail (Musfikir dkk., 2023).

4. *Deployment*

Setelah menyelesaikan tiga tahap awal dalam

pengembangan perangkat lunak sistem, langkah berikutnya adalah tahap implementasi. Tahap ini merupakan fase akhir dari proses penelitian, di mana hasil penelitian diimplementasikan untuk mengevaluasi efektivitasnya. Setelah sistem berhasil diimplementasikan, maka proses perancangan juga dianggap selesai (Musfikar dkk., 2023).

2.2.3 Collaborative Filtering

Metode Collaborative Filtering merupakan suatu teknik dalam mengevaluasi atau menyaring item berdasarkan persepsi dan opini dari pengguna lain. Proses ini melakukan penyaringan terhadap seluruh pengguna untuk memperoleh informasi yang digunakan untuk memberikan rekomendasi. *Collaborative Filtering* bekerja dengan menganalisis kemiripan karakteristik antar pengguna, sehingga dapat menyajikan informasi baru yang relevan. Sistem ini memberikan rekomendasi berdasarkan pola yang ditemukan dalam kelompok pengguna dengan karakteristik yang serupa. Dalam beberapa kasus, anggota kelompok dengan minat yang berbeda dapat menyumbangkan informasi baru yang berpotensi bermanfaat bagi anggota lainnya. Oleh karena itu, proses pemberian rekomendasi dalam metode ini dapat dibagi menjadi tiga langkah utama, yaitu: mengidentifikasi pengguna dengan kesamaan preferensi, membentuk kedekatan atau hubungan antar pengguna (*neighborhood*), serta menghitung prediksi berdasarkan hubungan yang telah ditentukan (Februariyanti dkk., 2021).

Collaborative Filtering terbagi menjadi dua kelas yaitu:

1. Item-to-item Collaborative Filtering

Item-based Collaborative Filtering memiliki pola yang serupa dengan *User-based Collaborative Filtering*. Namun, jika dalam pendekatan *User-based* sistem mencari hubungan atau korelasi antar pengguna, maka dalam *Item-based Collaborative Filtering*, sistem berfokus pada menemukan korelasi antar item yang

diminati oleh pengguna. Setelah korelasi antar item ditemukan, sistem kemudian merekomendasikan item yang memiliki keterkaitan kepada pengguna lainnya (Februariyanti dkk., 2021).

2.2.4 *Cosine Similarity Based Similarity*

Metode Cosine Similarity, atau dikenal juga sebagai *Vector-Based Similarity*, merupakan pendekatan yang banyak digunakan dalam sistem rekomendasi berbasis *Collaborative Filtering*, khususnya pendekatan *item-based*. Algoritma ini mengukur kesamaan antar dua vektor (misalnya dua item) berdasarkan sudut kosinus di antara keduanya, tanpa memperhatikan besar vektor (magnitudo), tetapi fokus pada orientasi atau arah vektor. Dalam konteks sistem rekomendasi, vektor biasanya merepresentasikan interaksi pengguna terhadap item, seperti rating atau riwayat pembelian. Jika dua item memiliki pola interaksi pengguna yang mirip, maka sudut antara vektornya kecil dan nilai cosine similarity mendekati 1 semakin mirip (Rosydah & Widiyaningtyas, 2024).

Rumus perhitungan *Cosine Similarity* antar item i dan j adalah:

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}},$$

Gambar 2. 3 Rumus *Cosine Similarity*
(Rosydah & Widiyaningtyas, 2024)

Di mana $\text{sim}(A, B)$ merupakan nilai *cosine similarity* antara dua vektor A dan B . Vektor A dan B dapat merepresentasikan dua item atau dua pengguna. Masing-masing elemen A_i dan B_i merupakan nilai rating (misalnya biner 0 atau 1) dari entitas ke- i terhadap item A dan item B (atau sebaliknya). Nilai *similarity* dihitung dengan membagi hasil *dot product* antara A dan B dengan hasil kali magnitudo masing-masing vektor, yaitu akar kuadrat dari jumlah kuadrat setiap

elemen pada vektor tersebut. Notasi i menunjukkan dimensi ke- i , yang bisa berupa indeks pengguna atau item tergantung konteks.

2.2.5 Prediksi *Collaborative Filtering* dengan *Cosine Similarity*

Setelah sistem menghitung nilai kemiripan antar item menggunakan *Cosine Similarity*, selanjutnya sistem memprediksi rating atau ketertarikan pengguna terhadap item yang belum pernah mereka interaksikan. Prediksi ini menggunakan metode *Weighted Sum*, yang memberikan bobot lebih tinggi kepada item-item yang memiliki kemiripan tinggi dengan item target.

Pada pendekatan *Cosine Similarity* prediksi dihitung dengan rumus berikut:

$$P(u, j) = \frac{\sum_{i \in I} (R_{u,i} * S_{ij})}{\sum_{i \in I} |S_{ij}|}$$

Gambar 2. 4 Rumus *Weighted Sum*
(Hartatik dkk, 2021)

Persamaan di atas digunakan dalam pendekatan item-based collaborative filtering untuk memperkirakan rating $P_{(u,j)}$ yaitu prediksi rating yang akan diberikan oleh pengguna u terhadap item j . Estimasi ini dihitung dengan menjumlahkan hasil perkalian antara rating aktual $R_{u,i}$ yang diberikan pengguna terhadap item i dan tingkat kesamaan S_{ij} antara item i dan item j , kemudian dibagi dengan jumlah total nilai absolut dari semua kesamaan $|S_{ij}|$. Himpunan I merepresentasikan kumpulan item yang telah diberi rating oleh pengguna u . Rumus ini mengasumsikan bahwa item yang mirip dengan item j dan telah diberi rating tinggi oleh pengguna u akan memberikan kontribusi lebih besar dalam prediksi rating terhadap item j (Hartatik dkk, 2021).

2.2.6 *Mean Absolute Error*

Tingkat akurasi dalam sistem rekomendasi diukur

berdasarkan nilai error yang dihasilkan. Dalam penelitian ini, digunakan persamaan *Mean Absolute Error* (MAE) untuk menentukan akurasi sistem. Persamaan ini termasuk dalam kategori *statistical accuracy metrics*, di mana MAE menghitung rata-rata selisih antara nilai prediksi dengan nilai aktual.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\bar{r}_i - r_i|$$

Gambar 2. 5 Rumus *Mean Absolute Error*

Di mana $P_{u,k}$ merupakan prediksi rating yang diberikan oleh pengguna u untuk item k , $R_{u,k}$ adalah rating aktual yang diberikan oleh pengguna u terhadap item k , dan N menunjukkan jumlah total pengguna.

2.2.7 Eisenhower matrix

Matrik *Eisenhower* adalah model manajemen waktu yang dikembangkan berdasarkan prinsip yang diperkenalkan oleh Presiden Amerika Serikat ke-34, Dwight D. Eisenhower. Model ini berfungsi sebagai alat bantu bagi pengguna dalam mengatur waktu dan menetapkan prioritas tugas dengan mempertimbangkan tingkat urgensi serta kepentingannya (Aisyah dkk., 2025).



Gambar 2. 6 *Matrix Eisenhower matrix*

Matrik *Eisenhower* membantu pengguna dalam mengklasifikasikan tugas-tugas mereka ke dalam empat kategori utama:

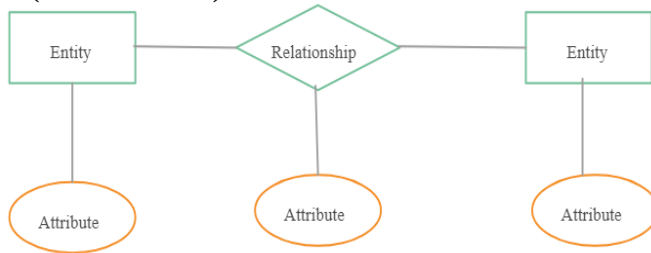
- Kuadran 1 – Mendesak dan Penting (*Do*): Tugas yang memerlukan penyelesaian segera tanpa penundaan.
- Kuadran 2 – Penting, tetapi Tidak Mendesak (*Plan*): Tugas yang harus direncanakan dengan baik dan dapat dijadwalkan untuk dikerjakan di lain waktu.
- Kuadran 3 – Tidak Penting, tetapi Mendesak (*Delegate*): Pekerjaan yang dapat dialihkan atau didelegasikan kepada orang lain.
- Kuadran 4 – Tidak Penting dan Tidak Mendesak (*Delete*): Aktivitas yang sebaiknya dikurangi atau dihilangkan karena tidak memberikan manfaat yang berarti.

Dengan menerapkan sistem ini, pengguna diharapkan dapat lebih efektif dalam menetapkan prioritas serta menghindari kebiasaan menunda pekerjaan (Aisyah dkk., 2025).

2.2.8 *Entity-Relationship Diagram*

Entity Relationship Diagram atau yang biasa dikenal dengan istilah ERD merupakan suatu diagram yang berbentuk dalam gambar beserta dengan simbol- simbol yang berfungsi untuk mengidentifikasi tiap entitas dan atribut yang berada didalamnya. Selain itu, ERD juga menjelaskan hubungan atau relasi dari entitas entitas tersebut. ERD berfungsi untuk memberikan gambaran terhadap konsep-konsep dan kerangka dasar dalam pembuatan software. Didalam ERD, laluakan mengenal beberapa istilah seperti *entitas*, *attributes* dan *entity relationship modeling*. *Entity* merupakan sebuah benda atau objek yang dapat dibedakan dari semua objek lainnya. Entitas terdiri dari berbagai kualitas yang mengidentifikasi atau membedakannya dari entitas lain. Setiap objek harus memiliki suatu properti yang unik yaitu *Primary Key*. Tujuan dari entities adalah untuk menampilkan sebuah *data store* yang memvisualisasikan proses data yang saling terhubung. Selain

itu entity juga dapat mengatur dan mengklasifikasikan data(Jiantono, 2023).



Gambar 2. 7 *Entity-Relationship Diagram*

Merujuk pada gambar 2.2 dapat dilihat bahwa Di dalam ERD terdapat 3 elemen dasar, yaitu entitas, atribut, dan relasi:

1. Entitas (*Entity*)
Entitas dapat berupa manusia, tempat, benda, atau kondisi mengenai data yang dibutuhkan. Simbol dari entitas berbentuk persegi panjang (Afiifah dkk., 2022).
2. Atribut (*Attribute*)
Atribut merupakan informasi yang terdapat dalam entitas. Sebuah entitas harus memiliki *primary key* sebagai ciri khas entitas dan atribut deskriptif. (Afiifah dkk., 2022).
3. Relasi (*Relationship*)
Relasi di dalam ERD merupakan hubungan antara dua atau lebih entitas. Simbol dari relasi berbentuk belah ketupat (Afiifah dkk., 2022). Relasi yang dapat dimiliki oleh ERD ada beberapa macam, yaitu:
 - 1) *One to One* Satu anggota entitas dapat berelasi dengan satu anggota entitas lain.
 - 2) *One to Many* Satu anggota entitas dapat berelasi dengan beberapa anggota entitas lain.
 - 3) *Many to Many* Beberapa anggota entitas dapat berelasi dengan beberapa anggota entitas lain.

2.2.9 Framework Codeigniter

Salah satu *framework* yang digunakan dalam pembuatan proyek akhir ini adalah *Codeigniter*. Tujuan menggunakan *framework codeigniter* karena untuk melakukan pengembangan program tidak perlu membuat kode dari awal sehingga dalam proses kerjanya pun terasa lebih cepat. Menurut Sallaby & Kanedi (2020) mengatakan bahwa *codeIgniter* adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman *PHP* yang bertujuan untuk memudahkan para *programmer* web untuk membuat atau mengembangkan aplikasi berbasis web. *CodeIgniter* memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya.

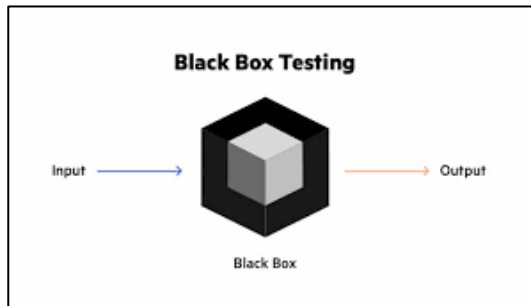


Gambar 2. 8 Codeigniter

2.2.10 Black Box Testing

Black Box Testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi. Berdasarkan pada spesifikasi kebutuhan perangkat lunak. Terdapat beberapa metode pengujian dalam pengujian black box seperti *equivalence partitioning*, *boundary value analysis*, *cause effect graph*, *comparison testing*, *random data selection*, *feature test*, *all-pair testing*, *fuzzing*, *orthogonal*

array testing, sample testing, robustness testing, behavior testing, performance testing, endurance testing, dan lain-lain (Dika Pratama & Noviarsyah Dadaprawira, 2023).



Gambar 2. 9 *Blackbox Testing*

Functional Testing berfokus pada spesifikasi pada fungsionalitas sistem berdasarkan input dan output, tanpa memperhatikan bagaimana sistem bekerja secara internal. Pengujian ini dilakukan dengan memasukkan input yang telah ditentukan dan memverifikasi apakah hasil yang dikeluarkan oleh sistem sesuai dengan yang diharapkan (Emanuella dkk., 2022).

2.2.11 *Usability Testing*

Usability Testing adalah metode yang digunakan untuk mengevaluasi pengalaman dan kebutuhan pengguna dalam berinteraksi dengan suatu aplikasi. Metode ini sangat penting karena pengguna memainkan peran yang krusial dalam proses pengembangan aplikasi. Pengalaman dan umpan balik yang diperoleh dari pengguna sebelumnya dapat memberikan arah yang jelas bagi pengembang untuk meningkatkan fungsionalitas dan kenyamanan aplikasi. Dengan mengidentifikasi masalah yang dihadapi pengguna selama penggunaan aplikasi, pengembang dapat membuat perbaikan yang lebih sesuai dengan kebutuhan pasar. Menurut Ridwan dkk., (2024) *Usability Testing* membantu dalam mengevaluasi sejauh mana aplikasi dapat digunakan dengan efisien, efektif, dan menyenangkan oleh pengguna.

Pengujian menggunakan *usability* dengan metode skala *likert*. Pengujian *usability* diukur dengan beberapa komponen umum *usability* yang dibagi ke dalam 5 aspek umum yaitu aspek *design*, aspek kemudahan, dan aspek kepuasan pengguna. Dari 5 aspek umum *usability* maka dilakukan penyusunan kuesioner kepada para pengguna sistem. Tujuan pengujian ini untuk mengukur kemudahan pengguna, kepuasan pengguna, dan tampilan desain sistem dalam menggunakan sistem (Ridwan dkk., 2024).

Tabel 2. 2 Bobot Presentasi Nilai *Usability Testing*

Skala <i>Likert</i>	Bobot Nilai
Sangat Setuju (SS)	5
Setuju (S)	4
Netral (N)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

2.2.12 User Acceptance Testing (UAT)

Dalam memastikan system berjalan dengan lancar *User Acceptance Testing (UAT)* sangat dibutuhkan, Menurut Chamida dkk. (2021) *User Acceptance Testing (UAT)* adalah pengujian interaksi antara *end-user* dan sistem secara langsung yang berfungsi untuk memverifikasi bahwa fitur telah berjalan sesuai dengan kebutuhan user tersebut. Pengujian UAT termasuk fase terakhir dalam proses pengujian pada sistem, yang dimana sistem telah selesai melalui tahap pengembangan. UAT menjadi salah satu rangkaian pengujian final dari perangkat lunak dan dilakukan sebelum dikembangkan dan diluncurkan.

Tabel 2. 3 Skala Pembobotan *User Acceptance Testing*

Skala	Keterangan	Skor	Presentase
SS	Sangat Sesuai	5	80 – 100%
S	Sesuai	4	60 - 79%
CS	Cukup Sesuai	3	40 – 59%
TS	Tidak Sesuai	2	20 – 39%
STS	Sangat Tidak Sesuai	1	0 – 19 %