# BAB II TINJAUAN PUSTAKA

## Penelitian Terdahulu

### Penelitian Pertama

Penelitian yang dilakukan oleh Robby Nugroho Setiawan (2021) dengan judul “*Pembuatan Sistem Informasi Topup Gaming Berbasis Website”* meneliti tentang bagaimana membuat sistem penjualan voucher game atau yang biasa disebut dengan *Topup Game*. Penelitian tersebut bertujuan untuk mengatasi masalah banyaknya penipuan dalam bidang topup game karena pihak pembeli harus memasukan username serta password dari akun game tersebut. Aplikasi ini dibangun menggunakan bahasa pemrograman Python dengan framework Django. Penelitian ini menggunakan SDLC *(Software Development Life Cycle)* metode *Waterfall*.

### Penelitian Kedua

Penelitian yang dilakukan oleh Fajar Setiawan, Hadi Zakaria (2022) dengan judul *“Rancang Bangun Sistem Pembelian Voucher Game Online Berbasis Website Menggunakan Teknologi MERN Stack Dengan Model Waterfall (Studi Kasus: Good Gaming Store)”* meneliti tentang bagaimana proses jual beli voucher game dan membuat sebuah sistem untuk mempermudah proses tersebut. Peneliti menemukan berbagai masalah dalam proses jual beli voucher game seperti pemesanan harus dilakukan manual melalui *whatsapp* kemudian pembeli juga harus menunggu konfirmasi pembayaran yang cukup lama. Sehingga peneliti membuat sistem pembelian voucher game ini untuk mengatasi masalah-masalah tersebut. Sistem ini dibangun dengan beberapa gabungan teknologi yang biasa disebut *MERN Stack (MongoDB, Express, React, Node)*.

### Penelitian Ketiga

Penelitian yang dilakukan oleh Rizky Dwi Nugroho (2019) dengan judul *“Rancang Bangun Sistem Informasi Layanan Antrian Sertifikasi Berbasis Web Menggunakan Metode Priority Service Pada Balai Riset Dan Standarisasi Industri Surabaya”* meneliti tentang bagaimana membuat sistem informasi layanan antrian sertifikasi dengan metode *priority service* yang mampu memanajemen antrian produk yang akan diuji dengan baik serta dapat mengelompokkan produk yang akan diuji berdasarkan jenis produknya. Penelitian tersebut bertujuan untuk mengatasi masalah antrian sertifikasi produk yang belum dimanajemen dengan baik serta belum adanya pengelompokan jenis produk jenis produk yang akan disertifikasi sehingga jenis produk yang memiliki masa waktu rentan terkena resiko kadaluarsa, rusak, dll. Aplikasi ini dibangun menggunakan bahasa pemrograman PHP. Penelitian ini menggunakan SDLC *(Software Development Life Cycle)* metode *Waterfall*.

### *Research Gap*

Berdasarkan penelitian yang pernah dilakukan sebelumnya, terdapat beberapa perbedaan (research gap) dengan penelitian yang akan dilakukan. Penilitian sebelumnya berfokus terhadap pembuatan *website* penjualan *voucher game* yang hanya menggunakan satu metode pembayaran yaitu paypal, sedangkan penelitian yang akan dilakukan akan menggunakan bermacam-macam metode pembayaran yang terhubung langsung dengan *payment gateway*. Kemudian pada penelitian sebelumnya berfokus pada pembuatan website untuk menyimpan data transaksi supaya tidak melalui pihak ketiga seperti *whatsapp* namun proses transaksi dan verifikasi pembayaran tetap dilakukan manual, sedangkan di penelitian yang akan dilakukan segala jenis transaksi dan verifikasi pembayaran akan diproses langsung oleh sistem. Kemudian pada penelitian sebelumnya penerapan antrian dilakukan untuk memanajemen produk fisik untuk menghindari kesalahan sortir sehingga jenis produk yang memiliki masa waktu rentan terkena resiko kadaluarsa, sedangkan pada penelitian ini antrian digunakan pada transaksi untuk menghindari kesalahan pemotongan saldo ketika banyak transaksi dilakukan secara bersamaan dalam satu waktu.



## Teori Terkait

### Sistem Informasi

Menurut (Anggraeni & Irviani, 2017), “sistem informasi merupakan suatu kombinasi teratur dari orang-orang, *hardware, software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi”.

Menurut (Suryantara, 2017), “sistem informasi adalah suatu sistem yang dibuat manusia dan terdiri atas komponen-komponen dalam organisasi untuk mencapai tujuan, yaitu menyajikan informasi”.

Menurut (Mahatmyo, 2014), “sistem informasi (*information system*) adalah serangkaian prosedur formal di mana data dikumpulkan, diproses menjadi informasi dan didistribusikan ke pengguna”.

### Antrian atau *Queue*

Menurut (Antono, 2010: 259-260), “Antrian merupakan sebuah sistem yang mencakup pelanggan yang datang dengan laju konstan atau bervariasi untuk mendapatkan pelayanan pada suatu fasilitas layanan. Jika pelanggan yang datang dapat memasuki fasilitas layanan, pelanggan dapat langsung dilayani. Jika pelanggan harus menunggu dilayani, pelanggan akan berpartisipasi membentuk antrian, dan akan berada dalam antrian hingga pelanggan mendapat giliran untuk dilayani. Pelanggan akan dilayani dengan laju layanan yang konstan atau bervariasi dan akhirnya meninggalkan sistem. Sistem antrian mencakup baik antrian dan fasilitas layanannya”.

### Disiplin Antrian

Menurut (Kakiay, 2009), Disiplin antrian adalah aturan di mana para pelanggan dilayani, atau disiplin pelayanan (service discipline) yang memuat urutan (order) para pelanggan menerima layanan. Ada 4 bentuk-bentuk disiplin antrian menurut urutan kedatangan antara lain adalah:

1. **First Come First Served (FCFS) atau First In First Out (FIFO)**, di mana pelanggan yang terlebih dahulu datang akan dilayani terlebih dahulu. Misalnya, antrian pada loket pembelian tiket bioskop, antrian pada loket pembelian tiket kereta api. Model ini hanya membicarakan kasus dalam keadaan Steady state. Beberapa karakteristik operasi: Intensitas lalu lintas, Periode sibuk, Distribusi peluang dari langganan dalam sistem, Jumlah rata-rata dalam sistem, Jumlah rata-rata dalam antrian, Jumlah rata-rata yang menerima layanan.
2. **Last Come First Served (LCFS) atau Last In First Out (LIFO)**, dimana pelanggan yang datang paling akhir akan dilayani terlebih dahulu. Misalnya, sistem antrian pada elevator untuk lanti yang sama, sistem bongkar muat barang dalam truk, pasien dalam kondisi kritis, walaupun dia datang paling akhir tetapi dia akan dilayani terlebih dahulu.
3. **Service In Random Order (SIRO) atau Random Selection for Service (RSS)**, dimana panggilan berdasarkan pada peluang secara random, jadi tidak menjadi permasalahan siapa yang lebih dahulu datang. Misalnya, pada arisan dimana penarikan berdasarkan nomor undian.
4. **Priority Service (PS)**, dimana prioritas pelayanan diberikan kepada pelanggan yang mempunyai prioritas lebih tinggi dibandingkan dengan pelanggan yang mempunyai prioritas yang lebih rendah, meskipun mungkin yang dahulu tiba di garis tunggu adalah yang terakhir datang. Hal ini mungkin disebabkan oleh beberapa hal, misalnya seseorang yang memiliki penyakit yang lebih berat dibandingkan orang lain pada suatu tempat praktek dokter, hubungan kekerabatan pelayan dan pelanggan potensial akan dilayani terlebih dahulu.

### *Flowchart*

Menurut (Barakbah, Karlita, & Ahsan, 2013), *Flowchart* adalah cara penulisan algoritma dengan menggunakan notasi grafis. *Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya.

Menurut (Kusuma, 2020), *flowchart* atau bagan alir adalah suatu model proses yang dinyatakan dalam simbol-simbol dan hubungan antar proses. Suatu diagram alir selalu dimulai dan diakhiri oleh terminasi. Antar proses dihubungkan dengan anak panah yang menggambarkan kelanjutan suatu proses.

**Tabel 2.1** Simbol *Flowchart*

| **SIMBOL** | **NAMA** | **FUNGSI** |
| --- | --- | --- |
|  | *Terminator* | Simbol awal (*Start*) / simbol akhir (*End*) |
|  | *Flow Line* | Simbol aliran / penghubung |
|  | Proses | Perhitungan / pengolahan |
|  | *Input*/*Output* Data | Mempresentasikan pembacaan data (*read*) / penulisan (*write*) |
|  | *Decision* | Simbol pernyataan pilihan, berisi suatu kondisi yang selalu menghasilkan 2 nilai keluaran yaitu benar atau salah |
|  | *Preparation* | Inisialisasi / pemberian nilai awal |
|  | *Predefined Process* | Proses menjalankan sub program fungsi / prosedur |
|  | *On Page Connector* | Penghubung *Flowchart* pada satu halaman |
|  | *Off Page Connector* | Penghubung *Flowchart* pada halaman berbeda |

### *Web Server*

Menurut (Kadir 2009:15) “MySQL merupakan *software* yang tergolong *database server* dan bersifat *open source*”. Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat *fleksibel* dan cepat.

Menurut (Supardi 2010:181) menjelaskan bahwa *Web Server* merupakan perangkat lunak yang mengatur atau mengelola program berdasarkan permintaan *browser* dan dikirimkan kembali ke *browser. Browser* sendiri merupakan perangkat lunak untuk menjalankan program atau *script web*. Setiap program *web server* bekerja dengan menerima permintaan HTTP dari klien, dan memberikan respon HTTP ke klien tersebut. Respon HTTP biasanya mengandung dokumen HTML tetapi dapat juga berupa berkas raw, gambar, dan berbagai jenis dokumen lainnya. Jika terjadi kesalahan permintaan dari klien atau terjadi masalah saat melayani klien maka web server akan mengirimkan respon kesalahan yang dapat berupa dokumen HTML atau teks yang memberi penjelasan penyebab terjadinya kesalahan.

### PHP *(Hypertext Preprosessor)*

Menurut (Arief, 2012) PHP (*Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML.

Menurut (Setiawan & Adams, 2017) PHP merupakan singkatan dari “*Hypertext Preprocessor*” yang merupakan sebuah bahasa pemrograman tingkat tinggi yang dipasang pada HTML. Sebagian besar sintaks dalam PHP mirip dengan bahasa C, Jawa, dan Perl, namun pada PHP ada beberapa fungsi yang lebih spesifik.

Menurut (Prasetyo, 2008) PHP (*Hypertext Preprosessor*) adalah *script* bersifat *server-side* yang ditambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari *Personal Home Page Tools*. *Script* ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman *web* tidak lagi bersifat statis, namun menjadi dinamis. Sifat *server-side* berarti pengerjaan *script* dilakukan di *server*, baru kemudian hasilnya dikirimkan ke *browser*.

### MySQL

Menurut Kadir (2009:15) “MySQL merupakan *software* yang tergolong *database* server dan bersifat *open source*”. Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat.

Menurut (Solichin, 2016), MySQL adalah sistem manajemen *Database* SQL yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database* MySQL mendukung beberapa fitur seperti *multi-threaded*, *multi-use*r dan SQL *Database managemen system* (DBMS). Salah satu keunggulan MySQL adalah kemudahan penggunaan dan pengelolaannya. Untuk mengakses database MySQL juga dapat dilakukan dengan berbagai MySQL.

### *Unified Modeling Language (UML)*

Menurut (Hendini, 2016), *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan, dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Tujuan dari UML sendiri adalah untuk menyediakan standar notasi yang dapat digunakan oleh semua metode orientasi objek dan untuk memilih dan mengintegrasikan elemen-elemen terbaik dari notasi-notasi pendahulu. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

#### **Use Case Diagram**

Menurut (Hendini, 2016), *Use case diagram* adalah diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

**Tabel 2.2** Simbol *Use Case Diagram*

| **SIMBOL** | **NAMA** | **FUNGSI** |
| --- | --- | --- |
|  | *Use Case* | *Use Case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja |
|  | *Actor* | Aktor adalah *Abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi atau menggunakan sistem. |
|  | *Association* | Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan  data. |
|  | *Association* | Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem |
|  | *Include Relationship* | *Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program. |
|  | *Extend Relationship* | *Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi. |

#### **Activity Diagram**

Menurut (Hendini, 2016), *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi, hingga berakhirnya aksi. *Activity diagram* juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan. *Activity diagram* adalah aktivitas-aktivitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Simbol-simbol yang digunakan dalam *Activity Diagram* yaitu:

Tabel 2.3 Simbol Activity Diagram

| **SIMBOL** | **NAMA** | **FUNGSI** |
| --- | --- | --- |
|  | *Start point* | *Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas. |
|  | *End Point* | *End Point*, akhir aktivitas. |
|  | *Activities* | *Activities*, menggambarkan suatu proses/kegiatan bisnis. |
|  | *Fork* | *Fork*/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |
|  | *Join* | *Join*/penggabungan digunakan untuk menunjukkan adanya dekomposisi. |
|  | *Decision Point* | *Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true* atau *false*. |
|  | *Swimlane* | *Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa. |

### *Entity Relationship Diagram (ERD)*

Menurut (Leman, 1998: 28) menyatakan bahwa ERD adalah diagram yang berfungsi untuk menggambarkan sistem yang terdiri dari hubungan entitas. Untuk menggambarkan sebuah ERD digunakan beberapa simbol. Pada dasarnya ada 3 (tiga) simbol yang digunakan, yaitu:

1. *Entity*

*Entity* merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entity ini biasanya digambarkan dengan persegi panjang.

1. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

1. Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut:

1. *One to One*

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

1. *One to Many*

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

1. *Many to Many*

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B. Begitu juga pada entitas B, dapat berhubungan dengan banyak entitas A.

ERD ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah entity dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang database. Untuk itu ERD dibagi menjadi dua jenis model, yaitu:

1. *Conceptual Data Model (CDM)*

*Conceptual Data Model (CDM)* adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

1. *Physical Data Model (PDM)*

*Physical Data Model (PDM)* adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

### PHPStrom

PHPStorm adalah sebuah *Integrated Development Environment (IDE)* yang dikembangkan oleh perusahaan JetBrains khusus untuk pengembangan aplikasi berbasis PHP. IDE adalah lingkungan yang menyediakan berbagai fitur dan alat untuk membantu para pengembang dalam mengkode, menguji, dan mengelola proyek perangkat lunak mereka dengan lebih efisien.

PHPStorm dirancang khusus untuk pengembangan menggunakan bahasa pemrograman PHP, tetapi juga mendukung bahasa pemrograman lainnya seperti HTML, CSS, JavaScript, dan banyak lagi. Dengan berbagai fitur seperti penyorotan sintaksis, pengutipan kode, analisis kode, *debugging*, pengaturan proyek, integrasi dengan sistem kontrol versi, dan banyak lagi, PHPStorm membantu pengembang dalam mengembangkan aplikasi web dengan cepat dan lebih produktif.

IDE ini memiliki antarmuka pengguna yang user-friendly dan beragam fitur yang dirancang untuk mempercepat alur kerja pengembangan. Selain itu, PHPStorm terus diperbarui oleh JetBrains untuk menjaga kompatibilitas dengan bahasa pemrograman yang terbaru dan standar industri dalam pengembangan web.

### *Payment Gateway*

Menurut (Sammy, 2022), Payment Gateway merupakan perantara proses pembayaran antara website penjualan dengan pihak bank. Payment Gateway adalah kunci penting dari transaksi online. Payment Gateway adalah saluran terenkripsi yang aman untuk mengirimkan detail transaksi dari pembeli ke bank untuk disetujui. Penggunaan Payment Gateway memberikan keuntungan seperti transaksi pembayaran lebih cepat dan efisien, banyak pilihan metode pembayaran, keamanan yang baik, dan kontrol administrasi untuk mempermudah pemilik bisnis.