**BAB I**

**PENDAHULUAN**

Pada bab ini akan dijelaskan mengenai latar belakang dari pembuatan *website* ini. Lalu akan dijelaskan front end, dan back end yang dipakai, metodologi yang akan diterapkan pada pembuatan website ini. Dijelaskan juga tujuan dari pembuatan website *multi*-transaksi untuk *gaming online* ini. Berikut adalah isi dari bab I.

1. **Latar Belakang**

Pada tahun 2020, *game* sudah banyak disukai dan bahkan sekarang menjadi salah satu olahraga, yaitu *E-Sport*. Sekarang peminat game khususnya gameonline *multiplayer* ini sudah mencapai 100 juta orang. Studi menunjukkan seperempat dari jumlah total *gamer* memiliki usia 16-24 tahun dan 25-34 tahun, dimana persentasenya masing-masing 27 persen. 24 persen lainnya tercatat berusia 35-44 tahun.[[1]](#footnote-1)

Di berbagai gameonline, contohnya *DOTA 2*, *Counter Strike : Global Offensive, Mobile Legend* pasti ada barang-barang yang bisa digunakan untuk akun pemain tersebut atau hanya sebagai aspek dekorasi saja. Ada juga gamer yang menggunakan barangnya untuk berinvestasi. Terdapat beberapa cara mudah untuk mendapatkan barang tersebut yaitu dengan cara masuk kedalam game saja atau seperti menyelesaikan beberapa misi dalam game. Terdapat juga barang-barang yang sulit didapatkan yaitu lewat lotere atau lebih sering disebut sebagai *gacha* yang kemungkinan untuk mendapatkannya sangat kecil.

Kebanyakan gameonline sekarang memiliki saldo *in-game*. Saldo dalam game dapat digunakan untuk membeli barang yang baru saja dikeluarkan oleh game dengan harga yang cukup mahal dan tidak bisa didapatkan dengan cara lain selain membelinya. Jadi, banyak gamemembuat fitur untuk pemain agar bisa membeli barang dengan mudah tetapi harganya tidak murah bahkan sampai berjuta-juta. Oleh karena harga yang tidak murah tersebut, banyak toko yang menjual *voucher* untuk mengisi saldo game dengan harga yang lebih murah dari pada harga yang ada di game.

Tetapi, banyak orang yang menjual barang dari game, masih terpisah dengan penjualan voucher. Karena dalam dunia bisnis, lebih baik fokus ke salah satu hal saja dari pada menjalankan keduanya. Bagaimana jika kedua fitur tersebut digabungkan menjadi satu, pasti pengunjung dari toko akan semakin bertambah dan penghasilan akan terus meningkat seiring dengan harga asli dari game tesebut. Semakin lengkapnya suatu toko, maka semakin banyak peminat yang mengunjunginya, apalagi toko yang menawarkan harga yang murah dan terjangkau oleh setiap kalangan gamer.

Dengan melihat situasi ini, terlintas ide untuk membuatwebsite multi-transaksi. Jadi, di website ini tidak hanya menjual barang dan voucher, tetapi juga menyediakan fitur bertukar barang antar *user* tetapi tetap dalam sistem jual dan beli. Kemudian, terdapat *channel* mengobrol untuk interaksi atau kesepakatan antar useryang ingin menjual dan membelu barang miliknya. Dengan adanya website ini, proses jual beli barang yang ada digame dapat menjadi lebih mudah dan ringkas yang artinya, user dapat membeli voucher, barang dalam game dan game sekaligus tanpa harus mengunjungi website yang berbeda.

1. **Tujuan**

Pada sub bab ini akan dibahas tujuan-tujuan dari pembuatan website ini. Website ini bertujuan untuk memberikan tempat para gamer untuk melakukan transaksi dan berkumpul dengan gamer lain. Dengan ini, mereka tidak perlu mencari – cari website atau *software* untuk berkumpul dan menjual barang di game yang diperlukan atau diinginkan. Gamer juga akan lebih mudah untuk saling berinteraksi satu dengan yang lain saat proses pembelian barang. Berikut poin-poin dari tujuan website ini.

1. Membuat gamer mendapatkan barang yang ada di game dengan mudah atau menjual barang yang ingin dijual dan juga menyediakan tempat untuk para gamer berkumpul serta berinteraksi dengan gamer lain atau *merchant*.
2. Menyediakan perlombaan atau *tournament.* Selain meningkatkan interaksi antar user, diadakannya tournament ini agar setiap gamer dapat menunjukan kemampuannya dalam memainkan game tersebut.
3. Membantu merchantmempromosikan barang yang akan dijual, karena pengunjung website ini yang mayoritas merupakan gamer.
4. **Teori Penunjang**

Teori penunjang merupakan penjelasan mengenai apa saja yang akan menjadi penunjang pada pembuatan website. Dalam hal ini adalah front-end dan back-end. Pada sub bab ini akan dijelaskan *framework* / CMS apa saja yang akan digunakan. Berikut framework serta CMS yang digunakan dalam pembuatan website.

* + 1. **Front-End**

Front-end merupakan bagian website yang menyuguhkan tampilan kepada user yang membuka website tersebut. Dalam front-end biasanya dibuat menggunakan *HTML*, *CSS, Javascript* sehingga tampilan pada website akan terlihat lebih menarik. Pada website ini, akan menggunakan *library* framework yang bernama Bootstrap untuk mempermudah dalam mendesain website. Framework adalah software pembantu dalam membuat tampilan website.

* Bootstrap

Bootstrap merupakan sebuah library framework CSS yang telah dibuat khusus uintuk mengembangkan front end sebuah website. Bootstrap juga dikenal sebagai salah satu framework CSS, HTML, Javascript yang begitu populer di kalangan website *developer* atau pengembang website. Dengan adanya bootstrap tersebut tentu saja membuat halaman website bisa menjadi lebih mudah dan juga website bisa menyesuaikan dengan ukuran *monitor* *device*. Baik jika di akses lewat ponsel, tablet ataupun desktop. Karena adanya bootstrap, membuat atau mendesain website tidak dilakukan mulai dari 0 (awal). Ada beberapa kelebihan dari bootstrap antara lain open source, dan juga bootstrap menyediakan banyak icon untuk pembuatan website.

* Bootstrap.css

Bootsrap.css adalah file css yang digunakan untuk membantu mengembangkan website. Berisi penjelasan untuk masing-masing perintah yang mduah dibaca.

* Bootstrap.min.css

Bootstrap.min.css adalah file css yang digunakan membantu mengembangkan website sama seperti Bootstrap.css hanya saja ukurannya lebih kecil karena telah di kompres.

* Bootstrap.responsive.css

Bootstrap.responsive.css merupakan file css yang digunakan untuk implementasi fitur responsive pada website. Disini berisi perintah-perintah yang disertai komen agar mudah dipahami

* Bootstrap.responsice.min.css

Bootstrap.responsice.min.css merupakan hasil kompres dari Bootstrap.responsive.css agar ukuran menjadi lebih kecil.

* Bootstrap.js

Bootstrap.js adalah file javascript yang digunakan untuk melengkapi framework dan bisa digunakan dalam proses pengembangan website.

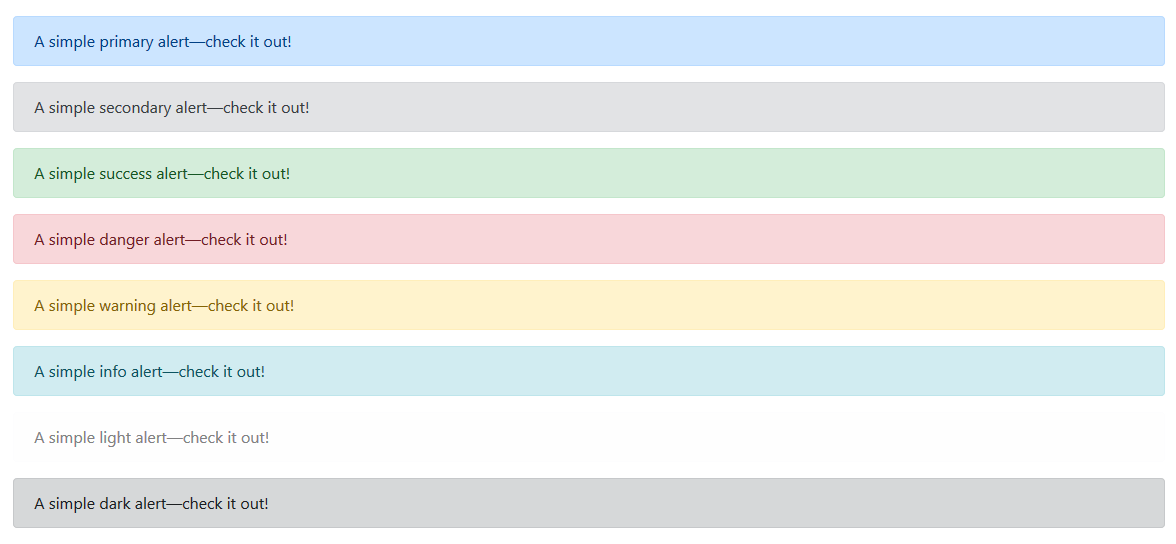
Untuk penulisan dari bootstrap juga cukup teratur. Jadi, user yang memakai tidak akan kebingungan saat menggunakan setiap komponennya. Ada banyak komponen yang disediakan oleh bootstrap. Jadi, jika ingin memakai komponen hanya perlu menyalin dari dokumentasi yang telah disediakan. Contoh potongan program pada bootstrap dapat dilihat pada Segmen Program 1.1 .

**Segmen Program 1.1 Penggunaan Bootsrap**

1. <div class="alert alert-primary" role="alert">
2. A simple primary alert—check it out!
3. </div>

**Segmen Program 1.1 Penggunaan Bootsrap (Lanjutan)**

1. <div class="alert alert-secondary" role="alert">
2. A simple secondary alert—check it out!
3. </div>
4. <div class="alert alert-success" role="alert">
5. A simple success alert—check it out!
6. </div>
7. <div class="alert alert-danger" role="alert">
8. A simple danger alert—check it out!
9. </div>
10. <div class="alert alert-warning" role="alert">
11. A simple warning alert—check it out!
12. </div>
13. <div class="alert alert-info" role="alert">
14. A simple info alert—check it out!
15. </div>
16. <div class="alert alert-light" role="alert">
17. A simple light alert—check it out!
18. </div>
19. <div class="alert alert-dark" role="alert">
20. A simple dark alert—check it out!
21. </div>

Pada Segmen Program 1.1 merupakan contoh dari penggunaan bootstrap yang digunakan dalam website. Komponen yang digunakan merupakan HTML. Perbedaannya, bootstrap telah menyediakan *class* sendiri. Jadi, pada setiap komponen dari bootstrap mempunyai nama class yang berbeda – beda, untuk contoh diatas meruapakan kode yang digunakan untuk membuat *alert*, maka digunakan nama class alert. Hasil yang ditulis dari Segmen Program 1.1 dapat dilihat pada Gambar 1.1.

**Gambar 1.1**

**Tampilan Alert menggunakan Bootstrap**

* + 1. **Back-End**

Back-end merpakan program yang menopang fitu-fitur dari sebuah website. Jika sebelumnya front-end fokus pada tampilan dari sebuah website, back-end fokus pada fungsionalitas dari sebuah website. Setiap back-end terdiri dari 3 (tiga) komponen yaitu server, database, dan aplikasi. Untuk mempermudah pembuatan back-end dapat digunakan juga framework seperti pada front-end, website ini akan menggunakan *CodeIgniter* untuk mempermudah dalam pengembangan back-end.

CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php agar lebih terstruktur. Seperti halnya framework lain, tujuan digunakan CodeIgniter ini yaitu untuk memudahkan programmer dalam mengembangkan website. CodeIgniter merupakan framework PHP dengan model MVC (*Model*, *View*, *Controller*) untuk mengembangkan website dengan menggunakan PHP yang dapat mempercepat dan mempermudah programmer untuk membuat sebuah aplikasi web.

Ada beberapa kelebihan dari framework CodeIgniter dari pada framework lainya. Pertama, performa dari CodeIgniter yang lebih cepat dibandingkan framework-framework lain. Dokumentasi yang disediakan untuk user juga lengkap. Banyak orang yang menggunakan framework ini jadi, para programmer dapar bertukar informasi serta wawasan baru terakait framework.

MVC merupakan teknik untuk memisahkan komponen utama menjadi tiga komponen yaitu Model, View, Controller. Pada pembuatan website ini akan digunakan konsep tersebut. Konsep MVC ini sering diterapkan dalam pembuatan aplikasi. Dengan adanya MVC ini, program akan lebih mudah dikelola. Berikut akan dijelaskan penjabaran dari MVC.

* Model

Model merupakan script program yang berperan untuk mengakses langsung database. Seperti, mengambil atau menginputkan ke database. Semua yang berkaitan tentang pengolahan databse di letakkan di model. Disini akan terdapat script yang nantinya akan dilempar ke controller sehingga dapat ditampilakan di view.

* View

View adalah script program yang berguna sebagai tampilan dalam sebuah website. Biasanya file ini berisi HTML, yang akan diatur oleh controller. View berguna untuk menerima data dari database dan kemudian ditampilkan kepada user. Jadi, tampilan pada *interface* akan dikumpulkan pada view untuk memisahkannya dengan controller dan model sehingga akan mempermudah programmer dalam mengembangkan website.

* Controller

Controller merupakan script yang digunakan untuk jembatan antara model dan view, yang artinya user tidak berhubungan dengan model secara langsung. Jadi, controller akan mengakses fungsi-fungsi yang ada pada model dan kemudian ditampilkan pada view. Controller diisini menggunakan bahasa pemrograman PHP.

* + 1. **Database**

Database merupakan kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang kemudian disimpan. Database yang akan digunakan untuk pembuatan website ini adalah MySQL yang merupakan basis data sumber terbuka yang paling popular dan banyak digunakan untuk aplikasi berbasis web seperti website dinamis dan e-commerce. Tahun 2013, MySQL merupakan basis data kedua yang paling banyak digunakan di dunia dan yang pertama untuk basis data sumber terbuka..[[2]](#footnote-2)

Dalam pembuatan database untuk website ini akan menggunaan bantuan *phpMyAdmin*. Kegunaan phpMyAdmin yaitu, [database](https://www.termasmedia.com/lainnya/software/69-pengertian-database.html) MySQL dan database MariaDB dapat dikelola lebih mudah yaitu dengan melalui antarmuka (interface) grafis. Aplikasi web ini ditulis menggunakan bahasa pemrograman PHP. Sebagaimana aplikasi-aplikasi lain untuk lingkungan web (aplikasi yang dibuka atau dijalankan menggunakan browser), phpMyAdmin juga mengandung unsur HTML/XHTML, CSS dan juga kode JavaScript. Dengan dukungan dari banyak developer dan translator, aplikasi web phpMyAdmin mengalami perkembangan yang cukup pesat dengan ketersediaan banyak pilihan bahasa. Sampai saat ini, ada kurang lebih 65 bahasa yang sudah didukung oleh aplikasi web phpMyAdmin.[[3]](#footnote-3)

* + 1. **API**

API adalah singkatan dari ***Application Programming* Interface**, dan memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API terdiri dari berbagai elemen seperti function, protocols, dan tools lainnya yang memungkinkan developer untuk membuat aplikasi. Tujuan penggunaan API adalah untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga developer tidak perlu membuat fitur yang serupa. Penerapan API akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya.[[4]](#footnote-4)

Dalam pembuatan website ini proses transaksi akan menggunakan API. API yang akan digunakan adalah *MidTrans*. [MidTrans](https://midtrans.com/) adalah salah satu layanan *payment gateway* yang digunakan dalam proses transaksi. MidTrans merupakan layanan penyedia payment getaway yang mudah dan praktis untuk digunakan. Penggunaan midtrans memerlukan beberapa langkah yang harus dijalankan. Berikut adalah langkah-langkah yang harus dilakukan untuk menggunakan midtrans:

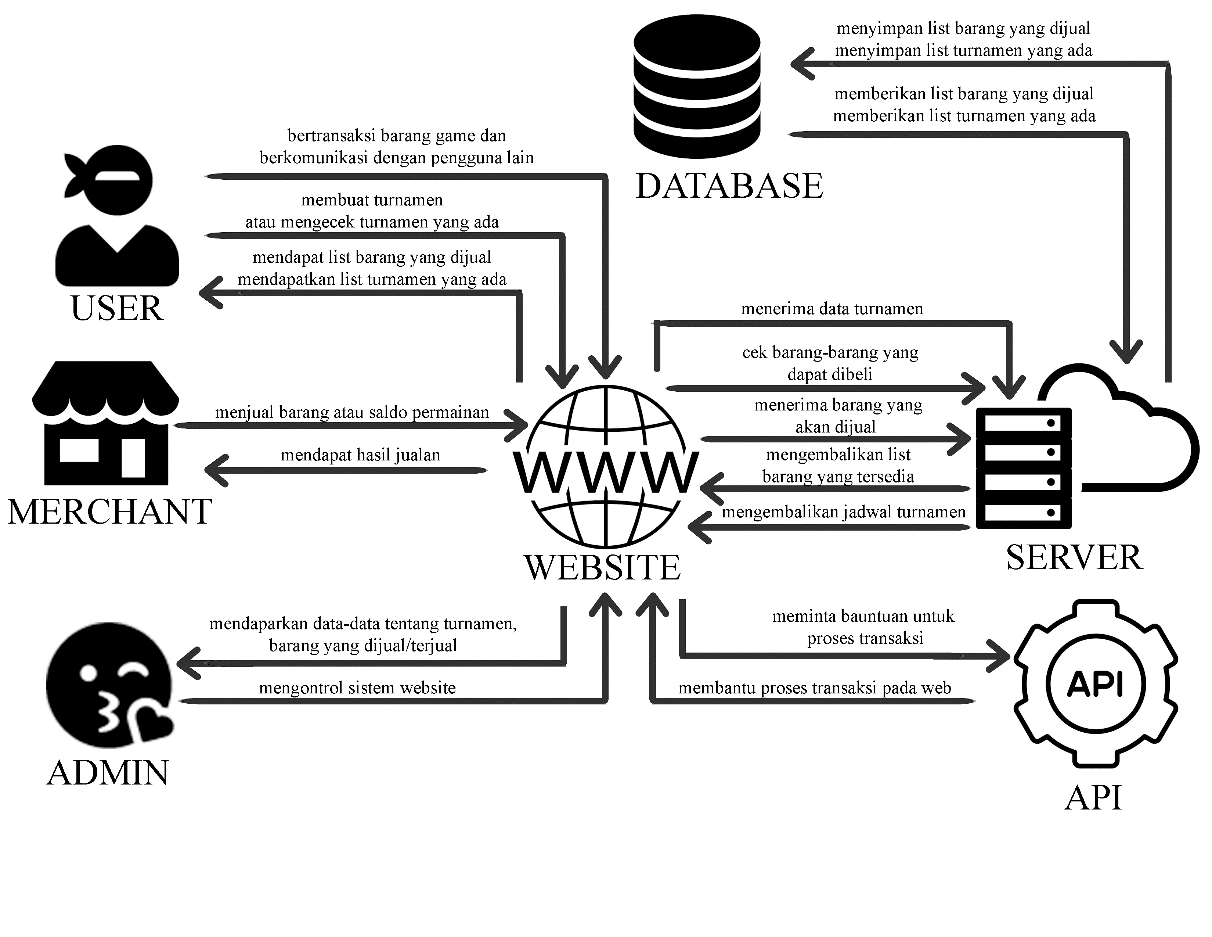
* + 1. Mendaftarkan web toko online dengan cara terhubung ke merchant admin portal MidTrans dalam mode testing
    2. Melanjutkan pendaftaran dengan melengkapi dokumen
    3. Mengintegrasikan teknis yang sudah tersedia ke website bisnis online/e-commerce dengan tim *developer.*

Kemudahan dari MidTrans

1. **Ruang Lingkup**

Ruang lingkup adalah sub bab yang akan menjelaskan mengenai sistem dari website ini. Tujuannya agar pembahasan lebih fokus dan tidak melebar kemana-mana. Disini akan dijelaskan juga batasan-batasan dari website ini serta fitur-fitur apa saja yang ada. Berikut akan dibahas ruang lingkup dari website ini.

* + 1. **Arsitektur Sistem**

Pada bagian arsitektur sistem ini akan dijelaskan mengenai struktur dan sistem dari website. Disini akan ditunjukan bagaimana hubungan antar komponen yang lebih spesifik dan secara strukur. Arsitektur sistem dari website ini dapat dilihat pada Gambar 1.2.

**Gambar 1.2**

**Arsitekstur Sistem**

Pada Gambar 1.2 dapat dilihat, terdapat 3 (tiga) aktor yang dapat menggunakan website ini. 3 (tiga) aktor tersebut adalah user, merchant, dan admin. Ketiga aktor ini mengakses website yang sama tetapi mereka akan mendapat tampilan yang berbeda-beda sesuai dengan *role* mereka. Tiap-tiap role ini juga memiliki fitur yang berbeda-beda dan memiliki keunggulannya masing-masing.

User akan mendapatkan fitur yang dicari oleh mereka saat mereka menemukan website ini. User dapat membeli barang sekaligus saldo atau voucher permainan yang diinginkan. Tidak hanya itu, user juga dapat mengadakan tournamen yang akan diikuti oleh user lainnya. Kemudian, setiap user dapat membuat toko yang akan digunakan untuk menjual barang yang nantinya akan disebut merchant. Merchant akan mendapatkan fitur layanan untuk membantu mereka menjualkan barang-barang game ataupun saldo game dan juga game yang mereka ingin jual kepada user. Kemudian yang terakhir, admin akan mengatur sistem-sistem yang ada agar tidak terjadi masalah-masalah yang tidak diinginkan.

Server bertugas untuk menyimpan data-data yang ada seperti data user dan merchant. Tetapi tidak hanya itu, server juga dapat menyimpan data barang-barang yang dijual merchant. Setelah itu, akan memberikannya sebagai list kepada user yang ingin membeli sesuatu. Kemudian, juga untuk menyimpan data-data tournament yang dibuat oleh user.

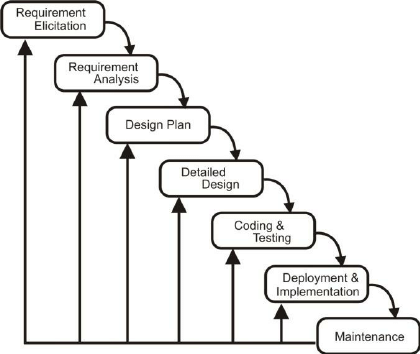
* + 1. **Batasan-batasan**

Batasan sistem merupakan suatu daerah yang membatasi antara satu sistem dengan sistem yang lain. Dalam setiap program atau website pasti ada sistem yang membatasi. Pada sub bab ini akan dijelaskan apa saja batasan-batasan dari website ini. Berikut batasan-batasan dari website yang dibuat.

* + 1. Saldo pada user tidak dapat ditarik kembali / diuangkan, dan transaksi hanya dapat dilakukan menggunakan saldo. Jadi, user harus mengisi saldo terlebih dahulu.
    2. Tidak dapat melakukan *call* dan *video* call jadi, hanya sebatas *chat*.

1. **Metodologi**

Metode *Iterative Waterfall* merupakan bentuk pengembangan dari Waterfall. Metode iterative Waterfall ini hampir sama dengan Waterfall. Terdapat beberapa hal yang membedakan yaitu, langkah-langkah yang telah dilakukan dapat diulang kembali. Jadi, apabila ada kesalahan pada langkah-langkah setelahnya, perulangan ini akah dapat digunakan untuk mendeteksi kesalahan yang ada. Jadi, model ini dapat mengurangi waktu yang dibutuhkan pengembang untuk mendeteksi dan memperbaiki kesalahan yang ada. Metode ini akan digunakan dalam pembuatan proyek ini. Gambaran dari metode iterative waterfall dapat dilihat pada Gambar 1.4.



**Gambar 1.4**

**Gambaran proses metode iterative Waterfall**

Dari Gambar 1.4 dapat dilihat proses yang dilakukan oleh metode ini, mulai dari analisa kebutuhah dari sistem. Dilanjutkan dengan desain dari sebuah sistem yang dibuat dan desain interface dari sistem tersebut. Pembuatan program akan dilakuakn setelah desain telah dibuat, setelah sebagian program telah dibuat dan sudah cukup untuk dilakukan uji coba maka akan dilakuakn uji coba. Tahap uji coba merupakan tahap yang akan sering terjadi perulangan karena akan sering ditemukan kesalahan dan akan diperbaiki sampai sistem benar-benar sempurna. Jika sistem sudah sempurna atau tidak terdapat kesalahan lagi, maka sistem akan dapat dirilis. Setelah sistem dirilis, akan dilakuakan pemeliharaan secara berkala terhadap sistem agar sistem berjalan dengan baik, pada tahap ini juga tidak menutup kemungkinan untuk melakukan perulangan bila terjadi kesalahan.

1. **Sistematika Pembahasan**

Pada sub bab ini akan di list apa saja bab yang ada pada buku. Kemudian, akan dijelaksan secara singkat juga apa saja isi yang ada pada setiap bab. Mulai dari bab satu (pendahuluan) sampai dengan bab tujuh (penutup). Berikut beberapa penjelasan singkat dari setiap bab yang ada.

* BAB I : PENDAHULUAN

Dalam bab ini dijelaskan tentang latar belakang, tujuan, ruang lingkup, teori penunjang, fitur-fitur, dan sistematika pembahasan.

* BAB II : ANALISA

Dalam bab ini menjelaskan tentang program pembanding, analisa masalah, analisa kebutuhan, deskripsi sistem baru, spesifikasi kebutuhan, laporan, serta fitur-fitur

* BAB III : DESAIN

Dalam bab ini dijelaskan tentang desain sistem, desain database, desain interface, desain table, dan desain algoritma,

* BAB IV : IMPLEMENTASI

Dalam bab ini dijelaskan tentang apa saja yang ada di website ini disertai dengan gambar dari website yang telah jadi.

* BAB V : UJI COBA

Dalam bab ini dijelaskan tentang hasil dari uji coba yang telah dilakukan.

* BAB VI : USER MANUAL

Dalam bab ini dijelaskan tentang cara penggunaan setiap fitur yang ada di website.

* BAB VII : PENUTUP

Dalam bab ini dijelaskan tentang kesimpulan dan saran dari setiap user yang telah mencoba website ini.

1. https://www.tek.id/insight/jumlah-gamer-di-indonesia-capai-100-juta-di-2020-b1U7v9c4A [↑](#footnote-ref-1)
2. https://www.termasmedia.com/lainnya/software/69-pengertian-database.html [↑](#footnote-ref-2)
3. https://www.termasmedia.com/database/phpmyadmin/131-mengenal-phpmyadmin-dan-kegunaannya.html [↑](#footnote-ref-3)
4. <https://www.codepolitan.com/mengenal-apa-itu-web-api-5a0c2855799c8>

   5 <https://www.dewaweb.com/blog/midtrans-payment-gateway/> [↑](#footnote-ref-4)