BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Teori ini membantu dalam memahami bagaimana website dirancang dan diimplementasikan untuk mendukung pembuatan Aplikasi Donasi *Online* Berbasis *Website* Menggunakan *API Payment Gateaway*.

Penelitian yang dilakukan Sri Mulandari dkk (2021) dalam jurnal penelitiannya yang berjudul ”Sistem Informasi Donasi *Online* Berbasis *Website*”. Menurutnya banyaknya masyarakat diberbagai daerah saling membantu dengan caranya masing-masing seperti galang dana di persimpangan lampu merah, keliling rumah warga, maupun ke beberapa instansi swasta atau donasi. *Website* donasi *online* ini dirancang untuk memberikan kemudahan dan membantu para donatur untuk dapat melakukan donasi secara *online* kepada masyarakat yang membutuhkan. Untuk pengembangan sistem informasi donasi *online* berbasis *website* kedepannya salah satunya seperti menghubungkan sistem dengan sistem pembayaran payment gateaway.[6]

Penelitian yang dilakukan Yoki Firmansyah dkk (2020) dalam jurnal penelitiannya yang berjudul ”Sistem Informasi ”Yukdonasi” Sebagai Media Penggalangan Donasi *Online* Berbasis *Website*”. Menurutnya masyarakat atau donatur khawatir karena tidak bisa mengecek apakah donasi yang diberikan tersampaikan dengan tepat sasaran, karena tidak bisa di cek dan

tidak ada media tempat mereka melihat donasi yang disalurkan. Dengan adanya aplikasi ini dapat mempermudah donatur mencari penggalang dana yang membutuhkan donasi tanpa pergi langsung ke lokasi penggalang dana tersebut.[7]

Penelitian yang dilakukan Hilda Amalia dkk (2017) dalam jurnal penelitiannya yang berjudul ”Sistem Informasi Pengolahan Dana Donasi”. Menurutnya suatu data yang tidak memiliki sistem informasi yang mendukung tidak akan bermanfaat atau tidak akan menjadi suatu informasi yang bisa disampaikan kepada sasaran penerima indormasi. Dengan adanya sistem informasi Pengolahan Dana Berbasis *Website* dapat memberi kemudahan untuk tidak hanya dari sisi masyarakat sebagai calon donatur ataupun donatur, tetapi juga memberikan kemudahan disisi Pesantren Yatama Az-Zikra sebagai *administrator*. Sesuai dengan tujuan pembuatan sistem informasi ini, berbgai informasi dapat disampaikan, diterima dan dikelola dengan mudah, cepat dan akurat.[8]

Penelitian yang dilakukan Deni Fajri dkk (2019) dalam jurnal penelitiannya yang berjudul ”Rancang Bangun Sistem Informasi Pengumpulan Dana Panti Asuhan menggunakan Metode *Crowdfunding* dengan *Model* Situs Donasi”. Menurutnya mekanisme metode *crowdfunding* ini aktor dimulai dari kreator (pencari dana) dan portal *crowdfunding* sebagai penghubung dan masyarakat sebagai donatur. Proses *crowdfunding* dimulai dengan kreator mencari dana dengan melakukan akses ke portal

crowdfunding. Setelah akses aktor melakukan pengisian pencarian dana pada portal *crowdfunding*. Jika donatur tertarik pada proyek tersebut maka dapat berpatisipasi untuk memberikan donasi dengan cara *transfer* ke rekening bank penyedia portal *crowdfunding*.[9]

Penelitian yang dilakukan Fabriyan Fandi Dwi Imaniawan (2020) dalam jurnal penelitiannya yang berjudul ”Sistem Informasi Penyaluran Donasi Berbasis *Web*”. Menurutnya Penerapan sistem informasi pada era *millenial* saat ini sangat penting dilakukan bagi sebuah lembaga ataupun organisasi sosial. Diperlukan adanya media untuk memenuhi kebutuhan akan informasi dan layanan, seperti *website*. Bagi sebuah lembaga maupun organisasi sosial, *website* saat ini memiliki peran yang penting. Permasalahan yang muncul selama ini terdapat di dalam pengelolaan transaksi donasi yang masih manual sehingga membutuhkan waktu yang lama. Hal ini menyebabkan lambatnya proses pembuatan laporan karena banyaknya donasi yang belum terkonfirmasi. Diharapkan dengan adanya sistem ini, pengelolaan dari transaksi penerimaan hingga penyaluran donasi dapat lebih efektif dan efisien sehingga berpengaruh kepada laporan keuangan yang dapat dipertanggungjawabkan secara transparan kepada donatur.[10]

2.2 Landasan Teori

2.2.1 JavaScript

JavaScript adalah bahasa yang digunakan untuk membuat program yang digunakan agar dokumen *HTML* yang ditampilkan pada

sebuah *Browser* menjadi lebih interaktif, tidak sekedar indah saja. *JavaScript* memberikan beberapa fungsionalitas ke dalam halaman *web*, sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antar muka *web*.[11]

Perhatikan gambar logo *Javascript* dibawah ini :

Gambar 2.1 Logo JavaScript

2.2.2 CSS

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen *web* yang berfungsi mengatur elemen *HTML* dengan berbagai *property* yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan. Sebagian orang menganggap *CSS* bukan termasuk salah satu bahasa pemrograman karena memang strukturnya yang sederhana, hanya berupa kumpulan-kumpulan aturan yang mengatur style elemen *HTML*.[12]

Perhatikan gambar logo *CSS* dibawah ini :

Gambar 2.2 Logo CSS

2.2.3 MySQL

MySQL adalah salah satu jenis *database* yang banyak digunakan untuk membuat aplikasi berbasis *web* yang dinamis. *MySQL* termasuk jenis *RDBMS* (*Relational Database Management Sistem*). *MySQL* ini mendukung Bahasa pemrograman *PHP*. *MySQL* juga mempunyai *query* atau bahasa *SQL*(*Structured Query Language*) yang simple dan menggunakan *escape character* yang sama dengan *PHP*.[13]

Perhatikan gambar logo *MySQL* dibawah ini :

Gambar 2.3 Logo MySQL

2.2.4 Payment Gateway (duitku.com)

Payment Gateway adalah salah satu cara untuk memproses transaksi elektronik. *Payment gateway* menyediakan alat - alat untuk memproses pembayaran antara *customer*, *businesses* dan *bank*. *Payment gateway* merupakan bagian terpenting dari suatu transaksi antar *customer*, *business*, dan lembaga - lembaga perbankan yang keduanya digunakan. *Payment Gateway* digunakan untuk memfasilitasi transaksi elektronik.[14]

Duitku.com merupakan *payment gateway* di Indonesia yang siap menghubungkan *website* bisnis *online* dengan berbagai macam

pembayaran, termasuk *e-wallet*. Selain itu ada juga pembayaran menggunakan kredit *online*, *virtual account*, sampai pembayaran di gerai retail seperti indomaret dan kantor pos.

Perhatikan gambar logo Duitku dibawah ini :

Gambar 2.4 Logo Duitku

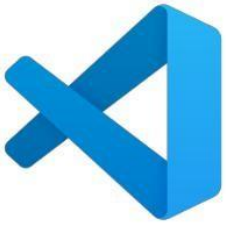
2.2.5 Hosting

Hosting merupakan tempat penyimpanan data *website* dimana didalamnya meliputi kapasitas penyimpanan, *bandwith* yang merupakan sebuah kapasitas yang di gunakan untuk mengukur jumlah pengunjung *website* serta *database*.[15]

2.2.6 Visual Studio Code

Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows.[16]

Perhatikan gambar logo *Visual Studi Code* dibawah ini :

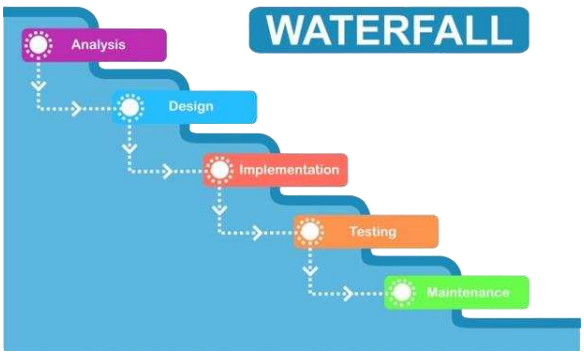


Gambar 2.5 Logo Visual Studio Code

2.2.7 Metode Waterfall

Model *waterfall* pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineerin*g (*SE*). saat ini model *waterfall* merupakan model pengembangan perangkat lunak yang sering digunakan. Model pengembangan ini melakukan pendekatan secara sistematis dan berurutan. Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.[17]

Perhatikan gambar metode *Waterfall* dibawah ini :



Gambar 2.6 Gambar Metode Waterfall

2.2.8 Github

Github merupakan *software hosting* untuk *open source* dengan menggunakan *tool Git*, *Github* juga di posisikan sebagai *Webhosting*. *Git* sendiri merupakan *Tool System Control* yang kegunaannya sebagai mengontrol berbgai *code* bahasa pemrograman. *Git* dipilih

karena dari sekian banyak *VCS* merupakan yang paling populer. Selain itu, mudah dan digunakan secara gratis serta *open source*.[18] Perhatikan gambar logo *Github* dibawah ini :

Gambar 2.7 Logo Github

2.2.9 UML (*Unified Modeling Languange*)

Unified Modeling Language atau lebih sering dikenal dengan sebutan *UML*, adalah salah satu metode dalam teknik rekayasa perangkat lunak yang digunakan untuk menggambarkan alur dan cara kerja sistem, fungsi, tujuan dan mekanisme kontrol sistem tersebut. Dalam teknik rekayasa perangkat lunak bidang analisis dan perancangan sistem informasi, saat ini lebih banyak menggunakan gabungan dari konsep pemrograman berorientasi objek dengan teknik pembuatan perangkat lunak, dimana suatu sistem dilihat sebagai objek tersendiri yang sudah mencakup data dan proses atau dapat bekerja secara mandiri dalam satu set sistem (*package*).

Dalam teknik perancangan sistem informasi, terdapat 4 model *UML* yang paling efektif penggunaanya untuk menggambarkan desain sistem, yaitu: *Use Case diagram*, *Class diagram*, *Sequence diagram*, dan *Behavioral State Machine diagram*. Empat teknik pemodelan *UML* ini juga disebut sebagai 4 teknik dasar (*core*) pemodelan

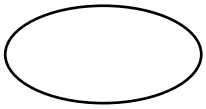
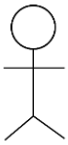
berbasis *UML*. Keempat teknik pemodelan *UML* ini telah mendominasi penggunaannya dalam proyek-proyek berorientasi objek.[19]

Dalam Perancangan beriorentasi objek berbasiskan *UML* menggunakan alat bantu sebagai berikut:

1. *Use Case Diagram*

   Use case diagram merupakan pemodelan untuk kelakuakan (*behavior*) sistem informasi yang akan dibuat. *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* bisa dilihat pada tabel dibawah.

   Tabel 2.1 Simbol Use Case Diagram

No	Simbol	Keterangan
1		*Use Case* : Abstraksi dan interaksi antara sistem dan aktor.
2		*Actor* : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case.*

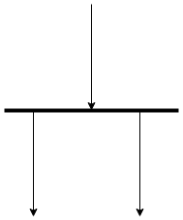
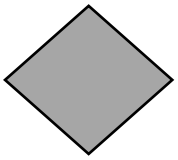
No	Simbol	Keterangan
3		Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
4		Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
5	<<include>>	*Include*, Menunjukkan spesialisasi *actor* untuk dapat berpartisipasi dengan *use case*.
6	<<extends>>	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

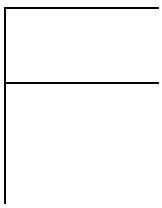
2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-

simbol yang digunakan dalam *activity diagram* bisa dilihat pada tabel dibawah.

Tabel 2.2 Simbol Activity Diagram

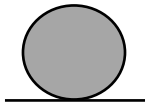
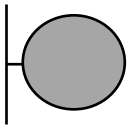
No	Simbol	Keterangan
1		*Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
2		*End Point*, akhir aktivitas.
3		*Activities*, menggambar kan suatu proses/kegiatan bisnis.
4		*Fork/*percabangan digunakan untuk menunjukan kegiatan yang dikakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5		*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true* atau *false.*

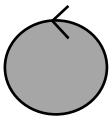
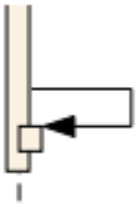
No	Simbol	Keterangan
6		*Swimlane*, pembagian *activity diagram* untuk menunjukkan siapa melakukan apa.

3. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* Bisa dilihat tabel di bawah.

Tabel 2.3 Sequence Diagram

No	Simbol	Keterangan
1		*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
2		*Boundary Class*, berisi kumpulan kelas yang menjadi *interfaces* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan *form entry* dan *form* cetak.

No	Simbol	Keterangan
3		*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
4		*Message*, simbol mengirim pesan antar *class.*
5		*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
6		*Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
7		*Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

2.2.10 Pengujian Sistem

Pengujian sistem adalah proses untuk mengecek apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai standar atau belum.[20]

Pengujian dalam perangkat lunak terbagi menjadi 2 yaitu *black box testing* dan *white box testing*.

1. *White Box Testing*

   White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam *output* yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya *White Box Testing* ini menguji dengan cara melihat *Pure Code* dari suatu aplikai/*software* yang diuji tanpa memperdulikan Tampilan atau *UI* dari aplikasi tersebut.

2. *Black Box Testing*

   Black Box Testing adalah pengujian yang dilakukan berdasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Black-box Testing* ini lebih menguji ke Tampilan Luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh *Customer*. Pengujian ini tidak melihat dan menguji *soucer code program*. *Black-box Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatianya hanya terfokus pada informasi *domain*