

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Perjanjian Kerja Sama**

Perjanjian kerja sama merupakan hasil perundingan antara serikat pekerja dan pengusaha yang memuat ketentuan kerja, hak, dan kewajiban antar kedua belah pihak. [1]

#### **2.2 Website**

*Website* merupakan sebuah media informasi yang ada di *internet*. *Website* tidak hanya dapat digunakan untuk penyebaran informasi namun bisa digunakan untuk membuat toko *online*.

*Website* adalah kumpulan dari halaman – halaman situs yang biasanya terangkum dalam sebuah domain atau sub domain, yang tempatnya berada didalam *World Wide Web* (www) di *internet*. [2]

Sebuah halaman *web* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*) dan hampir selalu bisa diakses melalui HTTP (protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*).

#### **2.3 HTML (*Hyper Text Markup Language*)**

*Hyper Text Markup Language* (HTML) adalah bahasa standar yang digunakan untuk menampilkan konten pada halaman *website*. [3] Fungsi – fungsi yang dapat dilakukan dengan bahasa pemograman HTML adalah:

1. Mengatur serta mendesain tampilan isi halaman *website*.
2. Membuat tabel pada halaman *website*.
3. Mempublikasikan halaman *website* secara *online*.
4. Membuat *form* yang dapat menjadi *input* serta menangani registrasi dan transaksi melalui *website*.
5. Menampilkan area gambar pada *browser*.

## 2.4 PHP

PHP merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk pengembang *web*. PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada akhir tahun 1994.

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *website*. *Website* yang baik hendaknya memiliki sifat dinamis dan interaktif maksudnya *website* tersebut bisa berupa tampilan kontennya sesuai kondisi tertentu dan dapat memberikan *feedback* bagi *user*. PHP merupakan bahasa pemrograman berjenis *server-side*. [4]

## 2.5 MySQL

MySQL adalah sebuah *database manajemen sistem* (DBMS) yang populer yang memiliki fungsi sebagai *relational database manajemen system* (RDBMS). MySQL *software* merupakan suatu aplikasi yang sifatnya *open source* serta *server* basis data MySQL memiliki kinerja yang baik, realibel, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems*. [5]

## 2.6 Framework

*Framework* adalah kumpulan *library* yang memiliki kemampuan dan fungsi yang berbeda satu sama lain yang dikembangkan oleh *developer* yang tergabung dalam komunitas *developer framework*. Pada *framework* terdapat fungsi yang dapat mengatasi permasalahan yang ada pada PHP *Native* yaitu mengatasi aktivitas atau pekerjaan yang berulang, sehingga pekerjaan dapat dilakukan lebih mudah dan cepat. [6]

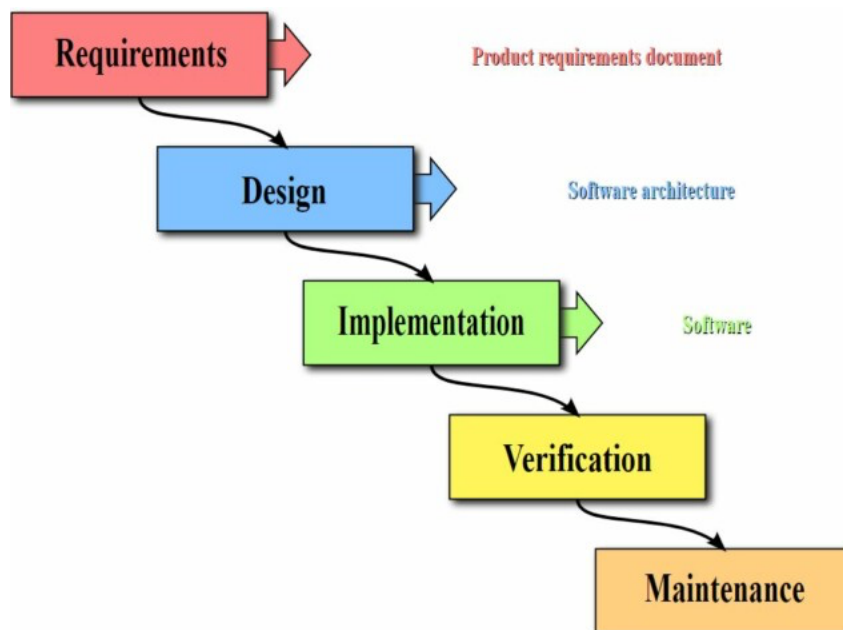
## 2.8 Laravel

*Laravel* merupakan *framework* yang dikembangkan oleh Taylor Otwell pada bulan Juni 2011 yang memiliki banyak pengguna. Pada *framework Laravel* terdapat fungsi – fungsi kode yang disediakan di *library* kemudian di *install* kedalam *Laravel*. Keuntungannya adalah penyebaran komunitas yang besar berdampak pada penemuan banyak *library* yang berbeda, beberapa peneliti

menemukan bahwa *library* di *Laravel* dapat cukup besar sehingga, mereka dapat menyelesaikan proyek pengembangan *web* dari skala rendah hingga menengah. [6]

## 2.9 Model Pengembangan Perangkat Lunak

Pada pengembangan perangkat lunak, penulis menggunakan metode air terjun (*waterfall*). Model SDLC air terjun (*waterfall*) sering disebut juga model sekuensial *linear* (*Sequential Linear*) atau alur hidup klasik (*Classic Life Cycle*). Model ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). [7]



**Gambar 2.1 Model Pengembangan Perangkat Lunak *Waterfall***

### 1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa saja yang dibutuhkan oleh *user*.

## 2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

## 3. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah sistem informasi sesuai dengan desain yang telah dibuat pada tahap desain.

## 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*Error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

## 5. Pendukung atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan bahwa sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan sistem informasi yang sudah ada, tapi tidak untuk membuat sistem informasi yang baru.

## 2.10 Quick Sort

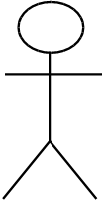
*Quick Sort* adalah algoritma pengurutan data yang sangat bergantung pada *elemen* pivot dengan cara kerja metodenya dengan mereduksi tahap demi tahap sehingga menjadi 2 bagian yang lebih kecil (*Divide and Conquer*). Elemen angka atau *n* yang besar akan dipartisi menjadi dua sub-elemen yang mana salah satunya berisi elemen yang lebih kecil dari pivot. Kemudian *quick sort* akan memanggil dirinya sendiri secara rekursif untuk mengurutkan elemen angka. [8]


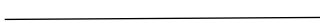
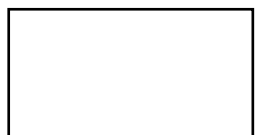
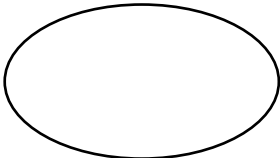
## 2.11 Use Case Diagram

*Use case* diagram adalah diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) dan sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case diagram* terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, sistem lain. Ataupun yang berinteraksi dengan system. [9]

*Use case* merupakan dokumen naratif yang mendeskripsikan kasus – kasus atau kejadian aktor dalam menggunakan sistem untuk menyelesaikan sebuah proses.

**Tabel 2.1 Simbol-Simbol pada Use case Diagram**

Simbol	Keterangan
Aktor 	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
Include	Hubungan dimana perubahan yang terjadi pada suatu elemen

	mandiri (dependent) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent)
<p>Association</p> 	Apa yang menghubungkan antara objek satu dengan objek yang lainnya
<p>Sistem</p> 	Menspesifikasikan paket yang menampilkan sistem secara terbatas
<p>Use Case</p> 	Mendesripsikan urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

## 2.12 Penelitian Sebelumnya

**Tabel 2.2 Literature Review / Penelitian Terdahulu**

No.	Judul	Penulis	Pembahasan	Tahun
1	Pengembangan Sistem Informasi Pendaftaran Beasiswa Peningkatan Prestasi Akademik (PPA) Jurusan Ilmu Komputer Fakultas	Dwi Sakethi, Febi Eka Febriansyah, Novita Sari.	Proses pendaftaran yang dilakukan selama ini setelah para mahasiswa yang mendaftar beasiswa mengumpulkan berkas pendaftaran	15

	<p>Matematika dan Ilmu Pengetahuan Alam Universitas Lampung dengan Menerapkan <i>Algoritma Sorting Quick Sort</i> dan <i>Selection Sort</i>.</p>		<p>ke pihak dekanat, data-data tersebut diserahkan kembali ke dekanat untuk diproses dan untuk menentukan siapa saja mahasiswa yang berhak menerima beasiswa.</p> <p>lam menu beasiswa terdapat 5 sub menu (sub menu lihat data pendaftaran, sub menu akumulasi skor pendaftaran beasiswa (dengan metode <i>quick sort</i>), sub menu akumulasi pendaftaran beasiswa (<i>selection sort</i>), sub menu lihat pengumuman beasiswa, dan sub menu data pendaftaran beasiswa lama).</p> <p>da menu pengaturan terdapat 4 sub</p>	
--	--	--	--	--

			<p>menu (sub menu pengaturan pendaftaran beasiswa, sub menu pengaturan skor beasiswa, sub menu pengaturan skor nilai bobot beasiswa, dan sub menu pengaturan akun mahasiswa).</p> <p>cara garis besar, dalam penelitian ini, <i>quick sort</i> digunakan untuk mengurutkan skor pendaftaran beasiswa.</p>	
2	<p>Aplikasi <i>Learning &amp; Test</i> Karyawan Dengan <i>Algoritma Quick Sort</i>.</p>	<p>Ndaru Ruseno, HS. Sulistiyowati.</p>	<p>Pada penelitian ini dibangun sebuah aplikasi learning &amp; test karyawan dengan menggunakan algoritma <i>quick sort</i>. <i>Algoritma Quick Sort</i> merupakan algoritma yang membandingkan suatu elemen</p>	22



		<p>(pivot) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen lain yang lebih kecil daripada pivot tersebut terletak di sebelah kirinya dan elemen-elemen lain yang lebih besar daripada pivot tersebut terletak di sebelah kanan.</p> <p>System informasi yang dibangun bertujuan untuk mengatasi masalah yang dihadapi oleh perusahaan yakni adanya situasi dan kondisi yang belum memungkinkan untuk melakukan tes karyawan secara tatap muka sehingga aplikasi dibuat dengan</p>	
--	--	--	--

			<p>tujuan untuk memudahkan para karyawan untuk tetap mengikuti test seleksi pengangkatan jabatan karyawan tanpa harus datang ke lokasi.</p>	
3	<p>Pengolahan Data Kuisisioner Pengguna <i>Website</i> Menggunakan Metode Pengurutan <i>Quick Sort</i> Guna Tercapainya Tujuan <i>Human Computer Interaction</i>.</p>	<p>Dwi Nugraheny.</p>	<p>Pada penelitian ini akan mengelola kuisisioner pengguna <i>website</i> di STTA menggunakan metode pengurutan <i>quick sort</i> guna mengetahui apakah <i>website</i> yang sudah ada memenuhi tujuan dari <i>Human Computer Interaction</i> (HCI) berdasarkan beberapa variabel dari <i>user interface</i> dan <i>user experience</i>. <i>Quick Sort</i> adalah algoritma <i>sorting</i> yang berdasarkan</p>	18

			<p>pembandingan dengan metoda <i>divide-and-conqueror</i>. <i>Quick sort</i> adalah metode pengurutan yang ideal beradaptasi untuk pengurutan dalam penyimpanan yang random.</p>	
4	<p>Rancang Bangun Aplikasi Tes <i>Toefl</i> Menggunakan Algoritma <i>Quick Sort</i> Berbasis Komputer.</p>	<p>Nuri David Maria Veronika, Yulia Darnita.</p>	<p>Bahasa Inggris merupakan salah satu bahasa yang paling penting, karena memiliki peran sebagai bahasa internasional. Namun untuk mengetahui tingkat pemahaman seseorang terhadap bahasa inggris diperlukan sebuah tes uji pemahaman bahasa inggris, salah satunya dengan mengikuti <i>Test Of English as a Foreign</i></p>	15

			<p><i>Language</i> (TOEFL). Cara kerja Algoritma <i>Quick Sort</i> adalah dengan membandingkan suatu elemen (pivot) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen lain yang lebih kecil daripada pivot tersebut terletak di sebelah kirinya dan elemen-elemen lain yang lebih besar daripada pivot tersebut terletak di sebelah kanan.</p>	
5	Rancang Bangun Sistem Informasi Lowongan Kerja di JPC Polinema dengan Metode <i>Quick Sort</i> .	Budi Harijanto, Deddy Kusbianto P, Julia Intan Amini.	Tempat penelitian dalam jurnal ini berada di JPC Polinema. JPC Polinema merupakan suatu lembaga atau wadah yang	18

			<p>membantu Politeknik Negeri Malang (Polinema) dalam mengelola para alumni dan pencari kerja untuk menginformasikan lowongan kerja yang tersedia. Akan tetapi sistem kerja JPC Polinema saat ini masihlah semi manual, hal ini membuat calon pelamar merasa sedikit kesulitan. Maka dari itu dibutuhkannya sebuah aplikasi yang dapat mengelolah batas waktu terakhir pendaftaran pekerjaan dengan tampilan yang selalu terupdate dan dapat menampilkan waktu yang mendekati masa</p>	
--	--	--	--	--

			terakhir pendaftaran lowongan pekerjaan secara terurut. Metode yang digunakan untuk mensorting semua informasi lowongan kerja yang masuk menggunakan metode <i>Quick Sort</i> .	
--	--	--	--	--

### 2.13 Pengujian *Black Box Testing*

*Black box testing* merupakan pengujian fungsional berdasarkan pada apa yang harus disistem lakukan. Pengujian yang dilakukan yaitu melihat apakah fungsi-fungsi pada perangkat lunak telah berjalan sesuai dengan perancangan. Keuntungan menggunakan *black box testing* yaitu penguji tidak perlu memiliki pengetahuan mengenai bahasa pemrograman tertentu dan pengujian ini dilakukan melalui sudut pandang pengguna sehingga dapat mengungkap kekurangan atau kesalahan dalam spesifikasi persyaratan. (Azima et al., 2018)