

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. IT Competition**

IT *Competition* merupakan sebuah ajang kompetisi yang diselenggarakan untuk mengembangkan serta mengeluarkan kemampuan para pesertanya dalam bidang Teknologi Informasi (IT). IT *Competition* dapat meliputi berbagai bidang, yaitu pemrograman, jaringan, multimedia, dan lain sebagainya. IT *Competition* dapat diikuti oleh siswa/i SMA/SMK sederajat, mahasiswa, bahkan masyarakat umum dapat mengikutinya.

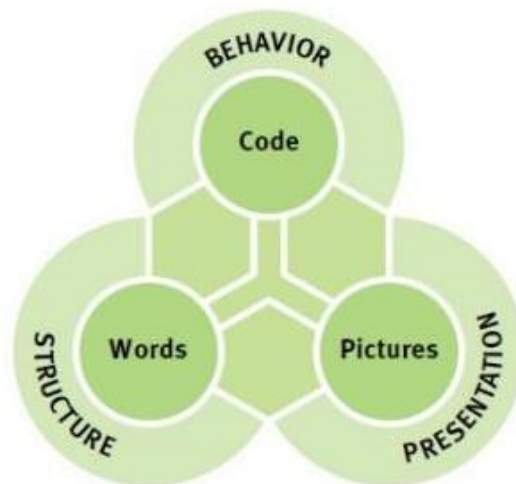
Program Studi Teknik Informatika Institut Informatika dan Bisnis Darmajaya rutin mengadakan perlombaan IT *Competition*, yaitu pada bidang Rekayasa Perangkat Lunak (RPL), Multimedia, dan Jaringan Komputer. Rekayasa Perangkat Lunak (RPL) berkaitan langsung dengan pengembangan sebuah perangkat lunak (*software*), Multimedia berkaitan langsung dengan pengolahan data multimedia, dan Jaringan Komputer berkaitan langsung dengan jaringan (*network*). Tahapan proses dari perlombaan IT *Competition* yang diselenggarakan oleh Program Studi Teknik Informatika adalah sebagai berikut.

1. Pendaftaran Peserta, yaitu merupakan proses awal yang bertujuan untuk mendapatkan informasi peserta dan menentukan syarat dan ketentuan yang harus dipatuhi oleh peserta.
2. Penilaian Karya, yaitu merupakan proses penilaian yang dilakukan oleh dewan juri berdasarkan kriteria yang telah ditentukan.
3. Pengumuman Pemenang, yaitu merupakan proses untuk mengumumkan nilai akhir dan pemenang dari masing-masing bidang perlombaan.

#### **2.2. Website**

Menurut (Veen, 2001) dalam (Sidik, 2019) *Website* merupakan sekumpulan halaman yang berisi informasi berbentuk digital. Informasi tersebut dapat berupa tulisan teks, gambar, suara, video, animasi ataupun gabungan dari semua elemen tersebut. *Website* dibentuk dari tiga komponen utama, yaitu *words* (teks), *pictures*

(gambar), dan *code* (koding). *Words* merupakan kata-kata yang menyusun konten dari sebuah *website* agar dapat dipahami sebagai sebuah informasi. *Pictures* merupakan gambar yang menghiasi kata-kata ataupun informasi berbentuk gambar yang dipadukan dengan kata-kata agar dapat dengan mudah dipahami. *Code* merupakan baris-baris perintah dalam bentuk kode-kode, contohnya seperti kode *HyperText Markup Language* (HTML), kode *Cascading Style Sheet* (CSS), dan kode lainnya yang digunakan untuk mengatur bagaimana informasi dalam bentuk teks maupun gambar dapat ditampilkan pada *software browser* baik pada komputer maupun *smartphone*. Komponen-komponen pembangun *website* tersebut dapat dilihat pada gambar 2.1.



Gambar 2. 1 Komponen Utama *Website*

Seiring dengan perkembangan internet, hubungan komponen *website* (*words*, *pictures*, dan *code*) mengalami perluasan ruang lingkup dan dikategorikan menjadi tiga bagian, yaitu *structure*, *presentation*, dan *behavior*. Ketiga bagian ini dapat dilihat pada gambar 2.1 di atas. Pengertian dari masing-masing bagian tersebut adalah sebagai berikut.

1. *Structure*, yaitu informasi yang ditekankan pada pengertian bagaimana informasi di dalam sebuah *website* disusun.
2. *Presentation*, yaitu bagaimana sebuah *website* ditampilkan secara visual yang menarik terhadap setiap orang yang mengakses *website* tersebut.

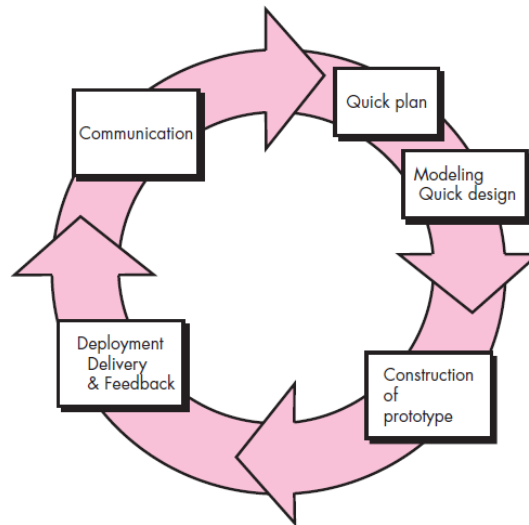
3. *Behavior*, yaitu bagaimana pengguna berinteraksi dengan *website* dan reaksi apa saja yang diharapkan muncul dari interaksi tersebut.

Menurut (Sklar, 2015) dalam (Sidik, 2019) secara umum *website* memiliki 4 fungsi utama, yaitu *website* sebagai komunikasi, *website* sebagai informasi, *website* sebagai transaksi, *website* sebagai hiburan. Pengertian dari keempat fungsi tersebut adalah sebagai berikut.

1. Fungsi Komunikasi, yaitu *website* yang fungsi utamanya untuk melakukan komunikasi dengan orang lain secara dinamis. Contohnya adalah Facebook, Gmail, Kaskus, dan lainnya.
2. Fungsi Informasi, yaitu *website* yang fungsi utamanya adalah untuk menyampaikan informasi dengan keterbaruan dan kebenaran dari informasi yang disampaikan. Contohnya adalah *website* berita, Wikipedia, dan lainnya.
3. Fungsi Transaksi, yaitu *website* yang fungsi utamanya adalah untuk melakukan transaksi secara *online* dan berhubungan langsung dengan keuangan. Contohnya adalah *website e-commerce*, PayPal, dan lainnya.
4. Fungsi Hiburan, yaitu *website* yang fungsi utamanya adalah untuk menghibur siapapun yang mengunjungi *website* tersebut. Contohnya adalah *website social media*, *website* permainan, dan lainnya.

### **2.3. Model Prototype**

Menurut (Pressman, 2010) dalam (Rusdiansyah, 2018) *prototype* adalah sebuah metode pengembangan *software* yang banyak digunakan pengembang agar dapat saling berinteraksi dengan *user* selama proses pembuatan sistem. Model *prototype* dapat dilihat pada gambar 2.2.



Gambar 2. 2 Model *Prototype*

Berdasarkan model *prototype* pada gambar 2.2 di atas maka dapat diuraikan pembahasan dari masing-masing tahap model tersebut adalah sebagai berikut (Basri & Devitra, 2017) dalam (Rusdiansyah, 2018).

1. *Communication* (Komunikasi)

Merupakan tahapan komunikasi tim pengembang perangkat lunak dengan para *stakeholder* untuk menentukan kebutuhan perangkat lunak yang saat itu diketahui dan untuk menggambarkan area-area untuk iterasi selanjutnya.

2. *Quick Plan* (Perencanaan Secara Cepat)

Merupakan tahapan perencanaan berdasarkan data-data yang telah didapatkan pada tahapan komunikasi, yaitu iterasi pembuatan prototipe dilakukan secara cepat.

3. *Modelling Quick Design* (Model Rancangan Cepat)

Merupakan tahapan memodelkan perencanaan hasil dari tahapan perencanaan secara cepat dengan menggunakan beberapa model berorientasi objek.

4. *Construction of Prototype* (Pembuatan Prototipe)

Merupakan tahapan pembuatan perangkat lunak dengan representasi aspek-aspek yang telah direncanakan pada tahapan-tahapan sebelumnya yang akan terlihat oleh para *end user*.

5. *Deployment Delivery & Feedback* (Penyerahan dan Umpan Balik) Merupakan tahapan penyerahan perangkat lunak kepada para stakeholder untuk dievaluasi dan memberikan umpan balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan.




#### 2.4. *Unified Modeling Language (UML)*


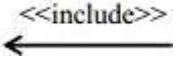
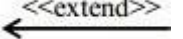
Menurut (M Teguh Prihandoyo, 2018) *Unified Modeling Language (UML)* merupakan salah satu metode pemodelan yang digunakan dalam perancangan dan pembuatan sebuah perangkat lunak (*software*) yang menerapkan konsep pemrograman berorientasi objek (OO). *Unified Modeling Language (UML)* juga merupakan sebuah standar penulisan atau semacam *blueprint* yang di dalamnya terdapat sebuah bisnis proses, dan penulisan kelas-kelas yang spesifik. Terdapat beberapa diagram *Unified Modeling Language (UML)* yang umum digunakan dalam melakukan pengembangan sebuah sistem adalah sebagai berikut (M Teguh Prihandoyo, 2018).

##### 1. *Use Case Diagram*

Merupakan gambaran umum dari fungsionalitas yang diharapkan dari sebuah sistem, dan dapat merepresentasikan interaksi antara aktor dengan sistem. Pada *use case diagram* terdapat aktor yang merupakan gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan. Pada *use case diagram* terdapat simbol-simbol yang merepresentasikan entitas-entitas. Simbol-simbol *use case diagram* dapat dilihat pada tabel 2.1.

Tabel 2. 1 Simbol *Use Case Diagram*







| Simbol  | Keterangan  |
|---|---|
|  | Aktor: Merepresentasikan manusia untuk berkomunikasi dengan <i>use case</i> |
|  | <i>Use Case</i> : Interaksi antara sistem dengan aktor.                     |
|  | <i>Association</i> : Penghubung antara aktor dengan <i>use case</i> .       |

|   |  |
|---|--|
|  | Generalisasi: Menunjukkan spesialisasi aktor untuk berpartisipasi dengan <i>use case</i> .                           |
|  | Menunjukkan suatu <i>use case</i> merupakan fungsionalitas dari <i>use case</i> lainnya.                             |
|  | Menunjukkan suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika kondisi terpenuhi. |

## 2. Activity Diagram

Merupakan gambaran alir dari aktivitas-aktivitas di dalam sistem yang berjalan. Pada *activity diagram* terdapat simbol-simbol yang merepresentasikan alur dari setiap aktivitas yang dapat dilakukan pada sistem. Simbol-simbol *activity diagram* dapat dilihat pada tabel 2.2.








Tabel 2. 2 Simbol *Activity Diagram*

| Simbol  | Nama            | Keterangan  |
|---|-----------------|---|
|  | <i>Start</i>    | Awal dari sebuah aktivitas.                               |
|  | <i>Activity</i> | Aktivitas yang dilakukan oleh sistem.                     |
|  | <i>Decision</i> | Percabangan untuk pilihan aktivitas yang lebih dari satu. |
|  | <i>Join</i>     | Penggabungan aktivitas menjadi satu.                      |
|  | <i>End</i>      | Akhir dari sebuah aktivitas.                              |
|  | Swimlane        | Pemisah antara aktivitas yang terjadi                     |

### 3. *Class Diagram*

Merupakan gambaran dari struktur dan deskripsi data dari sebuah *class*, *package*, dan objek yang saling berhubungan, diantaranya pewarisan, asosiasi, dan lain sebagainya. Pada *class diagram* terdapat simbol-simbol yang menggambarkan hubungan-hubungan antar *class*. Simbol-simbol pada *class diagram* dapat dilihat pada tabel 2.3.

Tabel 2. 3 Simbol *Class Diagram*

| Simbol  | Nama                    | Keterangan   |
|---|-------------------------|--|
|    | <i>Generalization</i>   | Hubungan objek anak ( <i>descendent</i> ) dan struktur data dari objek yang ada di atasnya objek ( <i>ancestor</i> ) |
|    | <i>Nary Association</i> | Untuk menghindari asosiasi dengan lebih dari 2 objek.  |
|  | <i>Class</i>            | Himpunan dari objek-objek yang berbagi atribut   |
|  | <i>Collaboration</i>    | Urutan aksi-aksi yang ditampilkan sistem dan menghasilkan hasil yang terukur.  |
|  | <i>Realization</i>      | Operasi yang dilakukan oleh objek.   |
|  | <i>Dependency</i>       | Hubungan perubahan pada elemen mandiri ( <i>independent</i> ) yang bergantung pada elemen yang tidak mandiri.        |
|  | <i>Association</i>      | Penghubung antara objek satu dengan objek lainnya.   |

#### 2.5. Pengujian *Blackbox*

Menurut (Vikasari, 2018) dalam (Nurudin et al., 2019) salah satu metode pengujian yang berfokus pada spesifikasi fungsionalitas dari perangkat lunak disebut *Black Box Testing* (Pengujian Kotak Hitam). Pengujian kotak hitam

memberikan gambaran atas sekumpulan kondisi masukan dan melakukan pengujian pada fungsional program. *Black Box Testing* digunakan untuk mendeteksi permasalahan sebagai berikut.

1. Fungsi yang salah atau hilang.
2. Kesalahan pada *interface* (antarmuka).
3. Kesalahan struktur data dan basis data.
4. Kesalahan fungsi.
5. Kesalahan deklarasi dan terminasi.

## 2.6. Penelitian Terdahulu

Penelitian mengenai perangkat lunak perlombaan berbasis *website* telah banyak dilakukan oleh peneliti-peneliti sebelumnya. Berikut peneliti sajikan beberapa referensi dari penelitian terdahulu yang telah dilakukan oleh peneliti sebelumnya dapat dilihat pada tabel 2.4.

Tabel 2. 4 Penelitian Terdahulu

| No | Judul, Peneliti, Tahun   | Metode           | Hasil  |
|----|--|------------------|--|
| 1  | Pengembangan Sistem Manajemen Pendaftaran Lomba Tingkat Pendidikan Menengah dan Tinggi (Guci et al., 2020) | <i>Waterfall</i> | Hasil kode program dari struktur algoritme yang ada dalam tahap perancangan, data yang berisi tentang Data Definition Language (DDL) sebagai proses eksekusi pembuatan basis data, dan antarmuka yang merupakan hasil dari tampilan yang ada di dalam sistem. Hasil dari tahap pengujian yang dilakukan untuk pengembangan sistem terbagi ke dalam tiga tahap pengujian. Tahap pertama |



|   |  |                  |  |
|---|--|------------------|--|
|   |  |                  | <p>merupakan pengujian unit yang menguji komponen method pada sebuah class dengan tidak mempedulikan faktor eksternalnya, kedua merupakan pengujian integrasi yang melakukan pengujian terhadap beberapa method berhubungan untuk dapat menjalankan sebuah fungsi, dan yang ketiga merupakan pengujian validasi untuk menguji seluruh kebutuhan fungsional sistem yang telah dilakukan spesifikasi kebutuhan. Dari pengujian yang dilakukan didapati hasil bahwa semua yang diharapkan telah 100% valid terhadap hasil pengujiannya.</p> |
| 2 | <p>Perancangan Sistem Informasi Perlombaan Berbasis <i>Website</i> untuk Kemudahan Penyampaian Informasi dan Pendaftaran Lomba (Fajar Maulana, 2022)</p> | <i>Waterfall</i> | <p>Sistem informasi perlombaan berbasis <i>website</i> ini dapat dengan mudah menginformasikan perlombaan yang akan diselenggarakan, yang mana berbagai macam perlombaan tersebut akan ditampilkan pada <i>user</i> (peserta). Sistem informasi perlombaan</p>   |

|   |  |               |  |
|---|--|---------------|--|
|   |  |               | berbasis <i>website</i> ini telah disediakan <i>form</i> pendaftaran peserta pada setiap halaman detail perlombaan, sehingga masyarakat dapat melakukan proses registrasi peserta secara <i>online</i> .   |
| 3 | Sistem Pendukung Keputusan Untuk Menentukan Peserta Terbaik Dalam Perlombaan Penulisan Kaligrafi Dengan Metode Topsis (Sena & Suparmadi, 2020) | <i>Topsis</i> | Hasil dari penelitian ini adalah sebuah sistem pendukung keputusan untuk menentukan peserta terbaik dalam perlombaan penulisan kaligrafi yang dikembangkan dengan menggunakan bahasa pemrograman PHP dan MySQL. Sistem yang telah dikembangkan dapat membantu peserta dan juga admin dalam keberlangsungan perlombaan. |
| 4 | Sistem Pendukung Keputusan Menentukan Lovebird Unggul dalam Perlombaan Menggunakan Metode AHP-Topsis (Amalia et al., 2019)                     | AHP Topsis    | Hasil dari penelitian ini adalah sebuah sistem pendukung keputusan dengan menerapkan metode AHP-Topsis untuk menentukan lovebird unggul dalam perlombaan sehingga para peserta dapat melihat transparansi nilai pada saat  |

|   |  |   |   |
|---|--|---|---|
|   |  |   | perlombaan. Hasil pengujian menunjukkan bahwa 5 data uji rill (98%) menunjukkan hasil keputusan seimbang antara keputusan sistem dengan keputusan manual.   |
| 5 | Sistem Pendukung Keputusan Menentukan Duta Sekolah untuk Lomba Kompetensi Siswa Menggunakan Metode <i>Elimination Et Choix Traduisant Ia Realite (Electre)</i> (Andika et al., 2019) | <i>Elimination Et Choix Traduisant Ia Realite (Electre)</i> | Hasil dari penelitian ini adalah sebuah sistem pengambilan keputusan pemilihan Siswa untuk Berpartisipasi dalam Lomba Kompetensi Siswa Bidang Desain Grafis pada Yayasan Perguruan Asy-Syafi'iyah Internasional Medan menggunakan Metode <i>Elimination Et Choix Traduisant la Realite (ELECTRE)</i> berhasil memberikan rekomendasi siswa peserta LKS kepada pihak YPI. Asy-Syafi'iyah Internasional Medan berdasarkan nilai dari masing-masing kriteria dan nilai bobot. Sistem pemilihan Siswa untuk Berpartisipasi dalam Lomba Kompetensi Siswa Bidang Desain Grafis menggunakan Metode <i>Elimination Et Choix</i> |

|   |  |  |  |
|---|--|--|--|
|   |  |  | <i>Traduisant la Realite (ELECTRE)</i> yang didapat dari hasil perhitungan nilai akhir adalah sesuai dengan perhitungan manual.  |
| 6 | Implementasi Sistem Pendukung Keputusan Seleksi Lomba Posyandu Menggunakan Metode <i>Simple Additive Weighting</i> (SAW) (Wijanegara & Saryanti, 2018) | <i>Simple Additive Weighting</i> (SAW) | Hasil dari penelitian ini adalah sebuah sistem dengan menggunakan metode SAW dapat melakukan perhitungan dengan beberapa kriteria dan sistem pendukung keputusan yang dibangun dapat mempercepat hasil seleksi perlombaan posyandu yang diselenggarakan. |