

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan empat tinjauan Pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu :

1. Penelitian yang dilakukan oleh (Sari, et al., 2018) yang berjudul “Pengembangan Aplikasi Mobile *E-Voting* Pilkades” penelitian tersebut mengulas tentang bagaimana mengembangkan sebuah aplikasi berbasis mobile yang akan digunakan untuk pemilihan kepala desa, aplikasi yang dikembangkan terdiri dari dua bagian yaitu *front-end* dan *back-end* dimana untuk *front-end* berbasis android sedangkan *back-end* berbasis web, penelitian tersebut menggunakan waterfall sebagai metode pengembangan aplikasi. Penelitian tersebut bertujuan untuk menggantikan sistem pemilihan secara konvensional yang memiliki kelemahan yaitu berupa, kecepatan waktu perhitungan, biaya percetakan kertas suara dan rentan terhadap penggelembungan perhitungan suara.
2. Penelitian yang dilakukan oleh (Lavarion & Yustanti, 2016) yang berjudul “Rancang Bangun E-Voting Berbasis Website Di Universitas Negeri Surabaya” Penelitian tersebut dilakukan bertujuan kurangnya peminat dalam proses pemilihan suara yang mengakibatkan suara yang terkumpul sangat sedikit karena pemungutan suara yang berlangsung di Universitas Negeri Surabaya masih menggunakan kertas, pencoblosan, dan pencelupan jari. Dalam rancang bangun e-voting ini dibangun dengan menggunakan

Bahasa pemrograman PHP dan basis data MySQL.

3. Penelitian yang dilakukan (Nasution, et al., 2020) yang berjudul “Rancang Bangun Aplikasi Pemilihan Pegawai Berprestasi Dengan Metode Profile Matching Berbasis Android Pada BPJS Ketenagakerjaan” dalam penelitian tersebut di jelaskan bahwa dalam pemilihan pegawai berprestasi tidak melibatkan dalam subjektif saja, melainkan ada beberapa pihak yang di libatkan. Hal tersebut menyebabkan ketidakefisienan waktu pengumpulan kuisisioner dan peneilaian kuesioner yang jumlahnya banyak. Sehingga di butuhkan aplikasi untuk melakukan pengambilan keputusan. Di dalam penelitan tersebut menggunakan metode *profile matching* yang di terapkan dalam sistem pengambil keputusan aplikasi dan metode pengembangan sistem fishbone dengan bahasa pemrograman java dalam pengembangan aplikasinya.
4. Penelitian yang dilakukan (Gelu, et al., 2022) yang berjudul “E-Voting Pemilihan Ketua BEM Universitas Timor Berbasis *Mobile* Menggunakan Metode *Waterfall*” dalam penelitian tersebut menjelaskan sulitnya pemungutan suara secara konvensional yang menggunakan kertas karena kurangnya minat mahasiswa Universitas Timor dalam mengikuti proses pemungutan suara sehingga di butuh kan aplikasi yang dapat di akses dimana saja untuk melakukan pemungutan suara, di dalam penelitian tersebut pengembangan aplikasi menggunakan metode waterfall, serta bahasa pemrograman kotlin untuk membuat program aplikasi berbasis mobile.

2.2 Pengertian Aplikasi

Menurut (Yuhefizar, 2008) Aplikasi adalah program yang sengaja dibuat dan di rancang untuk dapat membantu kebutuhan penggunanya dalam melakukan suatu aktifitas tertentu sedangkan menurut (Jogiyanto, 1999) Aplikasi merupakan suatu kelompok file yang saling berhubungan untuk melakukan kegiatan tertentu. Sehingga dapat di simpulkan bahwa aplikasi adalah kumpulan file yang saling berhubungan sehingga menjadi sebuah program yang di buat untuk membantu pengguna mengerjakan suatu kegiatan tertentu.

2.3 E-Voting

Menurut (Fahmi, et al., 2010) E-Voting (Electronic Voting) adalah proses pemilihan umum yang memungkinkan pemilih untuk mencatatkan pilihan secara elektronik, rahasia dan teramankan. mulai dari proses pendaftaran pemilih, pelaksanaan pemilihan, perhitungan suara, dan pengiriman hasil suara. E-Voting sendiri seharusnya dapat meningkatkan fleksibilitas dalam aktifitas pemilihan dimana pemilih tidak harus datang ke suatu lokasi melainkan menggunakan suatu program aplikasi yang dibuat secara khusus mengikuti peraturan pemilihan yang sudah di tetapkan.

2.4 Profile Matching

Profile Matching adalah sebuah metode sistem pengambil keputusan dengan mencocokkan profil dengan mengasumsikan bahwa terdapat tingkat variable prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukan dengan tingkat minimal atau maksimal yang harus dipenuhi atau dilewati (Kusrini, 2007). Terdapat beberapa tahapan dalam metode *Profile Matching* adalah sebagai berikut:

1. Menentukan bobot nilai Gap. Pada tahap ini ditentukan bobot nilai masing-masing aspek dengan menggunakan bobot nilai yang telah ditentukan bagi masing-masing aspek itu sendiri. Adapun inputan dari proses pembobotan ini adalah selisih dari setiap profil.
2. Langkah kedua dengan melakukan pemetaan Gap. Gap yang dimaksud adalah perbedaan antara profil siswa dengan profil tingkat keaktifan siswa.
3. Melakukan pencocokan dengan table bobot gap. Hasil Gap dari pengurangan profil siswa dan profil tingkat keaktifan mahasiswa bila dicocokkan dengan kolom selisih gap pada tabel bobot nilai yang dihasilkan sama. Tabel bobot nilai bisa dilihat pada Tabel 1.

Tabel 1. Bobot Nilai Gap

Gap No.	Selisih Gap	Bobot Nilai	Keterangan
1	0	5	Kompetensi sesuai dengan yang dibutuhkan
2	1	4.5	Kompetensi individu lebih 1 tingkat/level
3	-1	4	Kompetensi individu kurang 1 tingkat/level

Gap No.	Selisih Gap	Bobot Nilai	Keterangan
4	2	3.5	Kompetensi individu lebih 2 tingkat/level
5	-2	3	Kompetensi individu kurang 2 tingkat/level
6	3	2.5	Kompetensi individu lebih 3 tingkat/level
7	-3	2	Kompetensi individu kurang 3 tingkat/level
8	4	1.5	Kompetensi individu lebih 4 tingkat/level
9	-4	1	Kompetensi individu lebih 4 tingkat/level

4. Melakukan perhitungan *core factor* dan *secondary factor*. Setelah menentukan bobot nilai gap untuk kedua aspek yang dibutuhkan, kemudian tiap aspek dikelompokkan lagi menjadi dua kelompok yaitu *core factor* dan *secondary factor*.

$$NRC = \frac{\sum NC}{\sum IC} \dots\dots\dots(1)$$

Keterangan :

NRC = Nilai rata-rata core factor tiap aspek

$\sum NC$ = Jumlah total nilai core factor tiap aspek

$\sum IC$ = Jumlah item tiap aspek

$$NRS = \frac{\sum NS}{\sum IS} \dots\dots\dots(2)$$

Keterangan : NRS = Nilai rata-rata secondary factor tiap aspek

$\sum NS$ = Jumlah total nilai secondary factor tiap aspek

$\sum IS$ = Jumlah item tiap aspek

5. Perhitungan nilai total dengan rumus sebagai berikut :

Nilai total (N) merupakan (X)% nilai prosentase yang diinputkan dari NCI yaitu nilai rata-rata core factor ditambahkan dengan nilai-nilai prosentase yang diinputkan dari NSI yaitu rata-rata secondary factor. Perhitungan nilai total terlebih dahulu menentukan nilai persen yang dimasukkan yaitu core factor 60% dan secondary factor 40%. Kemudian nilai core factor dan secondary factor ini dijumlahkan sesuai dengan persamaan 1.

6. Perhitungan penentuan ranking dengan rumus sebagai berikut :

$$\text{Rangking} = \text{Ranking} = \%NCF + \%NSF \dots\dots\dots(4)$$

NCF mewakili nilai akhir aspek akademik dan NSF mewakili nilai akhir dari aspek non akademik.

2.5 PHP

PHP berasal dari kata *Hypertext Preprocessor* merupakan bahasa *scripting* yang bersifat *server-side rendering*, sifat tersebut dimaksudkan bahwa bahasa pemrograman PHP dilakukan di server, baru kemudian hasilnya dikirimkan ke browser (Kustiyaningsih, 2011). PHP banyak di implementasikan untuk pembuatan website dinamis dimana kode PHP di selipkan di antara *script* kode-kode html.

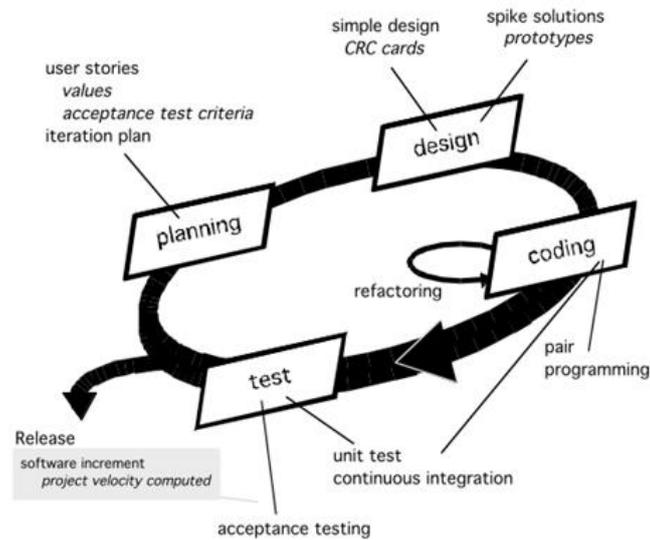
2.6 MySQL

MySQL adalah sebuah *Relational Database Management System* yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah yang sangat besar dan dapat di akses oleh banyak orang. MySQL juga salah satu software untuk database server yang banyak digunakan dan bersifat *open source* dan menggunakan SQL (Kustiyaningsih, 2011).

2.7 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah *Extreme Programming* model proses ini diciptakan dan dikembangkan oleh Kent Beck. *Extreme Programming (XP)* adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel. Walaupun menggunakan kata programming, XP tidak hanya berfokus pada coding tetapi seluruh area pengembangan perangkat lunak (Pressman, 2000).

Menurut Kent Beck, XP ringan, efisien, resiko rendah, mudah disesuaikan, dapat diprediksi dan mudah dikembangkan. Suatu model yang menekankan pada keterlibatan user secara langsung, pengujian dan *pay-as-you-go design*. Siklus *Extreme Programming* di gambar 2.1 berikut :



Gambar 2.1 Siklus Extreme Programming

1. *Planning*, pengumpulan user stories dari klien yang klien tetapkan prioritasnya. Setiap story ditetapkan harga dan lama pengembangan, jika terlalu besar, story dapat di pecah menjadi beberapa story yang lebih kecil. Periksa dan pertimbangkan resiko.
2. *Design*, menggunakan *Unified Modelling Language* (UML) serta memanfaatkan kartu CRC (*Class Responsibility-Collabolator*) untuk identifikasi dan mengatur class-class pada konsep *Object Oriented*. Jika menemui kesulitan, *prototype* dibangun (ini dinamakan *spike solution*). Lakukan refactoring, yaitu mengembangkan desain setelah pengkodean.

3. *Coding*, menyiapkan *unit test* sebelum pengkodean dipakai sebagai focus pemrograman untuk membuat program. *Pair programming* dilakukan untuk *real time problem solving* dan *realtime quality assurance*.
4. *Testing*, menggunakan *unit test* yang di persiapkan sebelum pengkodean

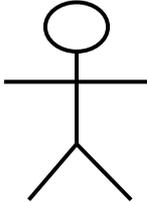
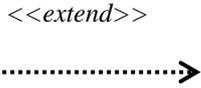
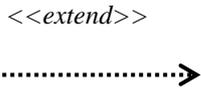
2.8 Unified Modelling Language (UML)

Menurut (Rosa & Salahuddin, 2013), UML (*Unified Modelling Language*) adalah Bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*):

1. Usecase Diagram

Menurut (Rosa & Salahuddin, 2013) *Use Case* adalah *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2 Simbol Use Case Diagram

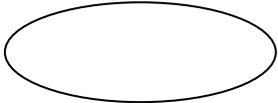
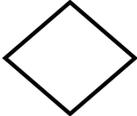
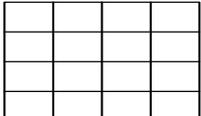
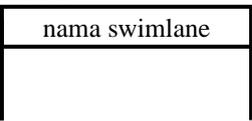
Simbol	Deskripsi
<p><i>Use Case</i></p> 	Menjelaskan tentang fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> pada sistem
<p>Aktor/<i>actor</i></p> 	Menjelaskan tentang orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor pada sistem.
<p>Asosiasi/<i>asociation</i></p> 	Menjelaskan tentang komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor pada sistem.
<p>Ekstensi/<i>exten</i></p> 	Menjelaskan tentang relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan pada sistem.
<p>Generalisasi/<i>generalization</i></p> 	Menjelaskan tentang hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p>Menggunakan/<i>Include/uses</i></p> 	Menjelaskan tentang relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini pada sistem.

2. Activity Diagram

Menurut (Rosa & Salahuddin, 2013) *activity* diagram adalah Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang

dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.3 .

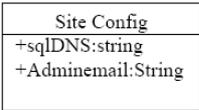
Tabel 2.3 Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Menjelaskan tentang status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal pada sistem.
Aktivitas 	Menjelaskan tentang aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja pada sistem.
Percabangan/ <i>decision</i> 	Menjelaskan tentang asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu pada sistem.
Penggabungan/ <i>join</i> 	Menjelaskan tentang asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu pada sistem.
Tabel 	Menjelaskan tentang suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis pada sistem.
Dokumen 	Menjelaskan tentang menunjukkan dokumen sumber atau laporan pada sistem.
Status akhir 	Menjelaskan tentang status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir pada sistem.
<i>Swimlane</i> 	Menjelaskan tentang memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi pada sistem.

3. Class Diagram

Menurut (Rosa & Salahuddin, 2013) *Class Diagram* adalah mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.4 berikut ini :

Tabel 2.4 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Menjelaskan tentang kelas pada struktur sistem yang akan dibuat.
Asosiasi/asociation 	Menjelaskan tentang relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> pada sistem.
Asosiasi berarah/ <i>directed association</i> 	Menjelaskan tentang relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> pada sistem.
Generalisasi 	Menjelaskan tentang relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) pada sistem.
Kebergantungan/ <i>dependecy</i> 	Menjelaskan tentang relasi antar kelas dengan makna kebergantungan antar kelas pada sistem.
Agregasi/agregation 	Menjelaskan tentang relasi antar kelas dengan makna semua bagian (<i>whole-part</i>) pada sistem.

2.9 Black Box Testing

Menurut (Pressman, 2010) *Black Box Testing* yang sering di sebut juga pengujian perilaku yang berfokus pada persyaratan fungsional dari sebuah perangkat lunak. Artinya pengujian *Black Box Testing* memungkinkan anda untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional program dan menguji setiap kondisi tersebut terpenuhi atau tidak, sehingga akan di dapatkan hasil yang menjadi acuan bahwa program yang diujikan memenuhi syarat atau tidak.

Pengujian *Black Box* berupaya untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang salah atau hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses basis data eksternal.
4. Kesalahan perilaku atau kinerja program.
5. Kesalahan inisiasi atau penghentian.

2.10 WebQual 4.0

WebQual 4.0 merupakan salah satu metode atau teknik pengukuran website berdasarkan persepsi *end user* atau pengguna akhir. *WebQual* dikembangkan sejak tahun 1998 dan telah mengalami interaksi dalam penyusunan dimensi dan butir pertanyaan, hingga versi terakhir yaitu *WebQual* 4.0 (Nada & Wibowo, 2015).

WebQual adalah pengukuran berdasarkan pada kualitas fungsi penyebaran (quality function deployment) atau biasa disebut dengan QFD, yang merupakan proses terstruktur dan disiplin yang menyediakan sarana untuk mengidentifikasi 12 dan

membawa aspirasi pelanggan melalui setiap tahap pengembangan produk dan atau jasa serta implementasinya. Aplikasi QFD berawal dari menangkap "suara pelanggan", dengan kata lain menemukan kebutuhan kualitas sesuai keinginan pengguna. Kualitas ini kemudian diberikan kembali ke pelanggan dan membentuk dasar dari evaluasi kualitas produk atau layanan. Pengguna juga diminta untuk menilai masing-masing kualitas, yang dapat membantu pemahaman tentang kualitas apa yang dianggap oleh pengguna paling penting dalam situasi tertentu.