

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian yang berkaitan dengan aplikasi ini diteliti oleh Azis, dkk, (2019). Penelitian ini menyatakan pemilihan secara manual memiliki permasalahan antara lain tingginya risiko dalam kesalahan perhitungan suara mengingat banyaknya jumlah surat suara yang harus dihitung dan rawan terjadinya kecurangan terhadap jumlah suara untuk kepentingan partai tertentu, dalam hal itu perlunya sistem pemilihan kepala desa yang dapat meningkatkan keamanan dalam pemilihan, dalam penelitian ini membangun beberapa fitur antara lain fitur *input* data pemilih tetap, data calon kepala desa, pencetakan kartu pemilih, dan rekapitulasi hasil pemungutan suara.

Penelitian lainnya yang berkaitan dengan penelitian ini diteliti oleh Gunawan (2017). Penelitian ini mengatakan dalam menangani permasalahan pada proses pencatatan data kehadiran konvensional telah dilakukan, salah satu cara untuk menangani menggunakan *ID Card* tetapi masih terdapat kekurangan seperti pengguna bisa melakukan pencatatan kehadiran tanpa harus hadir di tempat, karena *ID Card* masih bisa ditiptkan ke orang lain, data kehadiran masih diragukan, maka dari itu dengan penerapan *fingerprint* yang telah dilakukan di lokasi mitra pengabdian dapat menyelesaikan masalah “menitipkan absensi” yang masih dijumpai pada sistem pencatatan kehadiran konvensional.

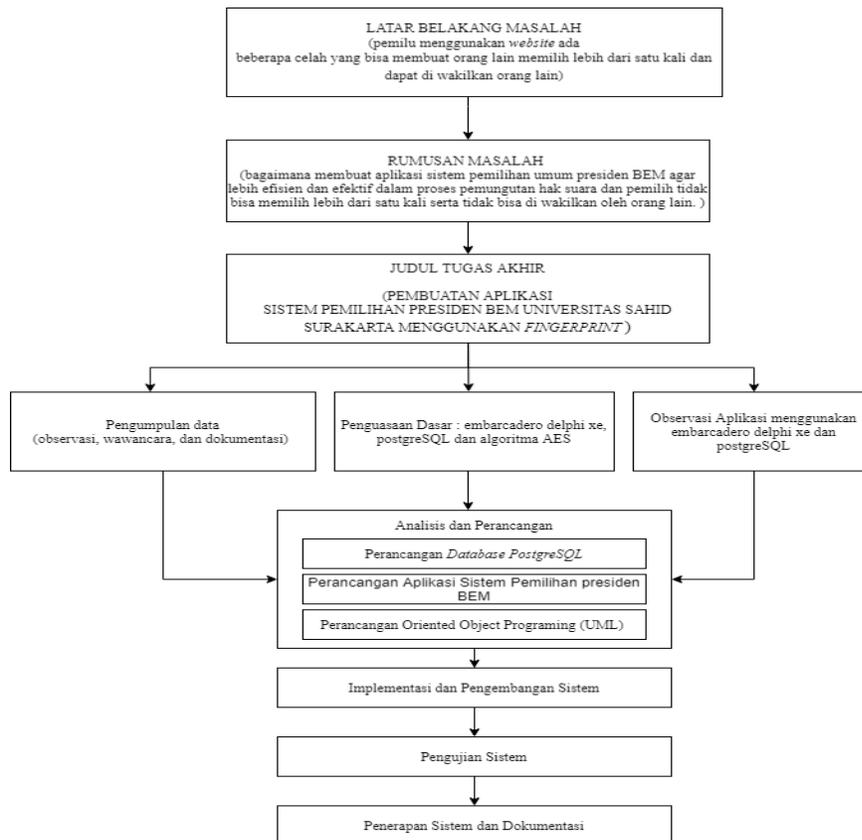
Penelitian yang berkaitan dengan penelitian ini diteliti oleh Alvian (2020). Penelitian ini mengatakan untuk pemungutan suara secara manual banyak membuang waktu yang begitu lama yaitu 15 menit. Pemungutan suara secara manual juga memiliki beberapa kendala, seperti penyimpanan dan pencarian data yang manual sehingga saat melakukan proses audit tidak secara instan, namun

masih menggunakan cara yang manual sehingga saat melakukan audit banyak data rekam yang hilang.

Penelitian lainnya yang berkaitan dengan penelitian ini diteliti oleh Isyanto, dkk ,(2019). Penelitian ini mengatakan *fingerprint* salah satu sistem keamanan yang dilengkapi dengan sistem pengenalan terhadap pemilik hak akses, agar benar-benar memiliki proteksi yang kuat. Sistem pengenalan tersebut menggunakan kunci pintu *fingerprint* yang hanya dapat diakses oleh pemilik hak akses hal ini dimaksudkan agar ruang tersebut hanya bisa dimasuki oleh orang yang berwenang saja.

Berdasarkan ketiga penelitian di atas, maka peneliti berinisiatif membuat aplikasi sistem pemilihan presiden BEM Universitas Sahid Surakarta Menggunakan *fingerprint*.

2.2. Kerangka pemikiran



Gambar 2.1. Diagram Kerangka Pemikiran

Keterangan Diagram Kerangka Pemikiran (Gambar 2.1):

1. Latar Belakang Masalah

Identifikasi adanya masalah yang menjadi sebuah permasalahan pada KPUM BEM Universitas Sahid Surakarta. Permasalahan yang ada pada KUM antara lain pemilih bisa memilih lebih dari satu kali yang menyebabkan data ganda, pemilihan bisa diwakilkan orang lain dan keamanan yang lemah, dari kendala tersebut dari pihak pemilu belum mendapat solusi untuk menyelesaikan permasalahan yang ada.

2. Rumusan Masalah

membuat aplikasi sistem pemilihan umum presiden BEM Universitas Sahid Surakarta menggunakan *fingerprint* agar lebih efisien dan efektif.

3. Judul Tugas Akhir

Judul yang sekiranya tepat untuk menangani permasalahan yang ada dan diambil dari pemilihan presiden BEM Universitas Sahid Surakarta, maka judul yang diambil adalah pembuatan aplikasi Sistem Pemilihan Presiden BEM Universitas Sahid Surakarta menggunakan *fingerprint*.

4. Pengumpulan Data

Semua data yang dibutuhkan dikumpulkan, baik melalui, observasi, wawancara dan dokumentasi. Pada observasi dilakukan di BEM Universitas Sahid Surakarta, untuk orang yang diwawancara antara lain Mayrica selaku panitia KPUM 2020/2021, bapak Fajar Alam Putra selaku kemahasiswaan Universitas Sahid Surakarta, Nadea Adisty Romadhona, Desti dan bela selaku mahasiswa Universitas Sahid Surakarta, sedangkan dokumentasi berupa laporan tugas akhir.

5. Penguasaan Dasar

Mempelajari Bahasa pemrograman Embarcadero Delphi XE, mempelajari *database* PostgreSQL, dan mempelajari algoritma AES, agar lebih mendalami sistem yang akan dibuat agar hasilnya lebih maksimal.

6. Observasi Aplikasi

Aplikasi yang sedang berjalan atau aplikasi serupa dilakukan pengamatan, observasi aplikasi yang sudah diamati seperti *website* KPUM Universitas Sahid Surakarta, aplikasi e-votting, dan aplikasi absensi menggunakan *fingerprint*. Observasi aplikasi baik dari karya ilmiah atau internet sehingga dijadikan referensi dalam membangun aplikasi.

7. Analisis dan Perancangan Sistem

Sistem yang akan dibangun dianalisis dan dirancang seperti bagaimana desainya dan apa saja fitur yang diperlukan sehingga dapat membantu memecahkan permasalahan yang ada. Perancangan sistem antara lain perancangan database PostgreSQL, perancangan aplikasi sistem pemilihan presiden BEM, dan perancangan oriented object programming (UML).

8. Implementasi Aplikasi

Membangun *database* dan aplikasi dari data-data yang sebelumnya sudah dirancang dengan kebutuhan yang diperlukan. Dalam membangun *database* menggunakan PostgreSQL sedangkan untuk membangun aplikasi menggunakan Embarcadero Delphi XE.

9. Pengujian Aplikasi

Menguji aplikasi yang telah dibangun bertujuan untuk mengetahui ada atau tidaknya kesalahan didalam aplikasi yang telah dibuat, pengujian aplikasi yang digunakan yaitu *Five View*.

10. Penerapan Sistem dan Dokumentasi

Sistem telah siap untuk digunakan dan sudah melewati tahap pengujian dan melakukan dokumentasi dari keseluruhan kegiatan penyusunan tugas akhir.

2.3. Teori-Teori Pendukung

2.3.1. Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer

dapat memproses *input* menjadi *output*. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melakukan tugas tertentu. Aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data (Muthohari, dkk, 2016).

2.3.2. E-Votting

E-Votting adalah suatu sistem pemilihan dimana data dicatat, disimpan, dan diproses dalam bentuk *informasi* digital. *E-votting* pada hakikatnya adalah pelaksanaan pemungutan suara yang dilakukan secara *elektronik* (digital) mulai dari proses pendaftaran pemilih, pelaksanaan pemilihan, penghitungan suara, dan pengiriman hasil suara (Priyono dan Dihan 2010).

2.3.3. Basis Data

Basis data adalah kumpulan *field-field* yang mempunyai kaitan antara satu *field* dengan *field* yang lain sehingga membentuk bangunan data untuk menginformasikan kondisi lalu lintas dalam bahasa tertentu. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup *informasi*. Dalam satu *file* terdapat *record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. Suatu sistem manajemen basis data berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Sistem manajemen basis data dan set program pengelola untuk menambah data, menghapus data, mengambil data dan membaca data (Bastanur, 2014).

2.3.4. Kriptografi

Kriptografi berasal dari dua kata Yunani, yaitu *kryptos* yang berarti rahasia dan *graphein* yang berarti menulis. Secara umum kriptografi bisa diartikan sebagai ilmu dan seni penyamaan yang bertujuan sebagai mengawasi keamanan dan kerahasiaan suatu data. Kriptografi mendukung keperluan dari 2 aspek keamanan *informasi*, yaitu *secrecy* (perlindungan tentang kerahasiaan data *informasi*) dan

authenticity (perlindungan tentang pemalsuan dan perubahan informasi yang tidak diharapkan). Secara singkat kriptografi adalah ilmu yang mempelajari teknis dimana berhubungan dengan aspek keamanan informasi, Kriptografi juga merupakan seni untuk menjaga keamanan sebuah pesan (*Cryptography is the art and science of keeping messages secure*) (Azizah, dkk, 2013).

2.3.4.1 Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) merupakan algoritma *cryptographic* yang dapat digunakan untuk mengamankan data. AES dipublikasikan oleh NIST (*National Institute of Standard and Technology*) pada tahun 2001 yang di pergunakan untuk menggantikan algoritma DES yang sudah dianggap kuno dan mudah dibobol oleh orang lain semenjak itu AES sering digunakan untuk mengamankan data. Algoritma AES adalah blok *chiphertext* simetrik yang dapat mengenkripsi (*encipher*) dan deskripsi (*decipher*) informasi. *Input* dan *output* dari algoritma AES terdiri dari urutan data yang sebesar 128 bit. Urutan data dalam satu kelompok adalah 128 bit yang bisa disebut juga sebagai blok data atau sebagai *plaintext* yang nantinya akan dienkripsi menjadi *ciphertext*. Enkripsi mengganti data yang tidak dapat lagi dibaca disebut *ciphertext*, sebaliknya deskripsi adalah merubah *ciphertext* data menjadi bentuk semula yang kita kenal sebagai *plaintext*. Algoritma AES menggunakan kunci kriptografi 128, 192, dan 256 bit untuk mengenkripsi dan deskripsi (Ibrahim, 2017).

2.3.5. Fingerprint

Pengenalan sidik jari merupakan teknologi yang amat sering dan umum digunakan oleh khalayak ramai dalam identifikasi identitas seseorang, bahkan telah menjadi teknologi yang cukup diandalkan karena efektivitas dan penggunaannya yang mudah. *Fingerprint* adalah alat yang digunakan untuk melakukan pemindaian sidik jari manusia. Perkembangan teknologi yang semakin canggih penerapan *fingerprint* untuk mengidentifikasi manusia telah diterapkan pada mesin absensi.

Teknik pembacaan sidik jari oleh mesin ke komputer berarti memindai sidik jari melalui suatu alat yaitu *fingerprint* agar dapat dibaca oleh program dengan cara

menempelkan jari kita ke sensor alat *fingerprint* tersebut. Pembacaan sidik jari pada penelitian ini menggunakan teknik *optic*, dengan teknik ini pola sidik jari direkam atau dipindai dengan menggunakan cahaya. Alat perekam (*fingerprint scanner*) yang digunakan adalah berupa kamera digital, di bawah *scan* area, terdapat lampu atau pemancar cahaya yang menerangi permukaan ujung jari. Hasil pantulan cahaya dari ujung jari tersebut ke dalam memori (Ngantung, dkk. 2014).

2.3.6. Diagram UML

UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek tujuan penggunaan UML adalah :

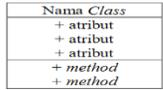
- 1) Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- 2) Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan.
- 3) Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

UML bisa juga berfungsi sebagai sebuah cetak biru (*blue print*) karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui *informasi* secara detail tentang koding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reverse engineering*) (Nugroho, 2010). Kesembilan jenis diagram dalam UML yang dapat digunakan yaitu :

a. Class Diagram

Class diagram bersifat statis yang memperlihatkan himpunan kelas-kelas, antar muka, kolaborasi-kolaborasi, serta relasi-relasi. Simbol-simbol *class* diagram bisa dilihat pada Tabel.2.4.

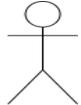
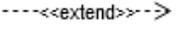
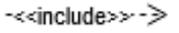
Tabel 2.1. Simbol-simbol *Class Diagram*

No	Simbol	Keterangan
1	<p>Class</p> 	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2	<p>Asosiasi</p> 	Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor
3	<p>Composition</p> 	sebuah class yang tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain.

b. *Use Case*

Diagram *Use Case* merupakan diagram bersifat statis yang memperlihatkan himpunan *use case* dan aktor-aktor. komponen sistem dalam *use case* diagram dapat dilihat pada Tabel 2.1.

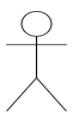
Tabel 2.2. Simbol-simbol *Use Case Diagram*

No	Simbol	Keterangan
1	<p>Actor</p> 	Orang proses, atau sistem lain yang berinteraksi dengan sistem <i>informasi</i> yang akan dibuat di luar sistem <i>informasi</i> yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2	<p>Use Case</p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
3	<p>Asosiasi</p> 	Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>
4	<p>Extend</p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i>
5	<p>Include</p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

c. *Sequence Diagram*

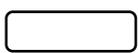
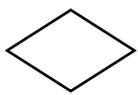
Diagram *Sequence* atau diagram urutan merupakan diagram yang bersifat dinamis. Diagram urutan merupakan diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu. Simbol-simbol *sequence* dapat dilihat pada Tabel 2.2.

Tabel 2.3. Simbol-simbol *Sequence diagram*

No	Simbol	Keterangan
1	Actor 	Orang proses, atau sistem lain yang berinteraksi dengan sistem <i>informasi</i> yang akan dibuat di luar sistem <i>informasi</i> yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2	Life Line 	Menyatakan Kehidupan suatu objek
3	Objek 	Menyatakan objek yang berinteraksi pesan
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5	Pesan 	Menyatakan suatu objek memanggil operasi yang ada pada objek lain atau dirinya sendiri

d. *Activity Diagram*

Tabel 2.4. Simbol-simbol *Activity diagram*

No	Simbol	Keterangan
1	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Activity 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Branch/Merge 	Asosiasi percabangan dimana lebih dari satu aktivitas dijadikan satu.
4	Final 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.

Pada *Activity Diagram* Tabel 2.3 menjelaskan bahwa *Activity* diagram adalah diagram yang bersifat dinamis, diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem.

e. *Component Diagram*

Simbol-simbol *component* diagram bisa dilihat pada Tabel 2.5.

Tabel 2.5. Simbol-simbol *Component Diagram*

No	Simbol	Keterangan
1	<p>Component</p> 	komponen sistem.
2	<p>Link</p> 	Menggambarkan relasi antar node

Component Diagram pada Tabel 2.5 adalah diagram yang bersifat statis yang memperlihatkan organisasi serta kebergantungan pada komponen-komponen yang telah ada sebelumnya.

f. *Deployment Diagram*

Deployment diagram adalah diagram yang bersifat statis yang memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run-time*). Diagram ini memuat simpul-simpul (*node*) beserta komponen-komponen yang ada didalamnya. Simbol-simbol *Deployment Diagram* bisa dilihat pada Tabel 2.6.

Tabel 2.6. Simbol-simbol *Deployment Diagram*

No	Simbol	Keterangan
1	<p>Node</p> 	Mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen.
2	<p>Link</p> 	Menggambarkan relasi antar <i>node</i>

2.3.7. Embarcadero Delphi XE

Delphi adalah salah satu bahasa pemrograman yang berbasis *Microsoft Windows* dan juga didesain untuk dapat memanfaatkan fasilitas *Microsoft Windows* dengan maksimal. Delphi dengan fasilitas *Object Oriented Programming* (OOP) menyediakan sebuah objek yang mudah untuk di gunakan. OOP adalah gabungan dari objek dan Bahasa pemograman. Objek adalah komponen-komponen yang dapat terlihat secara fisik, sedangkan untuk Bahasa pemograman dapat diartikan sebagai sekumpulan simbol atau kode yang digunakan untuk menjalankan komputer berdasarkan yang dimiliki. bahasa pemrograman dibuat untuk mengatasi suatu permasalahan yang harus diselesaikan oleh komputer, meskipun bahasa pemrograman berbeda namun memiliki fungsi dasar yang sama yaitu pengoperasian komputer, untuk pemograman *database*, Embarcadero Delphi XE 8 menyediakan *format database*, misalnya menggunakan *MS-Access*, *SyBase*, *Oracle*, *FoxPro*, *Informix*, *Paradox* dan *dBase* (Faslah, dkk, 2015)

2.3.8. Metode Pengujian *Five View*

Pengujian *Five View* sesuai dengan namanya terdiri dari 5 sudut pandang berbeda, dari sudut pandang yang berbeda bisa dilihat kekurangan dan kelebihan

dari sebuah aplikasi (Fitriani, dkk, 2019). Kelima sudut pandang tersebut antar lain yaitu:

2.3.8.1. *User View* (Pandangan Pengguna)

Sudut pandangan ini kualitas perangkat lunak akan dinilai berdasarkan kepuasan pengguna, apakah aplikasi sesuai dengan kebutuhan yang diperlukan oleh pengguna atau tidak.

2.3.8.2. *Manufacturing View* (Pandangan Manufaktur)

Pandangan ini berkaitan dengan faktor pada industri, apakah aplikasi memenuhi persyaratan atau tidak setiap penyimpangan dari persyaratan yang dinilai memungkinkan kualitas aplikasi konsep proses memakai peran kunci aplikasi yang dibuat dimana aplikasi harus orisinil sehingga biaya berkurang. Kualitas dapat secara bertahap dalam peningkatan perbaikan proses.

Proses pembangunan perangkat lunak memainkan peran kunci, dimana produk diproses sejak awal, sehingga untuk biaya pengembangan dan perawatan dapat diminimalisir. Apabila prosesnya berkualitas maka produknya juga berkualitas.

2.3.8.3. *Transcendental View* (Pandangan Transendental)

Pandangan ini akan menilai kualitas berdasarkan pengalaman para ahli. Kualitas yang dimaksud adalah sesuatu yang dapat dikenali atau diketahui tetapi tidak dapat didefinisikan, kualitas perangkat lunak bersifat subyektif dan tidak dapat dihitung dengan angka.

2.3.8.4. *Value-based View* (Pandangan Nilai)

Pandangan ini merupakan suatu gabungan antara keunggulan dan harga. Perangkat lunak akan diukur kualitasnya berdasarkan pada keunggulan, dan diukur nilainya berdasarkan pada harganya. Kualitas tidak akan dapat diukur jika

produk tidak membuat nilai ekonomis. pandangan *value-based* membuat *trade-off* antara harga dan kualitas.

2.3.8.5. Product View (Pandangan produk)

Sebuah produk diproduksi dengan sifat internal (misalnya bahan dan tindakan) yang baik, maka produk akan memiliki sifat eksternal atau *output* yang baik dan dapat dieksplorasi hubungan antara sifat internal dan kualitas eksternal.