



**IMPLEMENTASI TEST DRIVEN DEVELOPMENT PADA RANCANG
BANGUN APLIKASI E-VOTING PEMILIHAN RAYA UNIVERSITAS
DINAMIKA**

TUGAS AKHIR



Oleh:

SEBASTIANUS SEMBARA

17410100054

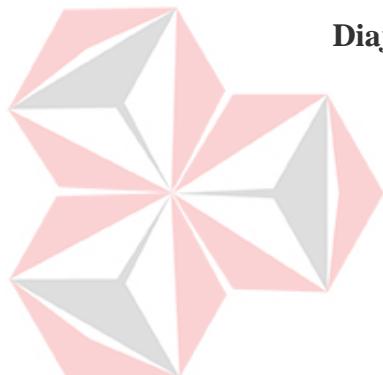
FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2022

**IMPLEMENTASI TEST DRIVEN DEVELOPMENT PADA RANCANG
BANGUN APLIKASI E-VOTING PEMILIHAN RAYA UNIVERSITAS
DINAMIKA**

TUGAS AKHIR



Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer

UNIVERSITAS
Dinamika

Oleh:

Nama : Sebastianus Sembara
NIM : 17410100054
Program Studi : S1 Sistem Informasi

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2022**

Tugas Akhir

**IMPLEMENTASI TEST DRIVEN DEVELOPMENT PADA RANCANG
BANGUN APLIKASI E-VOTING PEMILIHAN RAYA UNIVERSITAS
DINAMIKA**

Dipersiapkan dan disusun oleh
Sebastianus Sembara
NIM : 17410100054

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahasan

Pada: Selasa, 9 Agustus 2022

Susunan Dewan Pembahasan

Pembimbing

- I. Endra Rahmawati, M. Kom.

NIDN: 0712108701

- II. Sri Hariani Eko Wulandari, S. Kom., M.MT.

NIDN: 0726017801

Pembahasan

Dr. Bambang Hariadi, M.Pd.

NIDN: 0719106401


Digitally signed by
Endra Rahmawati
Date: 2022.08.18
13:45:38 +07'00'


Digitally signed
by Sri Hariani
Eko Wulandari
Date:
2022.08.18
13:30:21 +07'00'


Digitally signed by Universitas
Dinamika
DN: c4f2_b1-e6af-4a6a-
b3c9-4a2e9eUniversitas
Dinamika, cn:Universitas
Dinamika,
email: tsh@universitasdinamika.ac.id
Date: 2022.08.18 15:02:36 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana:


Digitally signed by
Universitas Dinamika
Date: 2022.08.19
15:28:24 +07'00'

Tri Sagirani, S. Kom., M.MT.

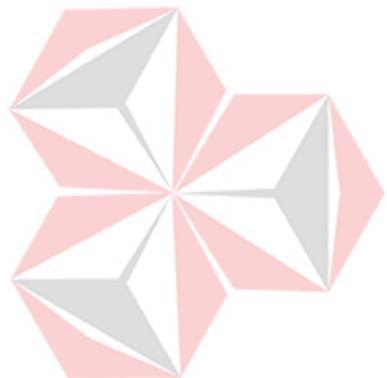
NIDN. 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

*“Alam tidak terburu-buru,
namun semuanya bisa tercapai”*

- Lao Tzu



UNIVERSITAS
Dinamika

Tugas Akhir ini ku persembahkan

kepada semua elemen Semesta

yang telah mendukungku hingga saat ini



UNIVERSITAS
Dinamika

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Sebastianus Sembara
NIM : 17410100054
Program Studi : S1 Sistem Informasi
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **IMPLEMENTASI TEST DRIVEN DEVELOPMENT
PADA RANCANG BANGUN APLIKASI E-VOTING
PEMILIHAN RAYA UNIVERSITAS DINAMIKA**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 6 Juli 2022

Yang Menyatakan



Sebastianus Sembara

NIM: 17410100054

ABSTRAK



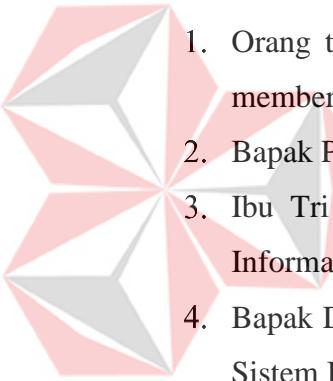
Pemilihan Raya Universitas Dinamika atau disebut juga PEMIRA adalah sarana pelaksanaan kedaulatan mahasiswa Universitas Dinamika. PEMIRA dilaksanakan untuk memilih Presiden dan Wakil Presiden Badan Eksekutif Mahasiswa (BEM) dan Ketua Umum Dewan Perwakilan Mahasiswa (DPM). Pelaksanaan PEMIRA dilakukan dengan menggunakan sebuah aplikasi *e-voting* yang dikembangkan oleh tim IT dari KPR Universitas Dinamika. Namun dilapangan, saat proses PEMIRA, ditemukan banyak sekali kesalahan pada aplikasi *e-voting* yang dikembangkan. Hal ini membuat tim KPR merasa kewalahan dan timbul kecemasan akan berkurangnya tingkat kepercayaan para pemilih atau partisipan yaitu Mahasiswa Universitas Dinamika pada KPR dalam pelaksanaan PEMIRA. Jika semua hal ini terus berlanjut, budaya PEMIRA di Universitas Dinamika akan hilang seiring waktu. Berdasarkan permasalahan tersebut, maka diperlukan sebuah metode pengembangan perangkat lunak yang dapat menghasilkan sebuah aplikasi *e-voting* PEMIRA Universitas Dinamika yang teruji atau minim akan *error*, *defect*, atau *bug*. Metode *Test Driven Development* (TDD) dipilih dalam penelitian ini karena proses pengembangan perangkat lunak yang menitikberatkan pada pengujian perangkat lunak berada diawal proses pengembangan. Proses pengujian akan menggunakan automasi test menggunakan *framework PHPUnit*. Peneliti juga akan melakukan pengujian *black box* dan survey penerimaan aplikasi menggunakan *user acceptance testing*. Hasil akhir penelitian ini menunjukkan pengujian TDD menggunakan *white box* dan pengujian secara manual menggunakan *black box* terbukti bahwa hasil pengujian aplikasi yang sekarang yaitu 100% sukses, dimana seluruh fungsi yaitu 13 fungsional yang ada mendapatkan hasil pengujian sukses. Hasil pengujian tersebut meningkat yang awalnya pada aplikasi lama sebesar 50% menjadi 100% sukses pada aplikasi baru. Hasil UAT juga menghasilkan rata rata nilai 89, 82% yang berarti bahwa aplikasi ini dapat diterima pengguna dengan baik dan siap membantu PEMIRA Universitas Dinamika.

Kata Kunci: *Test-driven development, E -Voting, PEMIRA Universitas Dinamika*

KATA PENGANTAR

Laporan Tugas Akhir dengan judul “**Implementasi Test Driven Development Pada Rancang Bangun Aplikasi E-Voting PEMIRA Universitas Dinamika**”. Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk persyaratan menyelesaikan Program Sarjana Komputer pada Program Studi S1 Sistem Informasi Universitas Dinamika.

Dalam melakukan penelitian maupun penyusunan laporan ini, penulis banyak mendapatkan dukungan dan dorongan dari berbagai pihak secara langsung maupun tidak langsung. Sehingga pada kesempatan ini, penulis menyampaikan ucapan terima kasih sebesar-besarnya kepada semua pihak yang telah membantu dalam menyelesaikan Laporan Tugas Akhir ini, terutama kepada:

- 
1. Orang tua tercinta serta keluarga yang selalu mendoakan, mendukung, dan memberikan semangat kepada setiap langkah dan aktivitas penulis.
 2. Bapak Prof. Dr. Budi Jatmiko, M. Pd selaku Rektor Universitas Dinamika.
 3. Ibu Tri Sagirani, S. Kom., M.MT. selaku Dekan Fakultas Teknologi dan Informatika
 4. Bapak Dr. Anjik Sukmaaji, S. Kom., M. Eng. selaku Ketua Program Studi S1 Sistem Informasi Universitas Dinamika.
 5. Ibu Endra Rahmawati, M. Kom. selaku Dosen Pembimbing 1 yang selalu membimbing, mendukung, dan memberikan motivasi kepada penulis dalam menyelesaikan tugas akhir ini.
 6. Ibu Sri Hariani Eko Wulandari, S. Kom., M. MT. selaku Dosen Pembimbing 2 dan juga selalu membimbing, mendukung, memberikan motivasi dan arahan kepada penulis dalam menyelesaikan tugas akhir ini.
 7. Bapak Dr. Bambang Hariadi, M.Pd. selaku Dosen Pembahas yang telah bersedia menjadi dosen pembahas dan memberikan saran serta, dukungan dalam penelitian tugas akhir ini.
 8. Ibu Norma Ningsih, S. ST., M.T. selaku Dosen yang telah membantu dan membimbing penulis khususnya pada materi pengujian perangkat lunak.

9. Teman-teman seperjuangan S1 Sistem Informasi khusunya Angkatan 2017 yang bersama-sama membantu, memberikan dukungan dan saran dari awal penelitian hingga pembuatan laporan ini.
10. Serta pihak-pihak yang tidak dapat disebutkan satu-persatu yang telah memberikan dukungan dan bantuan kepada penulis.

Penulis menyadari bahwa Tugas Akhir yang dikerjakan ini masih banyak terdapat kekurangan sehingga kritik yang bersifat membangun dan saran dari semua pihak sangatlah diharapkan agar aplikasi ini dapat diperbaiki menjadi lebih baik lagi. Semoga Tugas Akhir ini dapat diterima dan bermanfaat bagi penulis dan semua pihak.

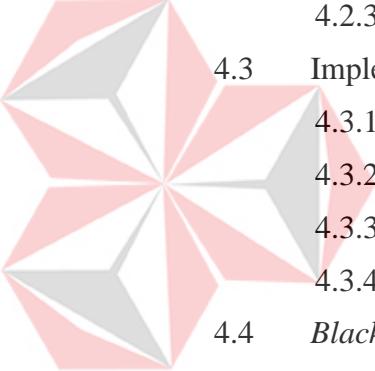
Surabaya, 9 Agustus 2022



DAFTAR ISI

Halaman

| | |
|--|------|
| ABSTRAK | vi |
| KATA PENGANTAR | vii |
| DAFTAR ISI..... | ix |
| DAFTAR TABEL..... | xi |
| DAFTAR GAMBAR | xiv |
| DAFTAR LAMPIRAN..... | xvii |
| BAB 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Batasan Masalah..... | 3 |
| 1.4 Tujuan Penelitian | 4 |
| 1.5 Manfaat Penelitian | 4 |
| BAB II LANDASAN TEORI | 5 |
| 2.1 Penelitian Terdahulu | 5 |
| 2.2 <i>Test-Driven Development</i> | 5 |
| 2.3 Pengujian Perangkat Lunak..... | 6 |
| 2.3.1 Metode Pengujian | 7 |
| 2.3.2 Tingkat Pengujian | 7 |
| 2.3.3 PHPUnit..... | 8 |
| 2.4 E - Voting | 8 |
| 2.5 Permira Universitas Dinamika | 8 |
| BAB III METODE PENELITIAN..... | 11 |
| 3.1 Tahap Awal | 12 |
| 3.1.1 Studi Literatur..... | 12 |
| 3.1.2 Observasi | 12 |
| 3.1.3 Identifikasi Masalah..... | 13 |
| 3.2 Tahap Pengembangan | 14 |
| 3.2.1 Analisis Sistem | 14 |
| 3.2.2 Desain Sistem | 21 |
| 3.2.3 <i>Test-Driven Development</i> | 22 |



| | | |
|-----------------------------------|--|----|
| 3.3 | Tahap Akhir | 23 |
| BAB IV HASIL DAN PEMBAHASAN | | 24 |
| 4.1 | Implementasi Desain Sistem | 24 |
| 4.1.1 | <i>Use Case System</i> | 24 |
| 4.1.2 | <i>Activity Diagram & Flow of Event</i> Pengelolaan Master Voting 25 | |
| 4.1.3 | <i>Activity Diagram & Flow of Event</i> Pendaftaran Calon Kandidat 26 | |
| 4.1.4 | <i>Activity Diagram & Flow of Event</i> Pemilihan Calon/ Voting .. 27 | |
| 4.1.5 | <i>Sequence Diagram</i> Pengelolaan Master Voting | 29 |
| 4.1.6 | Class Diagram..... | 30 |
| 4.2 | Implementasi <i>Test-Driven Development</i> | 31 |
| 4.2.1 | <i>Red Phase</i> | 31 |
| 4.2.2 | <i>Green Phase</i> | 36 |
| 4.2.3 | Refactoring Phase | 38 |
| 4.3 | Implementasi Sistem | 39 |
| 4.3.1 | Autentikasi | 39 |
| 4.3.2 | Halaman Pengelolaan Master Voting | 39 |
| 4.3.3 | Halaman Pendaftaran Calon Kandidat..... | 41 |
| 4.3.4 | Halaman Pemilihan Calon Kandidat/ Voting | 43 |
| 4.4 | <i>Black Box Testing</i> | 45 |
| 4.4.1 | Halaman Autentikasi | 46 |
| 4.4.2 | Halaman Pendaftaran Calon Kandidat..... | 46 |
| 4.4.3 | Halaman Pengelolaan Master Voting | 47 |
| 4.5 | User Acceptance Testing (UAT)..... | 48 |
| 4.6 | Pembahasan..... | 48 |
| BAB V PENUTUP..... | | 49 |
| 5.1 | Kesimpulan | 49 |
| 5.2 | Saran..... | 50 |
| DAFTAR PUSTAKA | | 51 |

DAFTAR TABEL

| | Halaman |
|---|---------|
| Tabel 3.1 Metode Penelitian | 11 |
| Tabel 3.2 Identifikasi Masalah dan Alternatif Solusi..... | 14 |
| Tabel 3.3 Identifikasi Pengguna..... | 16 |
| Tabel 3.4 Analisis Kebutuhan Pengguna | 16 |
| Tabel 3.5 Analisis Kebutuhan Fungsional | 17 |
| Tabel 3.6 Hak Akses | 19 |
| Tabel 4.1 <i>Flow of Event</i> Pengelolaan Voting | 25 |
| Tabel 4.2 <i>Flow of Event</i> Pendaftaran Calon Kandidat..... | 27 |
| Tabel 4.3 <i>Flow of Event</i> Pemilihan Calon/ Voting | 28 |
| Tabel 4.4 Implementasi TDD Pada Fase Red Phase..... | 35 |
| Tabel 4.5 Implementasi TDD Pada Fase <i>Green Phase</i> | 37 |
| Tabel 4.6 Struktur Kode Perangkat Lunak..... | 38 |
| Tabel 4.7 Desain <i>Testing</i> Halaman <i>Login</i> | 46 |
| Tabel 4.8 <i>Testing</i> Halaman Autentikasi/ <i>Login</i> | 46 |
| Tabel 4.9 Desain <i>Testing</i> Halaman Pendaftaran Calon Kandidat | 46 |
| Tabel 4.10 <i>Testing</i> Halaman Pendaftaran Calon Kandidat | 47 |
| Tabel 4.11 Desain <i>Testing</i> Halaman Pengelolaan Master Voting | 47 |
| Tabel 4.12 <i>Testing</i> Halaman Pengelolaan Master Voting..... | 47 |
| Tabel L1. 1 Hasil Uji Coba <i>e-voting</i> PEMIRA Sebelumnya | 52 |
| Tabel L2. 1 Penelitian Terdahulu..... | 54 |
| Tabel L4. 1 Interpretasi Skala <i>Linkert</i> | 60 |
| Tabel L5. 1 <i>Flow of Event</i> Autentikasi | 61 |
| Tabel L5. 2 <i>Flow of Event</i> Perekomendasi Mahasiswa Berpotensi | 62 |
| Tabel L5. 3 <i>Flow of Event</i> Pengelolaan Master Pengawas | 63 |
| Tabel L5. 4 <i>Flow of Event</i> Pengelolaan Master Panitia | 64 |
| Tabel L5. 5 <i>Flow of Event</i> Pengelolaan Master Peserta..... | 65 |
| Tabel L5. 6 <i>Flow of Event</i> Pengelolaan Master Calon Kandidat | 66 |
| Tabel L5. 7 <i>Flow of Event</i> Pengelolaan Master Artikel..... | 67 |
| Tabel L5. 8 <i>Flow of Event</i> Pengelolaan Aktifasi Status Voting | 68 |

| | |
|--|-----|
| Tabel L5. 9 <i>Flow of Event</i> Pengelolaan Aktifasi Status Voting | 69 |
| Tabel L5. 10 <i>Flow of Event</i> Pengelolaan Cetak Laporan..... | 70 |
| Tabel L11. 1 Desain <i>Testing</i> Halaman Perekendasian Mahasiswa Berpotensi | 98 |
| Tabel L11. 2 <i>Testing</i> Halaman Halaman Perekendasian Mahasiswa Berpotensi | 98 |
| Tabel L11. 3 Desain <i>Testing</i> Halaman Pengelolaan Master Pengawas | 98 |
| Tabel L11. 4 <i>Testing</i> Halaman Pengelolaan Master Pengawas | 99 |
| Tabel L11. 5 Desain <i>Testing</i> Halaman Pengelolaan Master Panitia | 99 |
| Tabel L11. 6 <i>Testing</i> Halaman Pengelolaan Master Panitia | 100 |
| Tabel L11. 7 Desain <i>Testing</i> Halaman Pengelolaan Master Peserta..... | 100 |
| Tabel L11. 8 <i>Testing</i> Halaman Pengelolaan Master Peserta..... | 100 |
| Tabel L11. 9 Desain <i>Testing</i> Halaman Pengelolaan Master Calon Kandidat | 101 |
| Tabel L11. 10 <i>Testing</i> Halaman Pengelolaan Master Calon Kandidat | 101 |
| Tabel L11. 11 Desain <i>Testing</i> Halaman Pengelolaan Master Artikel | 102 |
| Tabel L11. 12 <i>Testing</i> Halaman Pengelolaan Master Artikel | 102 |
| Tabel L11. 13 Desain <i>Testing</i> Aktifasi Status Voting..... | 103 |
| Tabel L11. 14 <i>Testing</i> Halaman Aktifasi Status Voting | 103 |
| Tabel L11. 15 Desain <i>Testing</i> Pemilihan Calon Kandidat..... | 103 |
| Tabel L11. 16 <i>Testing</i> Halaman Pemilihan Calon Kandidat..... | 104 |
| Tabel L11. 17 Desain <i>Testing</i> Halaman Pemilihan Sementara..... | 105 |
| Tabel L11. 18 <i>Testing</i> Halaman Hasil Pemilihan Sementara | 105 |
| Tabel L11. 19 Desain <i>Testing</i> Mencetak Laporan Voting | 105 |
| Tabel L11. 20 <i>Testing</i> Halaman Mencetak Laporan Voting | 105 |
| Tabel L12. 1 Daftar Pertanyaan <i>User Acceptance Test</i> (UAT) | 106 |
| Tabel L13. 1 Pengujian Kuesioner Soal 1..... | 107 |
| Tabel L13. 2 Pengujian Kuesioner Soal 2..... | 107 |
| Tabel L13. 3 Pengujian Kuesioner Soal 3..... | 108 |
| Tabel L13. 4 Pengujian Kuesioner Soal 4..... | 108 |
| Tabel L13. 5 Pengujian Kuesioner Soal 5..... | 109 |
| Tabel L13. 6 Pengujian Kuesioner Soal 6..... | 109 |
| Tabel L13. 7 Pengujian Kuesioner Soal 7..... | 110 |
| Tabel L13. 8 Pengujian Kuesioner Soal 8..... | 110 |

| | |
|---|-----|
| Tabel L13. 9 Pengujian Kuesioner Soal 9..... | 110 |
| Tabel L13. 10 Pengujian Kuesioner Soal 10..... | 111 |
| Tabel L14. 1 Perbandingan Hasil Uji Coba Aplikasi Baru dan Lama | 112 |



DAFTAR GAMBAR

Halaman

| | |
|---|----|
| Gambar 2.1 Metode Test Driven Development | 6 |
| Gambar 2.2 <i>Pop-up</i> PEMIRA Pada SICYCA..... | 10 |
| Gambar 3.1 <i>Use Case Bussines</i> PEMIRA Universitas Dinamika..... | 13 |
| Gambar 3.2 Diagram Input, Proses dan Output (1) | 20 |
| Gambar 3.3 Diagram Input, Proses dan Output (2) | 21 |
| Gambar 3.4 Diagram Alir Implementasi TDD Pada Pengembangan Aplikasi E-Voting PEMIRA Universitas Dinamika..... | 22 |
| Gambar 4.1 <i>Use Case System</i> E-Voting PEMIRA Universitas Dinamika..... | 24 |
| Gambar 4.2 <i>Activity Diagram</i> Pengelolaan data Voting..... | 25 |
| Gambar 4.3 <i>Activity Diagram</i> Pendaftaran Calon Kandidat..... | 26 |
| Gambar 4.4 <i>Activity Diagram</i> Pemilihan Calon/ Voting | 28 |
| Gambar 4.5 <i>Sequence Diagram</i> Pengelolaan Voting..... | 29 |
| Gambar 4.6 <i>Class Diagram</i> | 30 |
| Gambar 4.7 Test Case Daftar Voting | 31 |
| Gambar 4.8 <i>Test Case</i> Tambah Voting | 32 |
| Gambar 4.9 Kode Pengujian Pengelolaan Voting Destroy | 33 |
| Gambar 4.10 Kode Pengujian Pengelolaan Voting Update | 34 |
| Gambar 4.11 <i>Red Phase</i> - Hasil Uji Coba Kode Pengujian..... | 36 |
| Gambar 4.12 <i>Green Phase</i> - Hasil Uji Coba Kode Pengujian | 37 |
| Gambar 4.13 Struktur <i>Folder Model</i> | 38 |
| Gambar 4.14 Halaman Autentikasi | 39 |
| Gambar 4.15 Halaman Voting | 40 |
| Gambar 4.16 Halaman <i>Form</i> Tambah Data Voting..... | 40 |
| Gambar 4.17 Halaman <i>Form</i> Ubah Data Voting | 41 |
| Gambar 4.18 Halaman Pendaftaran Calon Kandidat | 41 |
| Gambar 4.19 Halaman Mengelola Visi dan Misi..... | 42 |
| Gambar 4.20 Halaman Mengelola Data Pasangan Calon | 43 |
| Gambar 4.21 Halaman Pengecekan Peserta Voting..... | 44 |
| Gambar 4.22 Halaman Masuk Peserta | 44 |



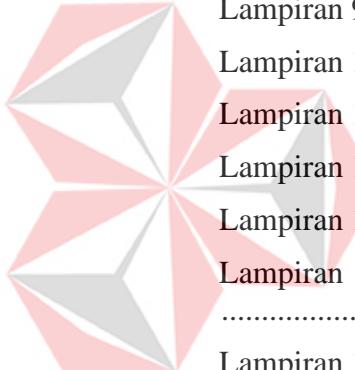
| | |
|--|----|
| Gambar 4.23 Halaman Pemilihan Kandidat/ Voting | 45 |
| Gambar L3. 1 Implementasi Pengujian Pada TDD..... | 56 |
| Gambar L4. 1 Layer Dalam <i>Testing</i> Perangkat Lunak | 57 |
| Gambar L5. 1 <i>Activity Diagram</i> Autentikasi | 61 |
| Gambar L5. 2 <i>Activity Diagram</i> Perekomendasi Mahasiswa Berpotensi | 62 |
| Gambar L5. 3 <i>Activity Diagram</i> Pengelolaan Master Pengawas | 63 |
| Gambar L5. 4 <i>Activity Diagram</i> Pengelolaan Master Panitia | 64 |
| Gambar L5. 5 <i>Activity Diagram</i> Pengelolaan Master Peserta..... | 65 |
| Gambar L5. 6 <i>Activity Diagram</i> Pengelolaan Master Calon Kandidat..... | 66 |
| Gambar L5. 7 <i>Activity Diagram</i> Pengelolaan Master Artikel..... | 67 |
| Gambar L5. 8 <i>Activity Diagram</i> Aktifasi Status Voting | 68 |
| Gambar L5. 9 <i>Activity Diagram</i> Pengelolaan Dashboard..... | 69 |
| Gambar L5. 10 <i>Activity Diagram</i> Pengelolaan Cetak Laporan..... | 70 |
| Gambar L6. 1 <i>Sequence Diagram</i> Autentikasi | 71 |
| Gambar L6. 2 <i>Sequence Diagram</i> Pendaftaran Calon Kandidat..... | 73 |
| Gambar L6. 3 <i>Sequence Diagram</i> Pengelolaan Master Pengawas | 74 |
| Gambar L6. 4 <i>Sequence Diagram</i> Pengelolaan Master Panitia | 75 |
| Gambar L6. 5 <i>Sequence Diagram</i> Pengelolaan Master Peserta..... | 76 |
| Gambar L7. 1 <i>Red Phase</i> - Hasil Uji Coba Kode Pengujian Autentikasi | 81 |
| Gambar L7. 2 <i>Red Phase</i> - Hasil Uji Coba Kode Perekomendasi Mahasiswa Berpotensi | 81 |
| Gambar L7. 3 <i>Red Phase</i> - Hasil Uji Coba Kode Pendaftaran Calon Kandidat ... | 81 |
| Gambar L7. 4 <i>Red Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Pengawas | 82 |
| Gambar L7. 5 <i>Red Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Panitia..... | 82 |
| Gambar L7. 6 <i>Red Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Peserta | 82 |
| Gambar L7. 7 <i>Red Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Calon Kandidat | 83 |
| Gambar L7. 8 <i>Red Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Artikel | 83 |
| Gambar L7. 9 <i>Red Phase</i> - Hasil Uji Coba Kode Aktifasi Status Voting..... | 83 |
| Gambar L7. 10 <i>Red Phase</i> - Hasil Uji Coba Kode Pemilihan Calon Kandidat/ Voting..... | 84 |
| Gambar L7. 11 <i>Red Phase</i> - Hasil Uji Coba Kode Aktifasi Status Voting..... | 84 |

| | |
|--|----|
| Gambar L7. 12 <i>Red Phase</i> - Hasil Uji Coba Kode Mencetak Laporan Voting | 84 |
| Gambar L8. 1 <i>Green Phase</i> - Hasil Uji Coba Kode Pengujian Autentikasi | 85 |
| Gambar L8. 2 <i>Green Phase</i> - Hasil Uji Coba Kode Perekomendasi Mahasiswa Berpotensi | 85 |
| Gambar L8. 3 <i>Green Phase</i> - Hasil Uji Coba Kode Halaman Pendaftaran Calon Kandidat..... | 85 |
| Gambar L8. 4 <i>Green Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Pengawas | 86 |
| Gambar L8. 5 <i>Green Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Panitia. | 86 |
| Gambar L8. 6 <i>Green Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Peserta.. | 87 |
| Gambar L8. 7 <i>Green Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Calon Kandidat | 87 |
| Gambar L8. 8 <i>Green Phase</i> - Hasil Uji Coba Kode Pengelolaan Master Artikel. | 88 |
| Gambar L8. 9 <i>Green Phase</i> - Hasil Uji Coba Kode Aktifasi Status Voting..... | 88 |
| Gambar L8. 10 <i>Green Phase</i> - Hasil Uji Coba Kode Pemilihan Calon Kandidat/ Voting..... | 88 |
| Gambar L8. 11 <i>Green Phase</i> - Hasil Uji Coba Kode Hasil Pemilihan Sementara | 88 |
| Gambar L8. 12 <i>Green Phase</i> - Hasil Uji Coba Kode Mencetak Laporan Voting | 89 |
| Gambar L9. 1 Struktur Folder <i>View</i> | 90 |
| Gambar L9. 2 Struktur Folder <i>Controllers</i> | 91 |
| Gambar L9. 3 Struktur Folder <i>Repositories</i> | 91 |
| Gambar L9. 4 Struktur Folder <i>Routes</i> | 92 |
| Gambar L9. 5 Struktur Folder <i>Tests</i> | 92 |
| Gambar L10. 1 Halaman Master Panitia..... | 93 |
| Gambar L10. 2 Halaman <i>Form</i> Tambah Data Panitia | 93 |
| Gambar L10. 3 Halaman <i>Form</i> Ubah Data Panitia..... | 94 |
| Gambar L10. 4 Halaman Master Peserta | 94 |
| Gambar L10. 5 Halaman <i>Form</i> Tambah Data Peserta..... | 95 |
| Gambar L10. 6 Halaman <i>Form</i> Ubah Data Peserta | 95 |
| Gambar L10. 7 Halaman Master Artikel..... | 96 |
| Gambar L10. 8 Halaman <i>Form</i> Tambah Data Artikel | 97 |
| Gambar L10. 9 Halaman <i>Form</i> Ubah Data Artikel..... | 97 |

DAFTAR LAMPIRAN

Halaman

| | |
|--|-----|
| Lampiran 1 Hasil Uji Coba Aplikasi E-Voting PEMIRA Universitas Dinamika Sebelumnya | 52 |
| Lampiran 2 Landasan Teori - Penelitian Terdahulu | 54 |
| Lampiran 3 Landasan Teori - Tahap <i>Test-Driven Development</i> | 55 |
| Lampiran 4 Landasan Teori - Tingkat Pengujian Perangkat Lunak | 57 |
| Lampiran 5 <i>Activity Diagram & Flow of Event</i> | 61 |
| Lampiran 6 <i>Sequence Diagram</i> | 71 |
| Lampiran 7 Implementasi TDD – Red Phase | 81 |
| Lampiran 8 Implementasi TDD – Green Phase | 85 |
| Lampiran 9 Implementasi TDD – Refactor | 90 |
| Lampiran 10 Implementasi Sistem..... | 93 |
| Lampiran 11 <i>Black Box Testing</i> | 98 |
| Lampiran 12 Draft Pertanyaan Kuesioner User Acceptance Testing (UAT) . | 106 |
| Lampiran 13 Hasil User Acceptance Testing (UAT)..... | 107 |
| Lampiran 14 Perbandingan Hasil Uji Coba Aplikasi E-Voting Lama dan Baru | 112 |
| Lampiran 15 Hasil Cek Plagiasi..... | 115 |
| Lampiran 16 Biodata Penulis | 116 |



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pengembangan perangkat lunak adalah sebuah proses yang menyangkut pengembangan suatu produk berupa perangkat lunak dalam bentuk sebuah penelitian, pengembangan baru, atau modifikasi sesuatu yang telah ada. Pengembangan perangkat lunak biasanya memiliki tujuan umum, yaitu memenuhi kebutuhan bisnis dan automasi pekerjaan atau tugas yang rumit. *Test-driven development* (TDD) adalah metode pengembangan perangkat lunak dengan langkah kerja membuat sebuah automasi pengujian untuk menghasilkan sebuah struktur kode yang terstruktur (Beck, 2002). Sedangkan menurut Karac & Turhan (2018) TDD adalah praktik dimana seorang pengembang perangkat lunak menulis kode pengujian terlebih dahulu sebelum pembuatan kode sumber. Dengan kata lain, TDD adalah sebuah metode yang menekankan pembuatan kode pengujian sebelum membuat kode perangkat lunak. Menurut Khanam & Ahsan (2017), TDD memiliki beberapa keunggulan, antara lain adalah (1) pengembang dapat dipandu oleh kode pengujian yang telah dibuat. Kode pengujian dibuat berdasarkan spesifikasi dari masing-masing fungsionalitas yang telah direncanakan, (2) Pengembang perangkat lunak dapat mengetahui lebih awal adanya *error*, *bug*, atau *defect* pada perangkat lunak yang sedang dikembangkan, karena pengujian kode yang ada diawali pengembangan, (3) Menghasilkan sebuah kode perangkat lunak yang terstruktur dan mudah dibaca. Adanya restrukturisasi kode pada TDD dapat membuat pengembang perangkat lunak dapat menghasilkan kode perangkat lunak yang terstruktur dan rapi (*clean code*). Hal ini juga dapat mempermudah seorang pengembang perangkat lunak dalam proses pemeliharaan perangkat lunak di kemudian hari.

Pemilihan Raya Universitas Dinamika atau disebut juga PEMIRA adalah sarana pelaksanaan kedaulatan mahasiswa Universitas Dinamika berdasarkan Undang-Undang Dasar Keluarga Mahasiswa Universitas Dinamika. PEMIRA dilaksanakan untuk memilih Presiden dan Wakil Presiden Badan Eksekutif



Mahasiswa (BEM) dan Ketua Umum Dewan Perwakilan Mahasiswa (DPM) (Dewan Perwakilan Mahasiswa, 2019a). PEMIRA diselenggarakan oleh Komisi Pemilihan Raya (KPR). KPR bernaung di bawah Dewan Perwakilan Mahasiswa (DPM) sebagai salah perwujudan tugas dan tanggung jawab DPM-U dalam menyelenggarakan pembentukan tim formatur KPR yang tertera dalam Pasal 8 Ayat (3a) UUD-KM UNDIKA/2019 tentang Organisasi Kemahasiswaan dan Kegiatan Mahasiswa dan Pasal 18 Ayat 5 UUD-KM UNDIKA/2019 tentang Tugas DPM-U. Segala kebijakan dan teknis pelaksanaan PEMIRA oleh KPR diatur dalam UU PEMIRA 2019. Pelaksanaan PEMIRA dilakukan dengan menggunakan sebuah aplikasi *e-voting* yang dikembangkan oleh tim IT dari KPR Universitas Dinamika. Aplikasi *e-voting* diharapkan dapat menjawab segala kebutuhan dari KPR selaku pantia dan Mahasiswa selaku partisipan dari PEMIRA ini. Namun dilapangan, saat proses PEMIRA, ditemukan banyak sekali kesalahan pada aplikasi *e-voting* yang dikembangkan. Hal ini membuat tim KPR merasa kewalahan dan timbul kecemasan akan berkurangnya tingkat kepercayaan para pemilih atau partisipan yaitu Mahasiswa Universitas Dinamika pada KPR dalam pelaksanaan PEMIRA. Penjelasan hasil uji coba aplikasi *e-voting* PEMIRA Universitas Dinamika sebelumnya dapat dilihat pada Lampiran 1. Dari Tabel Lampiran 1 dapat dilihat hasil uji coba aplikasi *e-voting* PEMIRA Universitas Dinamika sebelumnya dimana 10 dari 20 fungsional yang ada mendapatkan hasil pengujian gagal. Disini dapat dilihat bahwa 50% dari total keseluruhan aplikasi *e-voting* PEMIRA Universitas Dinamika mengalami masalah.

PEMIRA Universitas Dinamika juga sangat sepi peminat dalam hal partisipan maupun pendaftaran Calon Kandidat. Panitia KPR sering kali terjun langsung ke dalam lapangan untuk mencari Calon Kandidat. Pada PEMIRA tahun 2020, terbukti hanya ada 2 Calon Kandidat untuk pemilihan BEM dan 1 Calon Kandidat untuk pemilihan DPM. KPR juga kerap kali meminta bantuan kepada kemahasiswaan dan ketua program studi untuk membantu mencari Calon Kandidat yang cocok dan ingin menjadi presiden dan wakil presiden BEM maupun ketua DPM. Namun, kemahasiswaan dan ketua program studi juga bingung untuk menentukan mahasiswa yang layak untuk menjadi Calon Kandidat. Jika semua hal ini terus berlanjut, budaya PEMIRA di Universitas Dinamika akan hilang seiring

waktu. Dampak yang terjadi akibat PEMIRA tidak bisa terlaksana adalah Regenerasi tidak ada atau terhenti dan berimbas pada organisasi mahasiswa (ORMAWA) yang akan terbengkalai, karena tidak adanya badan atau organisasi yang merawat, mengakomodasi aspirasi secara sistematis, dan menampung ide gagasan dan membantu menyelanggarakan kegiatan ORMAWA.

Berdasarkan permasalahan tersebut, maka diperlukan sebuah metode pengembangan perangkat lunak yang dapat menghasilkan sebuah aplikasi *e-voting* PEMIRA Universitas Dinamika yang teruji atau minim akan *error*, *defect*, atau *bug*, sehingga Mahasiswa Universitas Dinamika dapat melaksanakan PEMIRA dengan baik. Metode *Test Driven Development* (TDD) dipilih dalam penelitian ini karena proses pengembangan perangkat lunak yang menitikberatkan pada pengujian perangkat lunak berada diawal proses pengembangan. Proses pengujian akan menggunakan automasi test menggunakan *framework PHPUnit*. Setiap fungsionalitas akan dipecah menjadi unit terkecil dan akan diuji hingga hasil uji menghasilkan keluaran sukses. Selain itu TDD juga menekankan perihal restrukturisasi kode perangkat lunak atau biasa disebut *code refactoring*. Tahap ini juga dapat menghasilkan struktur kode perangkat lunak yang terstruktur dan mudah dibaca, sehingga dapat dengan mudah sebuah aplikasi dapat dikembangkan dan dipelihara.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka rumusan masalah tugas akhir ini adalah bagaimana mengimplementasikan metode *test-driven development* pada rancang bangun aplikasi e-voting PEMIRA di Universitas Dinamika.

1.3 Batasan Masalah

Berdasarkan uraian diatas maka penulis membatasi pokok permasalahan yang akan dibahas, yakni sebagai berikut:

1. Pengembangan aplikasi menggunakan metode *test-driven development* dengan melakukan pengujian *unit test* menggunakan metode *white box testing* dan *framework PHPUnit*

2. Survey penerimaan aplikasi menggunakan *user acceptance test* (UAT) dengan metode *alpha-beta testing*
3. Studi Kasus yang digunakan dalam penelitian ini adalah aplikasi *e-voting* PEMIRA Universitas Dinamika
4. Pembuatan kode pengujian program hanya terbatas pada fungsionalitas sisi *back-end* aplikasi *e-voting* PEMIRA Universitas Dinamika
5. Keluaran dari penelitian ini adalah berupa aplikasi e-voting PEMIRA Universitas Dinamika yang telah diperbarui

1.4 Tujuan Penelitian

Berdasarkan latar belakang maka tujuan penelitian ini adalah mengimplementasikan metode *test-driven development* pada rancang bangun aplikasi e-voting PEMIRA Universitas Dinamika untuk mengetahui tingkat *error*, *defect*, dan *bug* pada aplikasi, sehingga dapat dikembangkan sebuah aplikasi baru dengan *error*, *bug*, dan *defect* yang minimal.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

1. Membantu pengembang aplikasi *e-voting* PEMIRA Universitas Dinamika untuk dapat mengetahui *error*, *bug* dan *defect* lebih awal
2. Memandu pengembang aplikasi *e-voting* PEMIRA Universitas Dinamika dalam proses pengembangan, karena semua spesifikasi yang ditargetkan akan dituangkan pada kode pengujian
3. Membantu pengembang aplikasi saat proses modifikasi spesifikasi aplikasi *e-voting* PEMIRA Universitas Dinamika, karena adanya bantuan *automated testing*
4. Menambah percaya diri pengembang aplikasi *e-voting* PEMIRA Universitas Dinamika, karena dapat membangun aplikasi yang teruji dengan baik
5. Meningkatkan tingkat partisipasi Mahasiswa Universitas Dinamika dalam mengikuti PEMIRA

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu panduan penulis dalam melakukan penelitian terhadap implementasi metode *test-driven development*, sehingga penulis dapat memperbanyak teori yang digunakan dalam penelitian yang dilakukan. Berikut ini adalah hasil penelitian terdahulu yang dapat dilihat pada Lampiran 2

2.2 *Test-Driven Development*

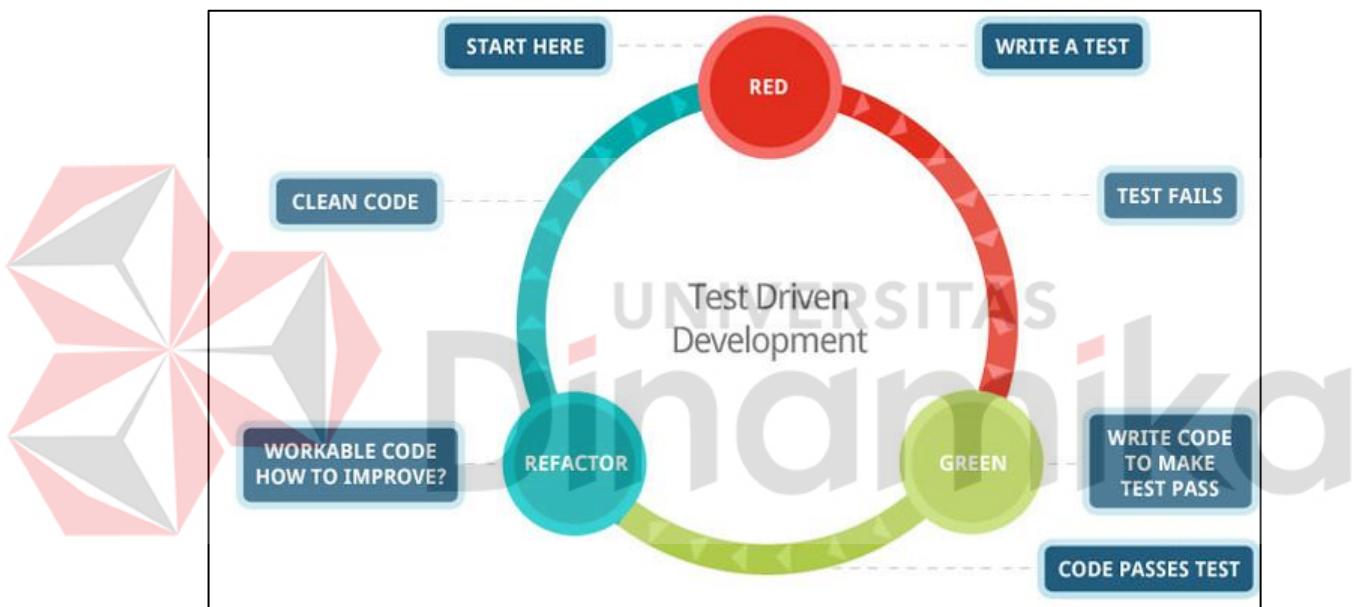
Menurut Beck (2002), *test-driven development* (TDD) adalah metode pengembangan perangkat lunak dengan membuat sebuah automasi pengujian untuk menghasilkan sebuah struktur kode yang terstruktur (*clean code*). Dalam pengembangan menggunakan TDD, kita diharuskan membuat kode pengujian, lalu membuat kode aplikasi, melakukan pengujian, hingga menghasilkan keluaran sukses pada proses pengujian.

Menurut Kumar & Bansal (2013), TDD adalah sebuah metode pengembangan perangkat lunak dengan membuat kode pengujian terlebih dahulu, dan mengetes kode aplikasi secara berkala selama proses pengembangan. Dan saat kode aplikasi memiliki keluaran sukses dalam pengujian, kode aplikasi akan direfaktorisasi untuk meningkatkan kualitas struktur kode. Siklus ini akan terus diulang hingga seluruh fungsionalitas diimplementasikan.

Dari pengertian diatas, dapat disimpulkan bahwa TTD adalah sebuah metode pengembangan perangkat lunak yang pengerjaannya adalah dengan membuat kode pengujian terlebih dahulu baru membuat kode aplikasi. Menurut Khanam & Ahsan (2017), Implementasi TDD memiliki beberapa manfaat, yaitu:

1. Mengetahui adanya *error*, *defect*, atau *bug* lebih awal sehingga dapat segera menangani hal tersebut

2. Meningkatkan kualitas kode aplikasi yang sedang dikembangkan, karena dengan adanya tahap refaktorisasi kode, pengembang dapat menghasilkan struktur kode yang lebih rapi dan efektif (*clean code*)
3. Memudahkan pengembang untuk melakukan pemeliharaan atau *maintenance* terhadap kode aplikasi yang dikembangkan
4. Meningkatkan kepercayaan diri seorang pengembang aplikasi, karena menghasilkan sebuah aplikasi yang teruji dan baik
5. Memandu seorang pengembang aplikasi dalam pengembangan aplikasi, karena adanya pembuatan kode aplikasi diawal, sehingga dapat memahami permintaan dan lingkup aplikasi dengan cepat



Gambar 2.1 Metode Test Driven Development
(Sumber: Kumar & Bansal, 2013)

Detail dari Metode Test-Driven Development dapat dilihat pada Lampiran 3.

2.3 Pengujian Perangkat Lunak

Menurut Hasibuan & Dirgahayu (2021) pengujian perangkat lunak merupakan proses meminimalkan peluang kegagalan dalam eksekusi perangkat lunak. Pengujian perangkat lunak bertujuan untuk menemukan suatu kesalahan dan kecacatan sebuah perangkat lunak yang sedang dikembangkan.

Menurut Romeo (2003), pengujian perangkat lunak adalah sebuah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan. Hal ini bertujuan untuk (1) verifikasi apakah telah berjalan dengan sebagaimana mestinya (menurut spesifikasi), (2) Mendeteksi *error*, (3) validasi apakah spesifikasi sudah memenuhi kebutuhan pengguna.

Dari pengertian diatas dapat disimpulkan bahwa pengujian perangkat lunak adalah sebuah proses menguji kesesuaian sebuah perangkat lunak, dengan berbagai macam kemungkinan agar pengembang perangkat lunak dapat mengetahui apakah perangkat lunak yang sedang dikembangkan telah sesuai dengan spesifikasi atau kebutuhan pengguna tanpa adanya kecacatan atau kesalahan.

2.3.1 Metode Pengujian

Metode pengujian digunakan untuk menemukan segala kemungkinan tertinggi menemukan *error*. Hal yang dilakukan adalah dengan cara membuat sebuah *test case*. Dengan membuat sebuah *test case*, penguji akan disediakan sebuah mekanisme untuk memastikan kelengkapan dari sebuah pengujian dan kemungkinan tertinggi untuk mendapatkan *error* pada sebuah perangkat lunak yang sedang dikembangkan (Romeo, 2003). Pada penelitian ini, penulis menggunakan 2 pendekatan dalam melakukan proses pengujian yaitu pendekatan *white box testing* dan *black box testing*. Pendekatan *white box testing* digunakan pada saat melakukan tes kode perangkat lunak menggunakan PHPUnit saat proses TDD, sedangkan pendekatan *black box testing* akan digunakan saat tahap *alpha testing* pada proses melakukan *user acceptance testing* (UAT).

2.3.2 Tingkat Pengujian

Terdapat 4 tingkatan layer dalam melakukan testing, yakni (1) *unit testing*, (2) *integration testing*, (3) *system testing*, dan (4) *acceptance testing* (Romeo, 2003). Tujuan dari tingkatan pengujian ini adalah untuk membuat pengujian perangkat lunak menjadi sistematis dan mudah untuk mengidentifikasi semua kemungkinan kasus pada setiap tingkatan uji. Detail tingkat pengujian dapat dilihat pada Lampiran 4

2.3.3 PHPUnit

PHPUnit adalah sebuah kerangka kerja pengujian perangkat lunak yang menggunakan Bahasa Pemrograman PHP (Bergmann, 2022). PHPUnit dibuat oleh Sebastian Bergmann dan di hosting pada Github. PHPUnit menggunakan arsitektur xUnit untuk kerangka pengujian *unit testing*. Proyek *Open Source* PHPUnit dimulai pada tanggal 25 Juni 2006 dan masih aktif dikembangkan hingga sekarang. Saat ini PHPUnit memiliki versi terbaru yakni versi 9.5 yang dirilis pada tanggal 18 Agustus 2021.

2.4 E - Voting

E-voting adalah sebuah sistem pemungutan suara berbasis elektronik, dimana data akan disimpan, diproses, dan diolah dalam bentuk informasi digital. Jadi mulai dari proses pendaftaran pemilih, calon pemilih, pelaksanaan pemilihan, penghitungan suara, dan laporan hasil suara akan dilakukan secara elektronik (Basri et al., 2021). Sedangkan menurut Hardjaloka & Simarmata (2016) Dalam istilah e-voting (pemungutan suara elektronik) kita biasanya memahami proses pemungutan suara yang memungkinkan pemilih untuk memberikan suara (aman dan rahasia) melalui Internet atau Intranet

Dari penjelasan diatas dapat disimpulkan *e-voting* adalah sebuah metode dimana semua proses pemilu dapat dilakukan secara daring melalui media elektronik, untuk menghemat biaya dan waktu, serta mengurangi kecurangan dalam proses voting itu sendiri karena semua dilakukan menggunakan komputer. Pada penilitian ini, penulis akan mengambil studi kasus aplikasi *e-voting* PEMIRA Universitas Dinamika, dimana *e-voting* ini dikembangkan untuk memilih Calon Kandidat ketua Dewan Perwakilan Mahasiswa, Presiden dan Wakil Presiden Mahasiswa Universitas Dinamika.

2.5 Permira Universitas Dinamika

Pemilihan Raya (PEMIRA) Universitas Dinamika adalah sarana pelaksanaan kedaulatan mahasiswa berdasarkan Undang Undang Dasar Keluarga Mahasiswa Universitas Dinamika. PEMIRA diadakan oleh Komisi Pemilihan Raya yang

diawasi oleh Badan Pengawas PEMIRA (BAWASRA). PEMIRA diselenggarakan untuk memilih Ketua Umum DPM-U, Presiden Mahasiswa, dan Wakil Presiden Mahasiswa (Dewan Perwakilan Mahasiswa, 2019b). PEMIRA diselenggarakan oleh Komisi Pemilihan Raya (KPR). KPR bernaung dibawah Dewan Perwakilan Mahasiswa (DPM) sebagai salah perwujudan tugas dan tanggung jawab DPM-U dalam menyelenggarakan pembentukan tim formatur KPR yang tertera dalam Pasal 8 Ayat (3a) UUD-KM UNDIKA/2019 tentang Organisasi Kemahasiswaan dan Kegiatan Mahasiswa dan Pasal 18 Ayat 5 UUD-KM UNDIKA/2019 tentang Tugas DPM-U. Segala kebijakan dan teknis pelaksanaan PEMIRA oleh KPR diatur dalam UU PEMIRA 2019.

Berikut ini adalah alur proses berlangsungnya PEMIRA di Universitas Dinamika:

1. Pendaftaran Calon Kandidat
2. Seleksi Calon Kandidat
3. Penetapan Calon
4. Pelaksanaan Kampanye
5. Pemungutan Suara
6. Perhitungan Suara
7. Penetapan Calon Terpilih



Proses pelaksanaan PEMIRA di Universitas Dinamika awalnya menggunakan SICYCA (*System Cyber Campus*). Fungsi dari proses voting PEMIRA disajikan dalam sebuah pop-up. Fungsi Pop-up adalah sebuah tampilan berupa halaman kecil yang tiba tiba muncul ketika mengunjungi sebuah website. Saat mahasiswa mengunjungi *SICYCA*, mereka akan langsung disuguhkan oleh tampilan calon ketua dan wakil BEM-U dan ketua DPM-U Universitas Dinamika. Dari pelaksanaan PEMIRA menggunakan *SICYCA* ini, ada beberapa mahasiswa atau pemilih yang secara tidak sengaja memilih paslon yang tidak diinginkan, karena terkejut atau sedang terburu buru untuk melihat jadwal kuliah, karena pada dasarnya *SICYCA* dipergunakan untuk melihat jadwal kelas dan matakuliah. Tampilan *pop-up* yang sederhana dan kurang jelas juga membuat antusias

mahasiswa dalam melakukan pemilihan ketua dan wakil BEM-U dan ketua DPM-U berkurang. *Pop-up* yang kecil dirasa kurang untuk memuat segala informasi terkait calon ketua dan wakil BEM-U dan ketua DPM-U, seperti biodata diri, riwayat prestasi, dan riwayat organisasi calon, yang dapat meyakinkan pemilih dalam memilih calon yang terbaik.



Gambar 2.2 *Pop-up* PEMIRA Pada SICYCA

(Sumber: Instagram @kprunivdinamika, diakses tanggal 08-09-2021)

Pada penelitian ini, penulis ingin membuat sebuah sistem yang mencakup seluruh proses PEMIRA di Universitas Dinamika. Mulai dari pendaftaran Calon Kandidat, seleksi Calon Kandidat, penetapan calon, kampanye, pemungutan dan perhitungan suara hingga penetapan calon terpilih. Namun pada proses seleksi Calon Kandidat hanya sebatas proses pengecekan dokumen prasyarat saja seperti surat keterangan aktif mahasiswa, surat keterangan aktif organisasi, transkrip nilai, bukti telah mengikuti LKM-TD, dan bukti *point* SSKM. Untuk proses wawancara dan tes tulis akan dilakukan diluar sistem.

BAB III

METODE PENELITIAN

Tahapan metode penelitian berisi tentang penjelasan mengenai tahapan atau langkah yang dikerjakan dalam menyusun dan menyelesaikan penelitian ini. Tahapan metode penelitian dilakukan dengan tujuan agar dapat mengerjakan penelitian dengan proses yang diperlukan agar dalam penggerjaan dapat dilakukan dengan terstruktur dan sistematis. Tahapan metode penelitian dapat dilihat pada Tabel 3.1

Tabel 3.1 Metode Penelitian

| No. | Metode Penelitian | Luaran |
|-----|--|--------------------------|
| 1. | Tahap Awal | |
| | I. Studi Literatur | |
| | II. Observasi | |
| | III. Identifikasi Masalah | |
| 2. | Tahap Pengembangan | |
| | I. Analisis | |
| | a. Identifikasi Pengguna | Dokumen User Requirement |
| | b. Identifikasi Kebutuhan Data | |
| | c. Analisis Kebutuhan Fungsional | |
| | d. Analisis Kebutuhan Non Fungsional | |
| | e. Analisis Kebutuhan Pengguna | |
| | II. Desain | |
| | a. Use Case Bisnis | Dokumen Desain Sistem |
| | b. Diagram Activity | |
| | c. Use Case Sistem | |
| | d. Flow of Event | |
| | e. Sequence Diagram | |
| | f. Class Diagram | |
| | III. TDD - Red Phase | |
| | a. Membuat Kode Pengujian | Kode Pengujian |
| | b. Melakukan Pengujian Perangkat Lunak Menggunakan Automated Test dengan Tools PHPUnit dan Metode <i>White Box Testing</i> | |
| | IV. TDD – Green Phase | |
| | a. Membuat Kode Perangkat Lunak | |
| | b. Melakukan Pengujian Perangkat Lunak Menggunakan Automated Test dengan Tools PHPUnit dan Metode <i>White Box Testing</i> | Kode Perangkat Lunak |
| | V. TDD – Refactoring Phase | |
| 3. | VI. Implementasi Sistem | Aplikasi <i>E-Voting</i> |
| 4. | Tahap Akhir | |
| | I. <i>Black Box Testing - Alpha Test</i> | Dokumen Hasil Pengujian |
| | II. <i>User Acceptance Testing (UAT) – Beta Test</i> | Dokumen UAT |
| | III. Membuat Laporan dan Dokumentasi | Dokumen Laporan |



3.1 Tahap Awal

Pada tahap awal terdapat tiga tahapan, yaitu studi literatur, Observasi, dan identifikasi masalah.

3.1.1 Studi Literatur

Studi literatur pada penelitian ini merupakan tahapan dalam mencari informasi yang diperoleh dari buku ilmiah, laporan penelitian, peraturan-peraturan, dan sumber-sumber tertulis baik tercetak maupun elektronik. Studi literatur yang dilakukan peneliti yaitu untuk memahami metode perancangan aplikasi *yaitu test-driven development*, sistem pemilihan umum raya di Universitas Dinamika, serta pengertian dari *e-voting* itu sendiri

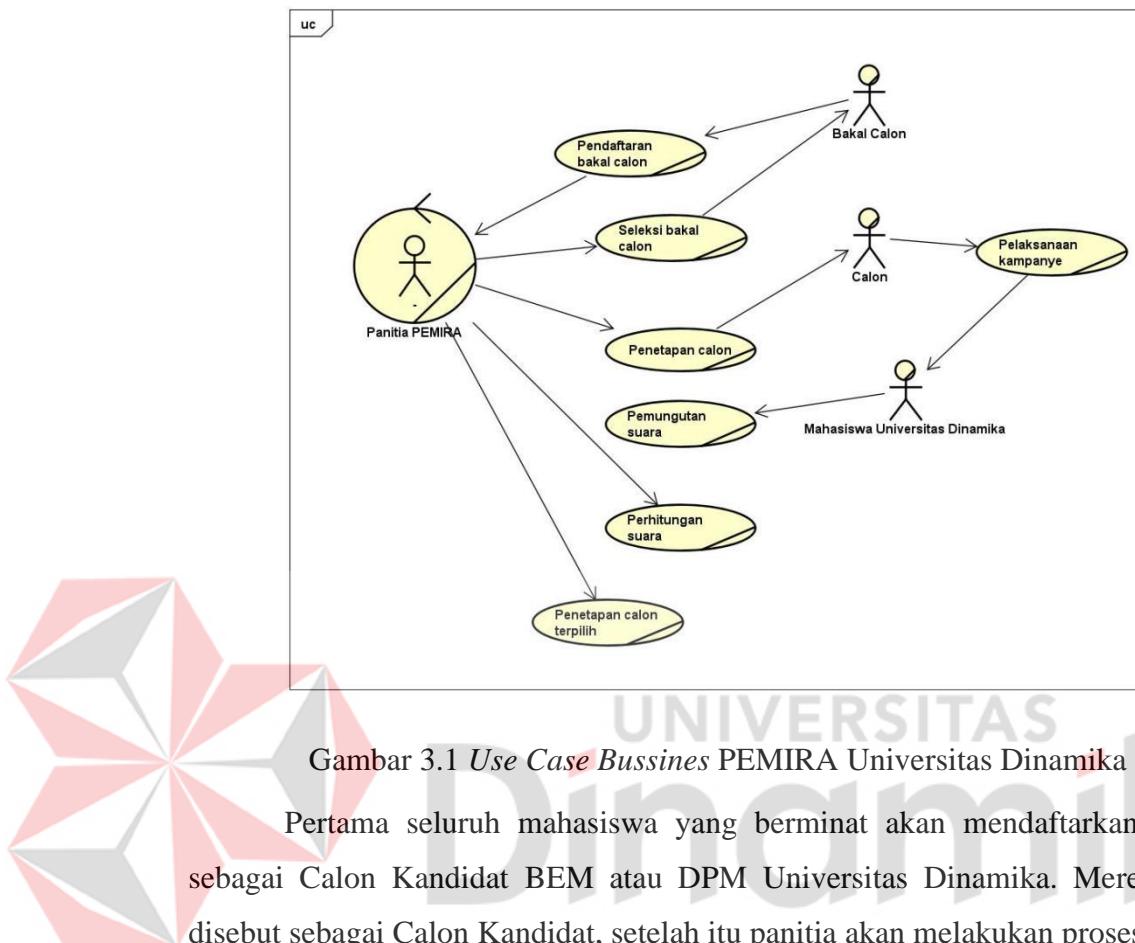
3.1.2 Observasi

Observasi atau pengamatan ini dilakukan pada proses bisnis yang terjadi pada proses PEMIRA di Universitas Dinamika. Penulis juga mengamati website atau aplikasi *e-voting* yang serupa. Hal ini bertujuan untuk bahan perbandingan dan refrensi dalam pembuatan sistem *e-voting* yang sesuai dengan proses bisnis PEMIRA di Universitas Dinamika. Dari hasil observasi mengenai proses bisnis PEMIRA di Universitas Dinamika, penulis membuat sebuah rancangan berupa *use case business diagram*. Dalam *use case business* yang akan dibuat pada proses PEMIRA di Universitas Dinamika, terdapat 7 proses bisnis utama antara lain:

1. Pendaftaran Calon Kandidat
2. Seleksi Calon Kandidat
3. Penetapan Calon
4. Pelaksanaan Kampanye
5. Pemungutan Suara
6. Perhitungan Suara
7. Penetapan Calon Terpilih

Dalam tujuh proses bisnis inti tersebut melibatkan satu *bussines worker* yaitu partai PEMIRA atau KPR dan tiga *bussines actor* yaitu Calon Kandidat, calon dan

peserta voting atau mahasiswa Universitas Dinamika. *Use case bussines* dapat dilihat pada gambar 3.1.



Gambar 3.1 *Use Case Bussines PEMIRA* Universitas Dinamika

Pertama seluruh mahasiswa yang berminat akan mendaftarkan dirinya sebagai Calon Kandidat BEM atau DPM Universitas Dinamika. Mereka akan disebut sebagai Calon Kandidat, setelah itu panitia akan melakukan proses seleksi, proses seleksi antara lain pengecekan dokumen syarat, tes tulis, dan yang terakhir adalah tes wawancara. Setelah mendapatkan penilaian, panitia PEMIRA akan mengumumkan calon calon yang akan maju kebabak selanjutnya, disini Calon Kandidat yang lolos akan disebut sebagai calon. Setelah itu seluruh calon akan melakukan kampanye kepada seluruh mahasiswa, mempromosikan dirinya sebagai calon yang layak menjadi pemimpin. Setelah pelaksanaan kampanye, seluruh mahasiswa Universitas Dinamika atau peserta voting akan melakukan pemungutan suara, setelah itu dilanjutkan dengan penghitungan suara dan penetapan calon terpilih sesuai dengan hasil suara pilihan yang masuk dan sah.

3.1.3 Identifikasi Masalah

Pada tahap ini dilakukan proses identifikasi masalah berdasarkan hasil analisis dan obeservasi. Hasil dari identifikasi masalah ini kemudian diolah kembali

pada proses Pengembangan. Berikut adalah hasil identifikasi masalah dan alternatif solusi yang dapat dilihat pada Tabel 3.2.

Tabel 3.2 Identifikasi Masalah dan Alternatif Solusi

| No | Masalah | Alternatif Solusi |
|----|---|---|
| 1. | Mahasiswa atau pemilih merasa dirugikan karena Proses PEMIRA di Universitas Dinamika yang dilakukan oleh pada <i>Sicyca (System Cyber Campus)</i> terkadang secara tidak langsung memilih paslon yang tidak diinginkan, karena terkejut atau sedang terburu buru untuk melihat jadwal matakuliah. | Membuat sistem pemilihan umum atau <i>e-voting</i> tersendiri dan terpisah dari <i>Sicyca (System Cyber Campus)</i> |
| 2. | Kurangnya minat mahasiswa sebagai pendaftar Calon Kandidat BEM dan DPM pada PEMIRA Universitas Dinamika. | Membuat sebuah sistem pendaftaran secara mandiri dan sistem perekомендasi mahasiswa berpotensi sebagai Calon Kandidat yang dapat diakses oleh kemahasiswaan atau kaprodi |
| 3. | Kurangnya minat para mahasiswa untuk mengikuti PEMIRA sebagai pemilih untuk menggunakan hak suaranya | Membuat sebuah portal kabar berita mengenai PEMIRA yang bisa diakses secara eksklusif oleh mahasiswa, guna meningkatkan antusias mahasiswa untuk menggunakan hak suaranya |
| 4. | KPR merasa kewalahan karena dalam saat proses PEMIRA tahun 2019 terdapat banyak sekali <i>error, bug, and defect</i> . Terdapat 50% dari total keseluruhan fungsionalitas aplikasi e-voting PEMIRA Universitas Dinamika mengalami masalah. sehingga mengurangi rasa percaya mahasiswa atau pemilih terhadap sistem PEMIRA | Menerapkan metode <i>test-driven development</i> dalam pengembangan aplikasi e-voting PEMIRA Universitas Dinamika |

3.2 Tahap Pengembangan

Pada tahap pengembangan terdapat beberapa tahapan, yaitu identifikasi data, identifikasi fungsional, identifikasi pengguna, analisis kebutuhan pengguna, analisis kebutuhan fungsional, analisis kebutuhan non fungsional, desain sistem, TDD, dan implementasi.

3.2.1 Analisis Sistem

Tahap analisis adalah tahapan yang digunakan untuk menguraikan suatu sistem informasi, dimana penulis akan mengidentifikasi dan mengevaluasi sebuah permasalahan sehingga menciptakan sebuah kebutuhan yang diharapkan sehingga dapat menciptakan sebuah solusi. Tahap tahap ini antara lain: Identifikasi Data,

Fungsional, Pengguna, Analisis kebutuhan pengguna, kebutuhan fungsional, dan kebutuhan non fungsional.

1. Identifikasi Data

Berdasarkan hasil observasi dan identifikasi permasalahan maka dapat dilakukan identifikasi kebutuhan data untuk sistem (perangkat lunak) yang akan dibuat. Kebutuhan data tersebut dapat didefinisikan sebagai berikut :

- a. Data Panitia
- b. Data Pengawas
- c. Data Peserta
- d. Data Mahasiswa Berpotensi
- e. Data Voting
- f. Data Calon Kandidat
- g. Data Suara Pilihan Masuk
- h. Data Artikel

2. Identifikasi Fungsional

Berdasarkan hasil observasi dan identifikasi permasalahan maka dapat dilakukan identifikasi kebutuhan fungsional untuk sistem (perangkat lunak) yang akan dibuat. Kebutuhan data tersebut dapat didefinisikan sebagai berikut:

- a. Autentikasi
- b. Perekondasian mahasiswa berpotensi
- c. Pendaftaran Calon Kandidat
- d. Pengelolaan master pengawas
- e. Pengelolaan master panitia
- f. Pengelolaan master peserta
- g. Pengelolaan master voting
- h. Pengelolaan master Calon Kandidat
- i. Pengelolaan master artikel
- j. Aktifasi status voting
- k. Pemilihan Calon Kandidat
- l. Hasil pemilihan sementara
- m. Mencetak laporan voting

3. Identifikasi Pengguna

Berdasarkan hasil observasi dan identifikasi permasalahan, maka dapat dilakukan identifikasi pengguna untuk sistem (perangkat lunak) yang akan dibuat. Identifikasi pengguna tersebut dapat dilihat pada Tabel 3.3.

Tabel 3.3 Identifikasi Pengguna

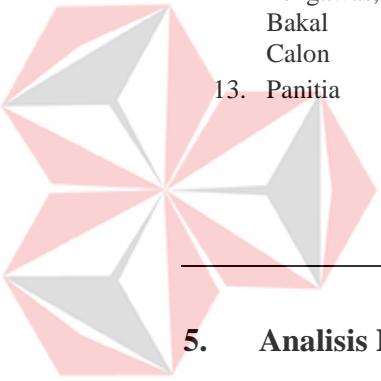
| No | Pengguna | Deskripsi |
|----|----------------|--|
| 1. | Pengawas | Pengawas voting adalah kemahasiswaan atau ketua program studi. Mereka memiliki kewenangan untuk memberikan perekomendasi mahasiswa berpotensi untuk menjadi Calon Kandidat |
| 2. | Calon Kandidat | Calon Kandidat adalah seluruh mahasiswa aktif Universitas Dinamika yang berminat untuk menjadi Calon Kandidat Presiden dan Wakil Presiden BEM dan Ketua DPM |
| 3. | Panitia | Panitia voting adalah KPR (Komisi Pemilihan Raya) Universitas Dinamika yang memiliki kewenangan penuh atas penyelenggaran PEMIRA di Universitas Dinamika |
| 4. | Peserta | Peserta voting adalah seluruh mahasiswa aktif Universitas Dinamika yang memiliki hak suara atas penyelenggaran PEMIRA di Universitas Dinamika |

4. Analisis Kebutuhan Pengguna

Berdasarkan hasil observasi dan identifikasi maka dapat dilakukan analisis kebutuhan pengguna untuk sistem (perangkat lunak) yang akan dibuat. Analisis kebutuhan pengguna tersebut dapat dilihat pada Tabel 3.4.

Tabel 3.4 Analisis Kebutuhan Pengguna

| No | Pengguna | Fungsional | Data | Informasi |
|----|---|--|--|---|
| 1. | Pengawas, Panitia, Calon Kandidat Peserta | Autentikasi | <ul style="list-style-type: none"> • Pengawas • Panitia • Calon Kandidat • Peserta | Mengetahui data pengawas, panitia, Calon Kandidat peserta yang dapat masuk kedalam sistem |
| 2. | Pengawas | Perekomendasi mahasiswa berpotensi | <ul style="list-style-type: none"> • Pengawas • Mahasiswa berpotensi | Mengetahui daftar mahasiswa berpotensi untuk menjadi Calon Kandidat |
| 3. | Calon Kandidat | Pendaftaran Kandidat | <ul style="list-style-type: none"> • Calon Kandidat | Mengetahui data Calon Kandidat |
| 4. | Panitia | Pengelolaan pengawas | <ul style="list-style-type: none"> • Panitia • Pengawas | Mengetahui data master pengawas |
| 5. | Panitia | Pengelolaan panitia | <ul style="list-style-type: none"> • Panitia | Mengetahui data master panitia |



| No | Pengguna | Fungsional | Data | Informasi |
|-----|---|-----------------------------------|--|--|
| 6. | Panitia | Pengelolaan master peserta | <ul style="list-style-type: none"> • Panitia • Peserta | Mengetahui data master peserta |
| 7. | Panitia | Pengelolaan master voting | <ul style="list-style-type: none"> • Panitia • Voting | Mengetahui data master voting |
| 8. | Panitia | Pengelolaan master Calon Kandidat | <ul style="list-style-type: none"> • Panitia • Voting • Calon Kandidat | Mengetahui data master Calon Kandidat |
| 9. | Panitia | Pengelolaan master artikel | <ul style="list-style-type: none"> • Panitia • Artikel | Mengetahui data master artikel |
| 10. | Panitia | Aktifasi status voting | <ul style="list-style-type: none"> • Panitia • Voting • Calon Kandidat | Mengetahui status voting (Periode voting) |
| 11. | Peserta | Pemilihan Calon Kandidat | <ul style="list-style-type: none"> • Peserta • Voting • Calon Kandidat | Mengetahui data Calon Kandidat (nama, visi-misi, pengalaman) |
| 12. | Peserta, Panitia, Pengawas, Bakal Calon | Hasil pemilihan sementara | <ul style="list-style-type: none"> • Voting • Calon Kandidat • Suara pilihan masuk | Mengetahui data suara pilihan masuk |
| 13. | Panitia | Mencetak laporan voting | <ul style="list-style-type: none"> • Panitia • Voting • Calon Kandidat • Suara pilihan masuk | Mengetahui hasil voting, pemenang voting, jumlah suara terkumpul tiap pasangan, jumlah peserta yang melakukan voting |

5. Analisis Kebutuhan Fungsional

Berdasarkan hasil observasi dan identifikasi permasalahan maka dapat dilakukan analisis kebutuhan fungsional untuk sistem (perangkat lunak) yang akan dibuat. Analisis kebutuhan fungsional tersebut dapat dilihat pada Tabel 3.5.

Tabel 3.5 Analisis Kebutuhan Fungsional

| No | Pengguna | Fungsional | Deskripsi |
|----|--|------------------------------------|--|
| 1. | Pengawas, Panitia, Peserta, Calon Kandidat | Autentikasi | Merupakan proses bagi pengawas, panitia, dan peserta untuk masuk ke dalam sistem. Pengawas dan panitia menggunakan email dan password, sedangkan peserta menggunakan token |
| 2. | Pengawas | Perekomendasi mahasiswa berpotensi | Merupakan proses bagi pengawas untuk dapat merekomendasikan mahasiswa berprestasi untuk menjadi Calon Kandidat BEM atau DPM |
| 3. | Calon Kandidat | Pendaftaran Calon Kandidat | Merupakan proses bagi Calon Kandidat untuk mendaftarkan diri secara mandiri |

| No | Pengguna | Fungsional | Deskripsi |
|-----|--|-----------------------------------|--|
| 4. | Panitia | Pengelolaan master pengawas | Merupakan proses pengelolaan data master pengawas voting bagi panitia meliputi <i>create</i> , <i>update</i> , dan <i>delete</i> |
| 5. | Panitia | Pengelolaan master panitia | Merupakan proses pengelolaan data master panitia voting bagi panitia meliputi <i>create</i> , <i>update</i> , dan <i>delete</i> |
| 6. | Panitia | Pengelolaan master peserta | Merupakan proses pengelolaan data master peserta voting bagi panitia meliputi <i>create</i> , <i>update</i> , dan <i>delete</i> |
| 7. | Panitia | Pengelolaan master voting | Merupakan proses pengelolaan data master voting bagi panitia meliputi <i>create</i> , <i>update</i> , dan <i>delete</i> |
| 8. | Panitia | Pengelolaan master Calon Kandidat | Merupakan proses pengelolaan data master Calon Kandidat bagi panitia meliputi <i>create</i> , <i>update</i> , dan <i>delete</i> |
| 9. | Panitia | Pengelolaan data master artikel | Merupakan proses pengelolaan data master artikel bagi panitia meliputi <i>create</i> , <i>update</i> , dan <i>delete</i> |
| 10. | Panitia | Aktifasi status voting | Merupakan proses mengaktifkan dan mengnonaktifkan status voting bagi panitia |
| 11. | Peserta | Pemilihan Calon Kandidat | Merupakan proses memilih Calon Kandidat yang dikehendaki oleh peserta voting |
| 12. | Peserta, Panitia, Pengawas, Calon Kandidat | Hasil pemilihan sementara | Merupakan proses melihat data suara pilihan masuk pada masing-masing Calon Kandidat bagi peserta |
| 13. | Panitia | Mencetak laporan voting | Merupakan proses bagi panitia untuk mencetak hasil voting berdasarkan data suara pilihan masuk |

6. Analisis Kebutuhan Non Fungsional

Berdasarkan hasil observasi dan identifikasi permasalahan maka dapat dilakukan analisis kebutuhan non fungsional untuk sistem (perangkat lunak) yang akan dibuat. Kebutuhan non fungsional tersebut dapat didefinisikan sebagai berikut:

a. Sistem Keamanan

Sistem harus dapat menjaga dan memastikan bahwa data aman dan dilindungi dari pihak yang tidak bertanggung jawab.

- 1) Panitia dan pengawas dapat memasukan informasi login berupa *email* dan *password* untuk dapat masuk kedalam aplikasi (*dashboard e-voting*)
- 2) Peserta Voting dapat memasukan token yang diberikan oleh sistem, untuk masuk kedalam aplikasi dan memulai proses voting. Hal ini memastikan

bahwa hanya peserta yang hanya memiliki hak suara yang dapat melakukan proses voting.

b. Hak Akses

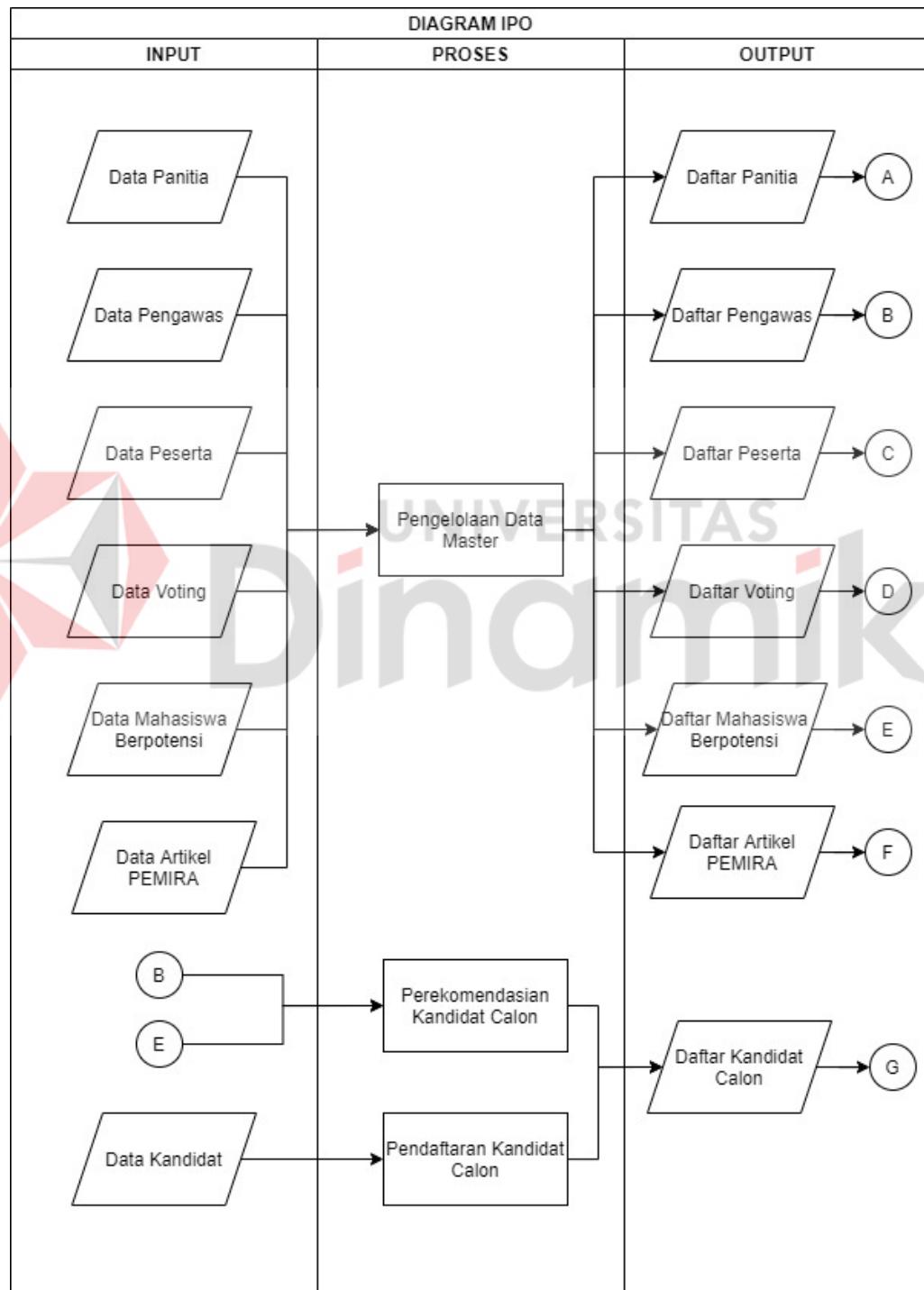
Pengaturan hak akses bertujuan untuk membatasi akses suatu user sesuai dengan fungsi yang telah direncanakan. Adapun pengaturan hak akses pada aplikasi voting adalah sebagai berikut:

Tabel 3.6 Hak Akses

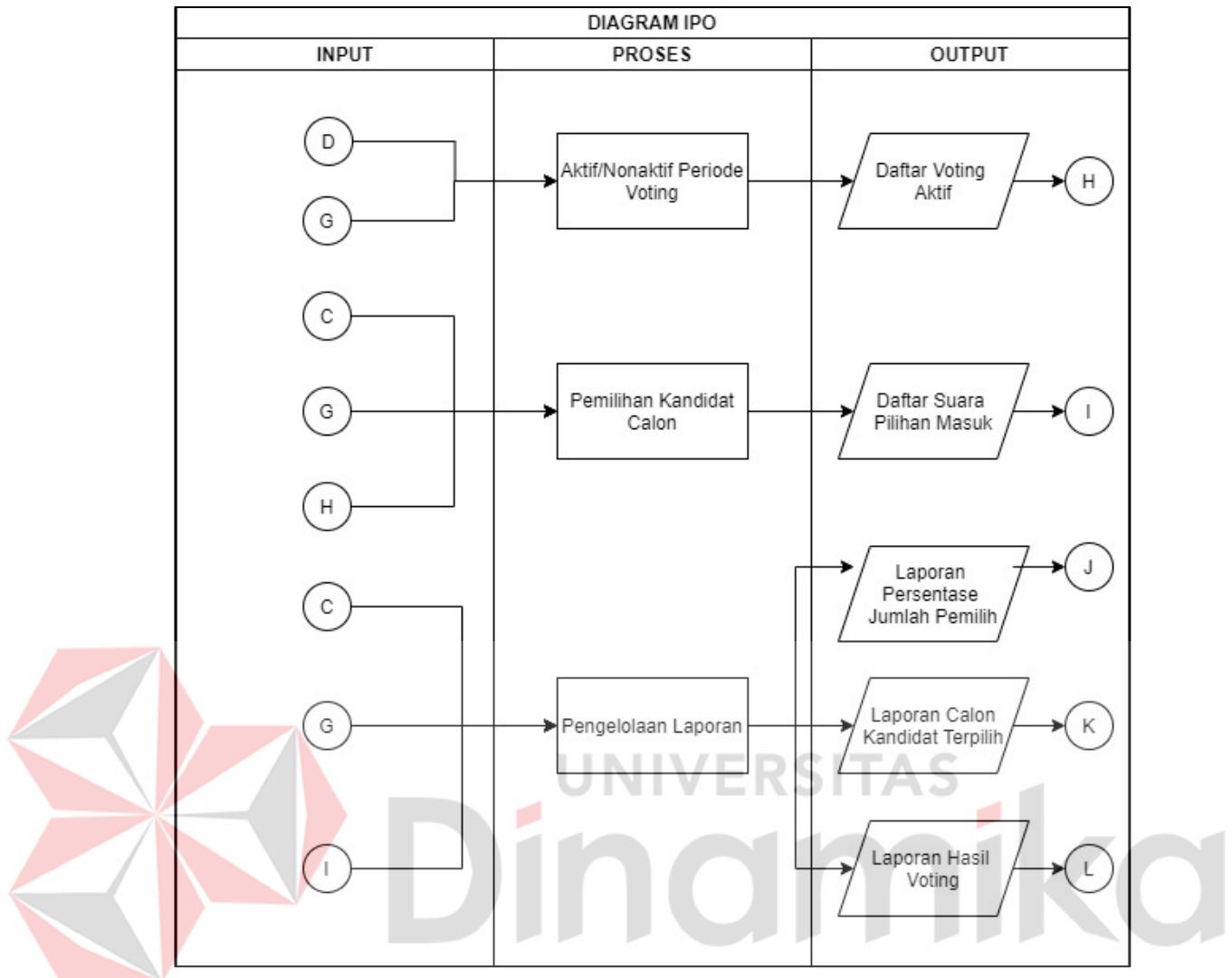
| No | Fungsionalitas | Hak Akses | | | |
|-----|------------------------------------|-----------|------------------------------|----------------|----------------|
| | | Pengawas | Panitia | Peserta Voting | Calon Kandidat |
| 1. | Autentikasi | Read | Create | Read | Read |
| 2. | Perekondasian mahasiswa berpotensi | Create | - | - | - |
| 3. | Pendaftaran Calon Kandidat | - | - | - | Create |
| 4. | Pengelolaan master pengawas | Read | Create, update, delete | Read | - |
| 5. | Pengelolaan master panitia | Read | Read | Create | - |
| 6. | Pengelolaan master peserta | Read | Create, update, delete | - | - |
| 7. | Pengelolaan master voting | Read | Create, update, delete | - | - |
| 8. | Pengelolaan master calon kandidat | Read | Create, update, delete | - | - |
| 9. | Pengelolaan master artikel | Read | Create, update, delete | Read | Read |
| 10. | Aktifasi status voting | Read | Create, update, delete | - | - |
| 11. | Pemilihan calon kandidat | - | - | Create | - |
| 12. | Hasil pemilihan sementara | Read | Read | Read | Read |
| 13. | Mencetak laporan voting | Read | Create, update, delete | - | - |

7. IPO Diagram

Pada tahap selanjutnya akan dilakukan permodelan sistem dari hasil analisis menggunakan permodelan IPO diagram untuk menggambarkan kebutuhan input, proses, dan output dari setiap fungsionalitas. IPO diagram dapat dilihat pada gambar 3.1 dan 3.2



Gambar 3.2 Diagram Input, Proses dan Output (1)

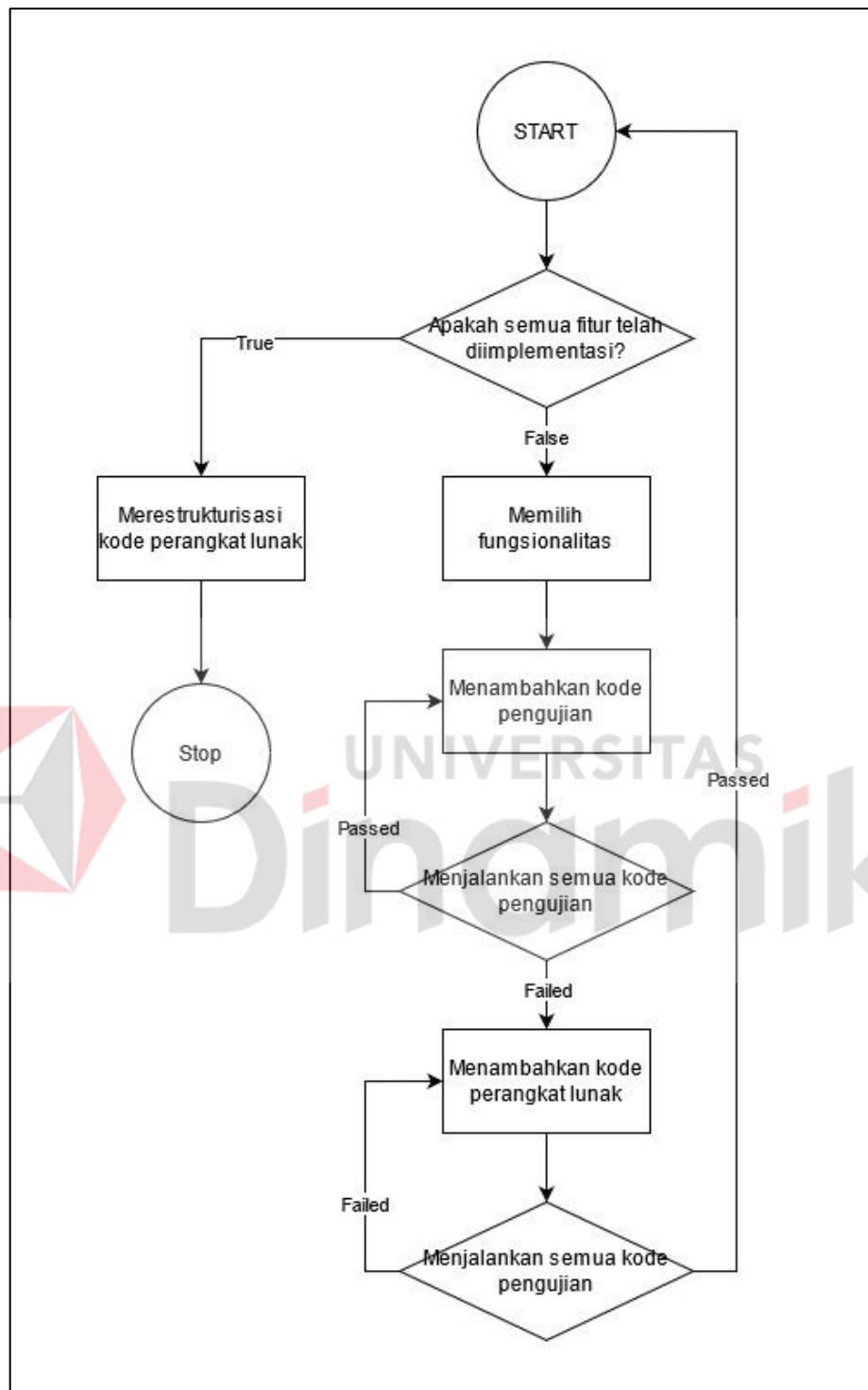


Gambar 3.3 Diagram Input, Proses dan Output (2)

3.2.2 Desain Sistem

Tahap desain adalah tahapan yang digunakan sebagai syarat kebutuhan sistem dalam sebuah perancangan aplikasi e-voting PEMIRA Universitas Dinamika. Pada tahap desain sistem, peneliti menggunakan pendekatan *Object Oriented* dengan menggunakan *use case diagram*, *sequence diagram* dan *class diagram*

3.2.3 Test-Driven Development



Gambar 3.4 Diagram Alir Implementasi TDD Pada Pengembangan Aplikasi E-Voting PEMIRA Universitas Dinamika

Pada proses ini akan dilakukan pembuatan aplikasi *e-voting* PEMIRA Universitas Dinamika, sesuai dengan analisis dan spesifikasi yang telah

direncanakan. Pengembangan aplikasi akan menggunakan Bahasa Pemrograman PHP dengan menggunakan *framework* Laravel Versi 8. Untuk alat pengujian kode perangkat lunak, penulis akan menggunakan *framework* PHPUnit. Langkah kerja implementasi TDD dalam pengembangan aplikasi e-voting PEMIRA Universitas Dinamika dapat dilihat pada Gambar 3.4.

Proses akan diawali dengan melihat apakah semua fungsionalitas pada aplikasi *e-voting* PEMIRA Universitas Dinamika telah berjalan sesuai dengan spesifikasi yang telah direncanakan. Jika belum sesuai, maka langkah awal, penulis akan membuat kode pengujian sesuai fungsionalitas yang ada, menjalankan kode pengujian, hingga semua pengujian mendapatkan keluaran “sukses”. Jika pengujian mendapatkan keluaran “gagal”, maka penulis akan memberikan revisi pada kode perangkat lunak, hingga pengujian mendapatkan keluaran “sukses”. Setelah semua fungsionalitas dibuatkan kode pengujian dan mendapat keluaran “sukses”, langkah berikutnya adalah merestrukturisasi kode perangkat lunak yang telah dibuat. Penulis akan merestrukturisasi kode dengan tujuan menghilangkan kode yang tidak terpakai tanpa mengubah perilaku kode tersebut. Hal ini juga agar kode perangkat lunak menjadi lebih terstruktur, menghilangkan duplikat kode, dan mudah terbaca.

3.3 Tahap Akhir

Pada tahap akhir peniliti akan membuat sebuah laporan penelitian dan dokumentasi aplikasi agar dapat mempermudahkan peneliti dan pengembang selanjutnya untuk melakuakan pemeliharaan dan pengembangan aplikasi e-voting PEMIRA Universitas Dinamika kedepannya.

BAB IV

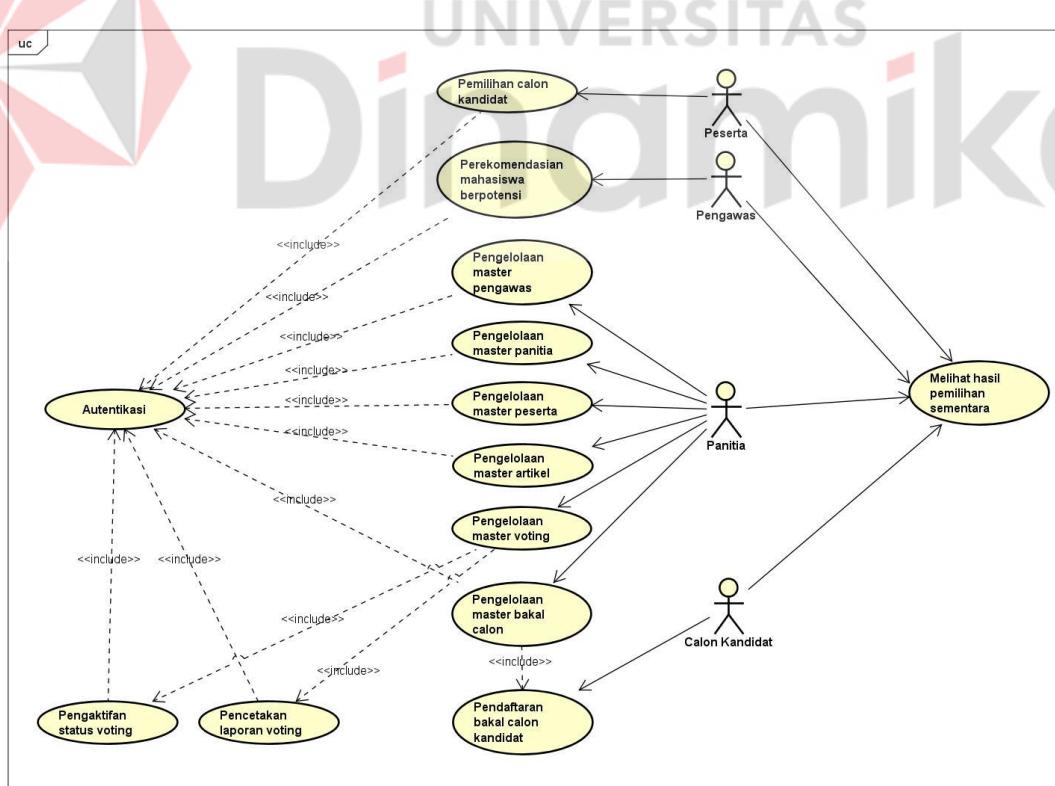
HASIL DAN PEMBAHASAN

4.1 Implementasi Desain Sistem

Pada implementasi desain sistem, telah selesai dilakukan proses analisis dan desain sistem berupa *use case system*, *activity diagram* & *flow of event*, *sequence diagram*, dan *class diagram*. Hasil *activity diagram* & *flow of event* secara detail dapat dilihat pada Lampiran 5. Untuk *sequence diagram* secara detail dapat dilihat pada Lampiran 6.

4.1.1 Use Case System

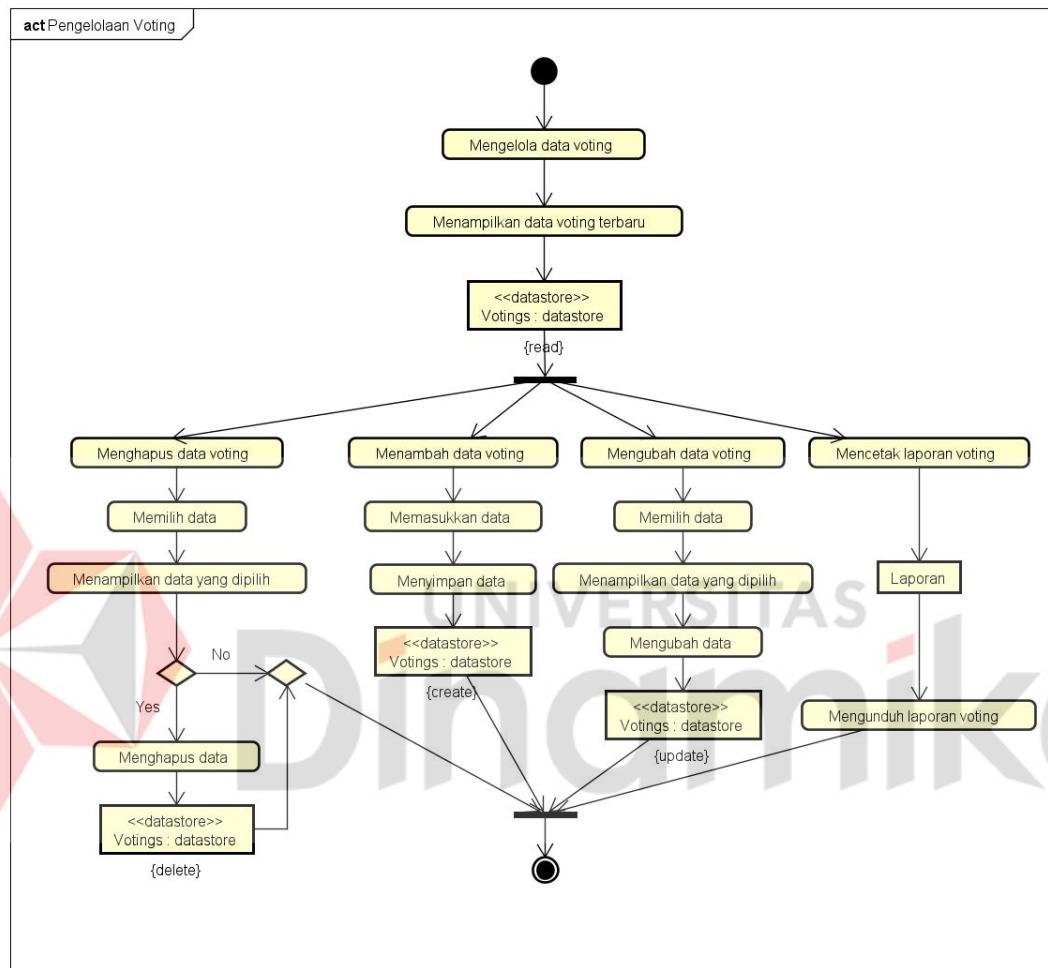
Dalam *use case system* memberikan gambaran tentang interaksi pengguna dengan sistem yang akan dibangun. Berikut ini adalah hasil perancangan berupa *use case system* aplikasi E-Voting PEMIRA Universitas Dinamika dapat dilihat pada Gambar 4.1.



Gambar 4.1 Use Case System E-Voting PEMIRA Universitas Dinamika

4.1.2 Activity Diagram & Flow of Event Pengelolaan Master Voting

Pada *use case* mengelola data voting, panitia dapat melakukan proses tambah data, ubah data, hapus data, dan cetak laporan. Alur dalam proses tersebut dijelaskan melalui *activity diagram* yang dapat dilihat pada Gambar 4.2.



Gambar 4.2 Activity Diagram Pengelolaan data Voting

Berikut ini adalah *flow of event* pengelolaan voting yang dapat dilihat pada Tabel 4.1.

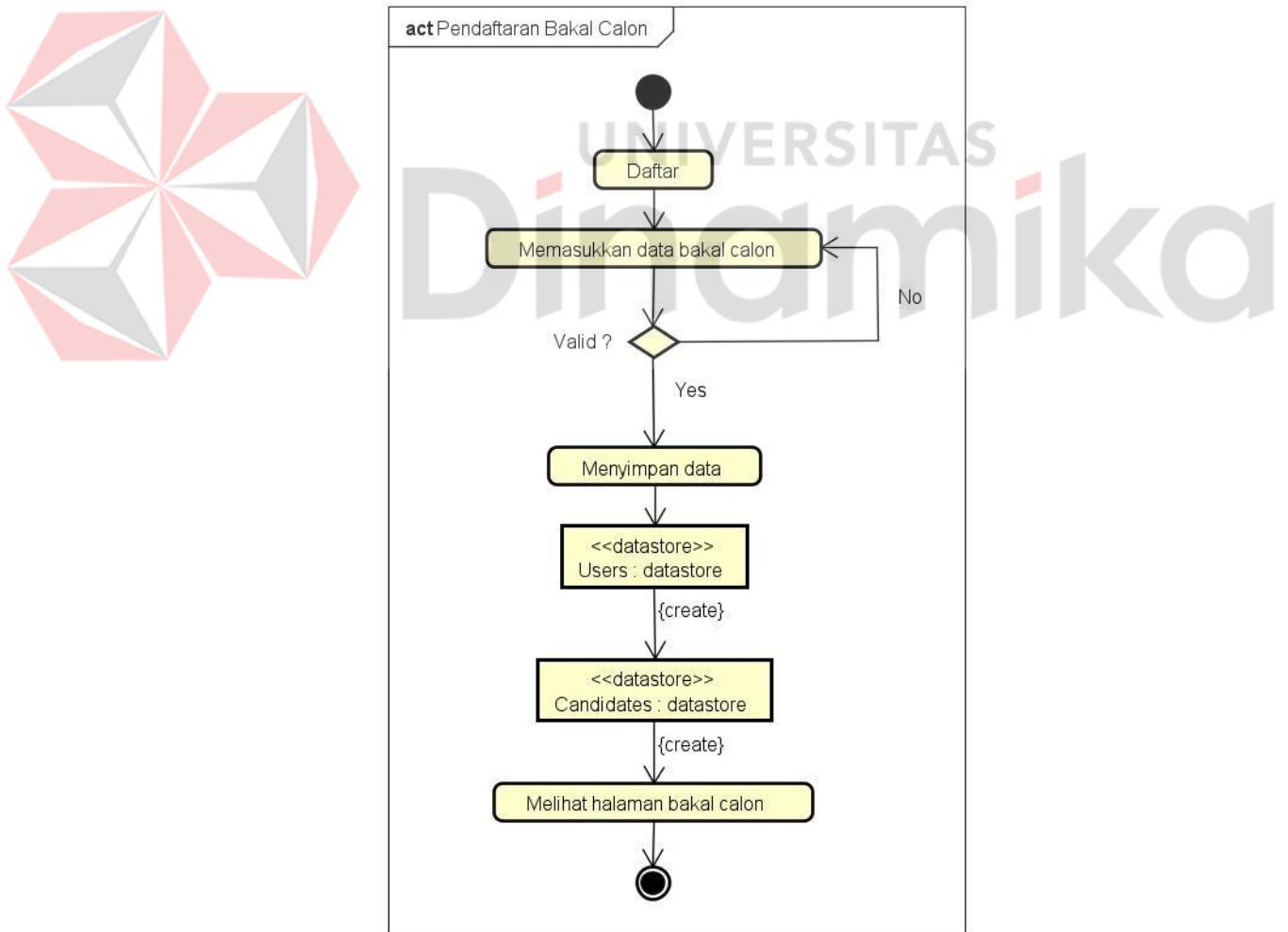
Tabel 4.1 Flow of Event Pengelolaan Voting

| | | |
|------------------------------|--|----------------------------|
| Deskripsi | Pengelolaan data voting | |
| Kondisi awal | Panitia sudah melakukan proses login | |
| Kondisi akhir | Panitia berhasil melakukan aktivitas pengelolaan data voting | |
| Aliran Kejadian Utama | | |
| No. | Panitia | Sistem |
| 1. | Memilih menu voting | Menampilkan halaman voting |

2. Memilih ingin menambah, mengubah, menghapus, atau mencetak data
3. Input data voting, apabila ingin menyimpan atau mengubah data. Klik “oke”, jika ingin menghapus data. Klik “cetak”, jika ingin mencetak laporan
- Jika memilih menambah, maka sistem akan menampilkan dialog “tambah voting”. Jika mengubah, maka sistem akan menampilkan dialog “ubah voting”. Jika menghapus, maka sistem akan menampilkan dialog “hapus data”. Jika mencetak, maka sistem akan menampilkan dialog “cetak data” Sistem memvalidasi, jika benar sistem akan menyimpan data kedalam *database*. Sistem akan menghapus data dalam database sesuai yang dipilih. Sistem akan menampilkan laporan untuk diunduh
-

4.1.3 Activity Diagram & Flow of Event Pendaftaran Calon Kandidat

Pada *use case* mengelola data voting, panitia dapat melakukan proses tambah data, ubah data, hapus data, dan cetak laporan. Alur dalam proses tersebut dijelaskan melalui *activity diagram* yang dapat dilihat pada Gambar 4.3.



Gambar 4.3 Activity Diagram Pendaftaran Calon Kandidat

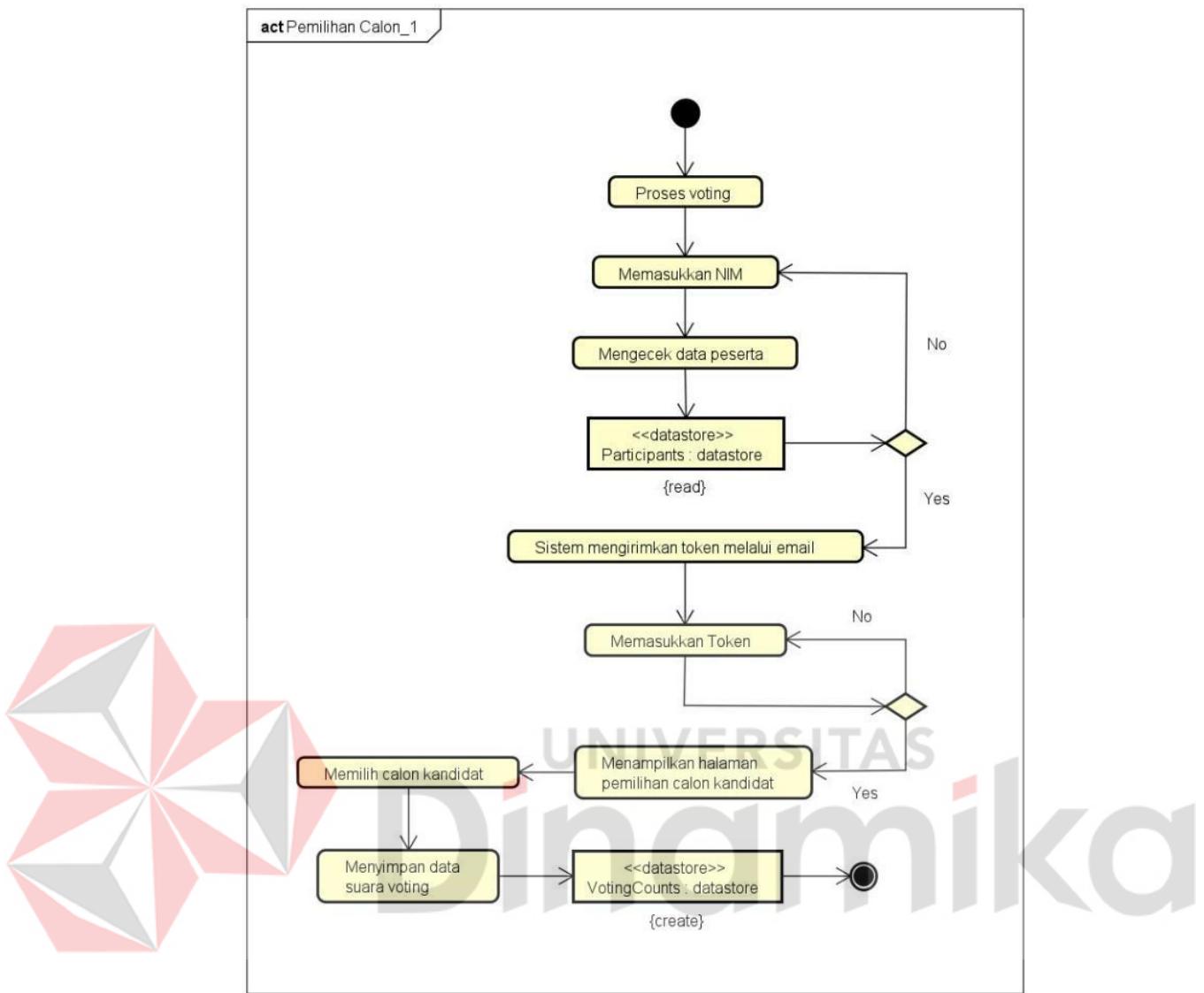
Berikut ini adalah *flow of event* pendaftaran calon kandidat yang dapat dilihat pada Tabel 4.2

Tabel 4.2 *Flow of Event* Pendaftaran Calon Kandidat

| Deskripsi | Pendaftaran Calon Kandidat | |
|-----------------------|---|---|
| Kondisi awal | Calon Kandidat belum memiliki akun yang terdaftar | |
| Kondisi akhir | Calon Kandidat memiliki akun untuk masuk kedalam aplikasi | |
| Aliran Kejadian Utama | | |
| No. | Calon Kandidat | Sistem |
| 1. | Memilih menu pendaftaran | Menampilkan halaman pendaftaran |
| 2. | Memasukan data <i>name</i> , <i>email</i> , <i>password</i> , <i>password confirmation organisasi</i> , dan <i>voting</i> . | Sistem melakukan <i>validasi inputan</i> , jika benar sistem akan menyimpan data dan akun berhasil terdaftar. Jika salah akan Kembali ke halaman pendaftaran dengan <i>notifikasi inputan</i> salah |

4.1.4 *Activity Diagram & Flow of Event* Pemilihan Calon/ Voting

Pada *use case* pemilihan calon atau proses voting, peserta diharuskan melakukan pengecekan data peserta, jika peserta terdaftar dalam daftar peserta voting, sistem akan mengirimkan token untuk peserta dapat masuk kedalam sistem aplikasi *e-voting* PEMIRA Universitas Dinamika, jika tidak terdaftar maka peserta tidak dapat masuk ke dalam sistem. Token akan dikirim melalui *email* kampus mahasiswa Universitas Dinamika sesuai NIM yang dimiliki. Setelah menerima token, peserta akan dapat masuk kedalam sistem dan dapat melakukan proses voting untuk memilih kandidat calon ketua Badan Eksekutif Mahasiswa (BEM) dan Dewan Perwakilan Mahasiswa (DPM). Alur dalam proses tersebut dijelaskan melalui *activity diagram* yang dapat dilihat pada Gambar 4.4.



Gambar 4.4 *Activity Diagram Pemilihan Calon/ Voting*

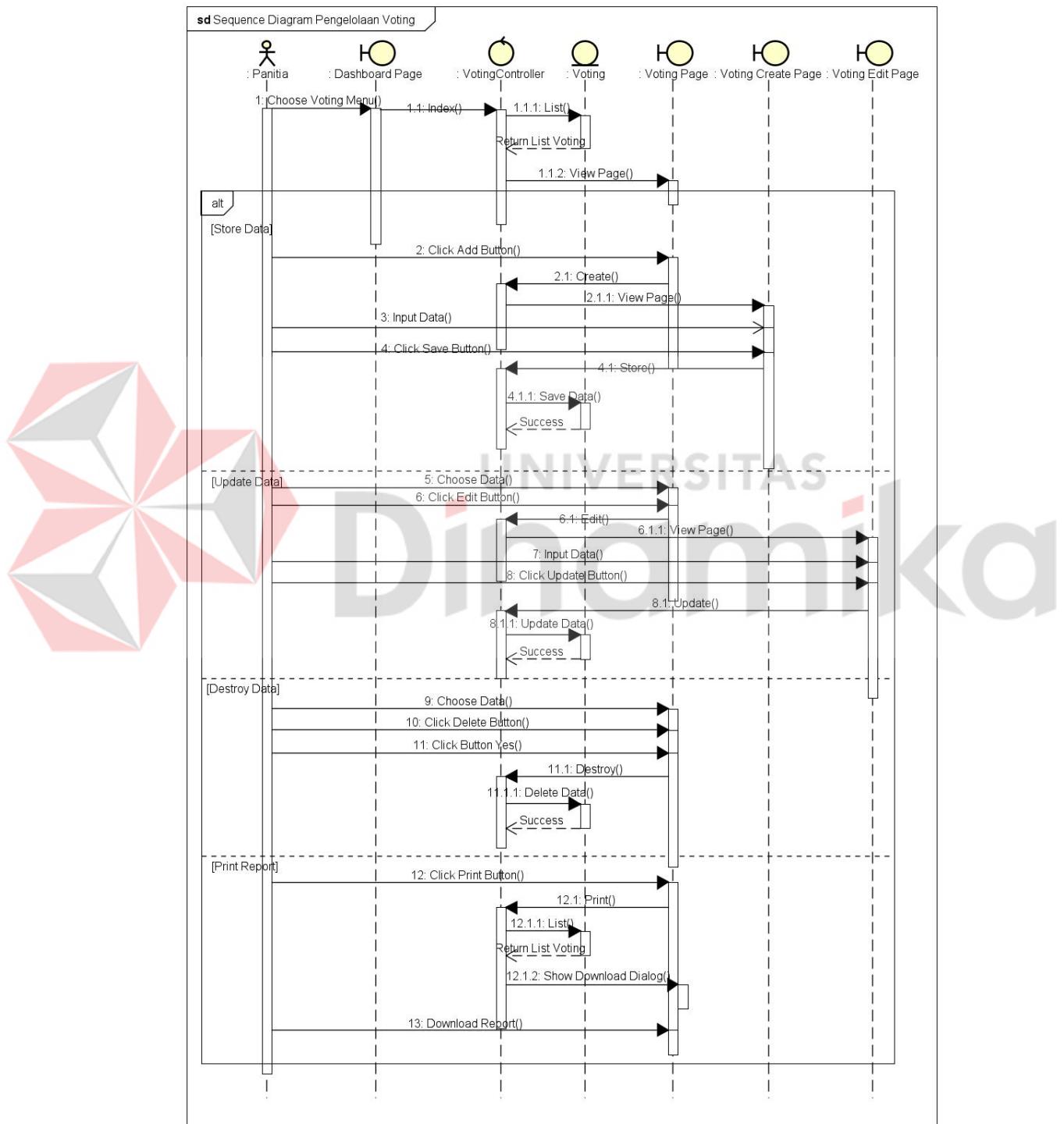
Berikut ini adalah *flow of event* pemilihan calon/voting yang dapat dilihat pada Tabel 4.3

Tabel 4.3 *Flow of Event Pemilihan Calon/ Voting*

| Deskripsi | Pemilihan Calon/ Voting | |
|------------------------------|---|--|
| Kondisi awal | Peserta terdaftar dalam daftar peserta voting | |
| Kondisi akhir | Peserta dapat melakukan proses voting | |
| <u>Aliran Kejadian Utama</u> | | |
| No. | Peserta | Sistem |
| 1. | Memilih menu voting | Menampilkan halaman cek daftar pemilih |
| 2. | Memasukan NIM | Sistem melakukan <i>validasi inputan</i> , jika benar sistem akan membuatkan token untuk masuk kedalam sistem. Jika salah akan kembali ke halaman cek daftar pemilih |

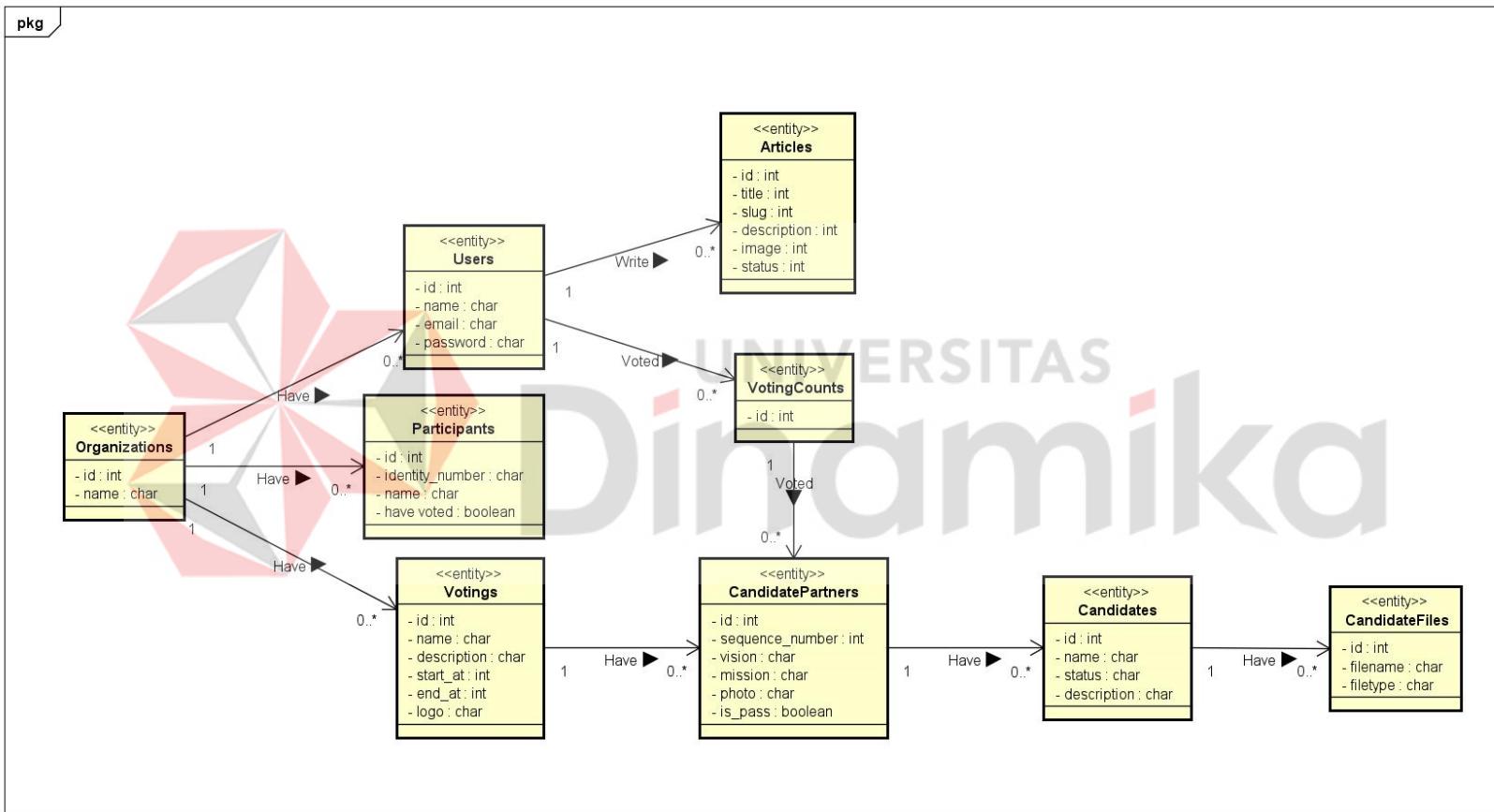
4.1.5 Sequence Diagram Pengelolaan Master Voting

Sequence diagram dibuat untuk menjelaskan secara detail *activity diagram* yang sudah dibuat sebelumnya. *Sequence diagram* untuk *use case* pengelolaan voting. Dapat dilihat pada Gambar 4.5



Gambar 4.5 *Sequence Diagram* Pengelolaan Voting

4.1.6 Class Diagram



Gambar 4.6 Class Diagram

4.2 Implementasi *Test-Driven Development*

Pada implementasi *Test-Driven Development*, telah dilakukan pada proses pengembangan e-voting PERMIRA Universitas Dinamika. Implementasi dilakukan dengan menerapkan langkah-langkah *test-driven development* yaitu (1) *red phase*, (2) *green phase*, dan (3) *refactoring phase*.

4.2.1 Red Phase

Fase *red phase* dimulai dengan melihat desain sistem yang berasal dari hasil analisis sistem. Setelah itu dilanjutkan dengan pembuatan kode pengujian, dan melakukan pengujian menggunakan metode *white box* dengan *automasi testing*.

1. Membuat Kode Pengujian

Kode pengujian dibuat menggunakan PHPUnit dengan *Framework* Laravel. Laravel telah menjadikan PHPUnit sebagai *default tools* untuk melakukan pengujian perangkat lunak.

```
public function test_voting_screen_can_be_rendered()
{
    $data = Organization::factory()
        ->has(User::factory(), 'users')
        ->create();

    $this->actingAs($data->users[0]);
    $this->get('/committee/voting')->assertOk();
}
```

Gambar 4.7 Test Case Daftar Voting

Berikut ini adalah implementasi pembuatan kode pengujian untuk kebutuhan fungsional pengelolaan data master voting. Pengelolaan data master voting mempunyai fitur yaitu melihat *list* data, menyimpan data, mengubah data, dan menghapus data voting. Pada Gambar 4.7 dapat dilihat bahwa terdapat fungsi dengan nama “*test_voting_screen_can_be_rendered*”. Fungsi ini berisi kode

pengujian untuk menguji fungsi menampilkan daftar voting yang tersimpan di database.



```

public function test_voting_create_screen_can_be_rendered()
{
    $data = Organization::factory()
        ->has(User::factory(), 'users')
        ->create();

    $this->actingAs($data->users[0]);
    $this->get('/committee/voting/create')
        ->assertStatus(200);
}

public function test_can_store_voting_data()
{
    Storage::fake('images');
    $data = Organization::factory()
        ->has(User::factory(), 'users')
        ->create();

    $this->actingAs($data->users[0]);
    $this->post('committee/voting', [
        'organization_id' => 1,
        'logo' => UploadedFile::fake()->image('logo.jpg'),
        'name' => "Badan Eksekutif Mahasiswa Vote",
        'description' => 'Lorem ipsum set dolor amet',
        'start_at' => Carbon::now(),
        'end_at' => Carbon::tomorrow(),
    ])->assertStatus(302);
    $data = Voting::query()->where('name', 'Badan Eksekutif Mahasiswa Vote')->first();

    Storage::disk('images')->assertExists('public/images/logo/' . $data->logo);
    $this->assertSame('Badan Eksekutif Mahasiswa Vote', $data->name);
    $this->assertSame('Lorem ipsum set dolor amet', $data->description);
}

public function test_cant_store_voting_with_invalid_data()
{
    Storage::fake('images');
    $data = Organization::factory()
        ->has(User::factory(), 'users')
        ->create();

    $this->actingAs($data->users[0]);
    $this->post('committee/voting', [
        'organization_id' => 1,
        'description' => 'Lorem ipsum set dolor amet',
        'start_at' => Carbon::now(),
        'end_at' => Carbon::tomorrow(),
    ])->assertInvalid([
        'logo', 'name'
    ]);
}

```

Gambar 4.8 *Test Case Tambah Voting*

Pada Gambar 4.8 dapat dilihat bahwa terdapat fungsi dengan nama “*test_voting_create_screen_can_be_rendered*”. Fungsi ini berisi kode pengujian

untuk menguji fungsi menampilkan *form* untuk menambah data voting. Fungsi “*test_can_stored_voting_data*” berisi kode pengujian untuk menguji fungsi menyimpan data voting dengan masukkan yang benar, dan fungsi “*test_can't_store_voting_data_with_invalid_data*” berisi kode pengujian untuk menguji fungsi menyimpan data voting dengan masukkan yang salah, dengan kata lain apakah sistem validasi pada aplikasi dapat bekerja dengan baik.



```
public function test_can_destroy_voting_data()
{
    Storage::fake('images');
    $auth = Organization::factory()
        ->has(User::factory(), 'users')
        ->create();

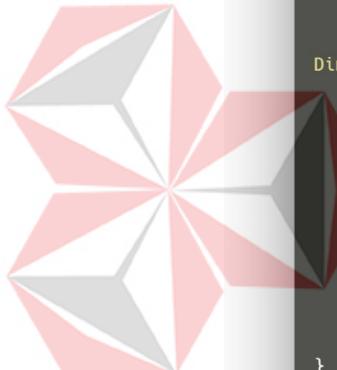
    $voting = Voting::factory()
        ->create();

    $this->actingAs($auth->users[0]);
    $this->assertDatabaseHas('votings', [
        'id' => $voting->id,
    ]);
    $this->delete("/committee/voting/{$voting->id}") ->assertRedirect('/committee/voting');

    $this->assertDatabaseMissing('votings', [
        'id' => $voting->id,
    ]);
}
```

Gambar 4.9 Kode Pengujian Pengelolaan Voting Destroy

Pada Gambar 4.9 dapat dilihat bahwa terdapat fungsi dengan nama “*test_can_destroy_voting*”. Fungsi ini berisi kode pengujian untuk menguji fungsi menghapus data voting yang ada pada database.



```
● ● ●

public function test_voting_edit_screen_can_be_rendered()
{
    $auth = Organization::factory()
        ->has(User::factory(), 'users')
        ->create();

    $voting = Voting::factory()
        ->create();

    $this->actingAs($auth->users[0]);
    $this->get("/committee/voting/{$voting->id}/edit")
        ->assertOk();
}

public function test_can_update_voting_data()
{
    $auth = Organization::factory()
        ->has(User::factory(), 'users')
        ->create();

    $voting = Voting::factory()
        ->create();

    $data = [
        'name' => 'PEMIRA BEM DAN DPM',
        'description' => 'Pemilihan ketua BEM dan DPM Universitas
Dinamika',
        'start_at' => Carbon::now(),
        'end_at' => Carbon::now()->addDays(3)
    ];

    $this->actingAs($auth->users[0]);
    $this->put("/committee/voting/{$voting->id}", $data)
        ->assertSessionHasNoErrors('name', 'description')
        ->assertRedirect('/committee/voting');

    $this->assertDatabaseHas('votings', [
        'name' => $data['name'],
        'description' => $data['description'],
    ]);
}

public function test_cant_update_voting_with_invalid_data()
{
    $auth = Organization::factory()
        ->has(User::factory(), 'users')
        ->create();

    $voting = Voting::factory()
        ->create();

    $data = [
        'name' => 'vote',
        'description' => 12312312312,
        'start_at' => 'hari ini',
        'end_at' => 'besok'
    ];

    $this->actingAs($auth->users[0]);
    $this->put("/committee/voting/{$voting->id}", $data)
        ->assertSessionHasErrors([
            'name', 'description', 'start_at', 'end_at'
        ]);
}
```

Gambar 4.10 Kode Pengujian Pengelolaan Voting Update

Pada Gambar 4.10 dapat dilihat bahwa terdapat fungsi dengan nama “*test_voting_edit_screen_can_be_rendered*”. Fungsi ini berisi kode pengujian untuk menguji fungsi menampilkan halaman *form* ubah data voting. Fungsi “*test_can_update_voting_data*” berisi kode pengujian untuk menguji fungsi mengubah data voting dengan masukkan yang benar, dan fungsi “*test_can't_update_voting_data_with_invalid_data*” berisi kode pengujian untuk menguji fungsi mengubah data voting dengan masukkan yang salah, dengan kata lain apakah sistem validasi pada aplikasi dapat bekerja dengan baik.

2. Pengujian Perangkat Lunak dengan Keluaran *Failed*

Setelah membuat rangkaian kode pengujian, langkah berikutnya adalah menjalankan kode pengujian tersebut. Berikut ini adalah daftar fungsionalitas yang akan dilakukan proses pengujian pada fase *green phase* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Implementasi TDD Pada Fase Red Phase

| No | Fungsional yang Akan diuji | Hasil |
|-----|------------------------------------|--------------|
| 1. | Autentikasi | Gambar L7.1 |
| 2. | Perekomendasi Mahasiswa Berpotensi | Gambar L7.2 |
| 3. | Pendaftaran Calon Kandidat | Gambar L7.3 |
| 4. | Pengelolaan Master Pengawas | Gambar L7.4 |
| 5. | Pengelolaan Master Panitia | Gambar L7.5 |
| 6. | Pengelolaan Master Peserta | Gambar L7.6 |
| 7. | Pengelolaan Master Voting | Gambar 4.11 |
| 8. | Pengelolaan Master Calon Kandidat | Gambar L7.7 |
| 9. | Pengelolaan Master Artikel | Gambar L7.8 |
| 10. | Aktifasi Status Voting | Gambar L7.9 |
| 11. | Pemilihan Calon Kandidat/ Voting | Gambar L7.10 |
| 12. | Hasil Pemilihan Sementara | Gambar L7.11 |
| 13. | Mencetak Laporan Voting | Gambar L7.12 |

Pada *red phase* hasil dari menjalankan kode pengujian adalah “*failed*”, karena penulis belum membuat kode perangkat lunak. Pembuatan kode perangkat lunak akan mengikuti panduan *test case* yang dibuat pada kode pengujian. Penulis akan membuat kode perangkat lunak dan kembali menjalankan kode pengujian hingga kode pengujian mengeluarkan hasil “*success*”. Berikut ini adalah hasil pengujian fase *red phase* pada fungsional pengelolaan master data voting dapat dilihat pada Gambar 4.11. Detail dari proses menjalankan kode pengujian dapat dilihat pada Lampiran 7.

```

semba@DESKTOP-4QTCC29 MINGW64 /c/laragon/www/votez (red-phase-voting)
$ php artisan test
Warning: TTY mode is not supported on Windows platform.

  FAIL  Tests\Feature\Committee\Voting\DestroyVotingTest
    ✘ can destroy user data

  FAIL  Tests\Feature\Committee\Voting\IndexVotingTest
    ✘ voting screen can be rendered

  FAIL  Tests\Feature\Committee\Voting\StoreVotingTest
    ✘ voting create screen can be rendered
    ✘ can store voting data
    ✘ cant store voting with invalid data

  FAIL  Tests\Feature\Committee\Voting\UpdateVotingTest
    ✘ voting edit screen can be rendered
    ✘ can update voting data
    ✘ cant update voting with invalid data

---

```

Gambar 4.11 *Red Phase* - Hasil Uji Coba Kode Pengujian

4.2.2 *Green Phase*

Fase *Green Phase* dimulai dengan membuat kode perangkat lunak sesuai dengan kode pengujian yang telah dibuat diawal. Fase ini akan berakhir ketika kode perangkat lunak yang telah kita buat memenuhi kode pengujian, dengan kata lain saat melakukan pengujian perangkat lunak, hasil yang akan didapat harus menerima keluaran “*success*”.

1. Pembuatan Kode Perangkat Lunak

Pembuatan kode perangkat lunak menggunakan bahasa pemrograman PHP dan *framework* Laravel. Pembuatan fungsionalitas mengacu pada setiap *test case* yang telah dibuat pada kode pengujian diawal.

2. Pengujian Perangkat Lunak dengan Keluaran Success

Setelah membuat rangkaian kode perangkat lunak, langkah berikutnya adalah menjalankan kode pengujian tersebut. Berikut ini adalah daftar fungsionalitas yang akan dilakukan proses pengujian pada fase *green phase* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Implementasi TDD Pada Fase *Green Phase*

| No | Fungsional yang Akan diuji | Hasil |
|-----|------------------------------------|--------------|
| 1. | Autentikasi | Gambar L8.1 |
| 2. | Perekondesian Mahasiswa Berpotensi | Gambar L8.2 |
| 3. | Pendaftaran Calon Kandidat | Gambar L8.3 |
| 4. | Pengelolaan Master Pengawas | Gambar L8.4 |
| 5. | Pengelolaan Master Panitia | Gambar L8.5 |
| 6. | Pengelolaan Master Peserta | Gambar L8.6 |
| 7. | Pengelolaan Master Voting | Gambar 4.12 |
| 8. | Pengelolaan Master Calon Kandidat | Gambar L8.7 |
| 9. | Pengelolaan Master Artikel | Gambar L8.8 |
| 10. | Aktifasi Status Voting | Gambar L8.9 |
| 11. | Pemilihan Calon Kandidat/ Voting | Gambar L8.10 |
| 12. | Hasil Pemilihan Sementara | Gambar L8.11 |
| 13. | Mencetak Laporan Voting | Gambar L8.12 |

Pada *green phase* hasil dari menjalankan kode pengujian harus sampai mengeluarkan hasil “*success*”. Hal ini menandakan semua fungsionalitas dalam proses pengkodean telah berhasil dibuat seluruhnya. Berikut ini hasil menjalankan kode pengujian pada fungsionalitas pengelolaan master data voting dapat dilihat pada gambar 4.12. Detail dari proses menjalankan kode pengujian dapat dilihat pada Lampiran 8.

```
semba@DESKTOP-4QTCC29 MINGW64 /c/laragon/www/votez (green-phase-voting)
$ php artisan test
Warning: TTY mode is not supported on Windows platform.

    PASS Tests\Feature\Committee\Voting\DestroyVotingTest
    ✓ can destroy user data

    PASS Tests\Feature\Committee\Voting\IndexVotingTest
    ✓ voting screen can be rendered

    PASS Tests\Feature\Committee\Voting\StoreVotingTest
    ✓ voting create screen can be rendered
    ✓ can store voting data
    ✓ cant store voting with invalid data

    PASS Tests\Feature\Committee\Voting\UpdateVotingTest
    ✓ voting edit screen can be rendered
    ✓ can update voting data
    ✓ cant update voting with invalid data

Tests: 8 passed
Time: 5.83s
```

Gambar 4.12 *Green Phase* - Hasil Uji Coba Kode Pengujian

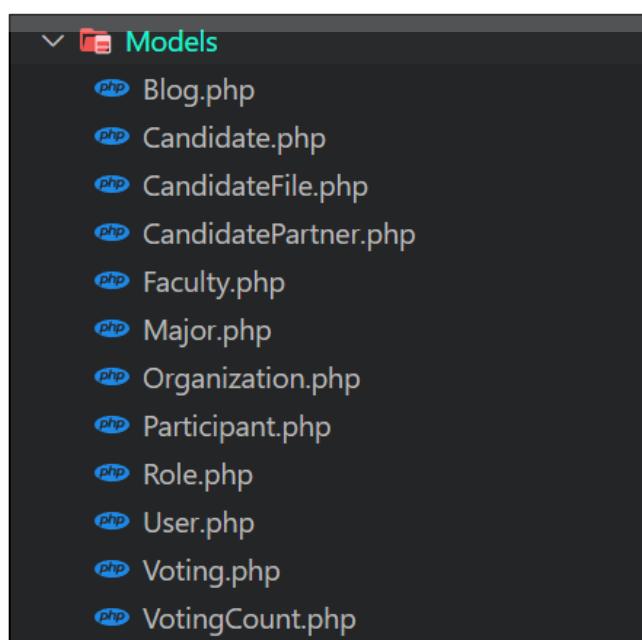
4.2.3 Refactoring Phase

Fase ini dimulai dengan merestrukturisasi kode perangkat lunak yang telah dibuat menjadi kode yang terstruktur dan mudah dibaca, tanpa merubah hasil atau keluaran hasil pengujian. Aplikasi *E-Voting PEMIRA* Universitas Dinamika dibuat menggunakan *framework* Laravel dengan *Repositori Pattern*. Sehingga struktur kode mudah dibaca, Berikut ini adalah struktur dari kode perangkat lunak Aplikasi *E-Voting PEMIRA* Universitas Dinamika:

Tabel 4.6 Struktur Kode Perangkat Lunak

| No | Folder | Deskripsi |
|----|---------------------|--|
| 1. | <i>Models</i> | Berisi kode yang berhubungan dengan tabel database |
| 2. | <i>Views</i> | Berisi kode yang berhubungan dengan desain antarmuka pengguna |
| 3. | <i>Controllers</i> | Berisi kode yang berhubungan dengan <i>bussines logic</i> |
| 4. | <i>Repositories</i> | Berisi kode yang berhubungan dengan <i>query logic</i> |
| 5. | <i>Routes</i> | Berisi kode yang berhubungan dengan pendaftaran <i>route</i> atau <i>url</i> |
| 6. | <i>Tests</i> | Berisi kode yang berhubungan dengan <i>use cases</i> pengujian |

Struktur *folder* model berisi file kode sumber yang berhubungan dengan tabel pada *database*. Setiap tabel pada *database* akan memiliki file model yang berbeda beda yang digunakan untuk berinteraksi dengan tabel tersebut (Otwell, 2022). Model memungkinkan untuk membuat sebuah berbagai macam *query*. Penjelasan detail masing masing *folder* dapat dilihat pada Lampiran 9



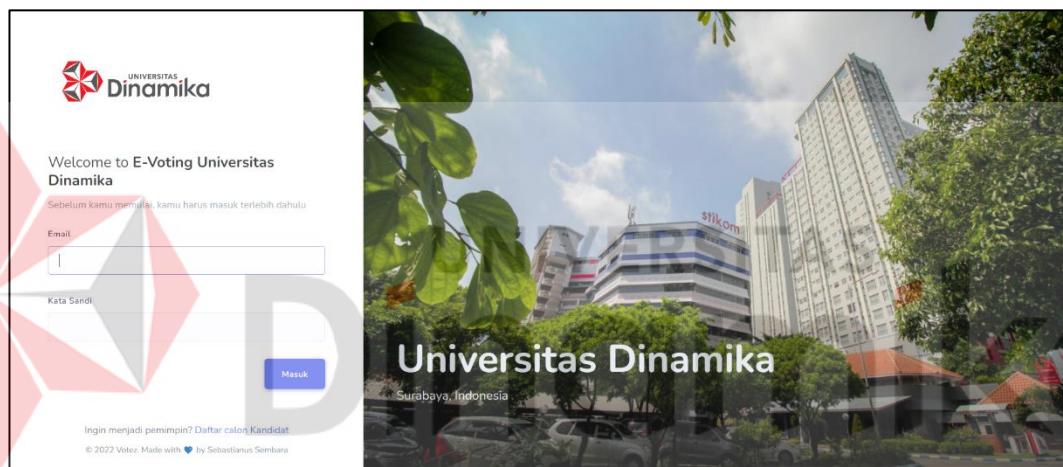
Gambar 4.13 Struktur *Folder* Model

4.3 Implementasi Sistem

Implementasi sistem dibuat sesuai dengan hasil analisis dan perancangan dengan mengikuti panduan kode pengujian yang telah dibuat. Berikut ini beberapa hasil implementasi sistem aplikasi *e-voting* PEMIRA Universitas Dinamika. Implementasi lebih detail diuraikan pada lampiran 8

4.3.1 Autentikasi

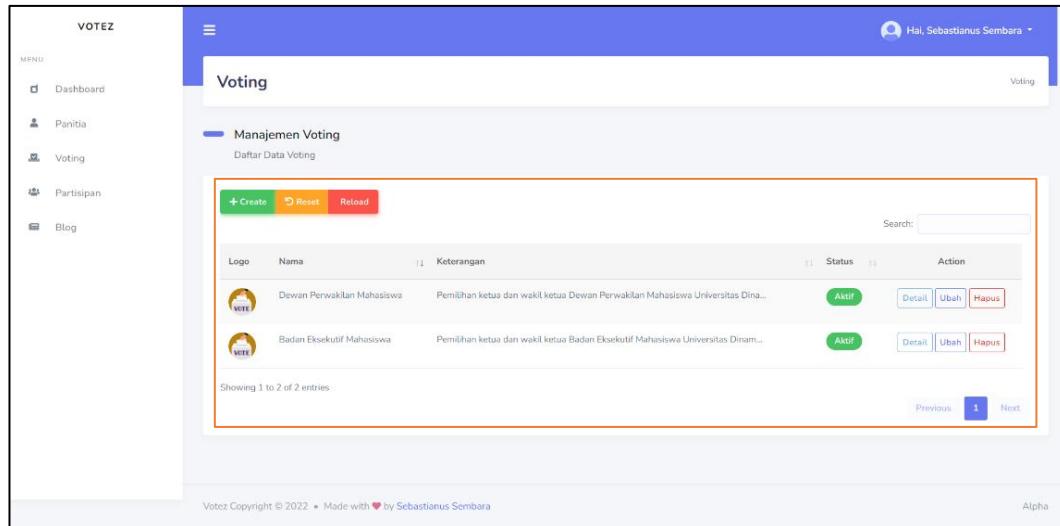
Untuk dapat mengakses aplikasi, setiap *user* harus melakukan autentikasi terlebih dahulu dengan memasukkan inputan *email* dan *password*. Halaman autentikasi digunakan untuk *user* yang mempunyai *role* pengawas, panitia, dan calon kandidat. Berikut ini adalah tampilan *login* pada Gambar 4.13.



Gambar 4.14 Halaman Autentikasi

4.3.2 Halaman Pengelolaan Master Voting

Berikut ini merupakan halaman pengelolaan master voting. Halaman ini digunakan oleh panitia untuk melakukan pengelolaan data master voting. Pada halaman voting, panitia dapat mengatur segala tentang voting. Mulai dari melihat daftar voting yang aktif dan tidak aktif, menambah data voting, mengubah data voting, dan menghapus data voting. Berikut ini adalah tampilan halaman pengelolaan data master voting.

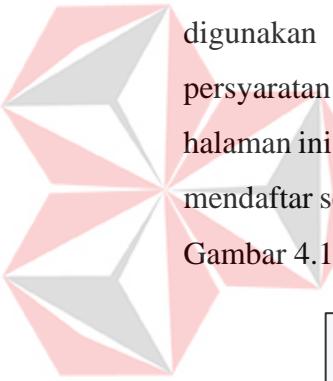


Gambar 4.15 Halaman Voting

Untuk melakukan proses tambah data voting, panitia harus menekan tombol *create*, lalu akan muncul halaman *form* tambah data voting, yang terdiri dari data logo, nama, deskripsi, tanggal mulai, dan tanggal berakhir voting. Berikut tampilan halaman *form* tambah data voting.

Gambar 4.16 Halaman Form Tambah Data Voting

Untuk melakukan proses ubah data voting, panitia harus dapat tombol ubah, lalu akan muncul halaman *form* ubah data voting. Berikut tampilan halaman *form* ubah data voting.



VOTEZ

MENU

- Dashboard
- Panitia
- Voting**
- Partisipan
- Blog

Logo Voting

Pilih File Tidak ada file yang dipilih

Nama Voting

Deskripsi

Pemilihan ketua dan wakil ketua Dewan Perwakilan Mahasiswa Universitas Dinamika

Tanggal Pelaksanaan

Tanggal Berakhir

Ubah **Kembali**

Gambar 4.17 Halaman *Form Ubah Data Voting*

4.3.3 Halaman Pendaftaran Calon Kandidat

Berikut ini merupakan halaman pendaftaran calon kandidat. Halaman ini digunakan Calon Kandidat untuk melakukan pendaftaran dan melengkapi persyaratan pendaftaran calon kandidat PEMIRA Universitas Dinamika. Pada halaman ini terdapat inputan Nama, Email, Kata Sandi, Organisasi dan pilihan ingin mendaftar sebagai calon kandidat di suatu voting. Detail halaman dapat dilihat pada Gambar 4.12.

Daftar Calon Kandidat

Nama

Email

Kata Sandi

Ulangi Kata Sandi

Organisasi

Voting

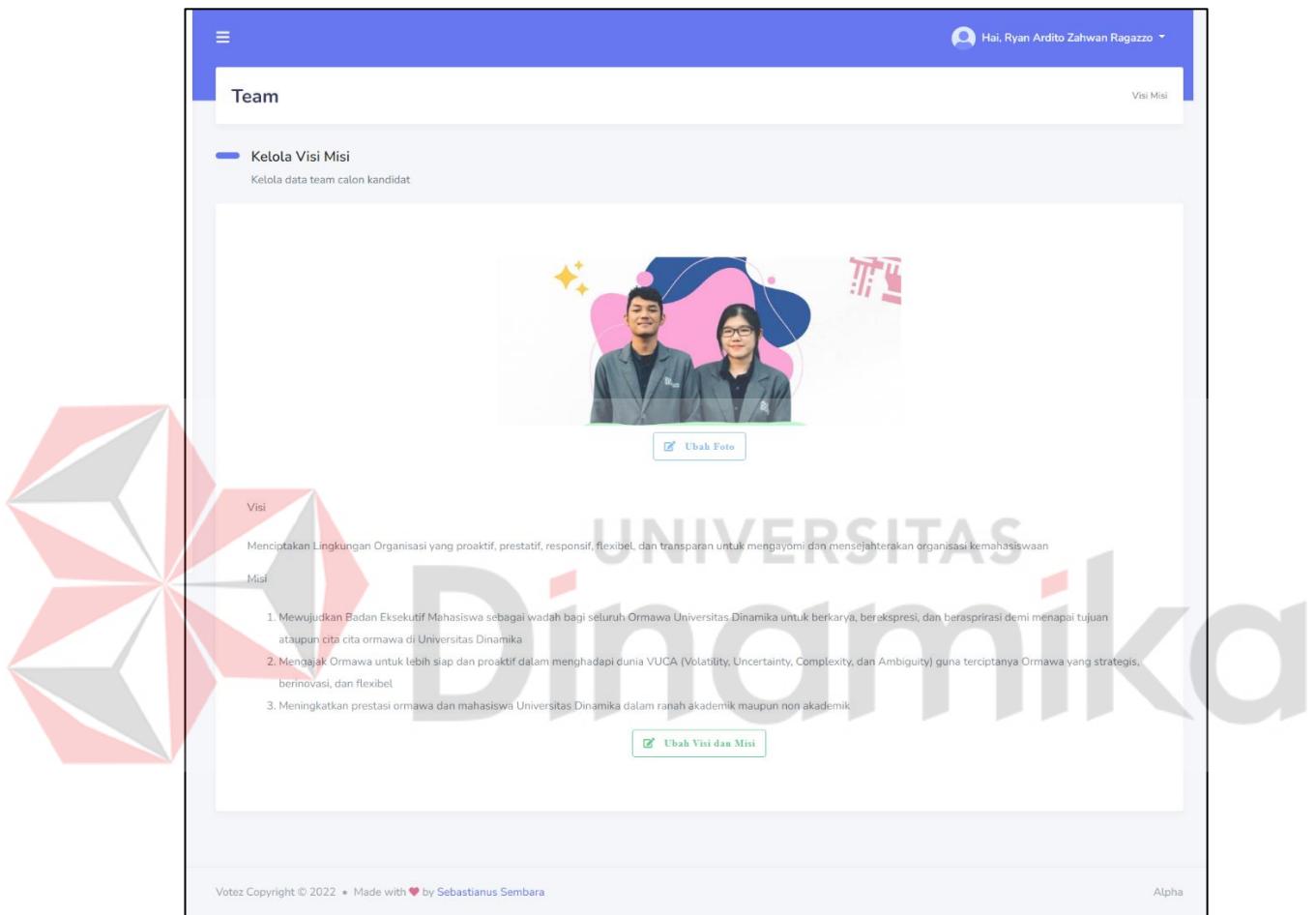
Pilih Komunitas

Register

© 2022 Votez. Made with ❤ by Sebastianus Sembara

Gambar 4.18 Halaman Pendaftaran Calon Kandidat

Setelah calon kandidat mendaftarkan akun, dia harus melengkapi data diri (personal) dan data tim (pasangan calon) yang menjadi persyaratan untuk menjadi seorang calon kandidat suatu voting. Data tim atau pasangan calon terdiri dari data visi dan misi calon kandidat, serta foto calon kandidat. Detail halaman pengelolaan data tim dapat dilihat pada Gambar 4.18



Gambar 4.19 Halaman Mengelola Visi dan Misi

Data personal adalah data diri setiap pasangan calon. Data tersebut antara lain adalah nama, nim, email, nomor *handphone*, jenis kelamin, alamat, tempat lahir, tanggal lahir, fakultas, program studi, semester, ipk terakhir, dan jumlah point sskm. Pada halaman ini juga dapat mengunggah berkas berkas persyaratan seperti surat keterangan aktif, transkrip nilai, sertifikat LKMM-TD, surat keterangan aktif organisasi, pengalaman organisasi (sk-aktif), dan surat bukti koalisi ormawa. Detail halaman pengelolaan data personal dapat diliha pada Gambar 4.19



Kelola Kandidat
Ubah data personal calon kandidat

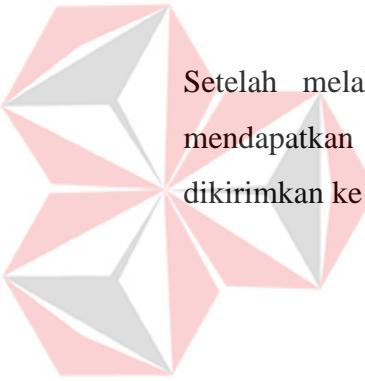
| | |
|--|--|
| Nama <input type="text" value="Ryan Ardito Zahwan Ragazzo"/> | Nim <input type="text" value="-"/> |
| Email <input type="text" value="-"/> | No. Handphone <input type="text" value=""/> |
| Voting <input type="text" value="Badan Eksekutif Mahasiswa"/> | Jabatan <input type="text" value="KETUA"/> |
| Jenis Kelamin <input type="radio"/> Laki Laki <input type="radio"/> Perempuan | Alamat <input type="text" value="-"/> |
| Tempat Lahir <input type="text" value="-"/> | Tanggal Lahir <input type="text" value="hh/bb/tttt"/> |
| Fakultas <input type="text" value="FAKULTAS TEKNOLOGI DAN INFORMATIKA"/> | Program Studi <input type="text" value="-"/> |
| Semester Sekarang <input type="text" value=""/> | IPK <input type="text" value=""/> |
| SSKM <input type="text" value=""/> | |

Ubah Kembali

Gambar 4.20 Halaman Mengelola Data Pasangan Calon

4.3.4 Halaman Pemilihan Calon Kandidat/ Voting

Berikut ini merupakan halaman pemilihan calon kandidat atau proses voting. Halaman ini digunakan partisipan atau peserta voting untuk melakukan proses voting atau pemilihan Calon Kandidat Ketua dan Wakil Ketua Badan Eksekutif Mahasiswa dan Ketua Dewan Perwakilan Mahasiswa Universitas Dinamika. Pertama peserta voting harus melakukan pengecekan data pada aplikasi *e-voting* dengan memasukkan NIM. Pengecekan ini bertujuan untuk memastikan bahwa peserta yang dapat masuk kedalam aplikasi adalah peserta yang terdaftar dan mempunya hak suara dalam proses voting. Detail halaman pengecekan peserta terdapat pada Gambar 4.20



VOTEZ

Cek Data Peserta

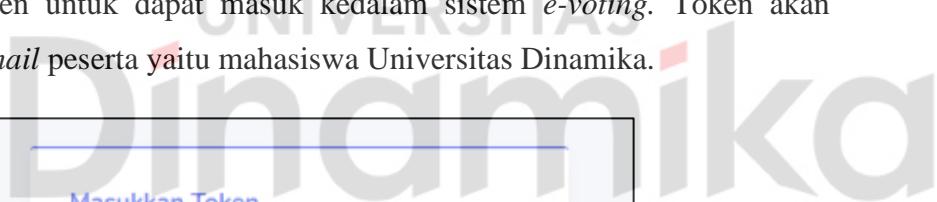
Nomor Identitas

Pilih Komunitas

Votez Copyright © 2022

Gambar 4.21 Halaman Pengecekan Peserta Voting

Setelah melakukan pengecekan data peserta, peserta yang terdaftar akan mendapatkan token untuk dapat masuk kedalam sistem *e-voting*. Token akan dikirimkan ke *e-mail* peserta yaitu mahasiswa Universitas Dinamika.



Masukkan Token

Silahkan Cek Email!, Sistem Mengirimkan Token Untuk Masuk Kedalam Sistem

NIM

Token

Belum memiliki token?

Gambar 4.22 Halaman Masuk Peserta

Setelah mendapatkan token, peserta voting akan dihadapkan dengan halaman voting dan dapat memilih kandidat ketua Badan Eksekutif Mahasiswa (BEM) dan Dewan Perwakilan Mahasiswa (DPM)



Gambar 4.23 Halaman Pemilihan Kandidat/ Voting

4.4 Black Box Testing

Pada tahap *alpha testing* dilakukan pengujian pada aplikasi e-voting PEMIRA Universitas Dinamika menggunakan metode *black box testing* yaitu mencoba keseluruhan fungsional dari aplikasi yang dikembangkan. Hasil keseluruhan *black box testing* pada keseluruhan fungsional aplikasi berbasis website 100 % berhasil atau sukses. Untuk lebih jelasnya hasil pengujian dapat dilihat pada lampiran 9.

4.4.1 Halaman Autentikasi

Pengawas, panitia dan Calon Kandidat dapat mengakses aplikasi dengan melakukan proses *login*. Berikut ini adalah desain *testing* untuk halaman *login* atau autentikasi.

Tabel 4.7 Desain *Testing* Halaman *Login*

| Nama Kolom | Inputan Data |
|------------|----------------|
| Email | me@sembara.xyz |
| Password | password123 |

Kemudian dilanjutkan *testing*, dengan desain *testing* yang dibuat sebelumnya. Berikut ini adalah tahapan pengujian halaman *login*.

Tabel 4.8 *Testing* Halaman Autentikasi/*Login*

| No | Dekripsi | Tahapan | Input | Output |
|----|-----------------------|--|--|--|
| 1. | Melakukan autentikasi | Memasukkan inputan <i>email</i> dan <i>password</i> dengan benar | <i>Email</i> dan <i>password</i> yang sesuai dan klik tombol “masuk” | Menampilkan halaman utama sesuai <i>role user</i> |
| 2. | Melakukan autentikasi | Memasukkan inputan <i>email</i> dan <i>password</i> dengan salah | <i>Email</i> dan <i>password</i> secara acak dan klik tombol “masuk” | Tetap berada di halaman <i>login</i> , dan muncul notifikasi “email atau password salah” |

4.4.2 Halaman Pendaftaran Calon Kandidat

Calon Kandidat dapat melakukan proses pendaftaran dengan mengakses halaman pendaftaran. Berikut ini adalah desain *testing* untuk halaman pendaftaran calon kandidat.

Tabel 4.9 Desain *Testing* Halaman Pendaftaran Calon Kandidat

| Nama Kolom | Inputan Data |
|------------|---------------------------|
| Nama | Sebastianus Sembara |
| Email | me@sembara.xyz |
| Password | password123 |
| Organisasi | Universitas Dinamika |
| Voting | Badan Eksekutif Mahasiswa |

Kemudian dilanjutkan *testing*, dengan desain *testing* yang dibuat sebelumnya. Berikut ini adalah tahapan pengujian halaman pendaftaran calon kandidat

Tabel 4.10 *Testing* Halaman Pendaftaran Calon Kandidat

| No | Dekripsi | Tahapan | Input | Output |
|----|-----------------------------|---|---|--|
| 1. | Membuka halaman pendaftaran | Menekan tombol “daftar calon kandidat” | Klik tombol “daftar calon kandidat” | Menampilkan halaman pendaftaran calon kandidat |
| 2. | Melakuakan pendaftaran | Memasukkan <i>inputan</i> pendaftaran yang sesuai, lalu menekan tombol “daftar” | Nama, <i>email</i> , <i>password</i> , organisasi, voting | Calon kandidat berhasil mendaftarkan diri |

4.4.3 Halaman Pengelolaan Master Voting

Panitia dapat melakukan proses pengelolaan master voting dengan mengakses menu voting pada *sidebar*. Panitia dapat melakukan proses tambah, ubah, dan hapus data voting. Berikut ini adalah desain *testing* untuk halaman pengelolaan master voting.

Tabel 4.11 Desain Testing Halaman Pengelolaan Master Voting

| Nama Kolom | Inputan Data |
|---------------|--|
| Organisasi | Universitas Dinamika |
| Nama | Badan Eksekutif Mahasiswa |
| Deskripsi | Pemilihan Ketua Badan Eksekutif Mahasiswa (BEM) Universitas Dinamika |
| Tanggal Mulai | 2022/06/30 |
| Tanggal Akhir | 2022/07/01 |

Kemudian dilanjutkan *testing*, dengan desain *testing* yang dibuat sebelumnya. Berikut ini adalah tahapan pengujian halaman pengelolaan master voting

Tabel 4.12 *Testing* Halaman Pengelolaan Master Voting

| No | Dekripsi | Tahapan | Input | Output |
|----|------------------------|--|---|---|
| 1. | Membuka halaman voting | Menekan menu voting pada <i>sidebar</i> | Klik menu voting pada <i>sidebar</i> | Menampilkan halaman pengelolaan master voting |
| 2. | Tambah data voting | Menekan tombol “create”, lalu memasukkan inputan yang sesuai, lalu tekan tombol “simpan” | Organisasi, nama, deskripsi, tanggal mulai, tanggal akhir | Data berhasil disimpan pada <i>database</i> |
| 3. | Ubah data voting | Memilih data yang ingin diubah, menekan tombol “update”, lalu memasukkan inputan yang sesuai, lalu tekan tombol “ubah” | Id, Organisasi, nama, deskripsi, tanggal mulai, tanggal akhir | Data berhasil dirubah pada <i>database</i> |

| No | Dekripsi | Tahapan | Input | Output |
|----|-------------------|--|-------|-------------------------------------|
| 4. | Hapus data voting | Memilih data yang ingin dihapus, menekan tombol “delete” | Id | Data berhasil dihapus pada database |

4.5 User Acceptance Testing (UAT)

Pada tahap *beta testing* dilakukan pengujian pada aplikasi e-voting PEMIRA Universitas Dinamika menggunakan metode *user acceptance testing*. Untuk pengujian ini akan dilakukan kepada pengguna. Sampel yang digunakan dalam penelitian ini adalah mahasiswa Universitas Dinamika yang diambil secara acak. Responden akan menjawab sepuluh pertanyaan yang diberikan terkait dengan aplikasi yang telah dibuat. Kuesioner menggunakan skala likert dari skala 1 sampai 5. Detail hasil kuesioner dapat dilihat pada Lampiran 11.

4.6 Pembahasan

Pada pembahasan implementasi *test-driven development* pada rancang bangun aplikasi Pemilihan Raya Universitas Dinamika, didapatkan hasil sebagai berikut:

1. Penelitian ini menghasilkan rancang bangun aplikasi *e-voting* PEMIRA Universitas Dinamika berbasis *website* menggunakan metode *Test-Driven Development*. Keluaran dari penelitian ini adalah berupa aplikasi *e-voting* PEMIRA Universitas Dinamika yang telah diperbarui
2. Aplikasi *e-voting* PEMIRA Universitas Dinamika dirancang dengan sistem empat *roles*, yaitu (1) pegawas voting untuk merekomendasikan mahasiswa berpotensi menjadi calon kandidat, (2) panitia voting untuk mengelola keseluruhan voting PEMIRA, (3) calon kandidat untuk mendaftar dan dipilih dan (4) peserta untuk memilih calon kandidat.
3. Dalam menggunakan aplikasi *e-voting* PEMIRA Universitas Dinamika ini diawali dengan panitia memasukkan data voting, lalu calon kandidat mendaftar, setelah itu panitia melakukan seleksi terhadap calon kandidat, lalu peserta yang terdaftar dapat melakukan pengecekan data, jika terdaftar mereka akan menerima email untuk mendapatkan *token*. Lalu peserta dapat memilih calon kandidat.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari implementasi *test driven development*, implementasi sistem, *black box testing*, dan user acceptance testing pada rancang bangun aplikasi *e-voting* PEMIRA Universitas Dinamika yang telah dilakukan, kesimpulan yang dapat ditarik adalah sebagai berikut:

1. Implementasi metode *test-driven development* berhasil dilakukan dan berdampak pada meningkatnya kualitas dari aplikasi *e-voting* PEMIRA Universitas Dinamika, dengan terbukti bahwa hasil pengujian aplikasi yang sekarang yaitu 100%, dimana seluruh fungsi yaitu 13 fungsional yang ada pada aplikasi baru mendapatkan hasil pengujian sukses. Hasil pengujian tersebut meningkat yang awalnya pada aplikasi lama sebesar 50% menjadi 100% sukses pada aplikasi baru. Hasil perbandingan dari uji coba aplikasi lama dan baru dapat dilihat pada Lampiran 14.
2. Aplikasi *e-voting* PEMIRA Universitas Dinamika ini dapat digunakan oleh 4 *user*, yaitu (1) pegawas voting untuk merekomendasikan mahasiswa berpotensi menjadi calon kandidat, (2) panitia voting untuk mengelola keseluruhan voting PEMIRA, (3) calon kandidat untuk mendaftar dan dipilih dan (4) peserta untuk memilih dalam proses voting PEMIRA Universitas Dinamika.
3. Menurut hasil pengujian yang telah dilakukan, aplikasi *e-voting* ini sudah siap untuk membantu Mahasiswa Universitas Dinamika dalam memeriahkan pesta demokrasi, yaitu PEMIRA Universitas Dinamika Surabaya. Memilih ketua dan wakil ketua Badan Eksekutif Mahasiswa (BEM) dan Ketua Dewan Perwakilan Mahasiswa (DPM).
4. Hasil *User Acceptance Test* (UAT) menghasilkan informasi bahwa aplikasi yang telah dirancang memiliki nilai valid atau sesuai dengan harapan yang diinginkan. Pengujian ini menghasilkan rata rata nilai 89,82% yang berarti bahwa aplikasi ini dapat diterima pengguna dengan baik.

5.2 Saran

Adapun saran yang dapat digunakan untuk pengembangan penelitian selanjutnya antara lain:

1. Menambahkan metode sistem pendukung keputusan untuk memilih mahasiswa berpotensi untuk menjadi ketua dan wakil Badan Eksekutif Mahasiswa (BEM) dan Ketua Dewan Perwakilan Mahasiswa (DPM) Universitas Dinamika
2. Mengembangkan UI/UX yang dapat lebih mudah dipahami oleh pengguna baik pengawas, panitia, calon kandidat, ataupun peserta voting PEMIRA
3. Menambahkan aplikasi versi mobile untuk pengembangan selanjutnya



DAFTAR PUSTAKA

- Basri, F. F., Yenti, A. N., & Taufiq, R. (2021). Rancang Bangun Aplikasi E-Vote Pemilihan Ketua RT Pada Perumahan Duta Asri Palem 7 Berbasis Web. *JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang*, 7(1), 245–251.
- Beck, K. (2002). *Test-Driven Development: By Example*. Addison-Wesley Professional.
- Bergmann, S. (2022). “*PHPUnit Official Website*”, <https://phpunit.de/>, diakses pada tanggal 25 April 2022 Pukul 13.00.
- Dewan Perwakilan Mahasiswa. (2019a). *Undang Undang Dasar Keluarga Mahasiswa Universitas Dinamika 2019 Pasal 8, Ayat 3a* (p. 5). Dewan Perwakilan Mahasiswa Universitas Dinamika.
- Dewan Perwakilan Mahasiswa. (2019b). *Undang-Undang Keluarga Mahasiswa No.1 Tahun 2019 Tentang Pemilihan Raya Universitas Dinamika*. STIKOM Surabaya.
- Hardjaloka, L., & Simarmata, V. M. (2016). E-Voting: Kebutuhan vs. Kesiapan (Menyongsong) E-Demokrasi. *Jurnal Konstitusi*, 8(4), 579–604.
- Hasibuan, A. N., & Dirgahayu, T. (2021). Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna. *Automata*, 2(1). <https://journal.uii.ac.id/AUTOMATA/article/view/17367>
- Jonathan, F., & Pakereng, M. A. I. (2021). Test-Driven Development pada Pengembangan Aplikasi Android untuk Memantau COVID-19. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 6(1), 20–24. <https://doi.org/10.31294/ijcit.v6i1.9502>
- Karac, I., & Turhan, B. (2018). *What Do We (Really) Know about Test-Driven Development?* <https://ieeexplore.ieee.org/abstract/document/8405634/>
- Khanam, Z., & Ahsan, M. N. (2017). Evaluating the Effectiveness of Test-Driven Development: Advantages and Pitfalls. In *International Journal of Applied Engineering Research* (Vol. 12, pp. 7705–7716).
- Kumar, S., & Bansal, S. (2013). Comparative Study of Test-Driven Development with Traditional Techniques. In *International Journal of Soft Computing and Engineering (IJSCE)* (Vol. 3, Issue 1, pp. 352–360).
- Masripah, S., & Ramayanti, L. (2020). Penerapan Pengujian Alpha Dan Beta Pada Aplikasi. *Jurnal Swabumi*, 8(1), 100–105.

- Otwell, T. (2022, July 1). “Laravel Official Website”, <https://laravel.com/docs/9.x/structure>, diakses pada tanggal 7 Juni 2022 Pukul 13.00.
- Raharja, P. A., Sunardi, S., & Imam, R. (2018). Perancangan E-voting Berbasis Mobile Menggunakan Metode Test-Driven Development. *Seminar Nasional PPM 2018 Universitas Negeri Surabaya*, 707–713.
- Romeo. (2003). *Testing dan Implementasi Sistem* (Pertama). STIKOM Surabaya.
- Subhiyakto, E. R., & Astuti, Y. P. (2020). Test-Driven Development (TDD) for Point-of-Sale System at Bicycle Shop. *Scientific Journal of Informatics*, 7(2), 238–246.
- Wardhono, W. S., & Kusuma, L. P. (2015). Evaluasi User Acceptance Augmented Reality Triage Mobile Pada Sistem Kedaruratan Medis. *Jurnal Sentar*, 978–979.

