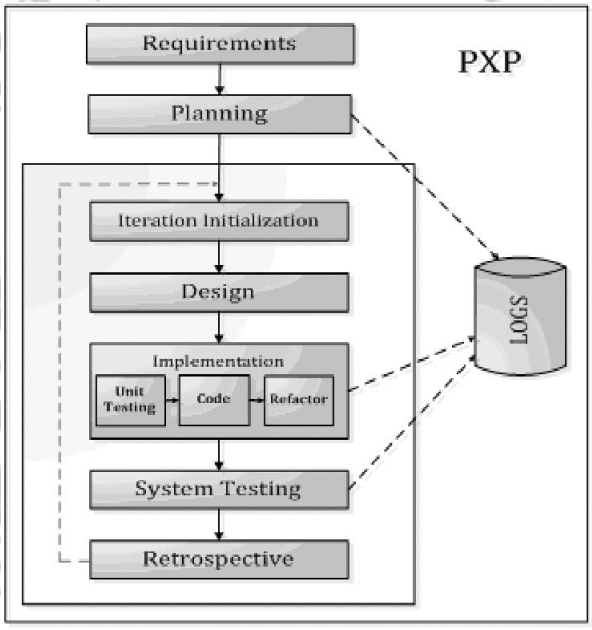
BAB III
METODOLOGI PENELITIAN

Penelitian ini memiliki beberapa tahapan penelitian yaitu requirements, planning, iteration initialization, design, Implementasi yang meliputi (*Unit Testing*, *Code* dan *Refactor*), Testing sistem, dan *retrospective*. Setelah tahapan *Personal Extreme Programming* selesai, akan dilakukan penambahan aspek keamanan yang digambarkan pada gambar 1.



Gambar 3. 1 SDLC Metode PXP dan Security

Pembangunan sistem E-voting ini dibangun dengan berbasis web menggunakan *framework* Laravel dengan bahasa pemrograman PHP dan *database* MySQL [12]. Rincian tahap metode pengembangan PXP dapat dilihat pada penjelasan berikut.

3.1 Requirements

Pengembangan mengumpulkan kebutuhan kebutuhan pada tahap ini. Pengumpulan kebutuhan tersebut dilakukan dengan cara wawancara dan diskusi bersama panitia pemilwa FK UB 2022. Kebutuhan-kebutuhan tersebut dijelaskan dalam bentuk *user stories* seperti tabel 1 dibawah.

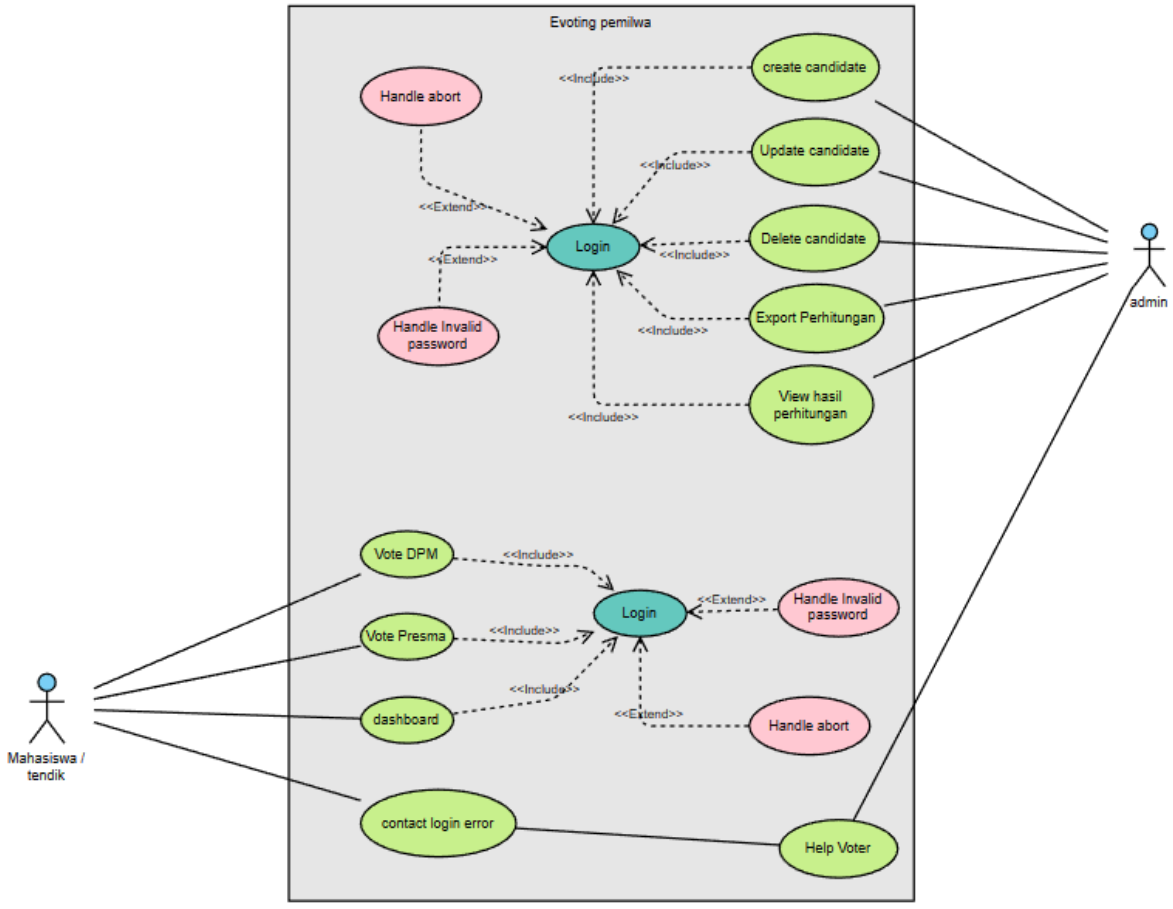
Tabel 3. 1 Kode User Stories

Kode user stories	Task
#LOGIN001	Login
#VOTING001	Melakukan Voting
#MANAGE001	Manage Data Calon, yang meliputi create, read, update, delete (CRUD)

Tabel 3. 2 Kode User Stories

User Stories	Task
User stories Mahasiswa #LOGIN001	1. Sebagai mahasiswa , saya ingin login dengan menggunakan akun dari kampus universitas brawijaya. 2. Sebagai mahasiswa, saya ingin yang bisa login hanya mahasiswa yang berasal dari fakultas kedokteran dan ilmu kesehatan. 3. Sebagai mahasiswa, saya ingin dapat menghubungi panitia ketika saya mengalami kendala ketika error login. 4. Sebagai pengguna aplikasi, saya ingin dapat mengganti kata sandi saya secara berkala untuk memastikan keamanan akun saya. 5. Sebagai pengguna aplikasi, saya ingin memastikan bahwa kata sandi saya tidak dapat dilihat oleh orang lain, bahkan oleh administrator sistem.
User Stories tenaga didik #LOGIN001	1. Sebagai tenaga didik, saya ingin login menggunakan token yang diberikan oleh developer
User Strories mahasiswa #	1. Sebagai mahasiswa, saya ingin voting hanya bisa dilakukan 1x di masing” calon

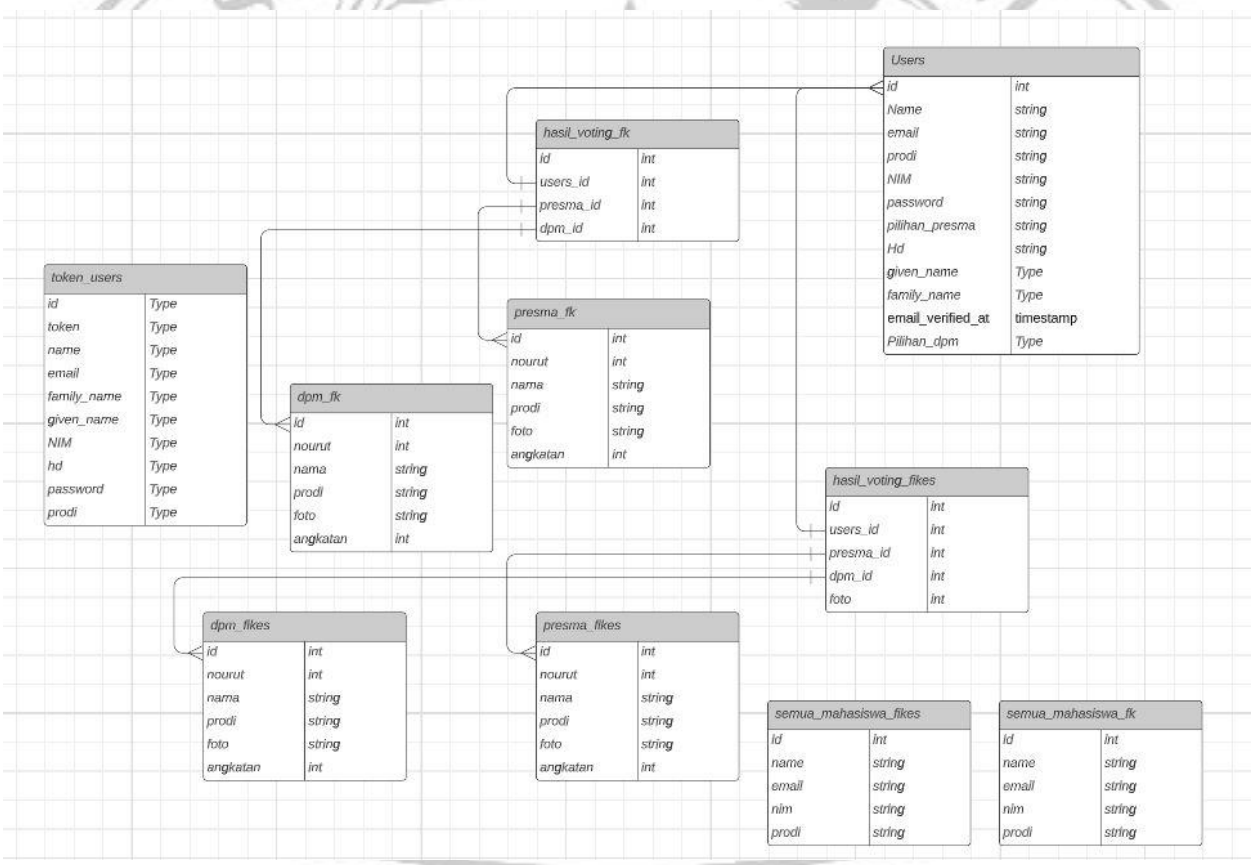
VOTING001	2. Sebagai mahasiswa, saya ingin dimunculkan notifikasi persetujuan ketika saya melakukan klik ke tombol pilih calon
User Stories Admin #MANAGE001	1. Sebagai panitia , saya ingin dapat menambahkan, edit, delete calon DPM dan Presma dengan aman. 2. Sebagai panitia, saya ingin dapat melihat hasil pemilihan secara langsung di halaman admin. 3. Sebagai panitia, saya ingin dapat melakukan export ke excel dari hasil pemilihan.



Gambar 3. 2 Use Case Diagram

Diagram use case untuk aplikasi e-voting Fakultas Kedokteran Universitas Brawijaya menggunakan Arsitektur Model View Controller (MVC) memperlihatkan interaksi antara aktor-aktor dan sistem dalam konteks pengembangan aplikasi tersebut. Sistem ini

bertujuan untuk memudahkan proses pemilihan elektronik di dalam fakultas. Terdapat beberapa aktor yang terlibat dalam sistem ini. Pertama, mahasiswa dan tenaga didik dapat sebagai pemilih, melakukan login, memilih kandidat yang diinginkan. Aktor terakhir adalah administrator sistem yang memiliki akses penuh ke sistem. Tugas administrator sistem meliputi manajemen akun pengguna, pengaturan hak akses, dan pemeliharaan rutin pada aplikasi. Dalam sistem ini, terdapat beberapa kasus penggunaan yang perlu diperhatikan. Mahasiswa dapat mendaftar sebagai pemilih, melakukan login ke sistem, memilih kandidat, dan melihat hasil pemilihan. Panitia pemilihan dapat mengelola kandidat, mengatur jadwal pemilihan, serta menghasilkan laporan hasil pemilihan. Dengan adanya diagram use case ini, interaksi antara aktor-aktor dan sistem dapat lebih jelas dan terorganisir, sehingga memudahkan pengembangan aplikasi e-voting ini.



Gambar 3. 3 ERD database

Setelah tergambarkan use case diagram, developer akan semakin mudah untuk membuat design database yang nantinya akan dibuat menjadi ERD seperti gambar 3 diatas. Use case diagram memberikan gambaran tentang fungsionalitas dan interaksi antara aktor

(pengguna) dengan sistem yang sedang dikembangkan. Dengan memahami use case diagram, developer dapat mengidentifikasi entitas-entitas utama yang terlibat dalam sistem, serta hubungan dan atribut yang terkait dengan entitas tersebut. Informasi yang terdapat dalam use case diagram, seperti aktor yang terlibat, aksi yang dilakukan, dan objek yang digunakan, dapat menjadi panduan dalam merancang struktur database yang efisien. Hal ini memungkinkan developer untuk menentukan tabel-tabel yang diperlukan dalam database, kolom-kolom yang harus ada, serta relasi dan keterhubungan antar tabel. Dengan menggunakan use case diagram sebagai dasar, developer dapat menggambarkan hubungan antara entitas-entitas tersebut secara lebih terstruktur dan menyeluruh dalam Entity-Relationship Diagram (ERD). ERD merupakan visualisasi dari struktur database yang menggambarkan entitas, atribut, dan hubungan antara entitas-entitas tersebut. Dengan memiliki ERD yang jelas, developer dapat menghindari kesalahan desain dan memastikan bahwa desain database sesuai dengan kebutuhan sistem yang akan dikembangkan.

Dari user stories yang ada pada tabel 2, terdapat beberapa *acceptance of criteria* dari beberapa fitur yang diinginkan oleh stakeholder yang mana dijelaskan pada tabel 3.

Tabel 3. 3 Acceptance of criteria

Fitur	Acceptance of criteria
Validasi login	**Scenario 1** ➢ Given : Mahasiswa mengisi email dengan menggunakan email UB ➢ When : format email @student.ub.ac.id ➢ Then : Berhasil **Scenario 2** ➢ Given : Mahasiswa mengisi email dengan menggunakan email selain email UB

	➢ When : @gmail.com ➢ Then : login error , harap menggunakan email ub **Scenario 3** ➢ Given : Mahasiswa login dengan menggunakan NIM selain fakultas kedokteran atau ilmu kesehatan ➢ When : parameter email yang terdaftar di *database* ub tidak menunjukkan bahwa mahasiswa / mahasiswi itu bukan berasal dari FIKES / FK ➢ Then : Login error, Anda tidak terdaftar di *database* kami

3 Planning

Dalam fase ini, pengembang menyusun serangkaian task yang akan dikerjakan pada tiap iterasi berdasarkan dari User Stories yang didapatkan [13][14]. Tabel 4 memecah beberapa user stories yang telah didapatkan menjadi beberapa task yang lebih spesifik.

Tabel 3. 4 Planning

Task	Estimasi Pengerjaan	Priority
Membuat sistem untuk login mahasiswa	3 hari	High
Hosting website	1 hari	Low

Membuat tampilan *user interface* untuk mahasiswa	2 hari	High
Membuat sistem belakang untuk menghandle mahasiswa voting	2 hari	Medium
Membuat sistem validasi login mahasiswa	2 hari	High
Membuat akun berupa token yang dapat digunakan oleh tenaga didik / mahasiswa ketika mengalami kendala saat login	1 hari	Medium
Membuat fitur untuk roles admin yang mencakup CRUD, perhitungan jumlah suara, export hasil ke excel	5 hari	High
Mengimplementasikan modul purifier untuk mengingkatkan sisi keamanan aplikasi	2 hari	Medium
Membuat tampilan responsive	1 hari	Low
Total pengerjaan (hari)	19 hari	

4 Iteration Initialization

Iteration Initialization merupakan tahap awal dari setiap iterasi yang akan dilaksanakan. Iterasi tersebut dimulai dengan pemilihan tugas, yang mana nantinya akan menjadi fokus utama dari iterasi tersebut [15][16]. Tugas-tugas tersebut didapatkan dari

hasil perencanaan pada tahap planning. Didapat dari tahap planning diatas terdapat beberapa task yang dibedakan berdasarkan tingkat priotitasnya yaitu *high*. *Medium* , dan *low*. Fokus utama diambil dari task yang memiliki tingkat prioritas *high,* yang mencakup:

1. Membuat sistem untuk login mahasiswa
2. Membuat tampilan *user interface* untuk mahasiswa
3. Membuat fitur untuk roles admin yang mencakup CRUD, perhitungan jumlah suara, export hasil ke excel
4. Membuat sistem validasi login mahasiswa

Ketika 4 task tersebut sudah selesai dikerjakan, maka selanjutnya akan dikerjakan task dengan tingkat prioritas *medium*, yang mencakup :

1. Membuat sistem belakang untuk menghandle mahasiswa voting
2. Membuat akun berupa token yang dapat digunakan oleh tenaga didik / mahasiswa ketika mengalami kendala saat login
3. Mengimplementasikan modul purifier untuk mengingkatkan sisi keamanan aplikasi

Yang terakhir jika kedua prioritas tersebut sudah selesai dikerjakan , maka akan dilanjutkan dengan task dengan prioritas *low*, yang mencakup :

1. Membuat tampilan responsive
2. Hosting website

5 Design

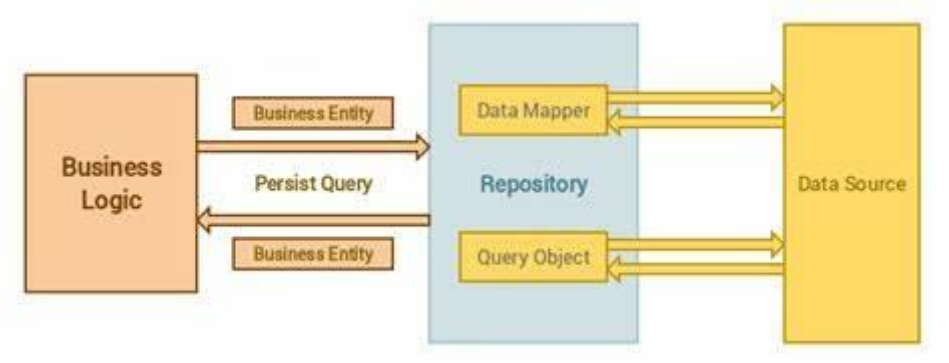
Pengembang memodelkan modul sistem yang akan diimplementasikan selama proses iterasi pada tahap desain [17]. Desain sistem tersebut hanya memenuhi kebutuhan klien yang didapatkan ketika proses *requirement*, tanpa adanya desain tambahan untuk kebutuhan user yang nantinya berubah. Pengembang membuat desain tersebut sebagai spike solution prototype yang merupakan skema desain *prototype* dari iterasi yang sedang dijalankan [18].

6 Implementation

Fase *Implementation* merupakan fase dimana kode pemrograman dibuat.

Pengembang mengimplementasikan objek-objek yang terdapat pada fase design. Fase ini terbagi lagi menjadi tiga tahap, yaitu *Unit Testing, Code Generation, dan Code Refactoring* [19].

a. *Unit testing* adalah pengujian fungsionalitas code program, pengujian ini berbeda dengan proses pengujian fungsionalitas pada metodologi pengembangan software lain dimana pengujian dilakukan setelah sistem selesai. Pada unit testing pengembang menulis sebagian code program di awal tahap implementasi lalu melakukan pengujian.
b. Tahap selanjutnya adalah *Code Generation*, di mana pengembang menghasilkan kode sumber yang sesuai dengan rancangan yang telah dibuat pada tahap sebelumnya. Dalam tahap ini, pengembang akan menulis kode untuk mengimplementasikan objek-objek yang telah dirancang sebelumnya. Pengembang akan membuat kode untuk mengimplementasikan objek-objek pada arsitektur Model View Controller, seperti kode untuk memproses input dari pengguna dan kode untuk menampilkan hasil voting dengan bantuan *Artisan command-line interface.* Beberapa perintah Artisan yang dapat membantu code generation, antara lain *make:model*, *make:controller*, *make:request*, *make:migration*, dan lain-lain.
c. Tahap terakhir dari fase implementasi adalah *Code Refactoring*, di mana pengembang melakukan perbaikan pada kode yang telah dibuat sebelumnya. Tujuan dari tahap ini adalah untuk meningkatkan kualitas dan keterbacaan kode, serta memperbaiki kesalahan dan menghilangkan redundansi. Tahap ini akan digunakan untuk memisahkan fungsi-fungsi yang terlalu kompleks menjadi fungsi-fungsi yang lebih sederhana dan mudah dipahami.



Gambar 3. 4 Repository Pattern

7 System Testing

System testing merupakan pengujian fungsuonalitas semua fitur hasil implementasi yang telah di lakukan selama proses iterasi [20]. *User Acceptance Test* disajikan sebagai hasil pengujian yang dilakukan oleh user. Sistem akan di-hosting agar dapat diakses secara online, mengingat tempat pengembang dan pengguna yang berbeda. Pengembang akan mengirimkan dokumen *User Acceptance Test* kepada user sebelum pengujian dilakukan untuk divalidasi. Saat pengujian dilakukan, pengembang akan menghubungkan perangkat user dengan perangkat milik pengembang melalui sistem remote. Hal ini bertujuan untuk memudahkan pengembang dalam memantau aktivitas user di perangkat mereka dan mengarahkan user ke fungsi sistem yang diimplementasikan dalam proses iterasi. Setelah itu, user akan memberikan verifikasi apakah fungsi tersebut sudah sesuai dengan keinginan user.

8 Retrospective

Fase terakhir dari proses iterasi tersebut adalah *retrospective*. Pengembang menganalisis jalannya setiap fase pengembangan modul, kesesuaian estimasi waktu pengerjaan, penyebab terjadinya keterlambatan ketika proses pengebangan dan mencegah hal tersebut terlulang kembali di iterasi selanjutnya [21].