

**Pengembangan Aplikasi *Automatic Scoring* Berbasis Web  
Untuk Penilaian Jawaban Teks pada Tugas dan Ujian  
*Online* dengan Metode *Personal eXtreme Programming***

**TUGAS AKHIR**

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1)  
di Program Studi Teknik Informatika, Jurusan Teknologi,  
Produksi dan Industri, Institut Teknologi Sumatera

Oleh:

**Markus Togi Fedrian Rivaldi Sinaga**

**118140037**



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI, PRODUKSI, DAN INDUSTRI  
INSTITUT TEKNOLOGI SUMATERA  
LAMPUNG SELATAN**

**2023**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Seiring pesatnya perkembangan teknologi informasi dan komunikasi, banyak bidang dalam kehidupan kita sehari-hari yang turut berkembang atau mengalami perubahan. Salah satunya adalah bidang pendidikan. Perubahan yang paling tampak pada bidang pendidikan tersebut adalah mulai ramainya praktik pembelajaran jarak jauh. Pembelajaran jarak jauh memanfaatkan teknologi informasi dan komunikasi seperti kelas virtual, konferensi video, dan forum diskusi *online* untuk memungkinkan terjadinya interaksi antara pengajar dan siswa meskipun mereka terpisah secara geografis. Seperti proses belajar mengajar pada umumnya, biasanya pada pembelajaran jarak jauh juga terdapat ujian. Ujian dilakukan sebagai salah satu cara untuk mengevaluasi pencapaian siswa dalam kegiatan belajar-mengajar [1]. Saat ini terdapat beragam platform ujian daring yang tersebar di internet seperti *Google Form*, *Quizziz*, *edBase*, *Testmoz*, dan sebagainya [2]. Namun seluruh platform yang disebutkan tadi, masih mengharuskan pengajar untuk melakukan penilaian jawaban satu persatu secara manual untuk soal tipe isian, yang mana hal ini tentu menyulitkan, dan tidak efisien dari segi waktu, di samping itu mengecek secara manual juga berpotensi mengakibatkan kesalahan dalam pengecekan. Untuk menjawab permasalahan tersebut, perlu dikembangkan dan diteliti suatu aplikasi berbasis *web* yang dilengkapi dengan suatu model kecerdasan buatan berupa sistem penilaian otomatis.

Sistem penilaian otomatis (*Automatic Scoring System*) sendiri merupakan sistem yang mampu melakukan proses konversi dari performa dalam penyelesaian tugas (umumnya di bidang pendidikan dan penelitian) menjadi berbagai level atau karakteristik kualitas kemampuan [3]. Kemudian, karena yang menjadi fokus dari penelitian ini adalah penilaian teks jawaban singkat, maka bidang terfokus yang paling sesuai adalah *Automated Essay Scoring* (AES), yang secara sederhana dapat dimaknai sebagai pemanfaatan kemampuan komputer untuk melakukan evaluasi dan penilaian pada kalimat yang diketik secara otomatis [4]. AES sendiri merupakan aplikasi dari Pengolahan Bahasa Alami (*Natural Language Processing*) yang merupakan salah satu

fokus bidang dari Kecerdasan Buatan (*Artificial Intelligence*) [4]. Implementasi dari AES sendiri sudah cukup banyak diteliti dan dibahas. Namun karena penelitian ini hanya berfokus pada pengembangan aplikasi berbasis web yang akan menjadi wadah bagi model AES, peneliti akan lebih banyak membahas terkait metode SDLC (*Software Development Life Cycle*). Terdapat beberapa model SDLC, mulai dari model-model yang bersifat *plan-driven* yang paling awal dikenali seperti model *waterfall*, *prototyping*, dan *iterative*, hingga model yang lebih ringkas, ringan dan fleksibel seperti model *agile*.

Model *agile* sendiri memiliki beragam variasi metodologi, mulai dari *Scrum*, *eXtreme Programming* (XP), hingga *Personal eXtreme Programming* (PXP). Pada penelitian ini sendiri, untuk pengembangan aplikasi berbasis *web*-nya digunakan metode *agile* PXP, yang merupakan pengembangan dari metode *agile* terdahulunya yaitu *eXtreme Programming* (XP) yang berfokus pada empat hal, yaitu : keterlibatan klien, pengujian berkelanjutan, pemrograman dengan tim kecil yang terpadu, serta siklus iterasi yang singkat. Sementara untuk metode PXP, disamping empat hal yang menjadi fokus dalam metode XP, terdapat pula perhatian yang besar pada keotonoman pengembang individu [10]. PXP sebagai salah satu metode dalam model SDLC *agile* dipilih karena memberikan kenyamanan dan fleksibilitas tinggi pada pengembangan perangkat lunak berukuran kecil bila dibandingkan dengan model SDLC lainnya, serta sifat PXP yang menekankan pada keotonoman pengembang individu yang memungkinkannya untuk bekerja sesuai dengan cara dan kecepatannya sendiri. Selain itu, PXP membuat jalur komunikasi antara pengembang dan klien menjadi lebih singkat sehingga akan lebih mudah dan cepat untuk merumuskan berbagai kebutuhan dari perangkat lunak, serta lebih mudah melacak dan memprediksi perubahan yang terjadi. Hal ini sesuai dengan kebutuhan pengembangan aplikasi penilaian teks otomatis berbasis web yang dapat dikatakan berukuran relatif kecil, dan berpotensi besar mengalami banyak penyesuaian selama proses pengembangan [11][12][13][14].

Di lain sisi, yang menjadi pertimbangan terbesar adalah batasan pada kemampuan peneliti yang sekaligus menjadi pengembang aplikasi dalam penelitian ini, dimana peneliti diharuskan untuk beradaptasi dan mempelajari beragam teknologi yang paling tepat untuk digunakan dalam pengembangan aplikasi, dan lebih mampu bekerja dengan maksimal dengan keterlibatan klien yang memadai namun mudah

dicapai. Hal ini sesuai dengan fleksibilitas, adaptabilitas, dan keotonoman yang tinggi bagi pengembang, yang bisa didapatkan dari metode *agile* model PXP.

Untuk mendukung metode SDLC yang dipilih, untuk metode evaluasi akan digunakan metode *Unit Testing* serta metode *White Box Testing* pada tahap pengembangan, dan metode *Black Box Testing* serta *System Usability Scale* (SUS) pada tahap evaluasi akhir.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah diidentifikasi, berikut adalah rumusan masalah yang dapat saya susun :

1. Bagaimana proses pengembangan aplikasi berbasis *web* dengan menggunakan metode SDLC *agile* model PXP?
2. Bagaimana metode *White Box Testing* digunakan dalam pengujian pada level *Unit Testing* untuk mengevaluasi dan menunjang kemampuan aplikasi dan proses pengembangannya?
3. Bagaimana performa aplikasi berbasis web yang dikembangkan dalam menggunakan kemampuan model AES yang telah dikembangkan berdasarkan hasil evaluasi menggunakan metode *Black Box Testing* pada level *Integration Testing* dan *System Usability Scale* pada level *System Testing*?

## 1.3 Tujuan Penelitian

Berikut adalah tujuan dari penelitian ini berdasarkan masalah-masalah yang telah dirumuskan :

1. Menganalisis penggunaan model SDLC *agile* metode PXP dalam proses pengembangan suatu aplikasi berbasis web.
2. Menganalisis penggunaan metode *White Box Testing* dalam pengujian pada level *Unit Testing* sebagai upaya untuk mengevaluasi dan menunjang kemampuan aplikasi dan proses pengembangannya.
3. Menganalisis performa aplikasi berbasis web saat menggunakan kemampuan model *Automated Scoring* dalam melakukan penilaian jawaban teks singkat secara otomatis melalui hasil evaluasi menggunakan metode *Black Box Testing* dan *System Usability Scale*.

#### 1.4 Batasan Masalah

Batasan masalah dalam penelitian ini perlu ditetapkan untuk kespesifikan tujuan dari penelitian, dan tidak membebani berbagai pihak yang terkait dengan penelitian ini, adapun rumusan masalahnya sebagai berikut :

1. Metode SDLC yang digunakan dalam pengembangan aplikasi adalah model *agile* metode PXP.
2. Penelitian hanya bertujuan untuk mengembangkan aplikasi berbasis web yang akan menjadi wadah untuk model AES, sehingga dokumen penelitian ini tidak akan membahas secara mendalam terkait model kecerdasan buatan yang digunakan.
3. Penelitian hanya berfokus dalam melakukan analisis, untuk perbandingan maupun peningkatan pengembangan lebih lanjut, disesuaikan dengan kemampuan peneliti dan waktu yang tersedia.
4. Aplikasi yang dikembangkan hanya akan dapat digunakan oleh admin, tim manajemen pengembangan, serta pengguna yang telah didaftarkan oleh admin.
5. Aplikasi berbasis web hanya menyesuaikan dengan karakteristik masukan dan luaran yang dibutuhkan oleh model kecerdasan buatan yang telah dikembangkan sebelumnya.

#### 1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut :

1. Memperkenalkan secara singkat metode-metode yang digunakan dalam penelitian ini kepada pembaca.
2. Membantu memudahkan pengembangan dan penelitian lebih lanjut di kemudian hari.
3. Bagi peneliti, sebagai sarana untuk memperdalam pengetahuan dan kemampuan di bidang teknologi informasi, khususnya pengolahan bahasa alami, dan pengembangan web.
4. Memenuhi tanggung jawab menyelesaikan tugas akhir sebagai prasyarat kelulusan.

## **1.6 Sistematika Penulisan**

Sistematika penulisan dokumen penelitian ini terdiri dari lima bab utama, yaitu sebagai berikut :

### **1.6.1 Bab I Pendahuluan**

Berisi gambaran umum terkait isi dari dokumen penelitian ini, antara lain, latar belakang, rumusan masalah, tujuan penelitian, batasan masalah penelitian, manfaat penelitian, dan sistematika penulisan dan penyusunan dokumen penelitian.

### **1.6.2 Bab II Tinjauan Pustaka**

Berisi informasi singkat terkait berbagai karya ilmiah yang dijadikan sumber acuan dalam penyusunan dokumen penelitian ini.

### **1.6.3 Bab III Metode Penelitian**

Berisi deskripsi rinci terkait berbagai metode yang digunakan dalam melakukan penelitian ini.

### **1.6.4 Bab IV Hasil Penelitian dan Pembahasan**

Berisi deskripsi rinci serta pembahasan menyeluruh terkait hasil dari penelitian yang telah dilakukan.

### **1.6.5 Bab V Kesimpulan dan Saran**

Berisi kesimpulan yang merangkum hasil analisis dari pembahasan hasil penelitian yang telah dilakukan

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Penelitian ini tentunya tidak terlepas dari bantuan dan ilham dari berbagai penelitian serupa maupun terkait terdahulu. Selain itu, berbagai penelitian terdahulu juga digunakan sebagai pembanding, dan acuan untuk meningkatkan hasil yang diharapkan. Berikut ini adalah berbagai penelitian terdahulu yang menjadi acuan bagi penelitian ini :

No	Judul	Peneliti	Tahun	Metode	Hasil	Pembeda
1	Implementasi Metode Personal Extreme Programming dalam Pengembangan Sistem Manajemen Transaksi Perusahaan (Studi Kasus: CV. Todjoe Sinar Group)	Muhammad Ulfi, Gita Indah Marthasari, Ilyas Nuryasin	2020	Metode SDLC : Personal eXtreme Programming	Sistem Informasi yang dapat memudahkan manajemen transaksi perusahaan	Implementasi metode SDLC akan dilakukan pada aplikasi yang memiliki kegunaan dan bidang studi kasus berbeda.
2	Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information System	Gita Indah Marthasari, Wildan Suharso, Frendy Ardiansyah	2018	Metode SDLC : Personal eXtreme Programming	Sistem Informasi yang dapat memudahkan manajemen perpustakaan	Implementasi metode SDLC akan dilakukan pada aplikasi yang memiliki kegunaan dan bidang studi kasus berbeda
3	Sistem Informasi Program Keluarga Harapan (PKH)	Abdullah Faqih Septiyanto, Wildan Suharso,	2020	Metode SDLC : Personal eXtreme Programming	Sistem informasi yang dapat menunjukkan data sederhana	Implementasi metode SDLC akan dilakukan pada aplikasi yang

No	Judul	Peneliti	Tahun	Metode	Hasil	Pembeda
	Menggunakan Metode Personal Extreme Programming dengan Metode Prioritas Ranking	Ilyas Nuryasin			dan mengunggah file	memiliki kegunaan dan bidang studi kasus berbeda
4	Implementing White Box Testing for Evaluating the Inner Logic Code of the Research, Staffs, and Library Information System of Institute of Informatics and Business Darmajaya	Anggi Andriyadi, Dona Yuliawati, Sushanty Saleh, Bobby Bachry	2020	Metode Evaluasi : White Box Testing	Pengujian berhasil dilakukan dan sistem lolos pengujian	Evaluasi akan dilakukan pada aplikasi yang memiliki kegunaan berbeda.
5						
6						

Penelitian pertama dilakukan oleh Muhammad Ulfi, Gita Indah Marthasari, dan Ilyas Nuryasin pada tahun 2020 dengan judul “Implementasi Metode *Personal Extreme Programming* dalam Pengembangan Sistem Manajemen Transaksi Perusahaan (Studi Kasus: CV. Toedjoe Sinar Group)” [6DP], pada penelitian ini menggunakan metode PXP. Luaran dari penelitian ini adalah Sistem Informasi yang dapat memudahkan manajemen transaksi perusahaan. Namun salah satu masalah yang terjadi adalah ketidaksesuaian waktu pengerjaan dengan waktu yang sebelumnya telah diestimasikan, akibat dari ketidak-familiar-an pengembang dengan salah satu permintaan (*requirement*) dari pengguna.

Pada penelitian kedua dilakukan oleh Gita Indah Marthasari, Wildan Suharso, dan Frendy Ardiansyah pada tahun 2018 dengan judul “*Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information System*” [8DP], pada penelitian ini digunakan metode PXP. Luaran dari penelitian ini adalah sebuah sistem



informasi yang dapat memudahkan manajemen perpustakaan. Kekurangan dari penelitian ini adalah meski telah menggunakan pendekatan MoSCoW untuk penentuan prioritas kebutuhan, namun tim pengembang tetap tidak dapat mencegah terjadinya waktu tunggu kala transisi antar iterasi.

Pada penelitian ketiga dilakukan oleh Abdullah Faqih Septiyanto, Wildan Suharso, dan Ilyas Nuryasin pada tahun 2020 dengan judul “Sistem Informasi Program Keluarga Harapan (PKH) Menggunakan Metode *Personal Extreme Programming* dengan Metode Prioritas Ranking” [9DP]. Pada penelitian ini digunakan metode SDLC model *Personal Extreme Programming* dengan Metode Prioritas Ranking. Luaran dari penelitian ini adalah sistem informasi yang dapat menunjukkan data dengan proses sederhana dan mengunggah file.

Pada penelitian keempat dilakukan oleh Anggi Andriyadi, Dona Yuliaty, Sushanty Saleh, dan Bobby Bachry pada tahun 2020 dengan judul penelitian “*Implementing White Box Testing for Evaluating the Inner Logic Code of the Research, Staffs, and Library Information System of Institute of Informatics and Business Darmajaya*” [10], pada penelitian ini dilakukan pengujian terhadap sistem dengan menggunakan metode evaluasi *White Box Testing*. Hasil dari penelitian ini pengujian yang berhasil dilakukan dan sistem lolos tahap pengujian. Namun, kekurangan dari penelitian ini adalah pengujian yang dilakukan sangat sederhana dan kurangnya informasi terkait penggunaan metode *white box testing*.

## 2.2 Dasar Teori

### 2.2.1. Automatic Scoring

Sistem penilaian otomatis (*Automatic Scoring System* biasa disingkat *Automatic Scoring*) merupakan sistem yang mampu melakukan proses konversi dari performa dalam penyelesaian tugas (umumnya di bidang pendidikan dan penelitian) menjadi berbagai level atau karakteristik kualitas kemampuan [3]. Dalam penelitian ini, yang menjadi fokus dari bidang penilaian otomatis adalah penilaian teks jawaban singkat, maka bidang terfokus yang paling sesuai adalah *Automated Essay Scoring* (AES), yang secara sederhana dapat dimaknai sebagai pemanfaatan kemampuan komputer untuk secara otomatis melakukan evaluasi dan penilaian pada kalimat yang diketik [4]. AES sendiri merupakan aplikasi dari Pengolahan Bahasa Alami (*Natural Language*

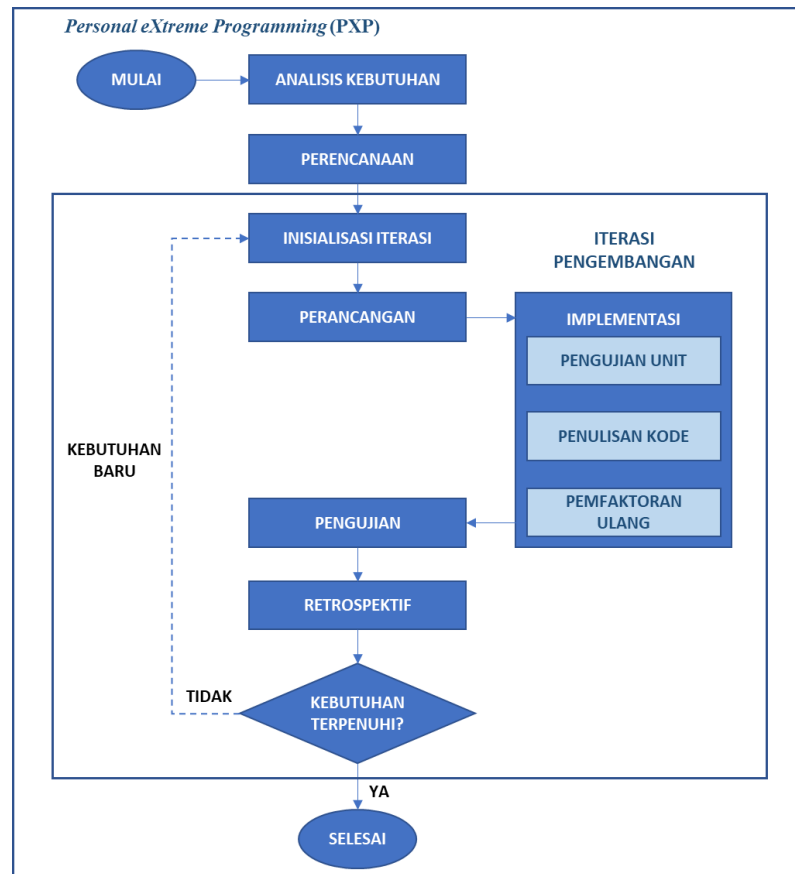
*Processing*) yang merupakan salah satu fokus bidang dari Kecerdasan Buatan (*Artificial Intelligence*) [4].

### **2.2.2. Agile Software Development Life Cycle Model**

*Agile* merupakan model daur hidup pengembangan perangkat lunak yang bersifat *incremental*. Model *agile* memberikan kemudahan dalam pengembangan perangkat lunak berskala kecil atau menengah ke bawah. Setiap tahapan *incremental*-nya sendiri berfokus untuk mengembangkan perangkat lunak secara cepat, bertahap, dan melibatkan pengguna secara langsung untuk menghasilkan luaran berkualitas tinggi. Metode *agile* memiliki beberapa variasi metode diantaranya adalah *Scrum*, *eXtreme Programming*, *Personal eXtreme Programming*, *Adaptive Software Development*, *Dynamic Systems Development Method*, dan *Agile Modeling* [19A].

### **2.2.3. Personal eXtreme Programming**

*Personal eXtreme Programming* (PXP) merupakan salah satu metode dari model SDLC *agile*, yang merupakan pengembangan dari metode terdahulunya yaitu *eXtreme Programming*, yang menitikberatkan pada sinergi sesama anggota tim pengembang berlingkup kecil. Untuk meningkatkan performa khususnya pada pengembang individu, PXP lebih menguntungkan karena pengembang dapat menentukan cara dan waktu bekerjanya sendiri, sehingga pengembang lebih mudah dalam melacak serta memprediksi perubahan yang akan terjadi [5DP]. Metode ini dipilih karena laju prosesnya yang relatif cepat, cocok untuk pengembangan perangkat lunak skala menengah ke bawah, dan fleksibilitasnya yang tinggi, serta pengembang tidak diharuskan untuk melakukan dokumentasi perencanaan matang secara menyeluruh terkait kebutuhan perangkat lunak yang bisa saja tidak dapat langsung diidentifikasi saat mengawali pengembangan.



PXP menuntut pengembang untuk bertanggung jawab pada setiap tugas dan perubahan yang terjadi. PXP memiliki beberapa tahapan dalam proses pengembangannya [21A]. Tahapan-tahapan tersebut dapat dilihat pada gambar 2.1 berikut. Rincian tahapan-tahapan pada PXP adalah sebagai berikut [22A].

### 1. Analisis Kebutuhan

Analisis Kebutuhan merupakan tahapan pengembang mengumpulkan kebutuhan dengan wawancara dan diskusi dengan *client*. Kebutuhan-kebutuhan yang diperoleh dituliskan dalam bentuk *user stories*.

### 2. Perencanaan

Pengembang menyusun dan membuat *task* yang akan dilaksanakan pada setiap iterasi berdasarkan *user stories* yang telah didapatkan. Pembagian *task* dilakukan berdasarkan prioritas dari *user stories* dan estimasi waktu pengerjaan.

### 3. Inisialisasi Iterasi

Tahap ini adalah tahap awal yang dilaksanakan untuk memulai *task* yang akan

dikerjakan. Tahap ini melakukan pemilihan tugas yang akan dijadikan fokus utama dari iterasi tersebut.

#### 4. Perancangan

Tahap ini merupakan tahap untuk memodelkan modul sistem yang akan diimplementasikan selama proses iterasi. Rancangan yang dibuat pengembang hanya berdasar dari kebutuhan *client* yang diperoleh pada tahap *requirement*.

#### 5. Implementasi

Tahap ini merupakan tahap pengimplementasian setiap objek pada tahap perancangan ke menjadi fitur / unit program. Tahap ini memiliki tiga sub-tahapan yaitu : **Pengujian Unit**, **Penulisan Kode**, dan **Pemfaktoran Ulang Kode**.

#### 6. Pengujian Sistem

Tahap ini merupakan tahapan pengujian fungsionalitas semua fitur yang ada dalam sistem. Hasil pengujian disajikan sebagai *User Acceptance Test*. Pengujian dilakukan ketika pengembang telah mengirimkan *user acceptance test* kepada *user* untuk dilakukan validasi.

#### 7. Retrospektif

Tahapan ini merupakan tahapan terakhir untuk setiap iterasi. Pada tahap ini pengembang melakukan analisis terhadap waktu pengembangan, estimasi waktu pengerjaan iterasi berikutnya, penyebab kesalahan atau gangguan seperti keterlambatan dan lain sebagainya dalam suatu iterasi untuk mencegah hal serupa terulang di iterasi selanjutnya.

#### 2.2.4. Unit Testing

Unit test biasanya ditulis oleh pengembang saat mereka menulis kode untuk suatu unit tertentu. Biasanya ditulis dalam bahasa pemrograman yang sama dengan perangkat lunak dan menggunakan kerangka atau pustaka pengujian yang menyediakan alat yang diperlukan untuk membuat dan menjalankan tes. Kerangka pengujian ini sering termasuk pustaka asserasi, yang memungkinkan pengembang untuk menulis kasus uji yang memeriksa output dari unit tertentu terhadap hasil yang diharapkan. Tes biasanya dijalankan secara otomatis dan terus menerus sebagai bagian

dari proses pembangunan perangkat lunak, dan hasilnya biasanya ditampilkan dalam alat pelari tes atau alat integrasi berkelanjutan.

#### **2.2.5. *White Box Testing***

*White Box Testing* merupakan salah satu metode evaluasi perangkat lunak yang melakukan pengujian dengan kondisi seluruh struktur internal, rancangan, dan detail implementasi dapat diakses oleh penguji. [12DP] Metode ini dipilih dengan asumsi bahwa yang menjadi penguji adalah *stakeholder* dari sistem ini sendiri, yang mana dalam kasus ini dosen pembimbing, yang memiliki pemahaman mendalam akan sistem yang akan dibangun.

#### **2.2.6. *Black Box Testing***

Black box testing adalah jenis pengujian perangkat lunak di mana fungsionalitas perangkat lunak tidak diketahui. Pengujian dilakukan tanpa pengetahuan internal produk yang diuji, artinya pengujian dilakukan dari sudut pandang pengguna luar. Tujuan dari pengujian ini adalah untuk memverifikasi apakah perangkat lunak dapat berfungsi dengan benar dan memenuhi persyaratan fungsional yang telah ditentukan tanpa perlu mengetahui bagaimana kode atau sistem di dalamnya bekerja.

Metode pengujian ini bertujuan untuk menemukan cacat dalam fungsionalitas, tampilan, dan kinerja perangkat lunak. Tes ini dilakukan dengan menguji input dan output dari perangkat lunak dan memeriksa apakah output yang dihasilkan sesuai dengan yang diharapkan. Beberapa teknik *black box testing* yang umum digunakan adalah pengujian fungsional, pengujian batas, dan pengujian penggunaan yang disarankan.

Dalam praktiknya, *black box testing* sering dilakukan oleh tim pengujian yang terpisah dari tim pengembangan perangkat lunak untuk memastikan bahwa pengujian dilakukan secara independen dan obyektif. Meskipun pengujian kotak hitam dapat membantu menemukan cacat pada perangkat lunak, metode ini memiliki keterbatasan karena tidak mengetahui detail internal sistem, sehingga beberapa masalah mungkin tidak dapat ditemukan melalui pengujian kotak hitam saja.

### **2.2.7. *System Usability Scale***

*System Usability Scale* (SUS) adalah alat pengukuran standar yang digunakan untuk mengevaluasi usability (kemudahan penggunaan) dari sebuah sistem, seperti perangkat lunak atau situs web. SUS terdiri dari 10 pernyataan yang dievaluasi oleh pengguna dengan skala Likert dari 1 hingga 5, di mana 1 adalah "sangat tidak setuju" dan 5 adalah "sangat setuju".

Pernyataan tersebut berkaitan dengan aspek usability seperti kemudahan penggunaan, efisiensi, dan kepuasan pengguna. Nilai SUS kemudian dihitung dengan mengalikan total skor dari semua pernyataan dengan 2,5 untuk menghasilkan skala dari 0 hingga 100. Semakin tinggi skor SUS, semakin baik usability sistem tersebut.

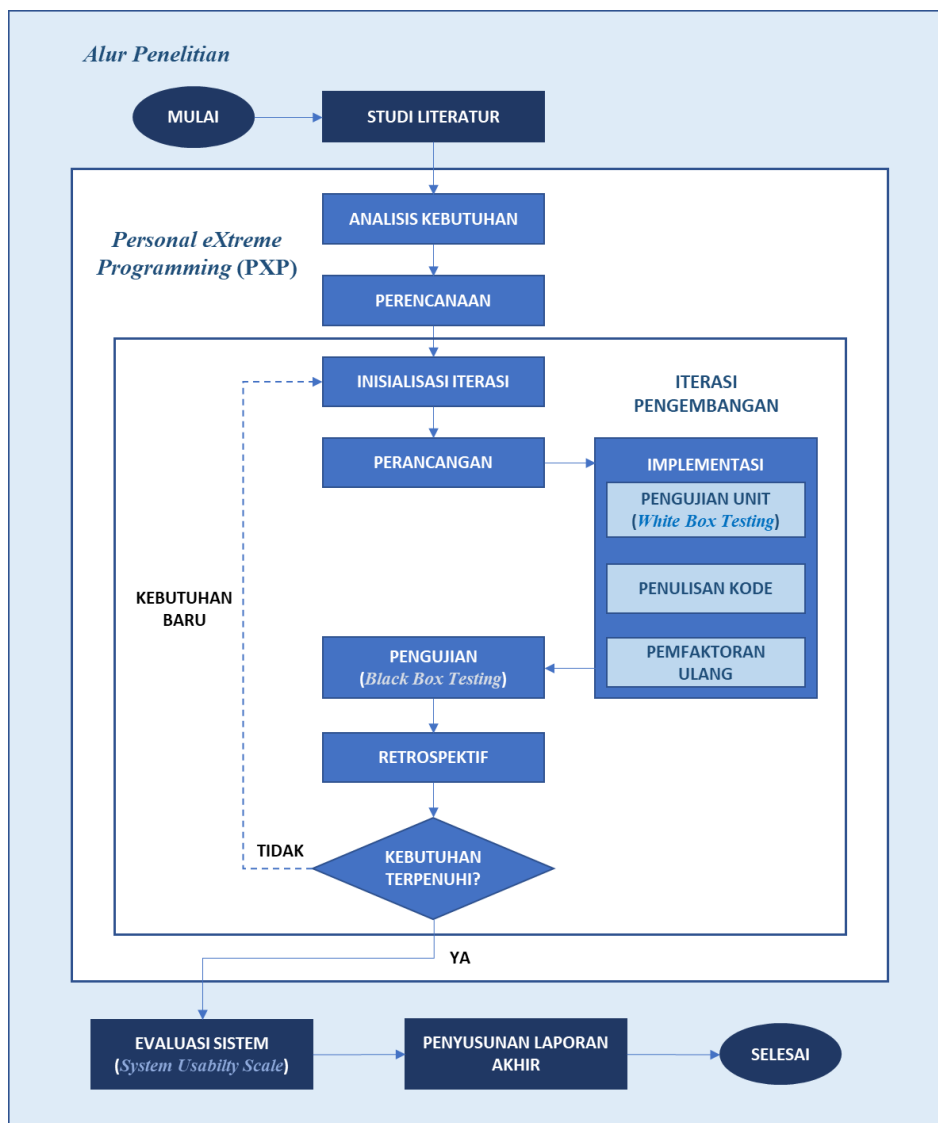
SUS telah digunakan secara luas oleh perusahaan dan peneliti untuk mengevaluasi usability berbagai produk dan layanan. Alat pengukuran ini mudah digunakan dan memberikan informasi yang bermanfaat dalam mengidentifikasi area yang perlu ditingkatkan dalam hal usability.

## BAB III

### METODE PENELITIAN

#### 3.1. Alur Penelitian

Alur penelitian merupakan tahapan pelaksanaan yang disusun untuk membantu mempermudah jalannya penelitian. Alur penelitian ini dituliskan dalam bentuk flowchart atau diagram alir yang menggambarkan semua tahapan dari awal hingga akhir. Diagram alir dapat dilihat dibawah ini :



Alur penelitian ini seperti yang ditunjukkan pada diagram alir di atas dimulai dari studi literatur lalu masuk ke tahapan PXP yaitu analisis kebutuhan, perencanaan, inisialisasi iterasi, perancangan, implementasi, pengujian integrasi, retrospektif, evaluasi sistem, penyusunan laporan akhir dan selesai.

### 3.2. Penjabaran Langkah Penelitian

Penelitian ini dilakukan dengan langkah-langkah yang sudah digambarkan pada gambar 3.1. Berikut merupakan uraian dari setiap langkah dalam alur penelitian :

#### 3.2.1. Studi Literatur

Pengembangan aplikasi AES berbasis web ini membutuhkan pemahaman teoritis mengenai bidang-bidang yang terkait dalam penelitian ini. Studi literatur dilakukan dengan mempelajari berbagai referensi baik dari jurnal, buku, dan situs-situs terpercaya. Pemahaman pengembang terhadap teoritis yang lebih baik diharapkan dapat membuat pelaksanaan penelitian yang dilakukan menjadi tepat guna, sehingga berbagai kesulitan dalam proses pengembangan dapat diatasi.

Jurnal yang dijadikan referensi dalam penelitian ini, beberapa diantaranya adalah; “Implementasi Metode Personal Extreme Programming dalam Pengembangan Sistem Manajemen Transaksi Perusahaan (Studi Kasus: CV. Tadjoe Sinar Group)” oleh Muhammad Ulfi dan rekan, dan “Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information System” oleh Gita Indah Marthasari dan rekan. Sementara beberapa buku yang dijadikan acuan utama dalam penelitian ini adalah buku yang berjudul “*Software Development, Design and Coding*”, yang ditulis oleh J. F. Dooley. Selain itu penelitian ini juga menggunakan salah satu dokumen tesis berjudul “The anatomy of the modern window manager”, oleh M. v. Deurzen.

#### 3.2.2. Metode PXP (1) : Analisis Kebutuhan

Analisis kebutuhan merupakan tahap pertama dalam metode PXP. Tahapan ini dijadikan pengembang untuk mengumpulkan kebutuhan-kebutuhan yang akan dituangkan kedalam sistem. Pengumpulan kebutuhan ini dilakukan dengan wawancara dan diskusi bersama pihak SMPN 10 Kotabumi. Kebutuhan-kebutuhan yang diperoleh dari hasil wawancara dituliskan dalam bentuk *user stories*.

#### 3.2.3. Metode PXP (2) : Perencanaan

Pada tahap perencanaan ini, pengembang menyusun tugas-tugas yang akan dikerjakan dalam setiap iterasi. Penyusunan tugas dilakukan berdasarkan *user stories* yang telah diperoleh. Pengembang menyusun kebutuhan berdasarkan estimasi waktu pengerjaan dan prioritas. Penyusunan tugas-tugas yang dilakukan pengembang disebut



dengan *practice planning game* [23A]. Penjabaran terkait penjabaran iterasi ini dapat dilihat pada rumus 3.1.

$$\text{Total Iterasi} = \frac{\text{Total stories point}}{\text{Velocity}}$$

Rumus 3.1. Total Iterasi

#### 3.2.4. Metode PXP (4) : Inisialisasi Iterasi

Inisiasi iterasi merupakan tahapan awal sebelum sebuah iterasi dimulai. Iterasi dimulai dengan pemilihan tugas yang menjadi fokus utama yang akan dikerjakan dari iterasi tersebut. Tugas yang dipilih diperoleh dari hasil perencanaan pada tahap sebelumnya yaitu tahapan perencanaan. Inisiasi iterasi ini akan menentukan nilai *velocity* untuk setiap iterasi.

#### 3.2.5. Metode PXP (5) : Perancangan

Tahapan perancangan, pengembang membuat model rancangan yang akan diimplementasikan selama proses iterasi. Desain yang dirancang hanya memenuhi kebutuhan pengguna yang diperoleh pada tahap analisis kebutuhan. Rancangan yang dibuat oleh pengembang merupakan *use case diagram* yaitu skema rancangan *prototype* aktor dan tugas yang dilakukan dari iterasi yang dijalankan [23A].

#### 3.2.6. Metode PXP (6) : Implementasi

Implementasi merupakan tahapan mengeksekusi desain yang dibuat pada tahap perancangan kedalam kode program sehingga dapat dipergunakan menjadi sistem pengadaan di Dinas Pertanian Toba. Tahapan implementasi memiliki tiga tahap yaitu *Unit Testing*, *Code Generation*, dan *Code Refactoring*. *Unit testing* merupakan pengujian fungsionalitas *code program* dimana sebagian *code program* dituliskan oleh pengembang di awal tahap pengembangan lalu dilakukan pengujian. *Unit testing* melakukan pengujian otomatis menggunakan *library PHP unit*. *Code generation* adalah *code program* setiap fitur yang telah lulus *unit testing* lalu dilanjutkan dengan melengkapi *code program* hingga selesai. Tahap terakhir adalah *refactoring* atau optimasi *code program* [23].

#### 3.2.7. Metode PXP (7) : Pengujian Integrasi

Pengujian sistem merupakan pengujian fungsionalitas yang dilakukan terhadap fitur-fitur yang telah diimplementasikan dari setiap iterasi. Pengujian dalam metode

*personal extreme programming* (PXP) ini dilakukan oleh pengguna dan hasil pengujian disajikan dalam bentuk *User Acceptance Test*. Pengujian dilakukan ketika sistem yang dibangun sudah di hosting terlebih dahulu. Dokumen *User Acceptance Test* diberikan kepada pengguna untuk proses validasi. Pengguna akan memberikan verifikasi terkait fungsi dari sistem yang diuji telah sesuai atau tidak sesuai dengan yang diinginkan.

### **3.2.8. Metode PXP (8) : Restropektif**

Retrospektif adalah tahapan terakhir dari proses iterasi. Pengembang melakukan analisis terhadap pengembangan sistem baik dari kesesuaian estimasi waktu pengerjaan, kendala yang menyebabkan keterlambatan, dan lain sebagainya. Analisis dilakukan untuk mencegah hal yang tersebut terulang kembali pada iterasi selanjutnya.

### **3.2.9. Evaluasi Sistem**

Pengujian sistem merupakan pengujian fungsionalitas yang dilakukan terhadap fitur-fitur yang telah diimplementasikan dari setiap iterasi. Pengujian dalam metode *personal extreme programming* (PXP) ini dilakukan oleh pengguna dan hasil pengujian disajikan dalam bentuk *User Acceptance Test*. Pengujian dilakukan ketika sistem yang dibangun sudah di hosting terlebih dahulu. Dokumen *User Acceptance Test* diberikan kepada pengguna untuk proses validasi. Pengguna akan memberikan verifikasi terkait fungsi dari sistem yang diuji telah sesuai atau tidak sesuai dengan yang diinginkan.

### **3.2.10. Penyusunan Laporan Akhir**

Penulisan laporan akhir adalah tahapan menuangkan hasil penelitian kedalam laporan. Laporan akhir ini akan dijadikan salah satu bukti dan syarat bahwa peneliti telah selesai melakukan penelitian terkait pengembangan aplikasi *Automatic Scoring* berbasis web.

## **3.3. Alat dan Bahan Tugas Akhir**

Perancangan dan pembangunan aplikasi yang dilakukan dalam penelitian ini membutuhkan alat dan bahan yang digunakan peneliti untuk menunjang penelitian.

### 3.3.1. Alat

Alat yang digunakan dalam penelitian sistem informasi pada Kantor Dinas Pertanian Kabupaten Toba adalah sebagai berikut:

1. *Software*
  - a. *Microsoft word*
  - b. *Microsoft visio*
  - c. *Sistem Operasi Windows 10*
  - d. *Visual studio code*
  - e. *Xampp*
  - f. *MySQL*
2. *Hardware*
  - a. *Laptop dengan prosesor intel celeron dengan RAM 4GB*
  - b. *Printer*
  - c. *Flashdisk*
  - d. *Smartphone*

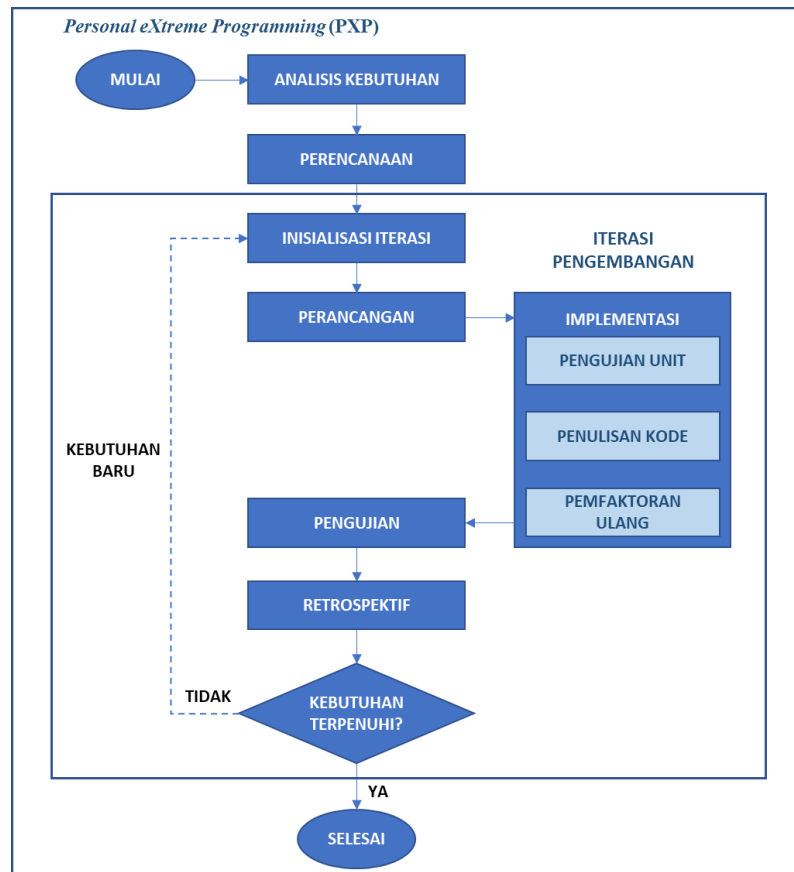
### 3.3.2. Bahan

Bahan penelitian yang digunakan peneliti adalah hasil dari wawancara dan observasi yang dilakukan. Bahan-bahan penelitian yang digunakan adalah sebagai berikut:

- a. Data penduduk yang merupakan anggota kelompok tani.
- b. Data nama-nama ketua kelompok tani.
- c. Data daftar bantuan yang akan dibagikan.
- d. Proposal yang pernah diajukan.

### 3.4. Metode Tugas Akhir

Metode penelitian yang dilakukan dalam pengembangan aplikasi *Automatic Scoring* berbasis web ini adalah *Personal Extreme Programming* (PXP). Metode PXP memiliki tahapan analisis kebutuhan, perencanaan, inisiasi iterasi, perancangan, implementasi, pengujian sistem dan retrospektif. Metode ini dapat dilihat pada gambar 3.3 berikut.



Tahapan metode PXP pada gambar 3.2 akan dijabarkan oleh pengembang sesuai dengan setiap proses yang dilakukan dalam pengembangan aplikasi *Automatic Scoring* berbasis web. Berikut adalah penjabaran tahapan metode PXP :

#### **3.4.1. Analisis Kebutuhan**

#### **3.4.2. Perencanaan**

#### **3.4.3. Insialisasi Iterasi**

#### **3.4.4. Perancangan**

#### **3.4.5. Implementasi**

#### **3.4.6. Pengujian Integrasi**

#### **3.4.7. Restropektif**

#### **3.5. Rancangan Pengujian**

1	3KP
2	4KP
3	6DL
4	7DL
5	8DL
6	9DL
7	10DL
8	11DL
9	12DL
10	13DL
11	14DL
12	SRI
13	AZIZAH
14	IYAWA