

**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

DAFTAR ISI

DAFTAR ISI.....	2
BAB I PENDAHULUAN.....	5
1.1 Latar Belakang Masalah	5
1.2 Rumusan Masalah.....	8
1.3 Tujuan Penelitian	8
1.4 Batasan Masalah	8
1.5 Manfaat Penelitian	9
1.6 Sistematika Penulisan	9
1.6.1 Bab I Pendahuluan.....	10
1.6.2 Bab II Tinjauan Pustaka	10
1.6.3 Bab III Metode Penelitian	10
1.6.4 Bab IV Hasil Penelitian dan Pembahasan	10
1.6.5 Bab V Kesimpulan dan Saran.....	10
BAB II TINJAUAN PUSTAKA	11
2.1 Tinjauan Pustaka.....	11
2.2 Dasar Teori	13
2.2.1. Aplikasi.....	13
2.2.2. <i>Automatic Scoring</i>	13
2.2.2. <i>Agile Software Development Life Cycle Model</i>	14
2.2.3. <i>Personal eXtreme Programming</i>	14
2.2.4. <i>Unified Modelling Language (UML)</i>	18
2.2.5. <i>Class Diagram</i>	18
2.2.6. <i>Use Case Diagram</i>	18
2.2.7. <i>Sequence Diagram</i>	18

2.2.8.	<i>Activity Diagram</i>	18
2.2.9.	<i>Entity Relationship Diagram (ERD)</i>	19
2.2.10.	<i>MoSCoW</i>	19
2.2.11.	<i>Unit Testing</i>	19
2.2.12.	<i>Black Box Testing</i>	20
2.2.13.	<i>System Usability Scale</i>	20
BAB III METODE PENELITIAN		23
3.1.	Alur Penelitian	23
3.2.	Penjabaran Langkah Penelitian	23
3.2.1.	Studi Literatur	24
3.2.2.	Metode PXP (1) : Analisis Kebutuhan	24
3.2.3.	Metode PXP (2) : Perencanaan	24
3.2.4.	Metode PXP (4) : Inisialisasi Iterasi	25
3.2.5.	Metode PXP (5) : Perancangan	25
3.2.6.	Metode PXP (6) : Implementasi	25
3.2.7.	Metode PXP (7) : Pengujian Sistem	25
3.2.8.	Metode PXP (8) : Restropektif	26
3.2.9.	Evaluasi Sistem	26
3.2.10.	Penyusunan Laporan Akhir	26
3.3.	Alat dan Bahan Tugas Akhir	26
3.3.1.	Alat	26
3.3.2.	Bahan	27
3.4.	Metode Tugas Akhir	27
3.4.1.	Analisis Kebutuhan	28
3.4.2.	Perencanaan	31
3.4.3.	Insialisasi Iterasi	33

3.4.4.	Perancangan.....	34
3.4.5.	Implementasi	39
3.4.6.	Pengujian Sistem	40
3.4.7.	Restrospektif.....	40
3.5.	Evaluasi Akhir Aplikasi.....	40
DAFTAR PUSTAKA		41
LAMPIRAN.....		43

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 tugas dan ujian. Tugas dan 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.

SMP Negeri 10 Kotabumi yang berlokasi di Jl. Alamsyah Ratu Prawira Negara, Kelurahan Kelapa Tujuh, Kecamatan Kotabumi Selatan, Kabupaten Lampung Utara, Provinsi Lampung tersebut merupakan salah satu sekolah yang menghadapi situasi tersebut. Berdasarkan hasil diskusi dengan Wakil Kepala Bidang Kurikulum SMP Negeri 10 Kotabumi, Ibu Eny Ros (rekaman wawancara dapat dilihat pada lampiran) peneliti mendapati bahwa SMP Negeri 10 Kotabumi sudah tidak baru lagi terhadap teknologi berupa *platform* kelas virtual seperti *Google Classroom* maupun *platform* formulir *online* seperti *Google Form*. Ibu Eny mengaku bahwa adanya *platform* sejenis sangat memudahkan bagi guru khususnya yang mengajar pada mata pelajaran yang bersifat hafalan konsep seperti ilmu pengetahuan sosial maupun bahasa.

Ibu Eny juga menyampaikan ketika masa pandemi COVID lalu, SMP Negeri 10 cukup kewalahan dalam beradaptasi dengan proses belajar mengajar dalam jaringan,

sehingga media yang digunakan untuk menunjang kegiatan belajar mengajar saat itu hanyalah media sosial percakapan *WhatsApp*, yang dirasa paling mudah dijangkau dan mudah digunakan. Namun beliau juga menyampaikan bahwa salah satu kelas di SMP Negeri 10 Kotabumi, saat ini sedang dalam masa uji coba, dimana untuk Penilaian Tengah Semester (PTS) dan Penilaian Akhir Semester (PAS) kelas tersebut akan dilaksanakan secara *online* dengan menggunakan layanan suatu penyedia jasa sistem informasi. Beliau juga menerangkan bahwa dalam beberapa waktu terakhir, terdapat beberapa penyedia jasa sistem informasi yang menawarkan layanan mereka, khususnya yang berbentuk *platform* ujian *online*. Namun dari apa yang dapat dirasakan pada praktiknya, menurut beliau beragam layanan tersebut masih belum cukup memudahkan, karena selain hanya mendukung penilaian otomatis pada soal bertipe pilihan ganda, namun juga tidak cukup memenuhi kebutuhan dari SMP Negeri 10, seperti fitur pengolahan data hasil ujian siswa, maupun fitur analisis data statistik dari hasil ujian tersebut, yang dirasa sangat dibutuhkan untuk mengevaluasi cara belajar maupun materi yang diajarkan atau diujikan. Untuk menjawab situasi tersebut, perlu dikembangkan dan diteliti suatu aplikasi berbasis *web* yang dilengkapi dengan suatu model kecerdasan buatan berupa sistem penilaian otomatis untuk tugas dan ujian di SMP Negeri 10 Kotabumi.

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) [5]. 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 [5]. 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 [6]. 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 [7]–[10].

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, akan digunakan metode evaluasi *Unit 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 performa aplikasi berbasis web yang dikembangkan dalam menggunakan kemampuan model AES yang telah dikembangkan berdasarkan hasil evaluasi menggunakan metode *Black Box Testing* dan metode *System Usability Scale*?

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 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. Tim manajemen pengembangan, pengguna, dan admin yang dimaksud pada poin sebelumnya bisa berasal dari lingkup Program Studi Teknik Informatika ITERA (khususnya pembimbing dan penguji penelitian ini), lingkup SMP Negeri 10 Kotabumi sebagai studi kasus penelitian ini, dan peneliti sendiri.
6. Aplikasi 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 SMP Negeri 10 Kotabumi, luaran penelitian ini berupa aplikasi penilaian otomatis berbasis web sebagai media yang dapat menunjang kemudahan kegiatan belajar mengajar.
4. Bagi peneliti, sebagai sarana untuk memperdalam pengetahuan dan kemampuan di bidang teknologi informasi, khususnya pengolahan bahasa alami, dan pengembangan web.
5. 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 <i>Personal Extreme Programming</i> dalam Pengembangan Sistem Manajemen Transaksi Perusahaan (Studi Kasus: CV. Todjoe Sinar Group)	Muhammad Ulfi, Gita Indah Marthasari, Ilyas Nuryasin	2020	Metode SDLC : <i>Personal eXtreme Programming</i>	Sistem Informasi yang dapat memudahkan manajemen transaksi perusahaan	Implementasi metode SDLC akan dilakukan pada aplikasi yang memiliki kegunaan dan bidang studi kasus berbeda.
2	<i>Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information System</i>	Gita Indah Marthasari, Wildan Suharso, Frendy Ardiansyah	2018	Metode SDLC : <i>Personal eXtreme Programming</i>	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) Menggunakan Metode <i>Personal Extreme Programming</i> dengan Metode	Abdullah Faqih Septiyanto, Wildan Suharso, Ilyas Nuryasin	2020	Metode SDLC : <i>Personal eXtreme Programming</i>	Sistem informasi yang dapat menunjukkan data sederhana dan mengunggah file	Implementasi metode SDLC akan dilakukan pada aplikasi yang memiliki kegunaan dan bidang studi kasus berbeda

No	Judul	Peneliti	Tahun	Metode	Hasil	Pembeda
	Prioritas Ranking					
4	Muster: Virtual Classroom For Students Using D-Jango	Mahesh Kancharla, Govinda Sai Kamisetty, Suraj Hussain Dudekula	2022	Model SDLC : Iterative	Sistem Ruang Kelas Virtual untuk media pembelajaran daring	Pembahasan yang menjadi fokus penelitian

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)” [11], 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*” [12], 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” [13]. 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 Mahesh Kancharla, Govinda Sai Kamisetty, dan Suraj Hussain Dudekula pada tahun 2022 dengan judul penelitian “Muster: *Virtual Classroom For Students Using D-Jango*” [14], pada penelitian ini dilakukan pengembangan kelas virtual. Hasil dari penelitian ini merupakan Sistem Ruang Kelas Virtual untuk media pembelajaran daring. Namun, kekurangan dari penelitian ini adalah kurangnya pembahasan terkait metode SDLC yang digunakan.

Berdasarkan penelitian sebelumnya, penulis mengajukan penelitian mengenai Pengembangan Aplikasi dengan kemampuan AES untuk penilaian jawaban teks pada tugas dan ujian *online* di SMP Negeri 10 Kotabumi. Perbedaan penelitian yang dilakukan peneliti dengan penelitian terkait adalah pada bidang studi kasus yang digunakan peneliti. Metode yang digunakan peneliti adalah *Personal Extreme Programming* (PXP) dan fitur yang menjadi pembeda dalam penelitian ini adalah fitur penilaian otomatis menggunakan bantuan model kecerdasan buatan.

2.2 Dasar Teori

2.2.1. Aplikasi

Dalam ilmu komputer, aplikasi (aplikasi perangkat lunak) merujuk pada program komputer yang dibangun untuk melakukan tugas spesifik tertentu. Aplikasi dapat berupa perangkat lunak yang dapat diinstal di perangkat seperti komputer, telepon seluler, atau tablet. Aplikasi perangkat lunak yang baik dibangun untuk menyelesaikan tugas-tugas tertentu dengan lingkup yang spesifik serta berfokus pada peningkatan efisiensi dan produktivitas.

2.2.2. 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*

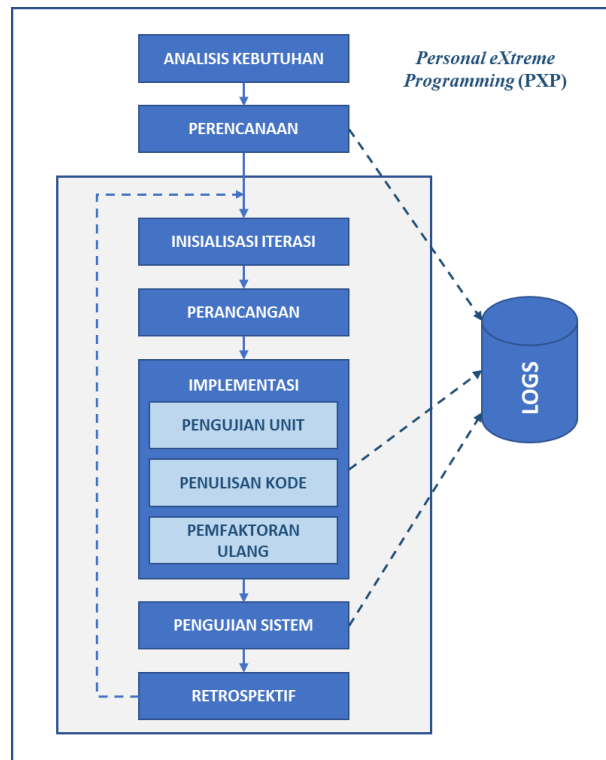
Processing) yang merupakan salah satu fokus bidang dari Kecerdasan Buatan (*Artificial Intelligence*) [4].

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* [15].

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 [7]. 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.



Gambar 2.2.2.1 Fase-fase dalam Metode PXP [6]

PXP menuntut pengembang untuk bertanggung jawab pada setiap tugas dan perubahan yang terjadi. PXP terdiri dari beberapa fase dalam proses pengembangan perangkat lunak. Tahapan-tahapan tersebut dapat dilihat pada gambar 2.1., dengan rincian dari tiap fasenya adalah sebagai berikut [6]:

1. Analisis Kebutuhan

Analisis Kebutuhan merupakan fase dimana pengembang mengumpulkan kebutuhan perangkat lunak melalui diskusi atau wawancara dengan *client*. Kebutuhan-kebutuhan yang diperoleh akan dituliskan dalam bentuk *user stories*.

2. Perencanaan

Pada fase perencanaan, pengembang menyusun dan membuat sekumpulan *task* yang akan dilaksanakan pada setiap iterasi berdasarkan kebutuhan yang disusun dari *user stories* yang telah didapatkan. Pembagian *task* dilakukan berdasarkan prioritas dari *user stories* dan estimasi waktu pengerjaan. Penyusunan tugas-tugas yang dilakukan pengembang disebut dengan *practice planning game* [16].

Pada fase ini, pengembang mengestimasi upaya yang dibutuhkan untuk mengimplemtasikan setiap *user story*, unit yang digunakan untuk menggambarkan besar kecilnya upaya itu disebut dengan *story point*. Nilai *story point* sebuah *user story* dipengaruhi oleh kompleksitas, ukuran pekerjaan, dan ketidakpastian resiko dari *user story* tersebut. Umumnya pengembang menggunakan deret *Fibonacci* sebagai rentang nilai *story point* untuk suatu *user story*, dimulai dari 1, 2, 3, 5, 8, 13, 21. Pada praktiknya pengembang biasanya membagi suatu *story* menjadi tugas-tugas kecil, ketika *story* tersebut dirasa cukup sulit hingga memiliki nilai *story point* 20. Namun untuk *story* yang diberikan estimasi *story point* kurang dari 20 tidak akan dilakukan pemecahan kepadanya.

Selain mengestimasi nilai *story point* untuk setiap *user story*, pada fase ini pengembang juga menentukan nilai *velocity*. Nilai *velocity* ini merupakan nilai estimasi banyaknya *story point* yang dapat diselesaikan pengembang dalam satu kali *sprint* (aktivitas pengerjaan dalam rentang waktu kerja tertentu). Pada penelitian ini, pengembang memiliki *velocity* sebesar 5 untuk setiap iterasi dengan *sprint-time* selama 10 hari kerja.

Setelah mengestimasi nilai *story point* dan nilai *velocity*, pengembang kemudian akan menghitung nilai Total Iterasi menggunakan rumus 2.1.

$$\boxed{\text{Total Iterasi} = \frac{\text{Total stories point}}{\text{Velocity}}} \quad (2.1)$$

Keterangan rumus 2.1 :

Total Iterasi = Jumlah iterasi yang terbentuk untuk semua *user stories*.

Total stories point = Jumlah total dari estimasi waktu pengerjaan *user stories*.

Velocity = Waktu yang dibutuhkan untuk mengerjakan setiap iterasi.

Pada tahap ini juga berbagai keputusan penting akan diambil seperti penentuan bahasa pemrograman, *framework* pengembangan, serta model aplikasi yang akan dikembangkan.

3. Inisialisasi Iterasi

Fase ini adalah fase awal yang dilaksanakan untuk memulai *task* yang akan dikerjakan. Fase ini melakukan pemilihan tugas yang akan dijadikan fokus utama dari iterasi yang sedang berlangsung.

4. Perancangan

Fase perancangan sendiri merupakan fase dimana pengembang memodelkan modul dan kelas yang nantinya akan diimplementasikan di dalam sistem selama proses iterasi yang berlangsung. Rancangan yang dibuat pengembang hanya mengacu kepada kebutuhan *client* yang diperoleh pada tahap *requirement*. Pengembang diberi kebebasan untuk memilih metode perancangan apa yang paling tepat, namun pada PXP sangat disarankan untuk membuat proses ini sesederhana mungkin.

4. Implementasi

Fase ini merupakan fase pengimplementasian setiap objek pada tahap perancangan menjadi fitur / unit program. Fase ini memiliki tiga sub-fase yaitu: **Pengujian Unit**, **Penulisan Kode**, dan **Pemfaktoran Ulang Kode**. Untuk dapat melanjutkan ke fase berikutnya dari fase implementasi ini, kode program yang dibuat harus dapat dijalankan tanpa error dan lolos sub-fase pengujian unit.

5. Pengujian Sistem

Pada fase ini dilakukan pengujian fungsionalitas semua fitur yang ada dalam sistem. Pada penelitian ini hasil pengujian disajikan melalui *Black Box Testing*. Pada fase ini, seluruh kesalahan pada sistem dicatat dan diperbaiki.

6. Retrospektif

Fase ini merupakan fase terakhir untuk setiap iterasi. Pada fase ini pengembang melakukan analisis terhadap estimasi waktu pengembangan yang dibuat ketika fase perencanaan dengan waktu pengembangan sesungguhnya pada iterasi yang sedang berlangsung, dan menganalisis potensi hambatan untuk menekan kemungkinan terjadinya kesalahan dalam memperkirakan waktu pengerjaan iterasi pada proyek berikutnya. Pada fase retrospektif ini pula, penyebab kesalahan atau gangguan dalam suatu iterasi dicatat dan dianalisis untuk mencegah hal serupa terulang di iterasi selanjutnya.

Di sepanjang seluruh proses, pengembang mengelola dan memelihara sejumlah catatan (*log*) yang berisi informasi terkait perencanaan tiap *task* dan durasi sesungguhnya ketika pelaksanaan *task* tersebut, jumlah dan detail kesalahan yang terjadi, serta saran peningkatan untuk iterasi maupun proyek berikutnya [6].

2.2.4. *Unified Modelling Language (UML)*

UML merupakan bahasa standar yang digunakan untuk menulis / menotasikan cetak biru perangkat lunak. UML dapat digunakan untuk tujuan visualisasi, penentuan, pembangunan, dan dokumentasi hal-hal yang berhubungan erat dengan sistem perangkat lunak. UML umumnya dinotasikan dalam bentuk diagram seperti : *Class Diagram*, *Use Case Diagram*, *Sequence Diagram*, *Activity Diagram*, dan sebagainya [17].

2.2.5. *Class Diagram*

Class Diagram merupakan salah diagram yang paling banyak digunakan dalam pemodelan sistem berbasis orientasi-kepada-objek. Diagram ini memvisualisasikan sekumpulan kelas dan antarmukanya, serta hubungan antara kelas-kelas tersebut [17].

2.2.6. *Use Case Diagram*

Use Case Diagram merupakan notasi dalam UML yang fokusnya berpusat pada pemodelan perilaku suatu sistem, subsistem, atau kelas, yang masing-masing menunjukkan sekumpulan kasus penggunaan, aktor, dan hubungan satu sama lain antara mereka [17].

2.2.7. *Sequence Diagram*

Sequence Diagram merupakan salah satu notasi yang bertipe ‘diagram interaksi’ dalam UML. Notasi ini menekankan pada pengurutan dari waktu pengiriman dan penerimaan data antar objek yang terjadi di dalam suatu sistem perangkat lunak [17].

2.2.8. *Activity Diagram*

Activity Diagram, merupakan notasi lainnya dari UML, yang digunakan untuk memodelkan aspek dinamis dari suatu sistem, seperti langkah berurutan dalam proses komputasi, maupun alur yang dilalui suatu objek dalam sistem, dari suatu kondisi ke kondisi lainnya dalam aliran kontrol sistem tersebut [17].

2.2.9. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram, merupakan notasi yang digunakan dalam pemodelan basis data untuk memvisualisasikan sekumpulan entitas, atributnya, dan relasi antara mereka.

2.2.10. *MoSCoW*

MoSCoW merupakan akronim dari *Must have*, *Should Have*, *Could Have*, dan *Won't Have*. MoSCoW merupakan aturan yang digunakan dalam menentukan prioritas pada kebutuhan dalam pengembangan perangkat lunak. *Must have* adalah bagian paling dasar dari sistem yang dibangun. *Should have* adalah bagian penting dari sistem yang dibangun namun ada jangka waktu tertentu yang digunakan untuk pengerjaannya. *Could have* adalah bagian sistem yang dapat dikeluarkan dari rencana apabila waktu yang dimiliki tidak ada lagi. *Won't have* adalah prioritas yang tidak terlalu dibutuhkan untuk sistem yang akan dibangun.

2.2.11. *Unit Testing*

With unit testing, you're testing individual methods and classes, but you're generally not testing larger configurations of the program. You're also not usually testing interfaces or library interactions—except those that your method might actually be using. Because you're doing unit testing, you know how all the methods are written, what the data is supposed to look like, what the method signatures are, and what the return values and types should be. This is known as white-box testing. It should really be called transparent-box testing, because the assumption is you can see all the details of the code being tested.

Dengan *Unit Testing*, pengembang melakukan pengujian terhadap metode atau class spesifik secara terpisah satu per satu. Dengan melakukan *unit testing*, pengembang diharuskan untuk memahami bagaimana metode dalam program ditulis, bagaimana bentuk data yang diolah dalam metode tersebut, apa keunikan dan tugas khusus dari metode tersebut, dan seperti apa tipe dan nilai luaran yang diharapkan dari metode tersebut [5].

Unit test 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 pengujian secara otomatis dan terus menerus sebagai bagian dari proses pembangunan perangkat lunak.

2.2.12. *Black Box Testing*

Black box testing adalah jenis pengujian perangkat lunak di mana fungsionalitas perangkat lunak tidak diketahui. Pengujian dilakukan tanpa pengetahuan internal terkait 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 masukan dan keluaran dari perangkat lunak dan memeriksa apakah masukan yang dihasilkan sesuai dengan yang diharapkan. 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 [5].

2.2.13. *System Usability Scale*

Metode *System Usability Scale* (SUS) merupakan metode yang menggunakan kuisioner untuk mengukur persepsi kegunaan perangkat lunak. Pengujian metode ini dilakukan setelah sistem telah selesai dibangun dan dikembangkan [18]. Metode SUS berisi 10 pertanyaan yang diberikan skala 1 sampai dengan skala 5. Pengertian skala yang dimaksud adalah 1 artinya sangat tidak setuju dan juga 5 yang mengartikan sangat setuju [19]. Metode SUS memiliki kriteria yang digunakan untuk mengelompokkan hasil kuisioner yang diperoleh dari responden. Adapun kriteria metode SUS adalah sebagai berikut [18]:

Tabel 2.0.1 Kriteria Metode SUS

SUS	Tingkatan	Kriteria
$x > 80,3$	A	Sangat Baik
$68 < x \leq 80,3$	B	Baik
$x = 68$	C	Cukup
$51 \leq x < 68$	D	Kurang
$X < 51$	E	Sangat Kurang

Pada tabel 2.1 telah dijelaskan bahwa hasil SUS diatas $> 80,3$ maka kriterianya sangat baik. Nilai terendah lebih kecil < 51 maka kriterianya sangat kurang.

Pertanyaan yang diberikan kepada responden akan dihitung skornya dengan menggunakan rumus metode SUS. Adapun rumus yang digunakan untuk menghitung skor dalam metode SUS dapat dilihat pada rumus 2.1

$$\tilde{x} = \frac{\sum x}{n} \quad (2.2)$$

Keterangan rumus 2.1:

\tilde{x} = Skor rata-rata

$\sum x$ = Jumlah Skor SUS

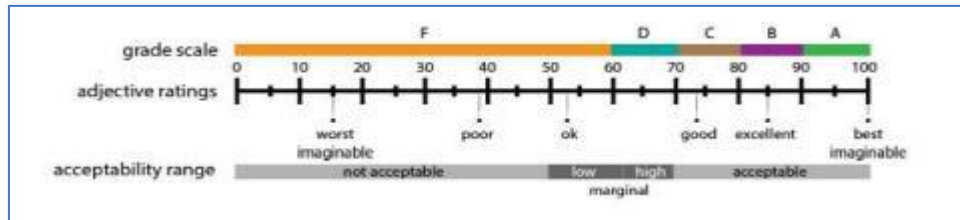
n = Jumlah responden

Perhitungan Skor dalam metode SUS menggunakan beberapa tahapan, diantaranya adalah sebagai berikut [20]:

Dari setiap soal yang bernomor ganjil, maka skor dikurangi 1 (X-1).

1. Pernyataan bernomor genap, nilainya akan dikurangi dengan 5 (5-X).
2. Hasil berkisar dari 0 hingga 4 atau bisa disimpulkan bahwa 4 adalah nilai paling positif.
3. Jumlahkan skor nilai untuk jawaban bernomor genap dan bernomor ganjil lalu kalikan jumlah proporsinya dengan 2.5.
4. Menghitung rata-rata jawaban instrument.

Penentuan hasil perhitungan SUS merujuk pada kepada tiga aspek utama, yaitu akseptabilitas (*acceptability range*), skala nilai (*adjective ratings*), dan tingkat kriteria sistem (*grade scale*). Akseptabilitas merupakan aspek yang menentukan penerimaan suatu sistem dalam kondisi seperti tidak dapat diterima (*not acceptable*), marjinal (*low or high*), dan dapat diterima (*acceptable*). Skala nilai digunakan untuk menentukan tingkat kualitas aplikasi yang terdiri dari tingkat A,B,C,D, dan E. Sementara tingkat kriteria sistem adalah yang menentukan kebergunaan sistem yang meliputi 6 tingkatan yaitu terburuk (*worst imaginable*), buruk (*poor*), oke, baik (*good*), sangat baik (*excellent*), dan istimewa (*best imaginable*) [21]. Pedoman umum interpretasi SUS score dapat dilihat pada gambar 2.2



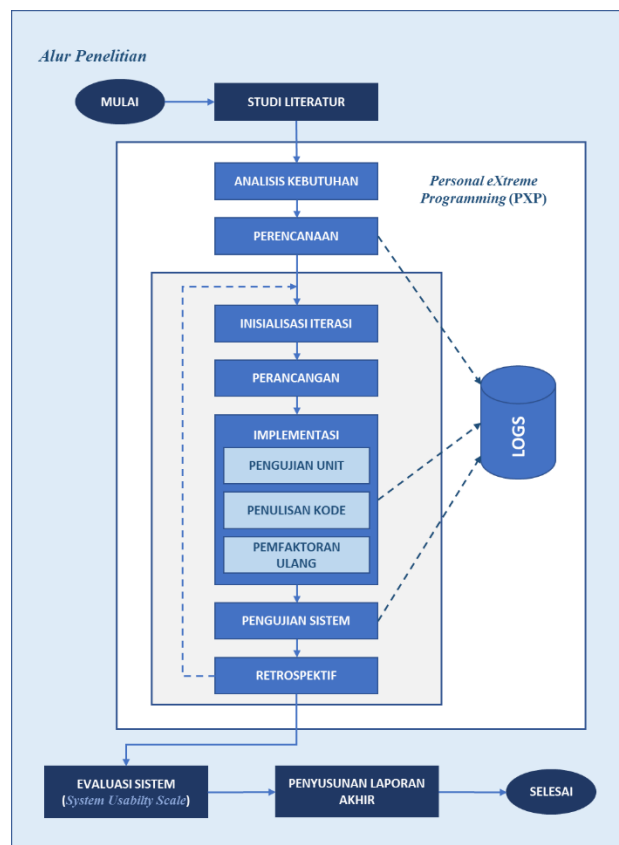
Gambar 2.2 Pedoman umum interpretasi SUS Score [22]

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 :



Gambar 3.1. Alur Penelitian

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 sistem, 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. Todjoe 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 (*tasks*) yang akan dikerjakan dalam setiap iterasi. Penyusunan tugas dilakukan berdasarkan kebutuhan fungsional dan non-fungsional yang dirumuskan dari *user stories* yang telah diperoleh. Pengembang menyusun kebutuhan berdasarkan estimasi waktu pengerjaan dan prioritas.

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 [16].

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* [16].

3.2.7. Metode PXP (7) : Pengujian 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.

3.2.8. 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 pengembangan aplikasi ini adalah sebagai berikut:

1. *Software*
 - a. *Microsoft word*

- b. *Sistem Operasi Windows 10*
- c. *Visual Studio Code*
- d. *MySQL*

2. *Hardware*

- a. *Laptop* dengan prosesor intel *celeron* dengan RAM 4GB
- b. *Printer*
- c. *Flashdisk*
- d. *Smartphone*

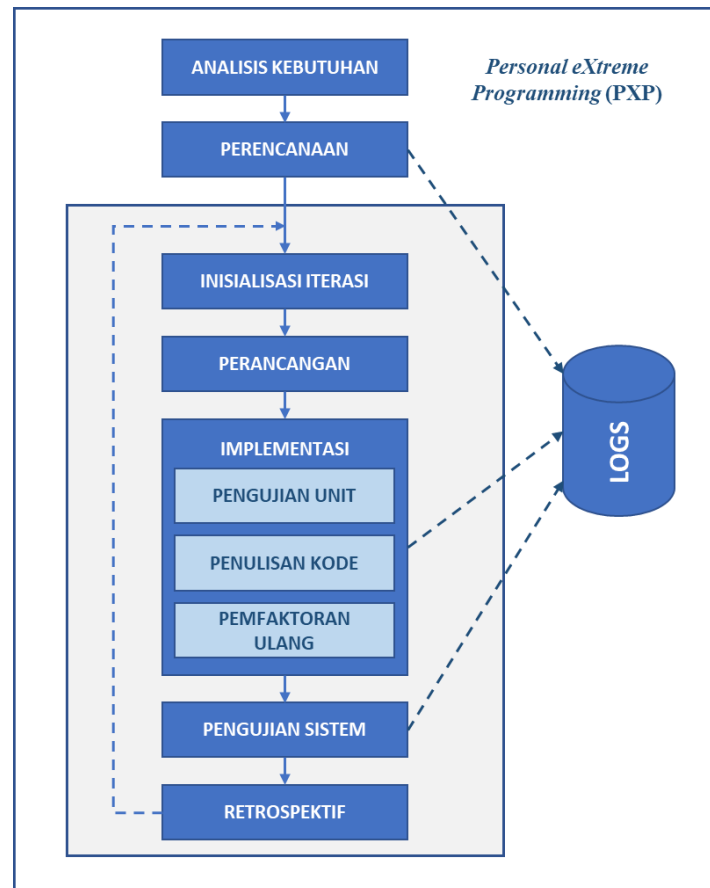
3.3.2. **Bahan**

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

- a. Data guru dan siswa di SMP Negeri 10 Kotabumi
- b. Data mata pelajaran yang akan dimasukkan ke dalam aplikasi yang dikembangkan.
- c. Data tugas dan ujian dengan soal beserta jawabannya yang akan dimasukkan ke dalam aplikasi yang dikembangkan.

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 3.2.



Gambar 3.4 Fase – fase dalam metode PXP

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

Tahap analisis kebutuhan ini diperoleh dari client melalui wawancara dan diskusi bersama ibu Eny Ros selaku Wakil Kepala Bidang Kurikulum di SMP Negeri 10 Kotabumi. Hasil yang didapatkan oleh pengembang dituliskan dalam bentuk user stories dengan format “Sebagai <jenis pengguna>, < saya ingin melakukan tindakan sesuatu>, sehingga <mendapatkan manfaat dari tindakan tersebut>”. Deskripsi dari pengguna atau aktor dapat dilihat pada tabel 3.1.

Tabel 3.1 Pengguna Sistem

No	Aktor	Deskripsi
1	Admin	Seseorang yang mengelola aplikasi yang dikembangkan
2	Kepala Sekolah	Seseorang yang menetapkan rencana, merumuskan kebijakan, dan memimpin penyelenggaraan program di SMP Negeri 10 Kotabumi.
3	Guru	Seseorang yang bertugas mengajar mata pelajaran tertentu dan mengevaluasi progres belajar siswa melalui tugas dan ujian.
4	Siswa	Seseorang yang meneri pengajaran dari guru dan bertanggung jawab untuk menyelesaikan tugas dan ujian yang diberikan oleh guru.

Berdasarkan hasil analisis kebutuhan untuk pengembangan aplikasi *Automatic Scoring* berbasis web untuk jawaban teks pada tugas dan ujian di SMP Negeri 10 Kotabumi menggunakan metode *Personal Extreme Programming* (XP) dituliskan *user stories* yang dapat dilihat pada Tabel 3.2.

Tabel 3.2 User Stories

Kode User Stories	User Stories
Story-01	Sebagai guru saya ingin dapat melakukan pengelolaan kelas sehingga saya bisa membagikan materi maupun tugas dan ujian kepada siswa.
Story-02	Sebagai guru saya ingin dapat melakukan pengelolaan tugas dan ujian bagi siswa, beserta dengan pengelolaan pertanyaan yang mendukung berbagai tipe soal, dan aplikasi dapat melakukan penilaian otomatis terhadap jawaban siswa.
Story-03	Sebagai guru saya ingin dapat melihat dan mengunduh hasil pengerjaan tugas dan ujian oleh siswa.
Story-04	Sebagai guru saya ingin dapat menganalisis progres belajar seorang siswa berdasarkan hasil pengerjaan tugas dan ujian siswa bersangkutan.
Story-05	Sebagai guru saya ingin dapat menganalisis materi yang diajarkan dan diujikan kepada siswa berdasarkan hasil pengerjaan tugas dan ujian oleh siswa dari satu kelas atau satu angkatan.

Berdasarkan *user stories* yang ditampilkan pada Tabel 3.2 maka disusun kebutuhan fungsional dan kebutuhan nonfungsional-nya seperti yang ditunjukkan pada Tabel 3.3 dan Tabel 3.4

Tabel 3.3 Kebutuhan Fungsional

No.	Aktor	Kebutuhan Fungsional
1	Admin	Dapat melakukan login dan logout
		Dapat menambah data akun
		Dapat melihat data akun
		Dapat mengubah data akun
		Dapat menghapus data akun
		Dapat mengimpor data akun
		Dapat melihat data kelas
2	Guru	Dapat melakukan login dan logout
		Dapat membuat kelas baru
		Dapat melihat data kelas
		Dapat mengubah data kelas
		Dapat membuat materi baru
		Dapat melihat materi
		Dapat mengubah materi
		Dapat menghapus materi
		Dapat membuat tugas / ujian baru
		Dapat melihat data tugas / ujian
		Dapat mengubah data tugas / ujian
		Dapat menghapus data tugas / ujian
		Dapat membuat soal dan kunci jawaban baru
		Dapat melihat data soal dan kunci jawaban
		Dapat mengubah data soal dan kunci jawaban
		Dapat menghapus data soal dan kunci jawaban
		Dapat mengimpor data soal dan kunci jawaban
		Dapat melihat data hasil pengerjaan tugas / ujian siswa
		Dapat mengunduh data hasil pengerjaan tugas / ujian siswa
		Dapat melihat data statistik hasil pengerjaan tugas dan ujian seluruh
		Dapat melihat rekap seluruh data hasil ujian siswa
		Dapat mengunduh rekap seluruh data hasil ujian siswa
		Dapat melihat profil
		Dapat mengubah profil
3	Siswa	Dapat melakukan login dan logout
		Dapat bergabung ke kelas
		Dapat melihat materi yang dibagikan guru
		Dapat mengunduh materi yang dibagikan guru
		Dapat melihat detail ujian
		Dapat mengerjakan ujian

No.	Aktor	Kebutuhan Fungsional
		Dapat mengirimkan jawaban ujian
		Dapat melihat data hasil ujian
		Dapat melihat profil
		Dapat mengubah profil

Tabel 3.4 Kebutuhan Non-Fungsional

No.	Kebutuhan Non-fungsional
1	Sistem dapat diakses 24 jam jika dan hanya jika perangkat terhubung dengan internet.
2	Hak akses (<i>role</i>) terbagi menjadi tiga yakni admin, guru dan siswa.
3	Setiap aktor akan diarahkan pada halaman utama yang berbeda sesuai dengan <i>role</i> -nya.
4	Sistem akan selalu memvalidasi apakah akun yang masuk terdaftar pada database dan sistem akan menolak proses login jika akun tidak terdaftar.

3.4.2. Perencanaan

asdasdasdasd

1. Estimasi *User Stories*

Estimasi pengerjaan yang telah ditentukan pengembang berdasarkan tingkat kesulitan pada setiap *user stories* dapat dilihat pada tabel 3.5.

Tabel 3.5. Estimasi *User Stories*

Kode Story	User Story	Stories Point	Estimasi Waktu (Hari)
Story-01	Sebagai guru saya ingin dapat melakukan pengelolaan kelas sehingga saya bisa membagikan materi maupun tugas dan ujian kepada siswa.	3	5
Story-02	Sebagai guru saya ingin dapat melakukan pengelolaan tugas dan ujian bagi siswa, beserta dengan pengelolaan pertanyaan yang mendukung berbagai tipe soal, dan aplikasi dapat melakukan penilaian otomatis terhadap jawaban siswa.	5	10
Story-03	Sebagai guru saya ingin dapat	2	4

Kode Story	User Story	Stories Point	Estimasi Waktu (Hari)
	melihat dan mengunduh hasil pengerjaan tugas dan ujian oleh siswa.		
Story-04	Sebagai guru saya ingin dapat menganalisis progres belajar seorang siswa berdasarkan hasil pengerjaan tugas dan ujian siswa bersangkutan.	2	4
Story-05	Sebagai guru saya ingin dapat menganalisis materi yang diajarkan dan diujikan kepada siswa berdasarkan hasil pengerjaan tugas dan ujian oleh siswa dari satu kelas atau satu angkatan.	3	6
Total		15	30

2. Prioritas User Stories

Untuk prioritas dari setiap *user story* sendiri, berikut adalah estimasi pengembang menggunakan aturan MoSCoW seperti ditampilkan pada Tabel .

Tabel 3.6 Prioritas User Story

Kode Story	User Story	Stories Point	Prioritas
Story-01	Sebagai guru saya ingin dapat melakukan pengelolaan kelas sehingga saya bisa membagikan materi maupun tugas dan ujian kepada siswa.	3	<i>Must have</i>
Story-02	Sebagai guru saya ingin dapat melakukan pengelolaan tugas dan ujian bagi siswa, beserta dengan pengelolaan pertanyaan yang mendukung berbagai tipe soal, dan aplikasi dapat melakukan penilaian otomatis terhadap jawaban siswa.	5	<i>Must have</i>
Story-03	Sebagai guru saya ingin dapat melihat dan mengunduh hasil pengerjaan tugas dan ujian oleh siswa.	2	<i>Must have</i>
Story-04	Sebagai guru saya ingin dapat menganalisis progres belajar	2	<i>Should have</i>

Kode Story	User Story	Stories Point	Prioritas
	seorang siswa berdasarkan hasil pengerjaan tugas dan ujian siswa bersangkutan.		
Story-05	Sebagai guru saya ingin dapat menganalisis materi yang diajarkan dan diujikan kepada siswa berdasarkan hasil pengerjaan tugas dan ujian oleh siswa dari satu kelas atau satu angkatan.	3	Should have

3. Perencanaan Iterasi

Pada tahap ini ditentukan banyaknya iterasi yang akan dilakukan dengan perhitungan menggunakan rumus 2.1 berdasarkan nilai *velocity* pengembang, dan nilai total *story points*.

$$\text{Total Iterasi} = \frac{15}{5}$$

$$\text{Total Iterasi} = 3$$

3.4.3. Insialisasi Iterasi

Tabel 3.6 Tabel Iterasi

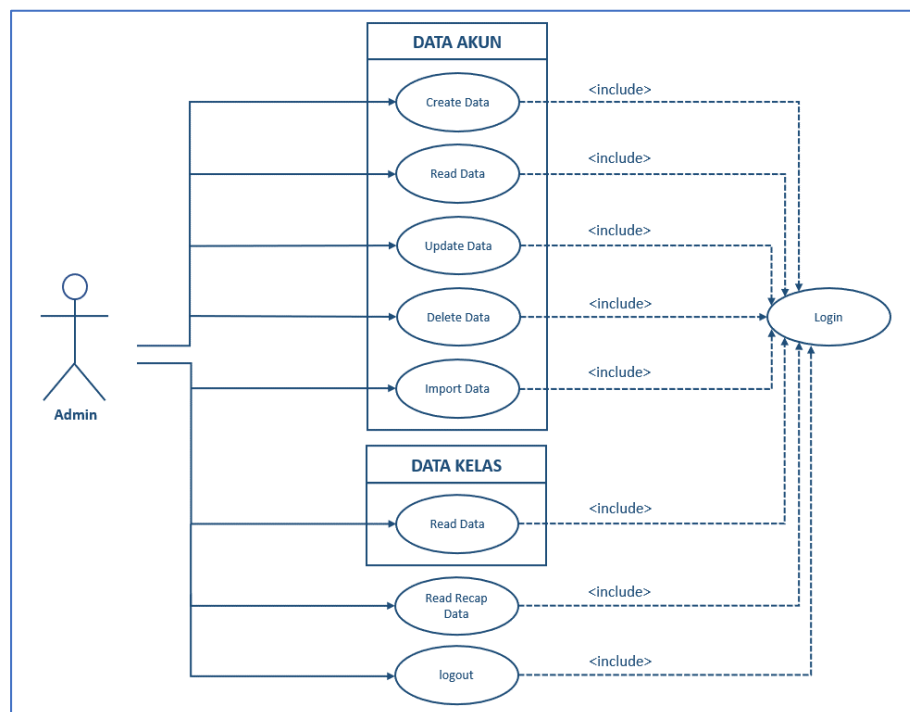
Iterasi 1			
User Story	Priority	Stories point	Estimasi waktu (Hari)
Sebagai guru saya ingin dapat melakukan pengelolaan tugas dan ujian bagi siswa, beserta dengan pengelolaan pertanyaan yang mendukung berbagai tipe soal, dan aplikasi dapat melakukan penilaian otomatis terhadap jawaban siswa.	Must have	5	10
Velocity		5	10
Iterasi 2			
User Story	Priority	Stories point	Estimasi waktu (Hari)
Sebagai guru saya ingin dapat melihat dan mengunduh hasil pengerjaan tugas dan ujian oleh siswa.	Must have	2	4
Sebagai guru saya ingin dapat melakukan pengelolaan kelas sehingga saya bisa membagikan materi maupun tugas dan ujian kepada siswa.	Must have	3	6
velocity		6	12

Iterasi 3			
User Story	Priority	Stories point	Estimasi waktu (Hari)
Sebagai guru saya ingin dapat menganalisis progres belajar seorang siswa berdasarkan hasil pengerjaan tugas dan ujian siswa bersangkutan.	Must have	2	4
Sebagai guru saya ingin dapat menganalisis materi yang diajarkan dan diujikan kepada siswa berdasarkan hasil pengerjaan tugas dan ujian oleh siswa dari satu kelas atau satu angkatan.	Must have	3	6
velocity	6	12	velocity

3.4.4. Perancangan

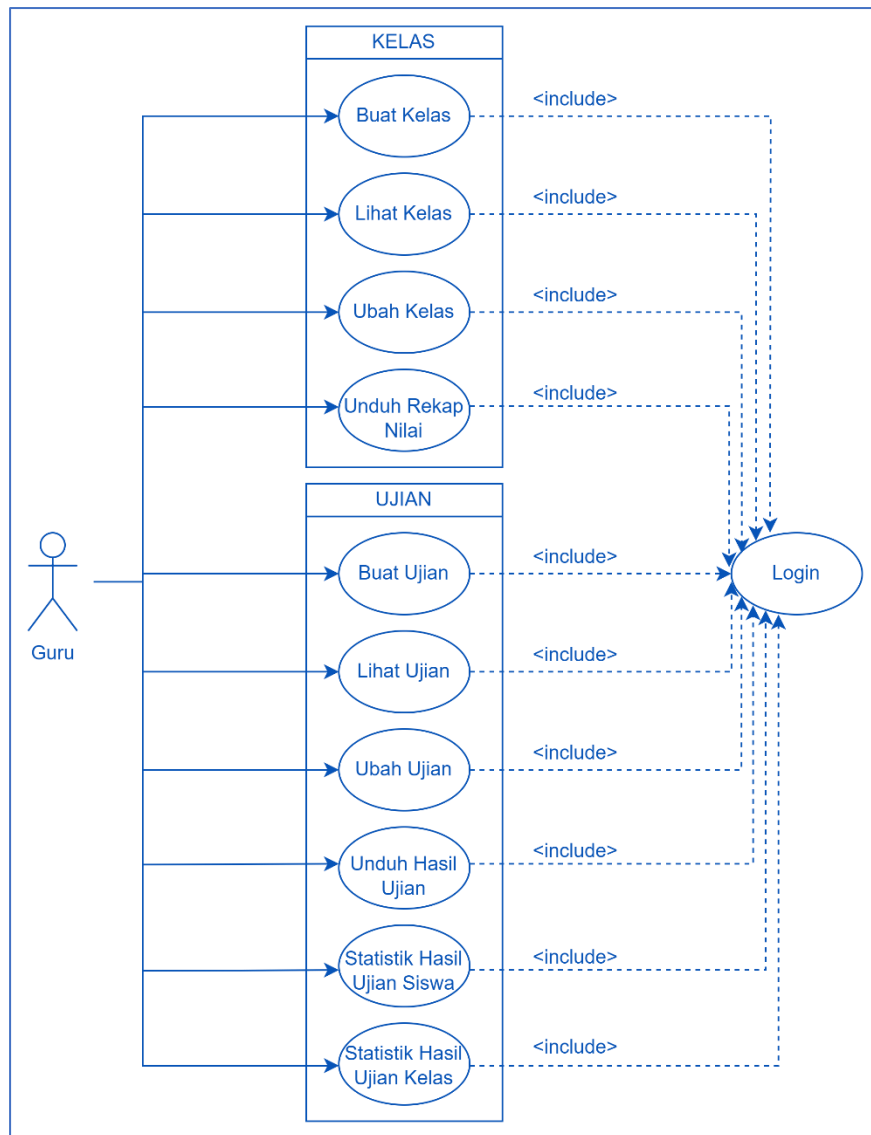
A. Use Case Diagram

1. Use Case Diagram – Admin

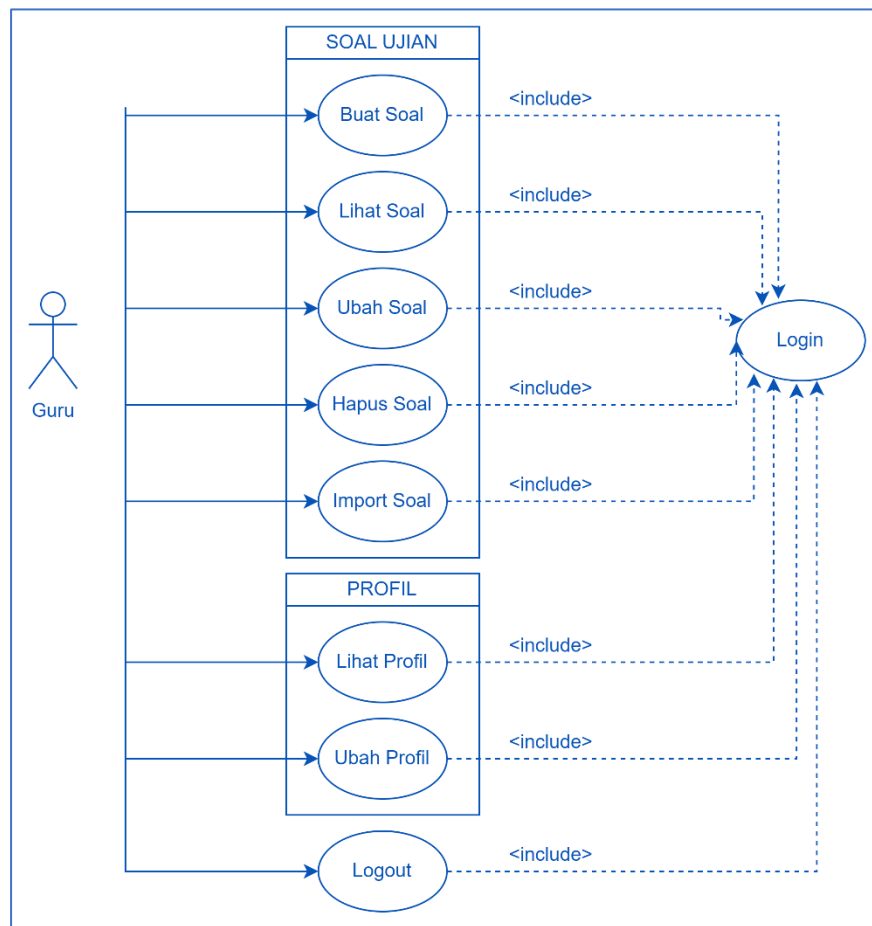


Gambar 3.4 Use Case Diagram Admin

2. Use Case Diagram – Guru

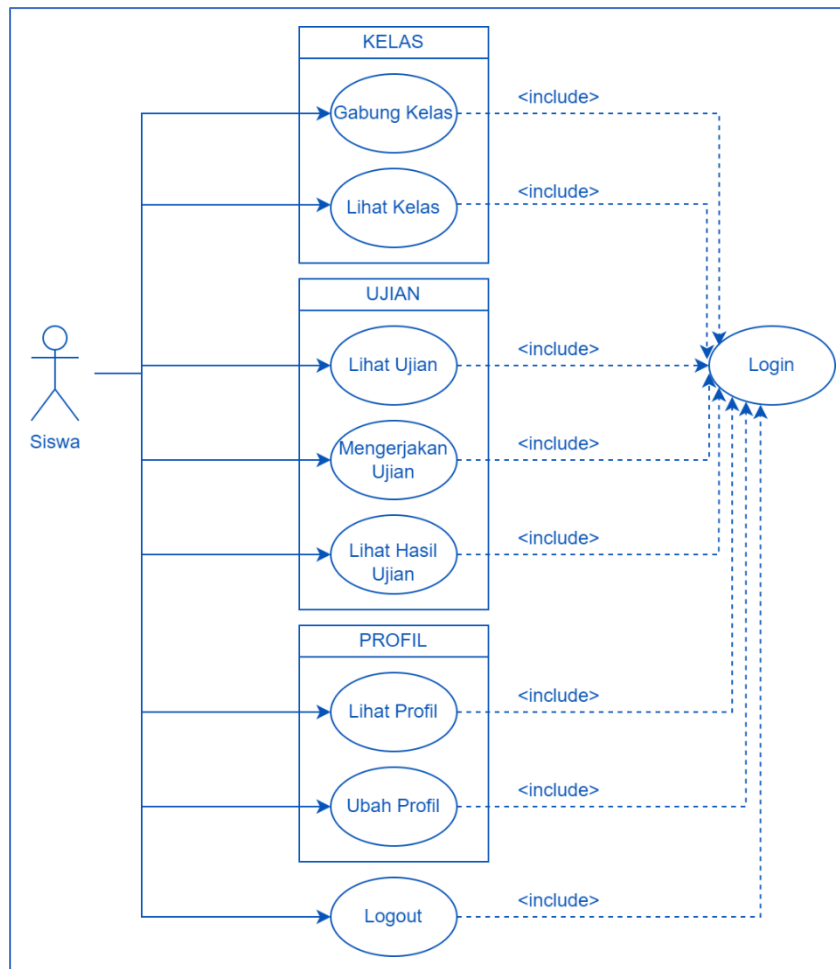


Gambar 3.4. Use Case Diagram Guru (1)



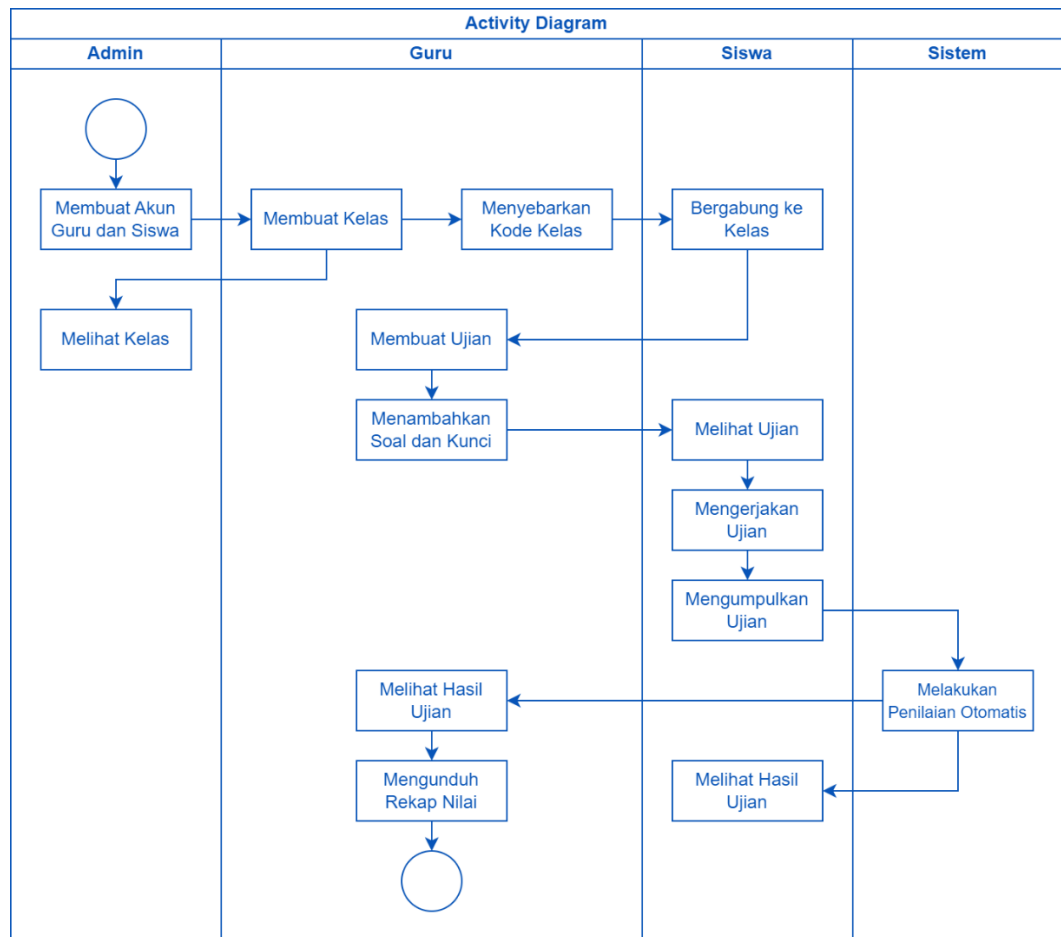
Gambar 3.4 Use Case Diagram Guru (2)

3. Use Case Diagram – Siswa



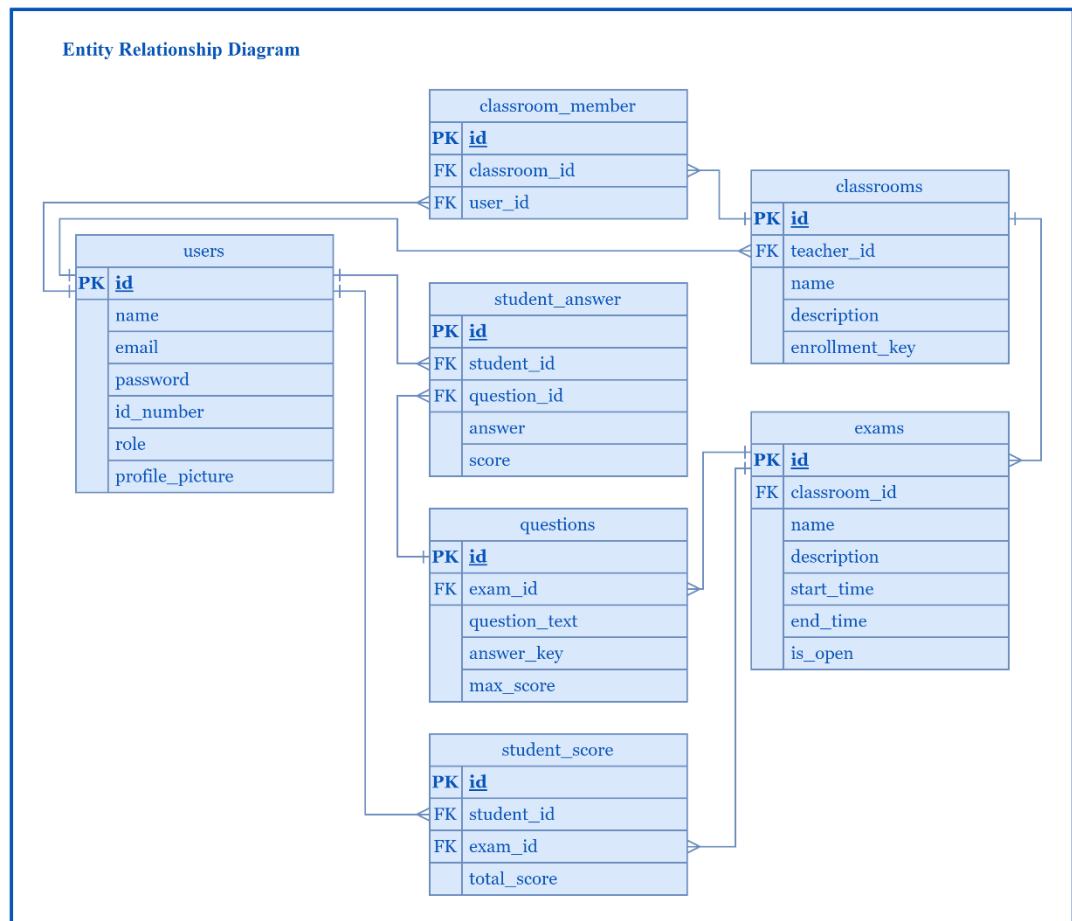
Gambar 3.4 Use Case Diagram Siswa

B. Activity Diagram



Gambar 3.4 Activity Diagram

C. ER – Diagram



Gambar 3.4 ER - Diagram

D. Rancangan Antarmuka

3.4.5. Implementasi

Implementasi merupakan proses menuliskan rancangan pada tahap perancangan kedalam code program. Implementasi dilakukan dengan pendekatan *Test Driven Development* (TDD). TDD memiliki tiga tahap diantaranya *unit testing*, *code generation*, dan *refactoring* yang akan diterapkan secara berulang pada masing-masing *user stories*. Tiga tahap TDD dibuat dalam bentuk implementasi kode dan dilakukan pengujian setelahnya. Bagian *refactoring* digunakan untuk optimasi kode sehingga dapat digunakan apabila diperlukan saja. Tahap implementasi ini digunakan untuk menuangkan semua hasil perancangan dari setiap iterasi kedalam kode sehingga sistem dapat digunakan oleh client dan diimplementasikan di SMP Negeri 10 Kotabumi.

3.4.6. Pengujian Sistem

Tahapan ini merupakan pengujian hasil implementasi menggunakan metode *Black Box Testing*. Pengujian dilakukan oleh pihak SMP Negeri 10 Kotabumi didampingi oleh pengembang. Klien menguji fitur apakah sesuai dengan kebutuhan awal pada tahap analisis dan perencanaan.

3.4.7. Restrospektif

Tahapan ini melakukan verifikasi terhadap semua *user stories* yang telah diimplementasikan dan dilakukan pengujian. Verifikasi dilakukan untuk perbandingan waktu estimasi dengan waktu realisasi sehingga dapat diketahui kendala-kendala penyebab *over* atau *under* estimasi pada pelaksanaan penelitian. Verifikasi ini bertujuan untuk mencegah perbedaan waktu estimasi pada penelitian selanjutnya.

3.5. Evaluasi Akhir Aplikasi


Untuk evaluasi akhir aplikasi yang dikembangkan akan digunakan metode *System Usabilty Scale* (SUS).

DAFTAR PUSTAKA

- [1] A. B. Sidiq and D. Kurniadi, "Perancangan Sistem Informasi Ujian Online Berbasis Web pada SMK N 1 Solok," *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, vol. 9, no. 2, p. 44, Jun. 2021, doi: 10.24036/voteteknika.v9i2.111521.
- [2] E. O. Choiri, "10 Aplikasi Ujian Online Terbaik, Gratis Pakai!," 2021. Qwords.com (accessed Nov. 16, 2022).
- [3] D. Yan, A. A. Rupp, and P. W. Foltz, *Handbook of Automated Scoring Theory into Practice*. Boca Raton: Taylor & Francis Group, LLC, 2020.
- [4] M. D. Shermis and J. C. Burstein, *Automated Essay Scoring: A Cross-Disciplinary Perspective*. New Jersey: Lawrence Erlbaum Associates, Inc., 2003.
- [5] J. F. Dooley, *Software Development, Design and Coding*. Berkeley, CA: Apress, 2017. doi: 10.1007/978-1-4842-3153-1.
- [6] Y. Dzhurov, I. Krasteva, and S. Ilieva, "Personal Extreme Programming – An Agile Process for Autonomous Developers," Jan. 2009.
- [7] M. van Deurzen, "The anatomy of the modern window manager," Bachelor thesis, Radboud University, Nijmegen, 2019.
- [8] G. E. Iyawa, "Personal Extreme Programming: Exploring Developers' Adoption," in *AMCIS 2020 Proceedings*, bepress, 2020.
- [9] S. A. Asri, I. G. M. A. Sunaya, P. M. Prihatini, and W. Setiawan, "Comparing Traditional and Agile Software Development Approaches: Case of Personal Extreme Programming," in *Proceedings of the International Conference on Science and Technology (ICST 2018)*, 2018. doi: 10.2991/icst-18.2018.116.
- [10] A. N. Yusril, I. Larasati, and P. Al Zukri, "Systematic Literature Review Analisis Metode Agile dalam Pengembangan Aplikasi Mobile," *Sistemasi : Jurnal Sistem Informasi*, vol. 10, no. 2, May 2021.
- [11] M. Ulfi, G. I. Marthasari, and I. Nuryasin, "Implementasi Metode Personal Extreme Programming dalam Pengembangan Sistem Manajemen Transaksi Perusahaan," *Jurnal Repositor*, vol. 2, no. 3, Mar. 2020.
- [12] G. Marthasari, W. Suharso, and F. A. Ardiansyah, "Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information

- System,” *Proceeding of the Electrical Engineering Computer Science and Informatics*, vol. 5, no. 1, Nov. 2018, doi: 10.11591/eecsi.v5.1701.
- [13] A. F. Septiyanto, W. Suharso, and I. Nuryasin, “Sistem Informasi Program Keluarga Harapan (PKH) Menggunakan Metode Personal Extreme Programming dengan Metode Prioritas Ranking,” *Jurnal Repositor*, vol. 2, no. 12, pp. 1671–1678, Dec. 2020, doi: 10.22219/repositor.v2i12.607.
- [14] M. Kancharla, G. S. Kamisetty, and S. H. Dudekula, “Muster: Virtual Classroom For Students Using D-Jango,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 5, May 2022.
- [15] A. S. Rosa and M. Salahuddin, *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Bandung Modula, 2011.
- [16] S. A. Asri and W. Setiawan, “Alternatif Penggunaan Model Pendekatan Agile pada Perancangan Sistem Informasi PKL Online,” *Jurnal Manajemen Teknologi dan Informatika*, vol. 5, no. 3, 2017.
- [17] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, 1st ed. Boston: Addison Wesley, 1998.
- [18] M. R. Sanjaya, A. Saputra, and D. Kurniawan, “Penerapan Metode System Usability Scale (SUS) Perangkat Lunak Daftar Hadir Di Pondok Pesantren Miftahul Jannah Berbasis Website,” *Jurnal Komputer Terapan*, no. Vol. 7 No. 1 (2021), pp. 120–132, Jun. 2021, doi: 10.35143/jkt.v7i1.4578.
- [19] Rasmila, “Evaluasi Website Dengan Menggunakan System Usability Scale (SUS) Pada Perguruan Tinggi Swasta di Palembang,” *JUSIFO : Jurnal Sistem Informasi*, vol. 4, no. 1, Jun. 2018.
- [20] J. Sauro, “Measuring Usability With The System Usability Scale (SUS),” *Measuring U*, Feb. 03, 2011.
- [21] F. Firmansyah, “Implementasi System Usability Scale pada Sistem Informasi Manajemen Anggaran dan Kegiatan di Badan Pusat Statistik,” *Technologia: Jurnal Ilmiah*, vol. 12, no. 3, p. 165, Jul. 2021, doi: 10.31602/tji.v12i3.5180.
- [22] A. Bangor, P. Kortum, and J. Miller, “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale,” *Journal of User Experience*, vol. 4, no. 3, 2009.

LAMPIRAN

No.	Lampiran
1	<p data-bbox="352 371 932 409">Folder Google Drive Rekaman Wawancara I</p>  A square QR code with a black and white pixelated pattern, used for linking to a Google Drive folder. It is enclosed in a thin blue rectangular border.