

**IMPLEMENTASI METODE EXTREME PROGRAMMING PADA
PEMBUATAN SISTEM INFORMASI PKL DAN PENGUJIAN
WHITE BOX SERTA COMPUTER SYSTEM USABILITY
QUESTIONNAIRE (CSUQ)**

SKRIPSI



Disusun Oleh :

Abdullah Al Fatih

21081010089

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

**UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAWA TIMUR**

SURABAYA

2025

Daftar Isi

Daftar Isi.....	i
Daftar Tabel.....	iii
Daftar Gambar.....	iv
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Fokus Penelitian	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
BAB II.....	6
TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terdahulu.....	6
2.2 Dasar Teori	7
2.2.1 Sistem Informasi	7
2.2.2 Praktek Kerja Lapangan.....	9
2.2.3 Next.js	9
2.2.4 Extreme Programming (XP)	10
2.2.5 White Box Testing	10
2.2.6 Flowchart	11
2.2.7 Flowgraph	13
2.2.8 Cyclomatic Complexity	13
2.2.9 Unified Modelling Language	14
2.2.10 Use Case Diagram.....	15
2.2.11 Activity Diagram	15
2.2.12 Sequence Diagram	15
2.2.13 Entity Relationship Diagram.....	16

2.2.14 Computer System Usability Questionnaire (CSUQ)	16
BAB III.....	17
DESAIN DAN IMPLEMENTASI SISTEM.....	17
3.1 Metode Penelitian dan Langkah-Langkah Penelitian.....	17
3.1.1 Studi Literatur	18
3.1.2 Analisis dan Perancangan Sistem	18
3.1.3 Pembuatan Sistem.....	18
3.1.4 Uji Coba Sistem	19
3.1.5 Survei Kegunaan.....	19
3.1.6 Evaluasi Hasil	19
3.1.7 Penyusunan Laporan.....	20
3.2 Desain Sistem	20
3.2.1 Use Case Diagram.....	20
3.2.2 Activity Diagram	22
3.2.3 Sequence Diagram	26
3.2.4 Entity Relationship Diagram.....	31
3.2.5 Langkah Pengujian dan Skenario Pengujian Sistem.....	32
3.2.6 Implementasi Pengujian White Box	34
3.2.7 Computer System Usability Questionnaire (CSUQ)	37
Daftar Pustaka	38

Daftar Tabel

Tabel 2.1 Simbol-simbol flowchart.....	13
Tabel 3.1 Skenario pengujian sistem	33
Tabel 3.2 Hubungan cyclomatic complexity dengan risiko.....	36
Tabel 3.3 Test case fungsi convertToHoursMinute	37

Daftar Gambar

Gambar 3.1 Langkah-langkah penelitian	18
Gambar 3.2 Alur siklus extreme programming.....	19
Gambar 3.3 Use case diagram sistem informasi PKL.....	21
Gambar 3.4 Activity diagram autentikasi	23
Gambar 3.5 Activity diagram pengajuan kegiatan PKL	24
Gambar 3.6 Activity diagram kegiatan bimbingan	25
Gambar 3.7 Activity diagram kegiatan penilaian.....	26
Gambar 3.8 Sequence diagram autentikasi mahasiswa.....	27
Gambar 3.9 Sequence diagram autentikasi dosen dan admin	28
Gambar 3.10 Sequence diagram pengajuan kegiatan PKL	29
Gambar 3.11 Sequence diagram kegiatan bimbingan	30
Gambar 3.12 Sequence diagram kegiatan penilaian	31
Gambar 3.13 Entity relationship diagram sistem informasi PKL	32
Gambar 3.14 Langkah pengujian	33
Gambar 3.15 Flowchart fungsi convertToHoursMinute	35
Gambar 3.16 Flowgraph fungsi convertToHoursMinute	35

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi, khususnya internet telah mengalami transformasi yang signifikan sejak awal kemunculannya hingga saat ini. Saat ini, internet sudah sangat umum digunakan di berbagai bidang kehidupan dan memberikan manfaat seperti akses informasi yang cepat [1]. Dengan berkembangnya internet, sistem informasi juga mengalami kemajuan pesat. Sistem informasi adalah sistem yang dapat didefinisikan dengan mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan informasi untuk tujuan tertentu [2]. Sistem informasi mengacu pada kombinasi dari teknologi, orang, dan proses yang digunakan untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi. Penerapan sistem informasi berguna untuk banyak hal seperti efisiensi operasional, pengintegrasian informasi, pemantauan kinerja, peningkatan pengambilan keputusan, dan lain sebagainya. Kampus UPN Veteran Jawa Timur memanfaatkan penerapan sistem informasi untuk pengintegrasian informasi dan efisiensi operasional praktek kerja lapangan.

Praktek Kerja Lapangan (PKL) adalah salah satu cara untuk mengintegrasikan secara sistematis dan sinkron antara program akademik dengan program penguasaan keahlian yang diperoleh melalui kegiatan kerja langsung di tempat kerja untuk mencapai tingkat keahlian tertentu [3]. PKL juga menjadi wadah bagi mahasiswa untuk mengenal tantangan di lingkungan industri atau organisasi, serta membantu mereka mempersiapkan diri untuk menghadapi tuntutan profesional setelah lulus. Dalam pelaksanaannya, mahasiswa ditempatkan di berbagai perusahaan atau institusi yang relevan dengan bidang studi mereka. PKL biasanya berlangsung selama beberapa bulan, tergantung pada kebijakan universitas. Selain itu, mahasiswa diwajibkan menyusun laporan yang berisi evaluasi pengalaman mereka selama menjalani PKL.

Pengelolaan data terkait PKL seperti penempatan, bimbingan, hingga evaluasi mahasiswa dapat menjadi tantangan tersendiri, dan di sinilah sistem informasi PKL memiliki peran penting untuk mengotomatisasi dan memudahkan seluruh proses tersebut. Sistem informasi PKL dirancang untuk membantu mempermudah berbagai proses administratif dan operasional kegiatan yang melibatkan beberapa pihak, seperti mahasiswa, dosen pembimbing, dan dosen penguji. Dengan adanya sistem ini, pengelolaan data terkait penempatan mahasiswa dapat dilakukan dengan lebih efisien, mulai dari pendaftaran, pemilihan lokasi PKL, persyaratan berkas penting hingga evaluasi kegiatan.

Penelitian sebelumnya oleh S. Thya Safitri dan Didi Supriyadi berjudul "Rancang Bangun Sistem Informasi Praktek Kerja Lapangan Berbasis Web dengan Metode Waterfall" [4] memiliki beberapa keterbatasan. Pertama, tidak dilakukan pengujian mendalam pada fitur-fitur utamanya. Kedua, tidak ada pengujian langsung kepada pengguna sistem. Ketiga, tidak dilakukan survei kegunaan untuk mengukur kualitas sistem. Untuk mengatasi kesenjangan ini, penelitian ini akan menggunakan metode White Box Testing dan Computer System Usability Questionnaire. Tujuannya adalah untuk memeriksa kecacatan dan mengukur kualitas sistem informasi PKL secara lebih komprehensif.

White Box Testing—juga dikenal sebagai "glass box testing"—adalah metode pengujian perangkat lunak yang berfokus pada alur kode program. Dalam metode ini, masukan dan keluaran diuji berdasarkan spesifikasi yang dibutuhkan [5]. White Box Testing terdiri dari beberapa teknik pengujian, antara lain control flow testing, path testing atau basic path testing, dan data flow testing. Dalam penelitian ini, penulis memilih menggunakan path testing. Metode ini menguji seluruh kemungkinan jalur dalam kode program, memastikan setiap jalur dilalui minimal satu kali. Path testing sangat efektif untuk mengevaluasi program-program yang kompleks [6].

Kegunaan adalah aspek penting dari keberhasilan sebuah program komputer. Kemudahan penggunaan merupakan aspek fundamental yang harus dicapai, mengacu

pada bagaimana seseorang dapat menggunakan program tersebut dengan mudah, terutama bagi mereka yang pertama kali menggunakan program tersebut. Computer System Usability Questionnaire atau CSUQ adalah kuesioner standar untuk mengukur kegunaan dari sudut pandang pengguna yang dikembangkan oleh IBM pada tahun 1995. CSUQ terdiri dari 16 pernyataan yang dinilai menggunakan skala nilai 1–7, di mana 1 berarti "sangat tidak setuju" dan 7 berarti "sangat setuju" [7].

Siklus hidup pengembangan perangkat lunak (SDLC) adalah metode yang memungkinkan pengembangan perangkat lunak secara sistematis. SDLC meningkatkan peluang penyelesaian proyek tepat waktu dan menjaga kualitas produk sesuai standar [8]. Penelitian ini menggunakan pendekatan Extreme Programming untuk pengembangan sistem informasi. Extreme Programming (XP) adalah metode pengembangan perangkat lunak yang didasarkan pada metodologi agile. XP dirancang untuk proyek yang sering mengalami perubahan dan memerlukan respons cepat terhadap kebutuhan klien yang berubah-ubah. Metode ini menekankan keterkaitan erat antara tahap perancangan dan implementasi, dengan fokus pada pembuatan kode dan pengujian berkelanjutan [9]. Penulis memilih metode ini karena sifat adaptifnya terhadap perubahan kebutuhan sistem informasi. Sifat ini sangat efektif untuk mengatasi kekurangan perencanaan matang sebelum pengembangan, namun tetap dapat menghasilkan sistem informasi berkualitas yang sesuai dengan keinginan pengguna.

Fokus penelitian ini adalah perancangan dan pembangunan sistem informasi Praktek Kerja Lapangan (PKL) dengan tujuan untuk mengotomatisasi dan mempermudah proses administrasi serta pengelolaan data PKL di perguruan tinggi. Sistem ini diharapkan dapat mengintegrasikan berbagai fungsi penting secara efisien, mulai dari pendaftaran mahasiswa, pemilihan lokasi PKL, pengelolaan berkas penting, hingga evaluasi kinerja mahasiswa. Selain itu, penelitian ini juga menekankan pengujian perangkat lunak dan survei kegunaan, dalam rangka menguji kegunaan dan kualitas dari sistem informasi. Penelitian ini diharapkan dapat bermanfaat sebagai acuan dalam perancangan dan pembangunan perangkat lunak.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, berikut merupakan rumusan masalah yang hendak penulis selesaikan:

1. Bagaimana merancang dan membangun sistem informasi yang dapat mengintegrasikan proses pendaftaran, penempatan, monitoring, dan evaluasi PKL secara efektif dan efisien menggunakan metode Extreme Programming?
2. Bagaimana memastikan sistem informasi PKL yang dikembangkan dapat diakses dengan mudah dan memiliki nilai guna oleh semua pihak yang terlibat, seperti mahasiswa, dosen pembimbing, dan dosen penguji?
3. Bagaimana implementasi White Box Testing dan Computer System Usability Questionnaire (CSUQ) untuk memastikan keandalan dan kelayakan sistem informasi PKL yang dibangun?

1.3 Fokus Penelitian

Fokus penelitian ini lebih diarahkan pada perancangan dan pembangunan sistem informasi. Penulis akan lebih mengeksplorasi praktek-praktek terbaik dalam merancang dan membangun sistem informasi PKL. Pengujian White Box Testing hanya terbatas pada modul-modul penting, seperti autentikasi, pendaftaran, evaluasi kinerja mahasiswa, dan lain sebagainya. Namun evaluasi keseluruhan sistem akan tetap dilakukan menggunakan metode Computer System Usability Questionnaire (CSUQ).

1.4 Tujuan Penelitian

Penulis melakukan perancangan dan pembuatan Sistem Informasi PKL bertujuan untuk:

1. Merancang dan membangun sistem informasi Praktek Kerja Lapangan (PKL) terintegrasi untuk mempermudah proses pendaftaran, penempatan, pemantauan, dan evaluasi mahasiswa.
2. Menguji keandalan dan memeriksa kecacatan sistem menggunakan metode White Box Testing.
3. Mengukur tingkat kegunaan sistem informasi PKL yang dikembangkan dengan metode Computer System Usability Questionnaire (CSUQ).
4. Menyediakan akses mudah bagi mahasiswa, dosen pembimbing, dan dosen penguji untuk memantau dan mengelola data Praktek Kerja Lapangan secara efektif dan efisien.

1.5 Manfaat Penelitian

- Bagi Penulis: Penelitian ini dapat memperluas wawasan serta memperdalam kemampuan di bidang pengembangan perangkat lunak.
- Bagi pembaca: Penelitian ini dapat mengenalkan dan menambah ketertarikan pembaca dengan dunia pengembangan perangkat lunak.
- Bagi Pihak Kampus: Penelitian ini dapat mengoptimalkan operasional kegiatan Praktek Kerja Lapangan (PKL) dan meningkatkan produktivitas secara keseluruhan.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu memiliki peran penting sebagai data pendukung dalam sebuah penelitian. Aspek krusial dari penelitian terdahulu adalah keterkaitan atau relevansinya dengan permasalahan yang diangkat dalam penelitian ini. Berikut adalah beberapa penelitian terdahulu yang penulis gunakan sebagai rujukan:

1. Penelitian "Pengembangan Sistem Informasi Praktik Kerja Lapangan (PKL) Siswa Berbasis Website Menggunakan Metode Extreme Programming (Studi Kasus: SMK Negeri 1 Sumenep)" yang dilakukan oleh M. Ro'if, Tri Afirianto, dan Satrio Hadi Wijoyo pada tahun 2024 menghasilkan sistem informasi PKL secara terstruktur dan komprehensif dengan metode Extreme Programming. Penelitian ini terdiri dari beberapa tahapan, yaitu identifikasi masalah, studi literatur, eksplorasi informasi, perencanaan, pengembangan, serta kesimpulan dan saran. Pengembangan sistem informasi terbagi menjadi 3 iterasi, yaitu iterasi 1 guru pembimbing, iterasi 2 panitia PKL, dan iterasi 3 kakomli dan siswa. Setiap iterasi melibatkan pengujian unit testing dan acceptance testing, diikuti dengan System Usability Scale untuk mengukur kualitas sistem informasi secara keseluruhan [10].
2. Penelitian "Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path" yang dilakukan oleh Citra Teguh Pratata, Ebenhaezer Mahardhika Asyer, Ima Prayudi, dan Aries Saifudin pada tahun 2020 menguji aplikasi Android dengan teknik basic path. Pengujian dilakukan secara sistematis dimulai dari tahap pembuatan flowchart, pembuatan flowgraph, menghitung cyclomatic complexity, penentuan independent path, dan melakukan unit test [11].

3. Penelitian “Evaluasi Usability Aplikasi English Competency Test (ECT) Menggunakan Metode Computer System Usability Questionner (CSUQ): Studi Kasus Pusat Bahasa Institut Teknologi Telkom Purwokerto” yang dilakukan oleh Atika Rahmadani Utami Br Ginting, Sarah Astiti, Khairun Nisa Meiah melakukan survei kegunaan menggunakan metode CSUQ yang terdiri dari 16 pernyataan. Penilaian dibagi menjadi 4 kategori, yaitu System Usefulness (SysUse), Information Quality (InfoQual), Interface Quality (IntQual) dan Overall Usability (OVERALL) [12].

Dari beberapa penelitian terdahulu di atas, penelitian pertama digunakan sebagai panduan untuk menerapkan metode Siklus Pengembangan Perangkat Lunak Extreme Programming (XP). Selain itu, penelitian ini akan menerapkan metode pengujian White Box, dengan mengacu pada penelitian kedua sebagai panduan. Untuk penerapan Computer System Usability Questionnaire (CSUQ), penulis akan berpedoman pada penelitian ketiga.

2.2 Dasar Teori

2.2.1 Sistem Informasi

Menurut Sutabri (2012), sistem informasi adalah sebuah sistem dalam suatu organisasi yang memadukan kebutuhan pengolahan transaksi harian dengan fungsi manajerial dan kegiatan strategis. Sistem ini mendukung operasi organisasi dan menyediakan laporan-laporan yang diperlukan kepada pihak luar tertentu. Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (building blocks). Komponen-komponen ini meliputi blok masukan, model, keluaran, teknologi, basis data, dan kendali. Sebagai suatu sistem, keenam blok tersebut berinteraksi satu sama lain, membentuk kesatuan yang berfungsi untuk mencapai tujuan tertentu.

- a. Blok masukan (input block)

Masukan mewakili data yang masuk ke dalam sistem informasi. Masukan ini mencakup metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

b. Blok Model (model block)

Blok model merupakan gabungan prosedur, logika, dan model matematika yang mengolah data input serta data tersimpan dalam basis data dengan metode tertentu untuk menghasilkan keluaran yang diinginkan.

c. Blok Keluaran (output block)

Produk dari sistem informasi berupa keluaran yang terdiri dari informasi berkualitas dan dokumentasi. Keluaran ini bermanfaat bagi semua tingkatan manajemen dan pengguna sistem.

d. Blok teknologi (technology block)

Teknologi merupakan "kotak peralatan" dalam sistem informasi. Teknologi digunakan untuk menerima masukan, menjalankan model, menyimpan dan mengakses data, mengolah keluaran, serta membantu kendali sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama: teknisi (brainware), perangkat lunak (software), dan perangkat keras (hardware).

e. Blok basis data (database block)

Basis data adalah kumpulan data yang saling terkait, tersimpan di perangkat keras komputer dan dimanipulasi menggunakan perangkat lunak. Penyimpanan data dalam basis data penting untuk penyediaan informasi selanjutnya. Data perlu diorganisasi dengan baik agar menghasilkan informasi berkualitas dan mengoptimalkan kapasitas penyimpanan. Untuk mengakses atau memanipulasi basis

data, digunakan perangkat lunak khusus yang disebut DBMS (Database Management System).

f. Blok kendali (control block)

Sistem informasi dapat terancam oleh berbagai faktor, mulai dari bencana alam, kebakaran, suhu ekstrem, air, debu, ketidakefisienan, hingga sabotase. Oleh karena itu, perlu dirancang dan diterapkan beberapa mekanisme pengendalian. Tujuannya adalah untuk mencegah kerusakan sistem atau memastikan penanganan yang cepat jika kerusakan telah terjadi [13].

2.2.2 Praktek Kerja Lapangan

Praktek Kerja Lapangan (PKL) adalah metode untuk mengintegrasikan program akademik dengan penguasaan keahlian secara sistematis dan sinkron. Keahlian ini diperoleh melalui pengalaman kerja langsung di tempat kerja untuk mencapai tingkat kompetensi tertentu. Tujuan pelaksanaan PKL meliputi: menambah pengetahuan dan wawasan keilmuan mahasiswa melalui pengalaman langsung di dunia kerja, mengaplikasikan keterampilan dan keahlian sesuai bidang ilmu yang dipelajari secara langsung sehingga mahasiswa dapat memahami ruang lingkup bidang kerja sesuai kompetensinya, membentuk sikap dan mental mahasiswa agar mampu dan berani menghadapi tantangan dunia kerja yang penuh persaingan. Selain itu, PKL juga memiliki berbagai manfaat lainnya [3].

2.2.3 Next.js

Next.js adalah kerangka kerja React ringan yang digunakan untuk mengembangkan aplikasi statis dan dirender oleh server. Kerangka kerja ini menggunakan struktur direktori sebagai metode perutean untuk halaman web, dengan "app" sebagai halaman default. Melalui penggunaan direktori halaman, Next.js menyediakan perutean otomatis, sementara sisi server menangani pengambilan data untuk setiap permintaan [14]. Next.js, yang dikembangkan oleh Vercel, dirancang

untuk menyederhanakan pembuatan aplikasi web yang dirender di server, dibuat secara statis, dan hibrida. Fitur-fitur utamanya meliputi pemisahan kode otomatis, render sisi server, pembuatan situs statis, dan router API bawaan. Secara kolektif, fitur-fitur ini meningkatkan kinerja dan SEO, memungkinkan pengembang untuk membuat aplikasi web yang efisien dan dapat diskalakan. Kerangka kerja ini mendukung baik rendering sisi server maupun pembuatan situs statis, memberikan fleksibilitas kepada pengembang untuk melakukan pra-rendering halaman baik pada waktu pembuatan atau waktu permintaan. Fleksibilitas ini menjadikan Next.js cocok untuk berbagai kasus penggunaan, mulai dari aplikasi dinamis hingga situs web statis [15].

2.2.4 Extreme Programming (XP)

Extreme Programming atau XP adalah metode pengembangan perangkat lunak berbasis pendekatan agile yang dirancang untuk proyek yang sering berubah dan membutuhkan respon cepat terhadap kebutuhan klien yang berubah. XP bersifat fleksibel, terutama untuk tim kecil dengan 2 hingga 10 anggota. Metode ini menekankan hubungan erat antara perancangan dan implementasi, dengan fokus pada pembuatan kode dan pengujian yang berkelanjutan. Aktivitas utama XP meliputi perencanaan, perancangan, pengkodean, dan pengujian. Praktek-praktek seperti pair programming (dua orang programmer bekerja pada kode yang sama), pengujian otomatis, dan revisi kode secara berkelanjutan merupakan karakteristik utama dari XP. Selain itu, nilai-nilai utama XP meliputi komunikasi, kesederhanaan, umpan balik, keberanian, dan saling menghargai antar anggota tim [9].

2.2.5 White Box Testing

Menurut Verma dkk. (2017), White Box Testing adalah teknik pengujian perangkat lunak yang berfokus pada struktur internal aplikasi. Metode ini menyelidiki logika internal, struktur kode, dan alur kontrol aplikasi. Teknik ini juga dikenal dengan nama lain seperti Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing, atau Structural Testing. Dalam white box testing, penguji harus memiliki pengetahuan mendalam tentang bahasa







pemrograman (source code) karena berkaitan dengan kerja internal aplikasi. Metode ini memerlukan informasi yang detail tentang aplikasi yang diuji.


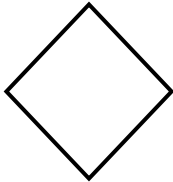
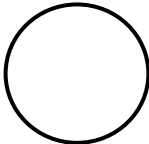
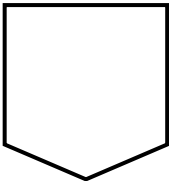
White Box Testing adalah metode pengujian yang diterapkan langsung pada kode sumber perangkat lunak, memeriksa semua jalur kode. Untuk melakukan teknik pengujian ini, penguji harus memiliki kemampuan pemrograman yang kuat. White Box Testing membutuhkan sumber daya manusia yang sangat terampil dan memiliki pengetahuan implementasi yang baik. White Box Testing terdiri dari beberapa teknik:

1. Control Flow Testing: Teknik pengujian struktural ini memeriksa aliran kontrol program, termasuk jalur sederhana dan rumit. Pengujian ini diterapkan pada semua perangkat lunak untuk menguji aliran kontrol. Ini adalah teknik mendasar yang berlaku pada program kecil dan subbagian dari program besar.
2. Path Testing/Basic Path Testing: Pengujian jalur ini menguji semua jalur program yang mungkin. Ini adalah teknik yang komprehensif, memastikan bahwa semua jalur dilalui setidaknya satu kali. Teknik cakupan jalur lebih baik daripada cakupan cabang dan cocok untuk menguji program yang kompleks.
3. Data Flow Testing: Pengujian aliran data berfokus pada titik-titik di mana variabel menerima nilai dan di mana nilai tersebut digunakan. Teknik ini dapat mengidentifikasi beberapa masalah seperti:
 - a. Variabel yang dideklarasikan tetapi tidak pernah digunakan dalam program.
 - b. Variabel yang digunakan dalam program tetapi tidak dideklarasikan.
 - c. Deklarasi ganda variabel sebelum digunakan [6].

2.2.6 Flowchart

Flowchart atau diagram alir adalah alat visual yang menggunakan simbol-simbol untuk mewakili komponen fisik dan/atau logika dari sistem informasi. Secara umum, diagram alir didefinisikan oleh standar ANSI/ISO sebagai representasi grafis dari definisi, analisis, atau metode penyelesaian masalah, di mana simbol digunakan untuk mewakili operasi, data, aliran, peralatan, dan lain-lain [16]. Berikut merupakan simbol-simbol flowchart yang umum digunakan [17]:

Simbol	Keterangan
	<p>Terminal: Menunjukkan awal atau akhir dari diagram alir.</p>
	<p>Input/Output: Menunjukkan operasi input atau output.</p>
	<p>Process: Mewakili tahapan/langkah dalam suatu proses.</p>
	<p>Predefined Process: Menunjukkan proses yang didefinisikan di tempat lain.</p>
	<p>Comment: Menunjukkan informasi tambahan tentang tahapan dalam suatu proses.</p>
	<p>Flow Line: Menunjukkan arah proses.</p>

	Document Input/Output: Menunjukkan operasi input dokumen atau output dokumen.
	Decision: Mewakili proses penentuan yang mempengaruhi langkah berikutnya dalam suatu proses.
	On-page Connector: Digunakan untuk menghubungkan bagian flowchart yang berada di halaman yang sama.
	Off-page Connector: Digunakan untuk menghubungkan bagian flowchart yang berada di halaman lain.

Tabel 2.1 Simbol-simbol flowchart

2.2.7 Flowgraph

Flowgraph menggambarkan jalannya logika dalam program. Flowgraph dapat dibuat dari kode program atau diagram alir sistem. Flowgraph menggunakan simbol lingkaran (node) dan panah (edge). Node mewakili pernyataan prosedural, sedangkan edge mewakili alur logika program [18].

2.2.8 Cyclomatic Complexity

Cyclomatic Complexity adalah pengukuran kuantitatif untuk mengetahui kompleksitas logika dalam program. Dalam pengujian whitebox menggunakan metode Basis Path, nilai Cyclomatic Complexity menentukan jumlah jalur independen dalam basis set program. Jumlah jalur independen ini menentukan jumlah kasus uji minimal

yang harus dilakukan untuk memastikan semua persyaratan pada jalur independen telah dieksekusi setidaknya satu kali [18]. Rumus untuk menghitung Cyclomatic Complexity adalah sebagai berikut:

$$V(G) = E - N + 2 \text{ atau } V(G) = P + 1$$

dimana:

$V(G)$ = Cyclomatic Complexity

E = jumlah edge pada flowgraph

N = jumlah node pada flowgraph

P = jumlah predicate node pada flowgraph

2.2.9 Unified Modelling Language

Menurut Jacobson dkk. (2021), Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan secara umum untuk menentukan, divisualisasikan, membangun, dan mendokumentasikan komponen sistem perangkat lunak. Bahasa ini digunakan untuk memahami, merancang, melacak, mengonfigurasi, memelihara, dan mengendalikan informasi tentang sistem tersebut. UML dimaksudkan untuk digunakan dalam semua metode pengembangan, tahapan siklus hidup, domain aplikasi, dan media. Bahasa pemodelan ini bertujuan untuk menyatukan pengalaman masa lalu tentang teknik pemodelan dan menggabungkan praktek terbaik perangkat lunak saat ini ke dalam pendekatan standar. UML mencakup konsep semantik, notasi, dan pedoman. Ia memiliki bagian statis, dinamis, lingkungan, dan organisasi.

UML menggunakan informasi tentang struktur statis dan perilaku dinamis dari sebuah sistem. Sistem dimodelkan sebagai kumpulan objek yang saling berinteraksi untuk memberikan manfaat bagi pengguna. Struktur statis menjelaskan jenis-jenis objek penting dan hubungan di antara mereka. Perilaku dinamis menjelaskan sejarah

objek dan komunikasi di antara mereka untuk mencapai tujuan. Memodelkan sistem dari beberapa sudut pandang yang saling terkait memungkinkan pemahaman yang lebih baik untuk berbagai tujuan [19].

2.2.10 Use Case Diagram

Use Case Diagram menangkap perilaku sistem, subsistem, atau kelas dari sudut pandang pengguna eksternal. Use Case Diagram termasuk ke dalam Unified Modelling Language (UML). Diagram ini membagi fungsionalitas sistem menjadi transaksi yang bermakna bagi aktor/pengguna ideal sebuah sistem. Bagian-bagian fungsionalitas interaktif ini disebut use case. Setiap use case menggambarkan interaksi dengan aktor sebagai rangkaian pesan antara sistem dan satu atau lebih aktor. Istilah "aktor" mencakup manusia, sistem lain, dan proses komputer [19].

2.2.11 Activity Diagram

Activity Graph adalah bentuk khusus dari mesin status yang digunakan untuk memodelkan komputasi dan alur kerja. Aktivitas-aktivitas dalam graf ini mewakili status-status pelaksanaan komputasi, bukan status-status objek biasa. Activity Graph mengasumsikan komputasi berjalan tanpa interupsi eksternal. Sebuah Activity Graph berisi status aktivitas yang mewakili eksekusi pernyataan dalam prosedur atau kinerja aktivitas dalam alur kerja. Transisi penyelesaian dijalankan ketika aktivitas sebelumnya selesai. Activity Diagram merupakan diagram yang menggambarkan Activity Graph. Activity Diagram termasuk ke dalam Unified Modelling Language (UML). Activity Diagram dapat mencakup cabang, serta percabangan kontrol ke dalam Concurrent Thread yang mewakili aktivitas yang dapat dilakukan secara bersamaan oleh objek atau orang yang berbeda. Activity Diagram memungkinkan kontrol konkuren selain kontrol sekuensial [19].

2.2.12 Sequence Diagram

Sequence Diagram menampilkan interaksi di dalam sistem sebagai bagan dua dimensi. Sumbu vertikal mewakili waktu, yang mengalir dari atas ke bawah halaman.

Sumbu horizontal menunjukkan peran pengklasifikasi, masing-masing mewakili objek dalam kolaborasi. Setiap peran pengklasifikasi direpresentasikan oleh kolom vertikal yang disebut garis hidup. Selama objek ada, perannya ditunjukkan dengan garis putus-putus. Saat prosedur pada objek aktif, garis hidup digambarkan sebagai garis ganda. Pesan ditampilkan sebagai panah dari garis hidup satu objek ke objek lainnya. Panah-panah ini disusun berdasarkan urutan waktu pada diagram. Sequence Diagram merupakan bagian dari Unified Modelling Language (UML) [19].

2.2.13 Entity Relationship Diagram

Entity-Relationship Diagram (ERD) merupakan teknik umum untuk mendesain struktur data dan sistem basis data dalam bentuk diagram. ERD digunakan untuk memodelkan data secara konseptual, membantu dalam merencanakan dan membangun basis data yang dapat diimplementasikan pada berbagai jenis Database Management System (DBMS). ERD menyajikan data dalam bentuk entitas, atribut, dan hubungan (relationship) antara entitas, yang merupakan representasi abstrak dari dunia nyata. ERD berfokus pada bagaimana simbol-simbol yang disimpan dalam basis data mencerminkan objek nyata, sehingga memastikan bahwa model tersebut adalah representasi yang benar dari dunia nyata [20].

2.2.14 Computer System Usability Questionnaire (CSUQ)

Computer System Usability Questionnaire (CSUQ) adalah alat penilaian kegunaan yang dikembangkan oleh James R. Lewis di IBM pada tahun 1995. CSUQ digunakan untuk memungkinkan praktisi menilai kegunaan produk atau layanan dari sudut pandang pengguna. Instrumen CSUQ terdiri dari 16 pernyataan yang dinilai berdasarkan skala 7 poin tingkat persetujuan. Penilaian CSUQ terbagi menjadi 4 kategori, yaitu System Usefulness (SYSUSE), Information Quality (INFOQUAL), Interface Quality (INTERQUAL), and Overall Usability (OVERALL) [7].

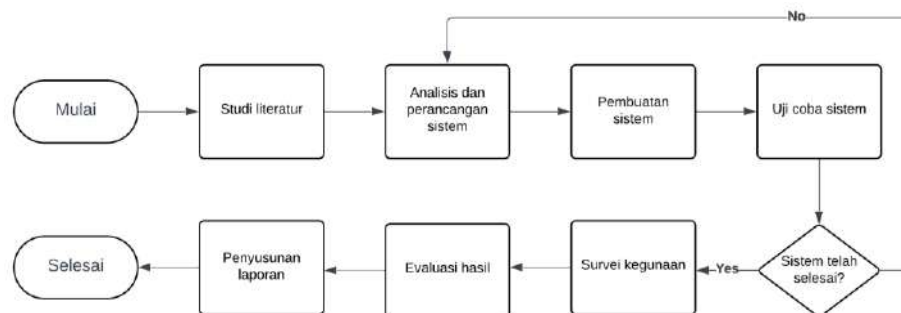
BAB III

DESAIN DAN IMPLEMENTASI SISTEM

Desain dan implementasi sistem berisi tentang prosedur penelitian dan rancangan sistem yang akan dibuat. Desain dan implementasi sistem terbagi menjadi 2 sub bab, yaitu metode penelitian dan desain sistem. Pada sub bab metode penelitian, penulis akan menjelaskan metode penelitian yang digunakan serta tahapan-tahapan penelitian secara ringkas. Diagram perancangan sistem hingga prosedur pengujian White Box dan Computer System Usability Questionnaire (CSUQ) akan dijelaskan pada sub bab desain sistem.

3.1 Metode Penelitian dan Langkah-Langkah Penelitian

Penelitian ini menggunakan metode Extreme Programming (XP) dan pengujian White Box untuk merancang dan membangun sistem informasi PKL. Pengujian difokuskan pada bagian-bagian penting sistem informasi PKL. Selain itu, penelitian ini juga menggunakan Computer System Usability Questionnaire (CSUQ) untuk mengevaluasi kegunaan sistem secara keseluruhan. Computer System Usability Questionnaire (CSUQ) dilakukan secara langsung oleh calon pengguna sistem. Langkah-langkah dalam penelitian ini ditunjukkan pada gambar 3.1.



Gambar 3.1 Langkah-langkah penelitian

3.1.1 Studi Literatur

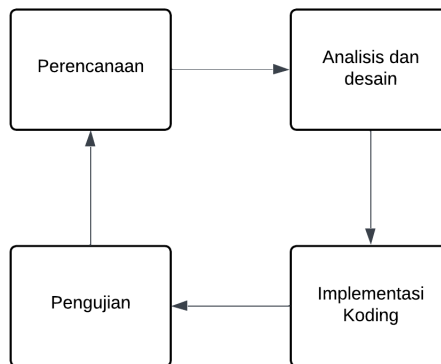
Langkah awal penelitian ini adalah melakukan studi literatur dengan tujuan untuk memiliki pengetahuan tentang konsep-konsep sebagai pondasi keberhasilan pengembangan sistem informasi PKL. Penulis melakukan studi literatur dengan melihat penelitian terdahulu, jurnal dan buku-buku serta literatur lainnya terkait dengan Next.js, Siklus Pengembangan Perangkat Lunak Extreme Programming (XP), Pengujian White Box, dan Computer System Usability Questionnaire (CSUQ). Penulis mempelajari praktek-praktek terbaik dari framework serta metode yang akan digunakan untuk penelitian ini.

3.1.2 Analisis dan Perancangan Sistem

Pada tahap ini penulis melakukan wawancara guna mengetahui kebutuhan sistem informasi PKL. Penulis berfokus untuk menggali informasi lebih dalam sekaligus memahami sistem yang akan dibuat secara keseluruhan. Selain itu, penulis juga membuat diagram-diagram terkait perancangan sistem informasi PKL, seperti use case diagram, activity diagram, sequence diagram dan entity relationship diagram.

3.1.3 Pembuatan Sistem

Pada pembuatan sistem penulis menggunakan metode Siklus Pengembangan Perangkat Lunak Extreme Programming (XP), yang artinya dilakukan secara iteratif. Iterasi dilakukan setiap pengembangan satu atau lebih fitur dalam siklus pendek, yaitu 2 minggu atau 1 bulan. Tahapan penelitian yang termasuk ke dalam siklus XP, yaitu analisis dan perancangan sistem, pembuatan sistem, dan uji coba sistem. Penerapan lebih lanjut tentang siklus XP ditunjukkan pada gambar 3.2.



Gambar 3.2 Alur siklus extreme programming

3.1.4 Uji Coba Sistem

Pada tahap uji coba sistem, penulis akan melakukan pengujian White Box pada sistem informasi PKL. Dengan pengujian White Box, penulis akan mengetahui lebih mudah jika terdapat sistem mengalami bug atau error. Pengujian sistem juga penting dilakukan sebelum survei kegunaan untuk memastikan sistem informasi PKL.

3.1.5 Survei Kegunaan

Survei kegunaan sistem informasi PKL akan dilakukan menggunakan Computer System Usability Questionnaire (CSUQ). Pada tahap ini penulis akan mengizinkan 30 orang untuk mencoba sistem informasi PKL secara langsung. Sampel pengguna tersebut terdiri dari dosen dan mahasiswa, penulis mencoba untuk menggunakan variasi latar belakang profesi untuk meningkatkan reliabilitas hasil.

3.1.6 Evaluasi Hasil

Evaluasi hasil dilakukan secara mendalam mulai dari pembuatan sistem, uji coba sistem, dan survei kegunaan. Penulis akan mengevaluasi sejauh mana sistem selaras dengan tujuan awal. Berdasarkan temuan evaluasi, penulis juga akan membuat rekomendasi perbaikan sistem di masa mendatang.

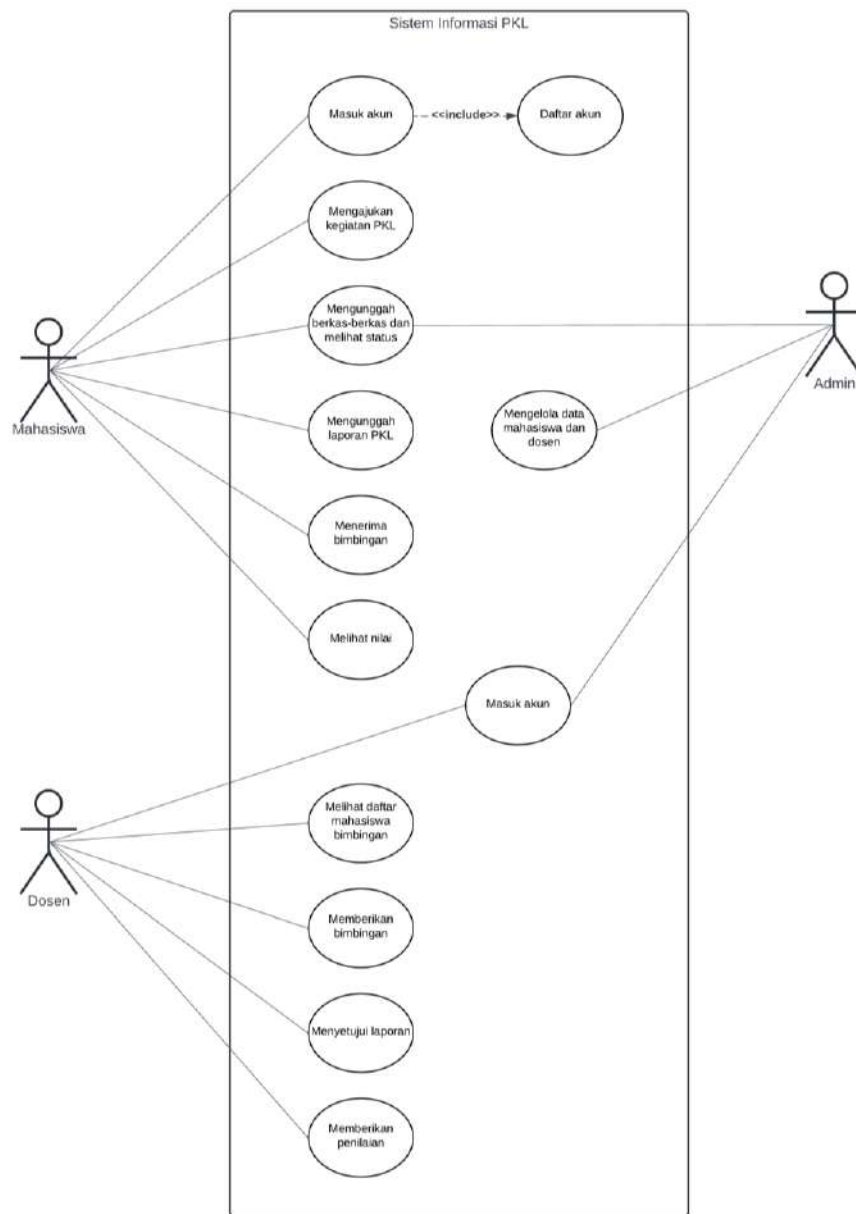
3.1.7 Penyusunan Laporan

Pada tahap ini, penulis sudah menyelesaikan sistem informasi PKL secara keseluruhan. Setelah itu, penulis akan berfokus pada penyusunan laporan. Penulis menyusun laporan secara komprehensif dan transparan berdasarkan tahapan-tahapan penelitian sebelumnya.

3.2 Desain Sistem

3.2.1 Use Case Diagram

Use Case Diagram menggambarkan interaksi antara aktor (pengguna) dan sistem yang sedang dikembangkan. Use Case Diagram bertujuan menjelaskan perilaku sistem secara keseluruhan tanpa mengungkapkan struktur internal dari sistem. Sebuah use case digambarkan dengan bentuk elips atau oval dengan nama di dalamnya. Use case dihubungkan dengan garis lurus ke aktor-aktor yang berinteraksi dengannya. Sebuah use case dapat mencakup perilaku dari use case lain sebagai bagian dari perilakunya sendiri. Hubungan ini disebut sebagai hubungan include [19]. Gambar 3.3 menampilkan Use Case Diagram untuk sistem informasi PKL.



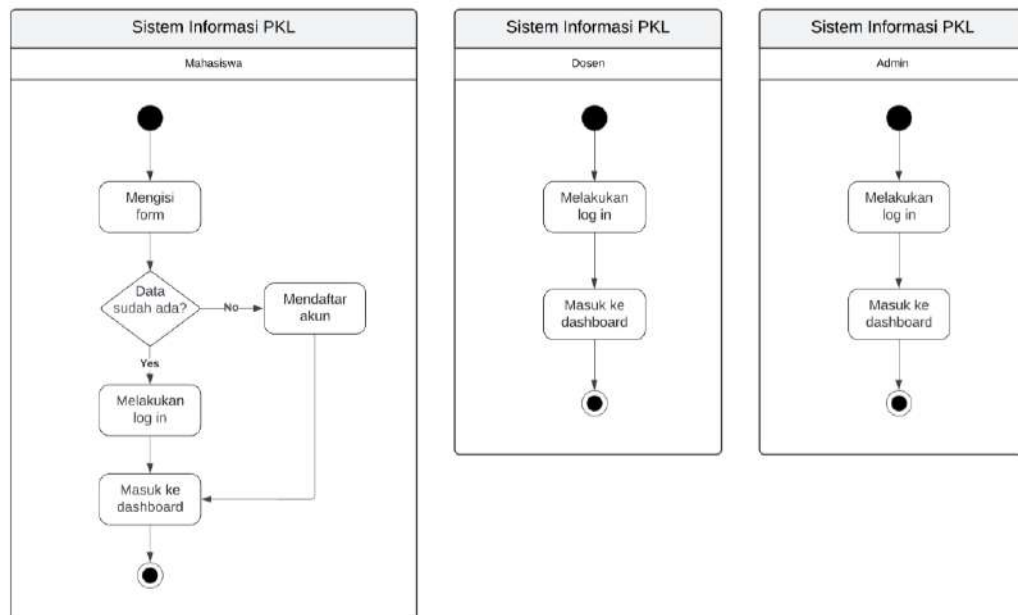
Gambar 3.3 Use case diagram sistem informasi PKL

Sistem informasi PKL terdiri dari 3 aktor atau pengguna yang berinteraksi satu sama lain, yaitu mahasiswa, dosen, dan admin. Aktor mahasiswa berperan sebagaimana mahasiswa, yaitu dapat membuat akun, mengajukan kegiatan PKL, mengunggah berkas-berkas keperluan kegiatan PKL sekaligus melihat status pengajuannya, mengunggah laporan PKL, menerima bimbingan, dan melihat nilai. Aktor dosen

memiliki 2 peran, yaitu sebagai dosen pembimbing dan sebagai dosen penguji. Dosen pembimbing dapat melihat daftar mahasiswa bimbingannya, memberikan bimbingan, hingga menyetujui laporan. Berbeda dengan dosen penguji yang dapat memberikan nilai terhadap laporan PKL mahasiswa. Admin memiliki kontrol penuh pada sistem informasi PKL ini. Admin memiliki tugas untuk memverifikasi pengajuan PKL mahasiswa. Selain itu, admin memiliki kontrol terhadap data mahasiswa dan dosen. Dalam sistem ini, dosen dan admin tidak perlu mendaftar akun, sementara mahasiswa perlu mendaftar akun terlebih dahulu di awal.

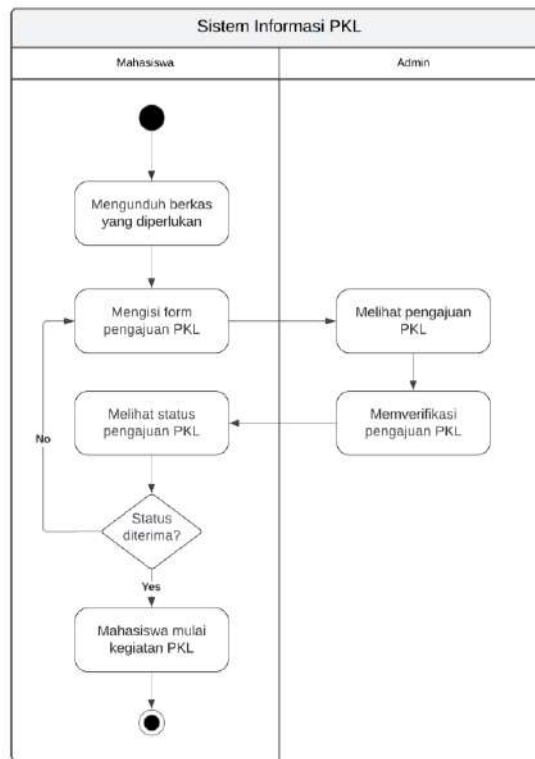
3.2.2 Activity Diagram

Activity Diagram menggambarkan alur proses atau langkah-langkah kerja dalam sistem. Diagram ini memperjelas urutan aktivitas dan pengambilan keputusan yang dilakukan oleh sistem atau pengguna selama proses tertentu. Simbol pada Activity Diagram memiliki sedikit kesamaan dengan flowchart (yang dijelaskan pada tabel 2.1), misalnya simbol proses dan decision. Namun, simbol “awal” digambarkan dengan lingkaran hitam kecil dan simbol “akhir” digambarkan dengan lingkaran yang memiliki lingkaran hitam di tengahnya. Di dalam Activity Diagram, area kerja aktor dipisahkan dengan garis sehingga membentuk kolom. Area kerja atau jalur ini disebut dengan swimlane [19]. Berikut ini adalah Activity Diagram untuk sistem informasi PKL:



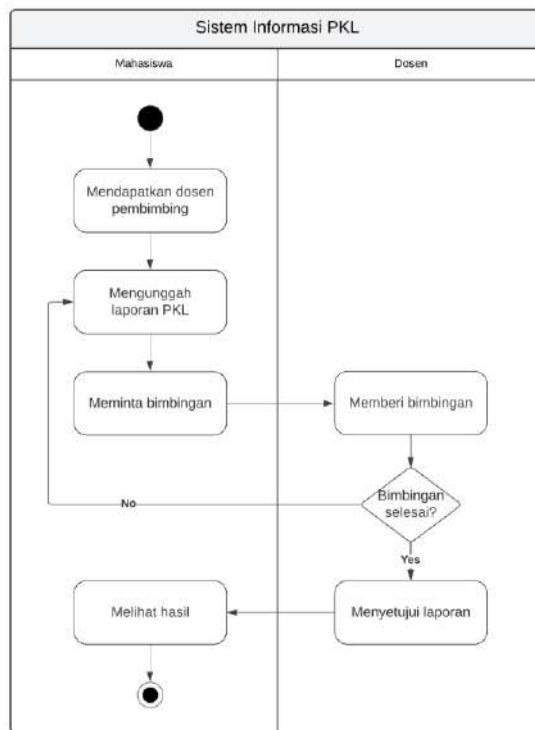
Gambar 3.4 Activity diagram autentikasi

Gambar 3.4 menampilkan Activity Diagram autentikasi masing-masing aktor. Ketika aktor mengisi data pada formulir login, sistem akan mengecek data akun. Jika data akun tersebut baru atau tidak ada di basis data, itu artinya pengguna tersebut adalah mahasiswa. Sistem akan otomatis menampilkan pesan kesalahan yang menginstruksikan mahasiswa untuk mendaftar akun terlebih dahulu. Jika mahasiswa sebelumnya sudah mendaftar akun, barulah mahasiswa dapat melakukan login menggunakan akun sebelumnya. Dosen dan admin tidak perlu mendaftar akun karena data akunnya sudah ada di basis data.



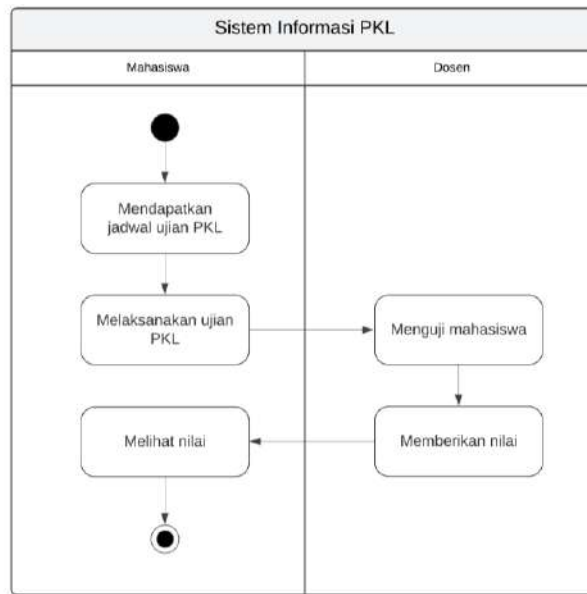
Gambar 3.5 Activity diagram pengajuan kegiatan PKL

Pada Activity Diagram pengajuan kegiatan PKL yang ditunjukkan pada gambar 3.5, mahasiswa perlu mengetahui berkas-berkas yang diperlukan untuk mendaftar pengajuan kegiatan PKL. Pada formulir ini mahasiswa perlu mengisi data terkait mitra tempat PKL dan mengunggah berkas-berkas tersebut. Admin kemudian akan memeriksa pengajuan tersebut dan mengonfirmasi apakah pengajuan dapat disetujui atau tidak. Jika pengajuan kegiatan PKL ditolak, mahasiswa dapat mengisi formulir kembali. Setelah pengajuan PKL diterima, mahasiswa diperbolehkan memulai kegiatan PKL.



Gambar 3.6 Activity diagram kegiatan bimbingan

Activity Diagram kegiatan bimbingan antara mahasiswa dan dosen ditunjukkan pada gambar 3.6. Dosen pembimbing ditentukan secara otomatis oleh koordinator PKL melalui akun admin. Mahasiswa dapat memulai bimbingan dengan mengunggah laporan PKL nya. Setelah itu, dosen akan memeriksa laporan PKL dan memberikan saran melalui formulir yang diisi. Mahasiswa dapat mengunggah kembali revisi laporan PKL. Jika tidak ada revisi lagi, dosen pembimbing dapat menyetujui laporan PKL tersebut dan mahasiswa dapat melihat hasil statusnya.

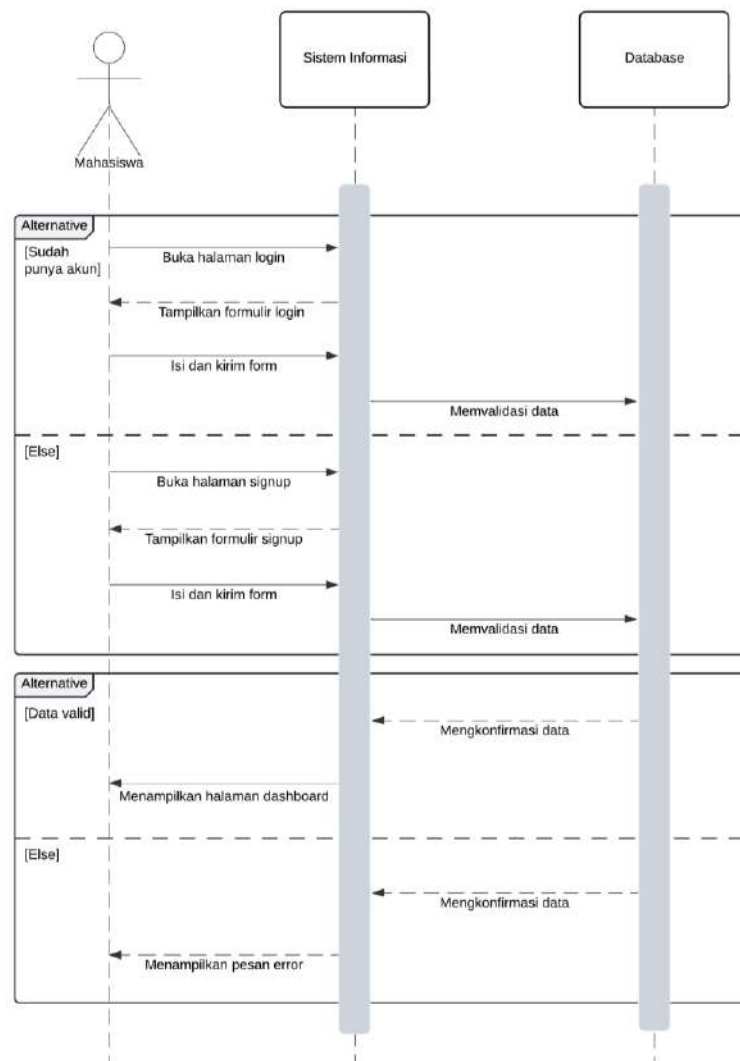


Gambar 3.7 Activity diagram kegiatan penilaian

Aktivitas penilaian antara mahasiswa dan dosen ditunjukkan pada gambar 3.7. Mahasiswa akan mendapat jadwal ujian PKL beserta dosen pengujinya. Mahasiswa kemudian dapat mengikuti ujian PKL dan dosen dapat memberikan nilai kepada mahasiswa di dasbor sistem. Nilai dapat dilihat langsung oleh mahasiswa.

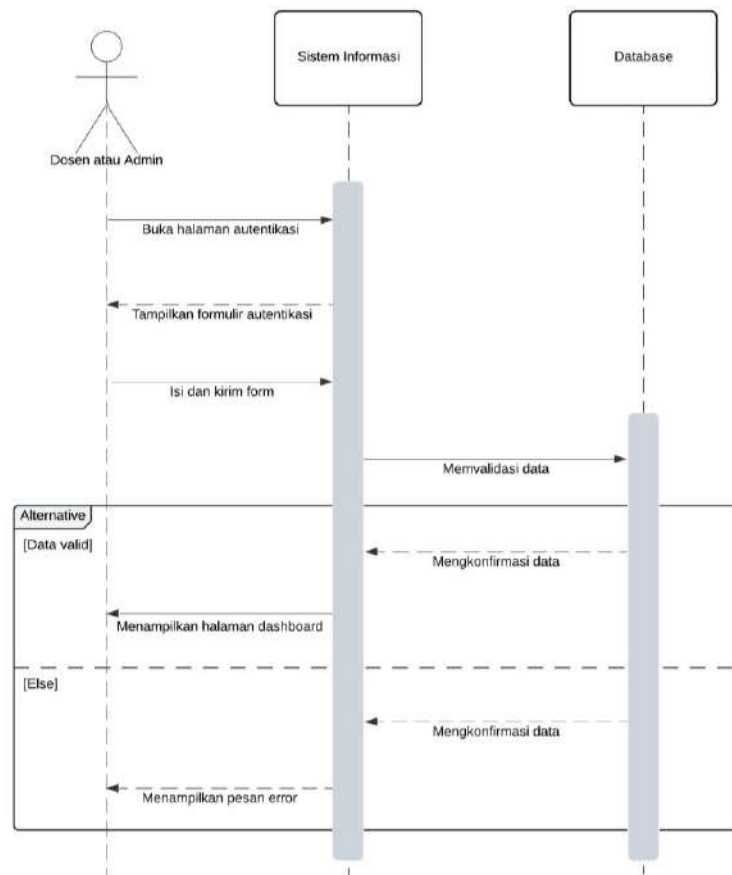
3.2.3 Sequence Diagram

Sequence Diagram menunjukkan rangkaian pesan yang disusun dalam urutan waktu. Terdapat garis vertikal yang merepresentasikan peran tersebut sepanjang waktu selama interaksi, disebut dengan lifeline. Pesan-pesan digambarkan sebagai panah yang menghubungkan lifeline tersebut. Pesan adalah komunikasi satu arah antara dua objek, yaitu aliran kontrol yang membawa informasi dari pengirim ke penerima. Pesan dapat berupa sinyal (respon) atau panggilan (permintaan) [19]. Berikut ini adalah Sequence Diagram untuk sistem informasi PKL:



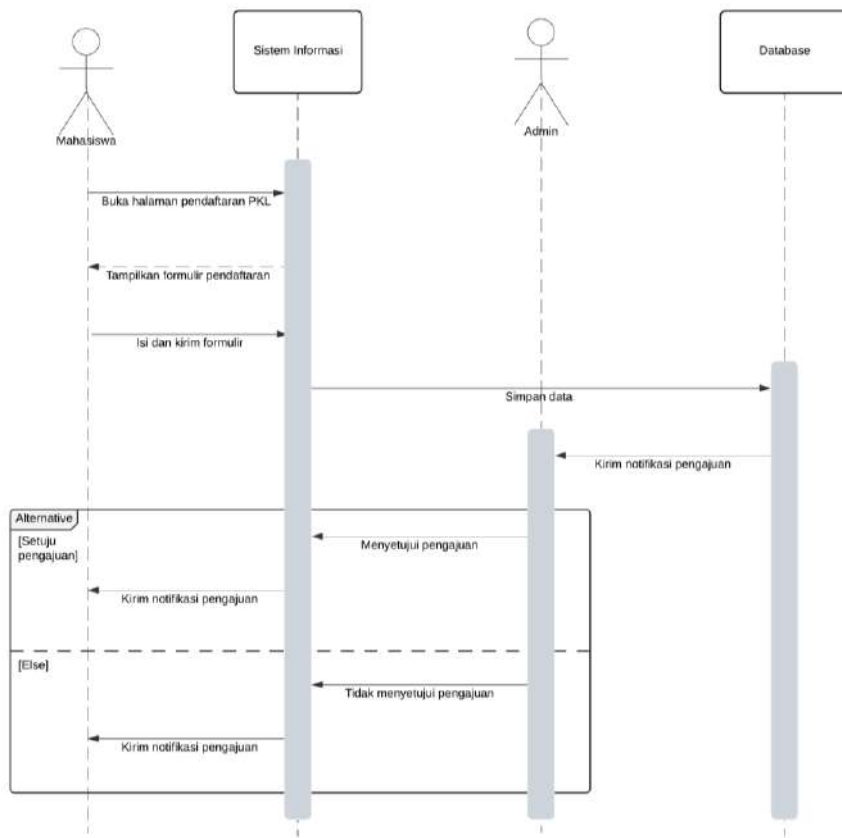
Gambar 3.8 Sequence diagram autentikasi mahasiswa

Gambar 3.8 menggambarkan proses login mahasiswa pada sistem. Sebelumnya, kotak “*alternative*” menunjukkan pengkodisian [21] dimana mahasiswa dapat memiliki 2 skenario yaitu belum punya akun dan sudah punya akun, begitu juga pengkodisian ketika data valid dan tidak valid. Setelah data dinyatakan valid, maka mahasiswa akan diarahkan ke dasbor.

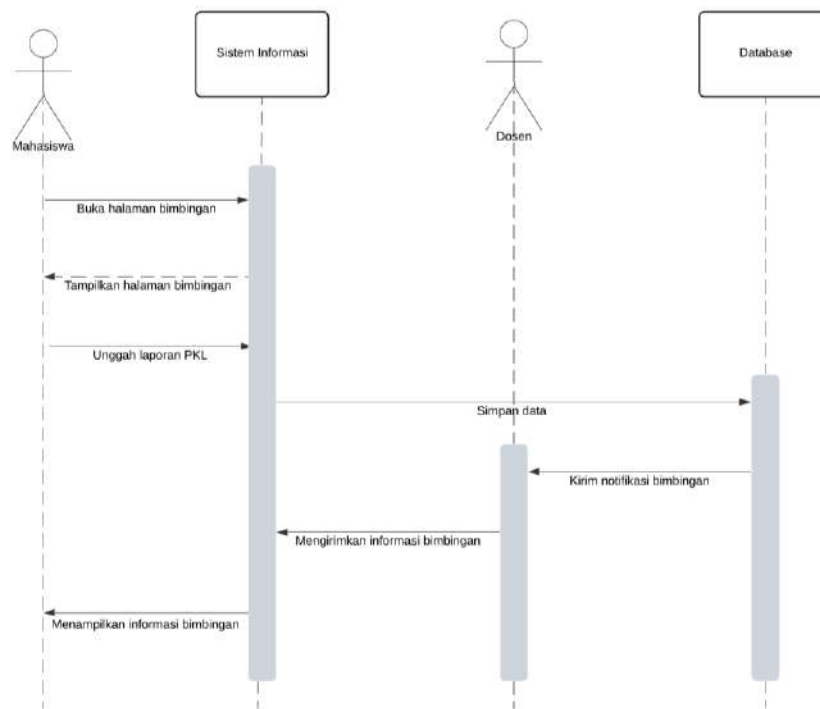


Gambar 3.9 Sequence diagram autentikasi dosen dan admin

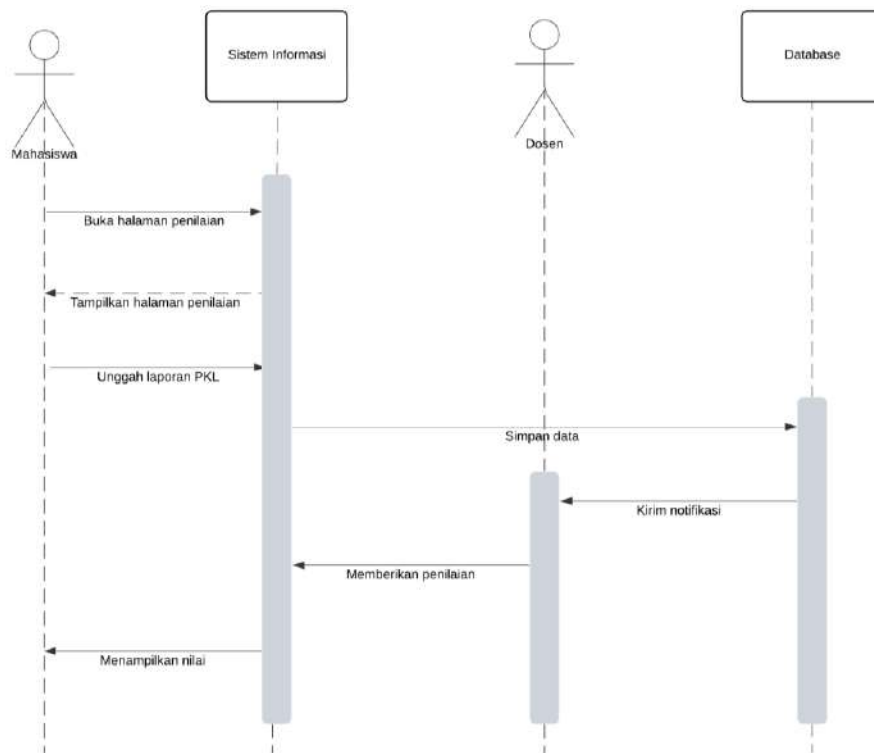
Gambar 3.9 menggambarkan proses login dosen pada sistem informasi PKL. Dosen tidak perlu mendaftar akun. Pada proses ini hanya terjadi pengkodisian ketika sistem memeriksa data.



Gambar 3.10 Sequence diagram pengajuan kegiatan PKL



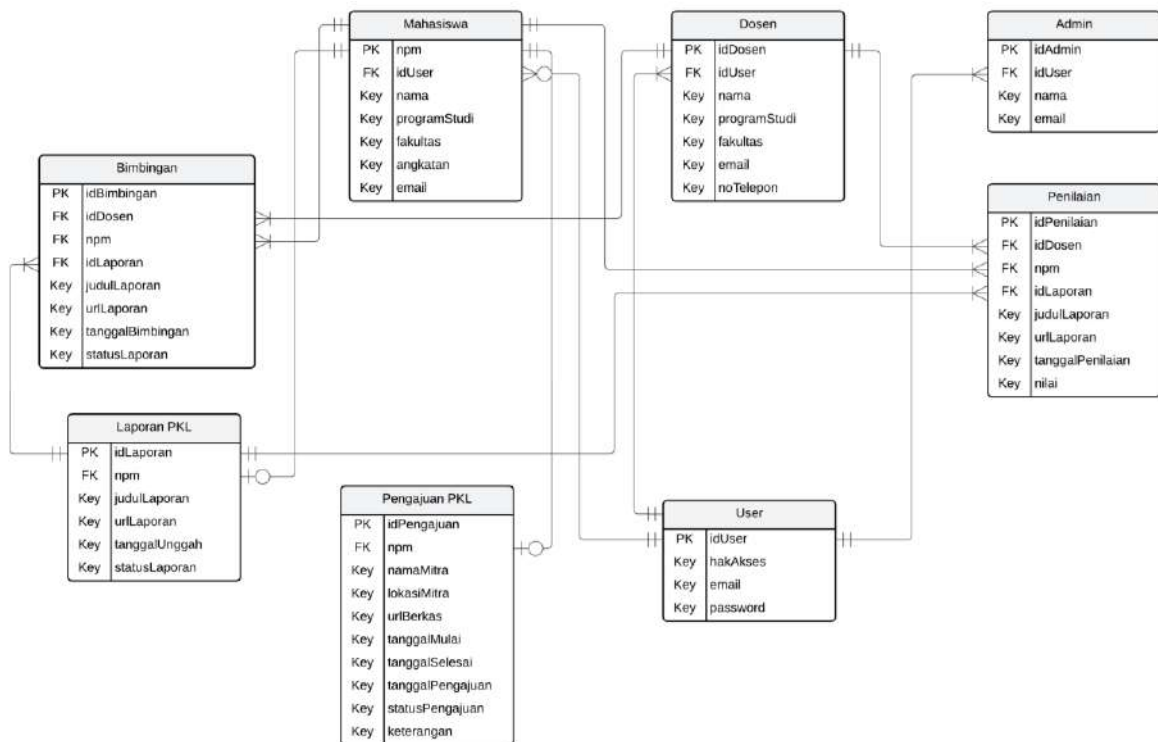
Gambar 3.11 Sequence diagram kegiatan bimbingan



Gambar 3.12 Sequence diagram kegiatan penilaian

3.2.4 Entity Relationship Diagram

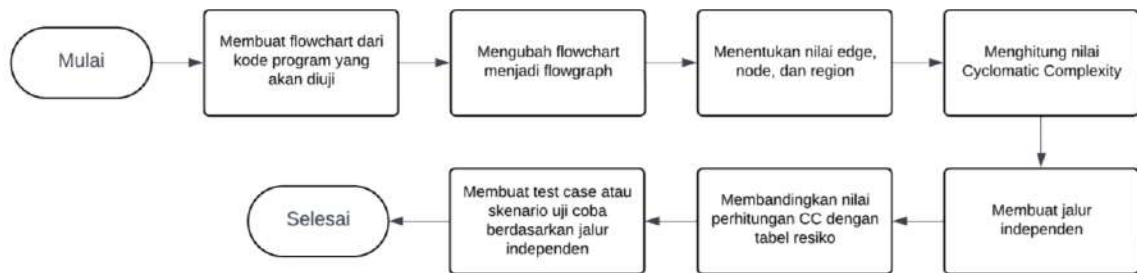
Entity Relationship Diagram atau ERD adalah sebuah diagram yang digunakan untuk mevisualisasikan data pada basis data [20]. Komponen-komponen yang ada pada ERD adalah entitas, hubungan, atribut, dan kardinalitas. Entitas adalah suatu hal yang dapat didefinisikan (seperti orang, objek, konsep, atau peristiwa) yang mampu menyimpan data tentang dirinya. Dalam diagram, entitas biasanya ditampilkan sebagai persegi panjang. Hubungan mendefinisikan bagaimana entitas terkait atau berkomunikasi satu sama lain. Hubungan digambarkan dengan anak panah. Atribut adalah karakteristik yang dimiliki oleh suatu entitas. Kardinalitas menjelaskan jumlah maksimum dan minimum dari hubungan antara entitas dalam model basis data. Ini menggambarkan seberapa banyak satu entitas dapat terhubung dengan entitas lain. Jenis-jenis kardinalitas adalah zero or one, many, one, one (and only one), zero or many, serta one or many [22]. Berikut adalah Entity Relationship Diagram pada sistem informasi PKL:



Gambar 3.13 Entity relationship diagram sistem informasi PKL

Pada Entity Relationship Diagram sistem informasi PKL terdapat 8 entitas, yaitu mahasiswa, dosen, admin, user, pengajuan PKL, bimbingan, laporan PKL, dan penilaian. Setiap entitas memiliki atribut masing-masing. Setiap entitas pada ERD sistem informasi PKL memiliki primary key atau PK, primary key adalah suatu atribut atau kumpulan atribut dalam basis data yang digunakan untuk mengidentifikasi setiap baris dalam tabel secara unik [23]. Entitas dapat memiliki foreign key (FK) yang merupakan atribut atau sekumpulan atribut dalam sebuah tabel yang digunakan untuk menghubungkan antara tabel tersebut dan tabel lain [23].

3.2.5 Langkah Pengujian dan Skenario Pengujian Sistem



Gambar 3.14 Langkah pengujian

Langkah pengujian pada penelitian ini ditunjukkan pada gambar 3.14. Langkah pengujian ini merupakan keseluruhan proses pengujian menggunakan white box testing dengan teknik basic path testing. Skenario pengujian sistem ini akan dilakukan berdasarkan hasil jalur independen. Data skenario pengujian sistem akan ditampilkan sebagai berikut:

No	Nama Skenario	Kegiatan	Target Hasil	Hasil	Keterangan
1	-	-	-	-	-
2	-	-	-	-	-
3	-	-	-	-	-

Tabel 3.1 Skenario pengujian sistem

Setiap hasil pengujian yang didapat nantinya akan ditulis berdasarkan tabel 3.1. Tabel skenario pengujian sistem memiliki beberapa kolom di antaranya adalah:

- Kolom “No” digunakan untuk penomoran pada setiap pengujian yang dilakukan.
- Kolom “Nama Skenario” digunakan untuk menulis nama skenario pengujian sistem yang dilakukan.
- Kolom “Kegiatan” digunakan untuk menjelaskan kegiatan yang dilakukan penguji selama satu skenario pengujian sistem.
- Kolom “Target Hasil” digunakan untuk menulis spesifikasi hasil yang diinginkan.

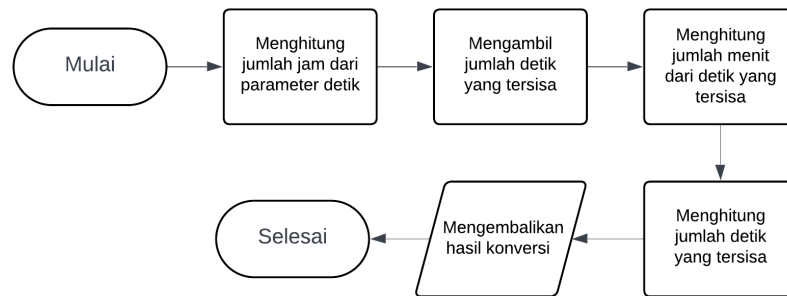
- e. Kolom “Hasil” digunakan untuk menulis hasil yang didapat saat pengujian sistem.
- f. Kolom “Keterangan” digunakan untuk memberi keterangan hasil pengujian sistem. Keterangan pengujian sistem dapat bernilai sesuai atau tidak sesuai. Sesuai memiliki arti nilai kolom “target hasil” dan “hasil” sama, sedangkan tidak sesuai memiliki arti bahwa nilai kolom “target hasil” dan “hasil” tidak sama.

3.2.6 Implementasi Pengujian White Box

Pada bagian ini akan dijelaskan contoh implementasi sederhana pengujian white box. Di bawah ini terdapat fungsi yang dapat mengubah jumlah detik menjadi format jam, menit, dan detik.

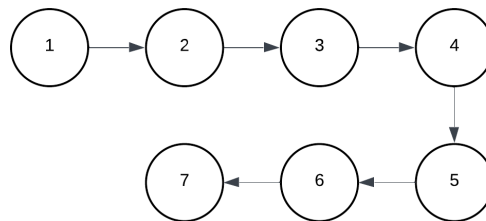
```
function convertToHoursMinutes(seconds) {  
  const hours = Math.floor(seconds / 3600);  
  const remainingSeconds = seconds % 3600;  
  const minutes = Math.floor(remainingSeconds / 60);  
  const finalSeconds = remainingSeconds % 60;  
  
  return `${hours}h ${minutes}m ${finalSeconds}s`;  
}
```

Dari kode tersebut selanjutnya akan diubah ke dalam bentuk flowchat, hasil perubahannya akan menjadi seperti berikut:



Gambar 3.15 Flowchart fungsi convertToHoursMinute

Langkah ketiga yaitu membuat representasi flowgraph dari bentuk flowchart yang sudah dibuat. Berikut adalah bentuk flowgraph nya:



Gambar 3.16 Flowgraph fungsi convertToHoursMinute

Berdasarkan flowgraph yang ditunjukkan pada gambar 3.16, maka diperoleh node berjumlah 7, edge berjumlah 6, region berjumlah 1, dan predicate node berjumlah 0. Langkah keempat yaitu menghitung Cyclomatic Complexity, perhitungannya adalah sebagai berikut:

a. $V(G) = E - N + 2$

$$V(G) = 6 - 7 + 2$$

$$V(G) = 1$$

b. $V(G) = P + 1$

$$V(G) = 0 + 1$$

$$V(G) = 1$$

c. Diketahui jumlah region = 1

Dari perhitungan sebelumnya, diperoleh jalur independen pada fungsi `convertToHoursMinute` berjumlah 1 antara lain sebagai berikut:

- a. 1 – 2 – 3 – 4 – 5 – 6 – 7

Setelah diketahui jalur independennya, maka langkah selanjutnya dilakukan perbandingan menggunakan tabel hubungan cyclomatic complexity dengan risiko seperti di bawah ini:

Nilai CC	Tipe Prosedur	Tingkat Risiko
1 – 4	Prosedur sederhana	Rendah
5 – 10	Prosedur terstruktur dengan baik dan stabil	Rendah
11 – 20	Prosedur lebih kompleks	Menengah
21 – 50	Prosedur kompleks dan kritis	Tinggi
> 50	Rentan kesalahan, sangat mengganggu, prosedur tidak dapat diuji	Sangat Tinggi

Tabel 3.2 Hubungan cyclomatic complexity dengan risiko

Menurut tabel risiko, fungsi `convertToHoursMinute` memiliki tingkat risiko yang rendah dengan prosedur sederhana karena memiliki jalur independen berjumlah 1. Setelah mengetahui hal tersebut, langkah berikutnya yaitu membuat test case atau skenario uji coba seperti berikut ini:

No	Nama Skenario	Kegiatan	Target Hasil	Hasil	Keterangan
----	---------------	----------	--------------	-------	------------

1	Uji fungsi convert to hours minute	Menguji konversi detik menjadi format jam, menit, dan detik	Berhasil mengonversikan dengan format jam, menit, dan detik	Berhasil mengonversikan dengan format jam, menit, dan detik	Sesuai
---	---	---	---	---	--------

Tabel 3.3 Test case fungsi convertToHoursMinute

3.2.7 Computer System Usability Questionnaire (CSUQ)

Setelah pengembangan sistem informasi PKL telah selesai, langkah selanjutnya adalah melakukan survei kegunaan pada sistem menggunakan Computer System Usability Questionnaire (CSUQ). Pada penelitian ini, penulis akan memberikan 16 pernyataan yang dapat dinilai oleh 30 pengguna menggunakan skala angka dari 1 hingga 7, dimana 1 berarti “sangat tidak setuju” dan 7 berarti “sangat setuju”. Penilaian kegunaan sistem terbagi menjadi 4 kategori, yaitu Usefulness (SYSUSE), Information Quality (INFOQUAL), Interface Quality (INTERQUAL), and Overall Usability (OVERALL). Skor dari setiap pernyataan akan dihitung jumlahnya, kemudian dihitung rata-ratanya untuk mengetahui nilai keseluruhan. Apabila nilai rata-ratanya berkisar di antara 5 – 7, itu mengindikasikan kepuasan pengguna ketika menggunakan sistem informasi [24].

Daftar Pustaka

- [1] A. G. Gani, "SEJARAH dan PERKEMBANGAN INTERNET DI INDONESIA."
- [2] E. Mufida, E. Rahmawati, dan H. Hertiana, "Rancang Bangun Sistem Informasi Inventory pada Salon Kecantikan," *Jurnal Mantik Penusa*, vol. 3, no. 3, hlm. 99–102, Des 2019.
- [3] M. Arifin, "Analisa dan Perancangan Sistem Informasi Praktek Kerja Lapangan pada Instansi/Perusahaan," vol. 5, Apr 2014.
- [4] S. T. Safitri dan D. Supriyadi, "Rancang Bangun Sistem Informasi Praktek Kerja Lapangan Berbasis Web dengan Metode Waterfall," *JURNAL INFOTEL - Informatika Telekomunikasi Elektronika*, vol. 7, no. 1, hlm. 69, Mei 2015, doi: 10.20895/infotel.v7i1.32.
- [5] C. T. Pratala, E. M. Asyer, I. Prayudi, dan A. Saifudin, "Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, hlm. 111, Jun 2020, doi: 10.32493/informatika.v5i2.4713.
- [6] A. Verma, A. Khatana, dan S. Chaudhary, "A Comparative Study of Black Box Testing and White Box Testing," *International Journal of Computer Sciences and Engineering*, vol. 5, no. 12, hlm. 301–304, Des 2017, doi: 10.26438/ijcse/v5i12.301304.
- [7] A. A. Al-Hassan, B. AlGhannam, M. Bin Naser, dan H. Alabdulrazzaq, "An Arabic Translation of the Computer System Usability Questionnaire (CSUQ) with Psychometric Evaluation Using Kuwait University Portal," *Int J Hum Comput Interact*, vol. 37, no. 20, hlm. 1981–1988, Des 2021, doi: 10.1080/10447318.2021.1926117.
- [8] A. Mishra dan D. Dubey, "A comparative study of different software development life cycle models in different scenarios," *International Journal of Advance research in computer science and management studies*, vol. 1, no. 5, 2013.
- [9] R. Fojtik, "Extreme Programming in development of specific software," *Procedia Comput Sci*, vol. 3, hlm. 1464–1468, 2011, doi: 10.1016/j.procs.2011.01.032.
- [10] M. Ro'if, T. Afirianto, dan S. H. Wijoyo, "Pengembangan Sistem Informasi Praktik Kerja Lapangan (PKL) Siswa Berbasis Website Menggunakan Metode Extreme Programming (Studi Kasus: SMK Negeri 1 Sumenep)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 1, hlm. 1–10, Feb 2024, doi: 10.25126/jtiik.20241116452.
- [11] C. T. Pratala, E. M. Asyer, I. Prayudi, dan A. Saifudin, "Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, hlm. 111, Jun 2020, doi: 10.32493/informatika.v5i2.4713.

- [12] Atika Rahmadani Utami Br Ginting, Sarah Astiti, dan Khairun Nisa Meiah, "Evaluasi Usability Aplikasi English Competency Test (ECT) Menggunakan Metode Computer System Usability Questionner (CSUQ): Studi Kasus Pusat Bahasa Institut Teknologi Telkom Purwokerto," *EduInovasi: Journal of Basic Educational Studies*, vol. 4, no. 3, hlm. 1485–1499, Sep 2024, doi: 10.47467/edu.v4i3.4294.
- [13] T. Sutabri, *Konsep sistem informasi*. Penerbit Andi, 2012.
- [14] H. A Jartarghar, G. Rao Salanke, A. K. A.R, S. G.S, dan S. Dalali, "React Apps with Server-Side Rendering: Next.js," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 14, no. 4, hlm. 25–29, Des 2022, doi: 10.54554/jtec.2022.14.04.005.
- [15] E. Kroon Celander dan A. Möllestål, "A Comparative Analysis of Next.js, SvelteKit, and Astro for E-commerce Web Development," 2024.
- [16] K. B. Lloyd dan J. Solak, "Flowcharting Techniques," dalam *Encyclopedia of Information Systems*, Elsevier, 2003, hlm. 331–344. doi: 10.1016/B0-12-227240-4/00071-X.
- [17] A. B. Chaudhuri, *Flowchart and algorithm basics: The art of programming*. Mercury Learning and Information, 2020.
- [18] C. P. C. Munaiseche dan G. C. Rorimpandey, "Penerapan Metode Basis Path Analysis dalam Pengujian White Box Sistem Pakar," *Prosiding SISFOTEK*, vol. 5, no. 1, hlm. 124–128, 2021.
- [19] L. Jacobson dan J. R. G. Booch, "The unified modeling language reference manual," 2021.
- [20] Q. Li dan Y.-L. Chen, "Entity-Relationship Diagram," dalam *Modeling and Analysis of Enterprise and Information Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, hlm. 125–139. doi: 10.1007/978-3-540-89556-5_6.
- [21] "UML Sequence Diagram Tutorial," <https://www.lucidchart.com/pages/uml-sequence-diagram>.
- [22] "What is an Entity Relationship Diagram (ERD)?," <https://www.lucidchart.com/pages/er-diagrams>.
- [23] J. L. Harrington, *Relational database design and implementation*. Morgan Kaufmann, 2016.
- [24] A. Acala dan H. Talirongan, "Assessing User Satisfaction and Usability of a University Portal: A Quantitative Study Utilizing the Computer System Usability Questionnaire (CSUQ)," *Psychology and Education: A Multidisciplinary Journal*, vol. 14, no. 4, hlm. 408–416, 2023, doi: 10.5281/zenodo.8426291.

