**Pengembangan Sistem Informasi Terpadu untuk Pelaporan Emisi Gas Rumah Kaca dengan Memanfaatkan Teknologi IoT dan Machine Learning pada Bidang Scope 1**

**Oleh :**

**RIFQI ABDULAZIZ**

**1202210103**



**PROGRAM STUDI STRATA 1 SISTEM INFORMASI**

**FAKULTAS REKAYASA INDUSTRI**

**UNIVERSITAS TELKOM**

**2024**

*<tambahkan sebuah halaman blank antara sampul dengan abstrak>*

ABSTRAK

Abstrak terdiri dari 500-800 kata. Abstrak menyajikan intisari dari penelitian yang meliputi latar belakang, permasalahan yang dikaji, dan metode yang digunakan. Selanjutnya, tuliskan ulasan singkat terkait hasil dan analisis penelitian, diikuti dengan kesimpulan dan manfaat yang dapat diambil dari penelitian.

Penulisan abstrak harus ringkas dan jelas, serta menggunakan bahasa yang dapat dipahami oleh pembaca dari kalangan umum tanpa mengurangi inti dari hasil tugas akhir. Penggunaan istilah-istilah yang terlalu teknis lebih baik dihindari. Jika istilah tersebut memang diperlukan, gunakan *font italic* untuk istilah tersebut. Diharapkan melalui abstrak ini, pembaca dapat memiliki gambaran umum terkait maksud, tujuan, atau hasil dari tugas akhir ini.

Kata kunci**—***sistem informasi, akademik, Telkom, aplikasi* [Sertakan kata kunci di sini sejumlah 3-5 kata kunci dengan format bold dan *italic*]

*ABSTRACT*

*Abstract consists of 500-800 words. Abstract summarizes the research/final project which consists of study’s background, problem statements, and methods. Describe briefly the study results and your analysis, followed by overall summary and the potential benefits people can obtain from this research / final project.*

*Abstract should be written in concise and clear fashion. Please use an easily understandable language so that readers from all background can understand the contents. It is recommended to avoid highly technical terms.*

*For Abstract in English, whole section is written in italic form.*

Keywords***—*** *information system, academics, Telkom, application [write your keywords here, 3- 5 words]*

LEMBAR PENGESAHAN

Tugas Akhir dengan judul :

**Pengembangan Sistem Informasi Terpadu untuk Pelaporan Emisi Gas Rumah Kaca dengan Memanfaatkan Teknologi IoT dan Machine Learning pada Bidang Scope 1**

Telah disetujui dan disahkan pada Sidang Tugas Akhir

Program Studi Strata 1 Teknik Industri / Sistem Informasi

Fakultas Rekayasa Industri Universitas Telkom

**Oleh :**

**RIFQI ABDULAZIZ**

**1202210103**

Bandung, <28 April 2025> **(Tanggal Selesai Revisi Terakhir)**

Disetujui oleh,

|  |  |
| --- | --- |
| Pembimbing 1, | Pembimbing 2, |
|  |  |
| Ir. Ahmad Musnansyah, M.Sc.  NIP : 20610001 | Dr. Sinung Suakanto,  NIP Pembimbing |

LEMBAR PERNYATAAN ORISINALITAS

|  |  |  |  |
| --- | --- | --- | --- |
| Put Your Formal Photo Here | Nama | : | Rifqi Abdulaziz |
| NIM | : | 1202210103 |
| Alamat | : | Cluster Pajajaran C/22, Taman Royal 2, Cipondoh, Tangerang, Banten |
| No. Tlp | : | 081212822148 |

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri. Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.

Bandung, 28 April 2025

(Tanda Tangan)

**(harus ttd basah mahasiswa)**

Rifqi Abdulaziz

Kata Pengantar

Alhamdulillah, puji syukur kepada Tuhan Yang Maha Esa. Tugas Akhir ini merupakan sebuah perjalanan di tahun terakhir studi sarjana saya di Program Studi Strata 1 Teknik Industri / Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom.

Ucapan terima kasih saya kepada:

1. Pembimbing Tugas Akhir saya, Ir. Ahmad Musnansyah, M.Sc. dan Ekky Novrizal Alam, S. , atas arahannya dan bimbingannya selama pengerjaan penelitian ini.
2. Divisi Teknologi Informasi – Perusahaan PT XYZ sebagai studi kasus dalam tugas akhir ini yang telah bersedia menyediakan sumber data mulai dari wawancara, observasi, dan dokumen untuk pelaksanaan penelitian.
3. Kementerian Pendidikan dan Kebudayaan Republik Indonesia melalui Beasiswa Bidikmisi sehingga saya dapat menempuh dan menyelesaikan pendidikan sarjana ini.

Terima kasih saya juga untuk seluruh pihak yang secara tidak langsung berkontribusi dalam penyelesaian Tugas Akhir ini.

Daftar Isi

ABSTRAK ii

*ABSTRACT* iii

LEMBAR PENGESAHAN iv

LEMBAR PERNYATAAN ORISINALITAS v

Kata Pengantar i

Daftar Isi ii

Daftar Gambar v

Daftar Tabel vi

Daftar Simbol vii

Daftar Istilah viii

Bab I PENDAHULUAN 1

I.1 Latar Belakang 1

I.2 Perumusan Masalah 2

I.3 Tujuan Penelitian 2

I.4 Manfaat Penelitian 3

Bab II TINJAUAN PUSTAKA 4

II.1 Literatur Penunjang Tugas Akhir 4

II.1.1 Emisi Gas Rumah Kaca (GRK) 4

II.1.2 Emisi Gas Rumah Kaca *Scope* 1 5

II.1.3 Sistem Informasi 7

II.1.4 *Real-Time Data Processing* 9

II.1.5 Dasbor dan KPI 10

II.1.6 Perhitungan Emisi Karbon 11

II.1.7 Internet of Things 12

II.1.8 Protokol MQTT (Message Queuing Telemetry Transport) 14

II.1.9 Unified Modelling Language (UML) 15

II.1.10 Use Case Diagram 16

II.1.11 Sequence Diagram 17

II.1.12 Class Diagram 19

II.1.13 Activity Diagram 20

II.1.14 REST API 21

II.1.15 Software Development Lifecycle 22

II.1.16 Software Development Method 24

Bab III Metodologi Penelitian 27

III.1 Kerangka Berpikir 27

III.2 Sistematika Penyelesaian Masalah 28

III.2.1 Identifikasi 29

III.2.1.1 Penetapan Latar Belakang 30

III.2.1.2 Penetapan Rumusan Masalah 30

III.2.1.3 Penetapan Tujuan Masalah 30

III.2.1.4 Studi Literatur 30

III.2.1.5 Analisis Kebutuhan 30

III.2.2 Tahap Pengembangan Sistem 31

III.2.2.1 Perencanaan 31

III.2.2.2 Analisis dan Desain 31

III.2.2.3 Pengembangan 31

III.2.2.4 Pengujian 32

III.2.2.5 Evaluasi 32

III.2.3 Tahap Kesimpulan dan Saran 32

III.3 Alasan Pemilihan Metode 32

III.4 Batasan dan Asumsi Penelitian 36

III.5 Rencana Jadwal Kegiatan 36

Bab IV Analisis dan Perancangan 40

IV.1 Analisis Proses Bisnis 40

IV.2 Desain Perangkat Lunak 40

IV.3 Spesifikasi Kebutuhan Teknologi 40

Bab V Implementasi dan Pengujian 41

V.1 Hasil Implementasi Perangkat Lunak 41

V.2 Proses dan Hasil Pengujian Perangkat Lunak 41

V.3 Evaluasi Akhir Solusi Sistem Informasi 41

Bab VI Kesimpulan dan Saran 42

VI.1 Kesimpulan 42

VI.2 Saran 42

Bab VII Daftar Pustaka 43

Daftar Gambar

[Gambar 1. Model Konseptual 27](#_Toc186293160)

[Gambar 2. Sistematika Penyelesaian Masalah 29](#_Toc186293161)

Daftar Tabel

[Tabel 1. Tabel Perbandingan Metode 35](#_Toc186293156)

[Tabel 2. Tabel Rencana Jadwal Kegiatan 39](#_Toc186293157)

Daftar Simbol

|  |  |  |
| --- | --- | --- |
| Simbol / Notasi | Deskripsi | Halaman pertama kali digunakan |
|  |  |  |
|  | *Fuzzy demand value* untuk permintaan di bawah rata-rata | 12 |
|  | *Fuzzy demand value* untuk permintaan di atas rata-rata | 12 |
| *a* | *Demand* minimal di dalam satu tahun | 12 |
| *b* | Rata-rata permintaan dalam satu tahun | 12 |
| *B* | *Production Reorder Point* | 13 |
| *c* | Permintaan maksimum di dalam satu tahun | 13 |
| *C* | Harga produksi satu unit | 13 |
| *H* | Biaya *holding* untuk satu unit per tahun | 13 |
| *m* | *Production Run* | 13 |
| *p* | *Production Rate* | 13 |
| *P* | Biaya setup untuk satu kali produksi | 13 |
| *Q* | Jumlah produksi optimal di tiap kali produksi | 13 |
| *r* | Angka / tingkat permintaan | 13 |
| *R* | Permintaan per tahun (dalam unit) | 13 |

**PS :**

**Urutkan dan klasifikasi simbol berdasarkan tipenya**

**Contoh bisa dilihat di atas**

Daftar Istilah

|  |  |  |  |
| --- | --- | --- | --- |
| Istilah |  | Deskripsi | Halaman pertama kali digunakan |
|  |  |  |  |
| GRK | : | Gas Rumah Kaca, merupakan gas yang dihasilkan oleh aktivitas manusia, seperti pembakaran bahan bakar fosil, asap pabrik, dan penggunaan alat transportasi. | 17 |
| CO2 | : |  | 18 |
| IoT | : |  | 19 |
| SDGs | : |  | 20 |
| *Waste* | : | Pemborosan atau segala aktivitas kerja yang tidak memberikan nilai tambah dalam proses transformasi *input* menjadi *output*. | 20 |
| CTQ | : | *Critical To Quality*, merupakan Atribut-atribut yang sangat penting untuk diperhatikan karena berkaitan langsung dengan kebutuhan dan kepuasan pelanggan. Merupakan elemen dari suatu produk, proses, atau praktek-praktek yang berdampak langsung pada kepuasan pelanggan. | 20 |
| Sev | : | *Severity*, merupakan suatu penilaian dari seberapa serius efek dari *mode* kegagalan potensial terhadap pelanggan. | 21 |
| Occ | : | *Occurence*, menunjukkan nilai keseringan suatu masalah yang terjadi karena *potential cause.* | 21 |
| Det | : | *Detection*, merupakan alat kontrol yang digunakan untuk mendeteksi *potential cause.* Identifikasi metode-metode yang diterapkan untuk mencegah atau mendeteksi penyebab dari *mode* kegagalan. | 21 |

# PENDAHULUAN

## Latar Belakang

Dalam konteks global, kekhawatiran terhadap emisi gas rumah kaca yang timbul dari berbagai aktivitas sehari-hari, mulai dari transportasi umum hingga kegiatan industri termasuk pabrik dan pertambangan, telah menjadi perhatian utama (Mikhaylov dkk., 2020). Ini mendorong para pemangku kepentingan di seluruh dunia untuk melakukan pengurangan emisi bahkan penekanan total dalam jangka panjang. Indonesia telah turut serta dalam proyeksi tersebut dengan pemerintah menetapkan target penurunan emisi gas rumah kaca sebesar 29% pada tahun 2030 dari upaya domestik, serta penurunan sebesar 41% melalui bantuan eksternal (Ditjen PPI KLHK, 2017). Sejalan dengan sasaran resmi penurunan emisi gas rumah kaca yang diumumkan oleh pemerintah Indonesia, sektor industri termasuk pertambangan diharapkan dapat memberikan kontribusi yang signifikan untuk mencapai tujuan tersebut (Ari Wibowo, 2013). Tekanan untuk mengurangi emisi ini tidak hanya berasal dari regulasi nasional tetapi juga dari komitmen global dalam melawan perubahan iklim (Malihah, 2022). Oleh karena itu, sektor pertambangan dituntut untuk mengadopsi pendekatan yang lebih efektif dalam pemantauan, manajemen, dan pelaporan emisi gas rumah kaca mereka (Pratama. MR dkk., 2023).

Dampaknya akan signifikan bagi industri pertambangan yang membantu secara substansial dalam emisi gas rumah kaca (Sekarini & Setiadi, 2022). Sistem pelaporan emisi gas rumah kaca di sektor pertambangan Indonesia saat ini menghadapi berbagai tantangan dalam penyusunannya (Arifiyanto & Sindu, 2020). Besarnya tantangan ini membuat sebagian industri pertambangan memiliki banyak tantangan dalam membuat laporan, sehingga di banyak lokasi pertambangan, data yang terkumpul tidak terintegrasi yang menyebabkan minimnya pengolahan dan analisis data untuk memprediksi emisi gas rumah kaca yang dihasilan, terutama dari emisi *scope 1* yang mana mencakup berbagai emisi yang dihasillkan langsung dari pertambangan tersebut (Azadi dkk., 2020).

Situasi ini memang disayangkan, mengingat kemajuan teknologi saat ini menawarkan berbagai alat yang dapat membantu mengurangi biaya sistem pelaporan, seperti sensor pendeteksi emisi gas rumah kaca yang dihasilkan oleh aktivitas pertambangan, seperti CO2 dan CH4 yang banyak dihasilkan dalam aktivitas tersebut (Adi Sucipto dkk., 2023). Oleh karena itu, ada kebutuhan mendesak untuk mengembangkan sistem yang mengintegrasikan fungsi-fungsi tersebut dalam satu platform yang dapat digunakan industri secara efisien, akurat, dan biaya yang terjangkau (Raza dkk., 2020).

Sistem ini diharapkan dapat menyelesaikan masalah pelaporan yang tidak terotomasi dikarenakan penghitungan emisi karbon yang masih manual, dan mampu memberikan gambaran seberapa besar jejak karbon yang dihasilkan dari industri kepada pihak pemangku kepentingan di perusahaan dan pemerintahan.

## Perumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan permasalahan untuk penelitian ini adalah:

1. Bagaimana cara merancang serta membangun sistem informasi guna pelaporan emisi gas rumah kaca dalam lingkup Scope 1 dengan menggunakan teknologi Internet of Things( IoT)?
2. Bagaimana cara mempersiapkan dashboard serta mengotomatisasi pelaporan periodik secara terjadwal?
3. Bagaimana auditor dari pemerintah serta internal industri bisa memantau emisi gas rumah kaca secara otomatis dari entitas yang diawasi?

## Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mengembangkan sistem informasi terpadu yang sanggup mengotomatiskan pelaporan emisi gas rumah kaca lingkup Scope 1, mencakup analisis, prediksi, serta pengolahan informasi yang disajikan dalam wujud dashboard berbasis informasi real-time.
2. Memanfaatkan teknologi IoT guna mengumpulkan informasi yang digunakan dalam proses analisis serta prediksi di dalam sistem, membolehkan pengolahan informasi secara otomatis serta efektif.
3. Mengukur daya guna dan akurasi dari sistem yang dirancang buat pelaporan emisi gas rumah kaca, guna menentukan sistem tersebut berperan sesuai dengan kebutuhan serta memberikan hasil yang akurat.

## Batasan Masalah

Dalam penelitian ini, ruang lingkup masalah dibatasi pada beberapa aspek berikut:

1. Fokus pada Emisi Gas Rumah Kaca Scope 1

Penelitian ini hanya berfokus pada emisi gas rumah kaca yang termasuk dalam kategori Scope 1, yaitu emisi langsung yang dihasilkan dari kegiatan operasional di area pertambangan, seperti pembakaran bahan bakar fosil pada alat berat dan pelepasan gas dari proses ekstraksi.

1. Pengembangan Sistem Terpadu

Sistem yang dikembangkan bertujuan untuk mengintegrasikan proses pemantauan, pengumpulan, dan pelaporan emisi gas rumah kaca dalam satu platform. Sistem ini difokuskan pada penerapan teknologi sensor untuk mendeteksi gas CO₂ dan CH₄ secara real-time.

1. Penggunaan Teknologi IoT

Batasan teknologi yang digunakan mencakup pemanfaatan Internet of Things (IoT) untuk pengumpulan data sensor serta pengolahan data secara otomatis dalam sistem berbasis cloud atau lokal.

1. Area Studi di Industri Pertambangan

Implementasi dan pengujian sistem difokuskan pada sektor industri pertambangan di Indonesia, mengingat sektor ini memberikan kontribusi besar terhadap emisi gas rumah kaca nasional.

1. Tidak Membahas Intervensi Pengurangan Emisi

Penelitian ini tidak membahas metode teknis untuk pengurangan emisi itu sendiri, seperti penerapan teknologi energi terbarukan atau perubahan proses produksi, melainkan hanya membahas aspek pemantauan, penghitungan, dan pelaporan emisi.

1. Parameter Emisi yang Diukur

Parameter yang menjadi fokus dalam sistem ini adalah konsentrasi dan estimasi kuantitatif emisi gas CO₂ dan CH₄ yang dilepaskan dari aktivitas operasional pertambangan.

1. Batasan Waktu

Sistem pelaporan yang dikembangkan dibatasi pada pengumpulan dan analisis data emisi dalam periode tertentu (misalnya harian atau bulanan), dan tidak mencakup proyeksi jangka panjang atau simulasi masa depan emisi.

## Manfaat Penelitian

Manfaat penelitian ini:

1. Bagi perusahaan, diharapkan penelitian ini dapat memberikan solusi efisien dan terjangkau untuk pelaporan emisi gas rumah kaca, mengurangi ketergantungan pada proses manual dan biaya audit yang tinggi, serta mendukung pemenuhan regulasi dan komitmen lingkungan..
2. Bagi auditor, diharapan penelitian ini dapat mempermudah pemantauan dan verifikasi emisi gas rumah kaca secara otomatis dan real-time, meningkatkan akurasi dan transparansi data dalam pengawasan industri pertambangan.

# TINJAUAN PUSTAKA

## Literatur Penunjang Tugas Akhir

### Emisi Gas Rumah Kaca (GRK)

Emisi gas rumah kaca (GRK) menjadi isu lingkungan yang semakin mendesak, terutama di negara berkembang seperti Indonesia. Dalam hal ini, pengelolaan emisi GRK sangat penting untuk mendukung keberlanjutan dan upaya mitigasi perubahan iklim. Laporan keberlanjutan perusahaan di Indonesia mulai memasukkan pengungkapan emisi GRK sebagai bagian dari informasi lingkungan yang disampaikan, sejalan dengan perkembangan standar Global Reporting Initiative (GRI) yang membantu perusahaan dalam mengidentifikasi dan melaporkan emisi GRK mereka (MR, 2023a). Hal ini mencerminkan peningkatan kesadaran tentang pentingnya transparansi dalam pengelolaan emisi GRK di kalangan pelaku bisnis.

Sektor pengelolaan sampah di Indonesia memberikan kontribusi besar terhadap emisi GRK. Penelitian menunjukkan bahwa emisi GRK dari pengangkutan sampah di Kota Kediri mencapai 56,89 x 10^3 ton per tahun, jauh lebih tinggi dibandingkan dengan pengomposan yang dapat menghindari emisi sebesar -19,37 x 10^3 ton per tahun (Ardisty Sitogasa, 2022a). Metode penghitungan emisi ini mengikuti pedoman IPCC 2006, yang sudah diterima secara luas untuk memperkirakan emisi dari berbagai tahapan pengelolaan sampah, termasuk pengangkutan dan penimbunan (Chaerul dkk., 2020a). Oleh karena itu, penting bagi pemerintah daerah untuk mengimplementasikan strategi mitigasi yang efektif dalam pengelolaan sampah untuk mengurangi emisi GRK.

Selain sektor pengelolaan sampah, sektor pertanian dan peternakan juga menjadi kontributor utama emisi GRK di Indonesia (Ufresti Praptyanti dkk., 2022a). Di Surabaya, misalnya, sektor ini memiliki target pengurangan emisi sebesar 0,008 gigaton pada tahun 2020. Penelitian di Kabupaten Tuban menunjukkan bahwa emisi GRK dari sektor pertanian mencapai 1.665,67 Gg CO2-eq, dengan subsektor pertanian dan peternakan masing-masing berkontribusi 1.092,50 Gg CO2-eq dan 573,17 Gg CO2-eq (Mustikaningrum dkk., 2021). Hal ini menunjukkan bahwa sektor pertanian dan peternakan memerlukan perhatian lebih dalam upaya pengurangan emisi GRK.

Penerapan kebijakan seperti pajak karbon juga dianggap sebagai langkah strategis untuk menurunkan emisi GRK di Indonesia. Pajak karbon dapat memberikan kontribusi pada pendapatan negara sekaligus mendorong transisi menuju teknologi ramah lingkungan (Ihsan, 2023a). Oleh karena itu, pengelolaan emisi GRK di berbagai sektor, termasuk pengelolaan sampah, pertanian, dan industri, sangat penting untuk mencapai target nasional dalam mengurangi dampak perubahan iklim. Selain itu, akuntansi karbon di perusahaan dapat membantu dalam mengurangi emisi GRK dan memperluas daerah penyerapan karbon, yang krusial untuk mitigasi perubahan iklim (Rahmasari, 2024a).

Sehingga, disimpulkan bahwa pengelolaan emisi gas rumah kaca di Indonesia memerlukan pendekatan yang komprehensif dan kolaboratif antara pemerintah, sektor swasta, dan masyarakat. Dengan mengintegrasikan pengungkapan emisi GRK dalam laporan keberlanjutan, serta menerapkan kebijakan yang mendukung pengurangan emisi, Indonesia dapat mengambil langkah signifikan menuju keberlanjutan dan mitigasi perubahan iklim. Upaya ini tidak hanya akan membantu mengurangi dampak negatif dari perubahan iklim, tetapi juga akan berkontribusi pada pembangunan ekonomi yang lebih berkelanjutan dan ramah lingkungan di masa depan.

### Emisi Gas Rumah Kaca *Scope* 1

Emisi gas rumah kaca (GRK) menjadi salah satu isu lingkungan yang mendesak saat ini, yang dikelompokkan dalam tiga kategori, yakni Scope 1, Scope 2, dan Scope 3. Scope 1 mencakup emisi yang dihasilkan langsung dari sumber yang dimiliki atau dikelola oleh perusahaan, seperti emisi dari pembakaran bahan bakar, proses industri, dan kendaraan operasional. Di Indonesia, pengelolaan emisi GRK Scope 1 sangat penting, mengingat kontribusi sektor-sektor seperti industri, transportasi, dan pengelolaan sampah terhadap total emisi nasional. Penelitian mengungkapkan bahwa sektor industri, terutama yang berhubungan dengan pengolahan sumber daya alam, berkontribusi besar terhadap emisi GRK, yang dapat dihitung dan dilaporkan dengan berbagai metode, termasuk yang direkomendasikan oleh IPCC (Chaerul dkk., 2020b; Ufresti Praptyanti dkk., 2022b).

Tantangan utama dalam penghitungan emisi GRK Scope 1 adalah keakuratan data yang digunakan. Penghitungan yang tepat sangat penting untuk memastikan bahwa laporan emisi mencerminkan kenyataan operasional perusahaan. Contohnya, pada pengelolaan sampah, emisi GRK dapat terjadi pada berbagai tahapan, dari pengumpulan hingga pembuangan akhir, dengan faktor emisi yang berbeda pada setiap tahapan (Anifah dkk., 2021; Ardisty Sitogasa, 2022b). Oleh karena itu, perusahaan perlu melakukan inventarisasi emisi secara menyeluruh dan menggunakan data yang valid untuk perhitungan emisi mereka. Penelitian di Kota Kediri menunjukkan bahwa emisi dari pengangkutan sampah lebih tinggi dibandingkan dengan pengolahan sampah, yang menunjukkan perlunya strategi mitigasi yang lebih baik dalam pengelolaan sampah (Ardisty Sitogasa, 2022b).

Laporan keberlanjutan perusahaan juga memainkan peran penting dalam pengungkapan emisi GRK Scope 1. Laporan ini tidak hanya mencakup data kuantitatif tentang emisi, tetapi juga menjelaskan langkah-langkah yang diambil perusahaan untuk mengurangi emisi tersebut. Penggunaan standar Global Reporting Initiative (GRI) dapat membantu perusahaan menyusun laporan yang transparan dan akuntabel (MR, 2023). Dengan demikian, pengungkapan emisi GRK dalam laporan keberlanjutan dapat meningkatkan kesadaran publik dan mendorong perusahaan untuk lebih proaktif dalam mengurangi jejak karbon mereka.

Selain itu, kebijakan pemerintah sangat mempengaruhi pengelolaan emisi GRK Scope 1. Misalnya, penerapan pajak karbon di Indonesia dapat mendorong perusahaan untuk mengurangi emisi melalui insentif ekonomi (Ihsan, 2023b; Rahmasari, 2024b). Kebijakan ini bertujuan tidak hanya untuk mengurangi emisi, tetapi juga meningkatkan pendapatan negara dan mendukung transisi ke teknologi yang lebih ramah lingkungan. Oleh karena itu, kolaborasi antara sektor publik dan swasta sangat penting dalam mencapai target pengurangan emisi yang telah ditetapkan dalam Rencana Aksi Nasional Penurunan Emisi Gas Rumah Kaca (RAN-GRK) (Nugrahayu dkk., 2017).

Secara keseluruhan, pengelolaan emisi gas rumah kaca Scope 1 di Indonesia memerlukan pendekatan yang menyeluruh, melibatkan perhitungan yang akurat, pengungkapan yang transparan, dan kebijakan yang mendukung. Dengan memahami sumber emisi dan menerapkan strategi mitigasi yang efektif, diharapkan Indonesia dapat mengurangi emisi GRK secara signifikan dan berkontribusi dalam upaya global mengatasi perubahan iklim.

### Sistem Informasi

Sistem informasi merupakan suatu entitas yang sangat penting dalam berbagai bidang kehidupan modern, terutama dalam organisasi dan pendidikan. Sebagai suatu sistem yang terdiri dari berbagai komponen yang saling terhubung, sistem informasi memiliki peran dalam mengumpulkan, menyimpan, mengolah, dan mendistribusikan informasi yang diperlukan untuk mendukung pengambilan keputusan. Dalam dunia pendidikan, Sistem Informasi Akademik (SIAKAD) adalah contoh penting dari penerapan sistem informasi yang memiliki peran utama dalam mengatur dan mengelola data terkait kegiatan akademik, seperti pendaftaran mahasiswa, pengelolaan nilai, dan penyampaian informasi kepada mahasiswa (Suryawan & Prihandoko, 2018). Evaluasi penerapan SIAKAD sangat diperlukan untuk memastikan bahwa sistem tersebut dapat memenuhi kebutuhan penggunanya dan memberikan kepuasan yang optimal (Suryawan & Prihandoko, 2018).

Lebih lanjut, sistem informasi juga digunakan di berbagai sektor lainnya, seperti kesehatan dan manajemen organisasi. Di rumah sakit, Sistem Informasi Manajemen Rumah Sakit (SIMRS) berfungsi untuk meningkatkan kualitas pelayanan kesehatan dengan mengelola data pasien dan laporan rumah sakit secara lebih efisien (Serhalawan, 2023; Supriyono, 2020). Evaluasi terhadap SIMRS juga penting untuk mengidentifikasi kendala yang mungkin terjadi pada pengguna maupun organisasi, serta untuk meningkatkan kinerja sistem (Serhalawan, 2023; Supriyono, 2020). Selain itu, dalam organisasi bisnis, sistem informasi manajemen memainkan peran penting dalam perencanaan, pengelolaan, dan pengawasan hasil kerja untuk mencapai tujuan organisasi dengan lebih efektif (Tri Putra, 2019).

Pengembangan sistem informasi melibatkan berbagai metode dan pendekatan, seperti metode waterfall dan Scrum, yang digunakan untuk merancang dan menerapkan sistem yang sesuai dengan kebutuhan pengguna (Sallaby & Kanedi, 2020; Syarif Hidayatullah, 2023). Proses ini mencakup analisis kebutuhan pengguna yang menjadi langkah krusial, di mana pemahaman yang mendalam tentang kebutuhan dan ekspektasi pengguna akan membantu dalam merancang sistem yang lebih efisien dan efektif (Syarif Hidayatullah, 2023). Selain itu, budaya organisasi juga berpengaruh pada keberhasilan sistem informasi, karena sistem yang dibangun harus mencerminkan nilai-nilai dan praktik yang ada dalam organisasi tersebut (Rahayu & Anggadini, 2014).

Di era digital saat ini, penerapan sistem informasi berbasis web semakin berkembang, memberikan kemudahan bagi pengguna dalam mengakses dan mengelola informasi. Sebagai contoh, sistem informasi yang dirancang untuk organisasi mahasiswa atau lembaga pendidikan dapat meningkatkan efisiensi dalam mengelola kegiatan dan keuangan (Sihotang & Yutanto, 2021; Wulandari, 2020). Dengan memanfaatkan teknologi informasi, organisasi dapat mengoptimalkan proses bisnis mereka, meningkatkan transparansi, serta mendukung pengambilan keputusan yang lebih baik (Widiarta, 2024). Oleh karena itu, sangat penting bagi setiap organisasi untuk terus melakukan evaluasi dan pengembangan sistem informasi agar dapat mengikuti perkembangan kebutuhan dan teknologi yang ada.

Secara keseluruhan, sistem informasi memegang peranan yang sangat vital dalam meningkatkan efisiensi dan efektivitas berbagai kegiatan dalam organisasi. Dengan pemahaman yang baik tentang komponen dan fungsi sistem informasi, serta penerapan metode yang tepat dalam pengembangannya, organisasi dapat mencapai tujuan mereka dengan lebih baik. Evaluasi yang berkelanjutan terhadap sistem informasi yang ada juga menjadi faktor kunci untuk memastikan bahwa sistem tersebut tetap relevan dan bermanfaat bagi penggunanya (Serhalawan, 2023; Supriyono, 2020; Suryawan & Prihandoko, 2018).

### *Real-Time Data Processing*

Proses data secara real-time telah menjadi area penting dalam ilmu komputer, terutama dengan meningkatnya permintaan untuk penanganan data yang cepat di berbagai aplikasi. Paradigma pemrosesan ini mengintegrasikan pengambilan data, pemrosesan, dan ekspor secara mulus, memungkinkan sistem untuk merespons secara cepat terhadap aliran data yang masuk (Maske & Prasad, 2015). Pentingnya pemrosesan data real-time ditekankan oleh penerapannya di berbagai bidang seperti komputasi awan, di mana hal ini mendukung pengelolaan data sensor secara efisien dan meningkatkan skalabilitas infrastruktur pemrosesan data (Kamburugamuve dkk., 2015). Selain itu, peralihan dari pemrosesan batch tradisional ke pemrosesan aliran data sangat berperan dalam pengelolaan data dengan volume tinggi secara real-time, yang penting untuk aplikasi yang membutuhkan wawasan dan tindakan segera (Ma & Yang, 2017; Shahrivari, 2014).

Dalam konteks big data, pemrosesan real-time menghadirkan tantangan unik, terutama terkait dengan kecepatan dan kebutuhan sistem yang mampu menganalisis data saat data tersebut dihasilkan (Hammoodi dkk., 2018; “Real-Time Data Stream Processing - Challenges and Perspectives,” 2017). Integrasi teknologi canggih seperti mesin pemrosesan peristiwa terdistribusi (DSPE) telah memfasilitasi pengembangan arsitektur yang skalabel untuk mengelola kompleksitas aliran data real-time (Kamburugamuve dkk., 2015). Selain itu, penerapan proses ETL (Extract, Transform, Load) otomatis sangat penting untuk memastikan bahwa perubahan data dapat tertangkap dan diproses hampir secara real-time, yang mendukung pengambilan keputusan yang tepat waktu dalam analitik bisnis (Biswas dkk., 2019, 2022). Kemampuan ini sangat relevan di lingkungan di mana data dari berbagai sumber, seperti media sosial dan jaringan sensor, harus digabungkan dan dianalisis dengan cepat untuk menghasilkan wawasan yang dapat ditindaklanjuti (Ounacer dkk., 2017).

Kemajuan terbaru dalam pembelajaran mesin dan komputasi aliran data semakin meningkatkan efisiensi sistem pemrosesan data real-time. Misalnya, teknik seperti pembelajaran mendalam telah digunakan untuk mensintesis dan menganalisis aliran data multimodal, yang meningkatkan akurasi dan kecepatan interpretasi data (Haleem dkk., 2023). Selain itu, pengembangan kerangka kerja yang memanfaatkan komputasi awan untuk analitik real-time telah merevolusi cara industri mengelola dan memanfaatkan data, terutama dalam aplikasi pengendalian proses dan pemantauan (Goldin dkk., 2017). Seiring dengan terus berkembangnya lanskap pembangkitan data, kebutuhan akan solusi pemrosesan real-time yang tangguh akan semakin mendesak, sehingga memerlukan penelitian dan inovasi yang berkelanjutan di bidang ini (Shi & Li, 2017; Xuan Phan dkk., 2011).

### Dasbor dan KPI

Dasbor dan KPI (Key Performance Indicators) merupakan elemen penting dalam manajemen kinerja organisasi yang berfungsi untuk memantau dan mengevaluasi efektivitas strategi yang diterapkan. Sebagai alat visualisasi data, dasbor memungkinkan manajer untuk melihat informasi kinerja secara langsung, yang sangat bermanfaat dalam pengambilan keputusan yang cepat dan akurat. Penelitian menunjukkan bahwa penggunaan dasbor dalam manajemen penjualan di perusahaan farmasi tidak hanya mendukung pemantauan kinerja tetapi juga membantu pengambilan keputusan yang lebih baik berdasarkan indikator kinerja yang relevan (“Developing a Sales Dashboard With Power BI – A Case Study in a Pharmaceutical Company,” 2024). Dasbor yang dirancang dengan baik dapat menampilkan metrik yang diperlukan untuk menilai kinerja organisasi, sehingga manajer dapat mengidentifikasi area yang membutuhkan perhatian lebih (Najem, 2023).

Di sisi lain, KPI adalah ukuran kuantitatif yang digunakan untuk menilai sejauh mana organisasi berhasil mencapai tujuan strategisnya. Dalam pengembangan produk baru, penggunaan dasbor pemasaran yang menampilkan KPI dapat mempercepat implementasi strategi pemasaran (Durmuşoğlu dkk., 2022). Penelitian juga menunjukkan bahwa penggunaan KPI yang tepat dapat meningkatkan daya saing perusahaan, terutama bagi UKM, dengan memberikan wawasan yang lebih mendalam tentang kinerja dan area yang perlu diperbaiki (Zaitsev, 2023). Oleh karena itu, penting bagi organisasi untuk mengembangkan KPI yang relevan dan terukur, serta memastikan bahwa data yang digunakan untuk menghitung KPI tersebut akurat dan dapat diandalkan (Rabiei & Almasi, 2022).

Desain dasbor yang efektif juga memerlukan perhatian terhadap aspek visualisasi data. Dasbor yang baik harus dapat menyajikan informasi dengan cara yang mudah dipahami, sehingga pengguna dapat dengan cepat menangkap informasi penting dan membuat keputusan berdasarkan data tersebut (Mohd Zainuddin dkk., 2022). Dalam hal ini, penelitian menunjukkan bahwa desain visual yang baik dapat meningkatkan pemahaman pengguna terhadap KPI dan metrik lainnya, yang pada gilirannya dapat meningkatkan kinerja organisasi secara keseluruhan (Pappas & Whitman, 2011). Selain itu, integrasi data dari berbagai sumber juga sangat penting untuk memberikan gambaran menyeluruh tentang kinerja organisasi, memungkinkan manajer untuk melakukan analisis yang lebih mendalam (Rachansa, 2024).

Dalam praktiknya, tantangan yang sering muncul dalam pengembangan dasbor dan KPI meliputi pengumpulan data yang tepat, integrasi sistem, dan penyajian informasi yang efektif (Rabiei & Almasi, 2022). Oleh karena itu, organisasi perlu mengembangkan strategi yang jelas untuk mengatasi tantangan-tantangan ini, termasuk pelatihan bagi staf mengenai penggunaan dasbor dan pemahaman KPI yang relevan (Wilbanks & Langford, 2014). Dengan demikian, dasbor dan KPI tidak hanya berfungsi sebagai alat pengukuran, tetapi juga sebagai instrumen strategis yang dapat membantu organisasi mencapai tujuan jangka panjangnya.

Secara keseluruhan, dasbor dan KPI memiliki peran yang sangat penting dalam manajemen kinerja organisasi. Dengan memanfaatkan teknologi dan metodologi yang tepat, organisasi dapat meningkatkan efektivitas pengambilan keputusan dan mencapai hasil yang lebih baik. Oleh karena itu, investasi dalam pengembangan dasbor dan KPI yang efektif harus menjadi prioritas bagi setiap organisasi yang ingin tetap kompetitif di pasar yang semakin kompleks.

### Perhitungan Emisi Karbon

Berdasarkan perhitungan oleh Penman dkk. (2006) pada IPCC Guideline tahun 2006, berikut adalah perumusan perhitungannya.

Di mana:

* *E*: Emisi karbon total (dalam ton CO2​ eq.)
* ​: Aktivitas (misalnya, konsumsi bahan bakar dalam TJ atau GJ)
* : Faktor emisi untuk jenis bahan bakar tertentu (dalam ton CO2​/TJ)
* *CF*: Faktor koreksi oksidasi atau efisiensi (biasanya antara 0.98 hingga 1.0, tergantung pada teknologi yang digunakan).

Di dalam penelitian oleh Tumara (2015) menyajikan pendekatan metodologis untuk menghitung emisi karbon dioksida (CO₂) yang merujuk pada pedoman Intergovernmental Panel on Climate Change (IPCC) tahun 2006. Fokus utama kajian ini adalah pada emisi yang dihasilkan dari aktivitas manusia, termasuk sektor energi, proses industri, perubahan penggunaan lahan, dan pengelolaan limbah.

### Internet of Things

Internet of Things (IoT) merupakan suatu paradigma yang mengubah cara kita berinteraksi dengan dunia fisik, di mana berbagai perangkat fisik yang dilengkapi dengan sensor dan konektivitas jaringan dapat berkomunikasi dan saling terhubung melalui Internet. Ekosistem yang terhubung ini mencakup beragam perangkat, termasuk sensor, kamera, mesin industri, serta benda-benda sehari-hari seperti ponsel pintar dan perangkat wearable. Perkembangan perangkat IoT sangat pesat, dengan perkiraan jumlah perangkat yang meningkat dari 7,74 miliar pada tahun 2019 menjadi sekitar 25,44 miliar pada tahun 2030 (Dawod dkk., 2022; Gubbi dkk., 2013). Pertumbuhan yang cepat ini menunjukkan peran penting IoT dalam masyarakat modern, yang memungkinkan pengumpulan dan analisis data dalam jumlah besar, yang dapat dimanfaatkan untuk menciptakan aplikasi inovatif di berbagai sektor seperti kesehatan, kota pintar, dan otomatisasi industri (Sattar dkk., 2017; Ullah dkk., 2020).

Arsitektur IoT ditandai dengan kemampuannya untuk mengintegrasikan perangkat dan sistem yang heterogen, memungkinkan komunikasi dan pertukaran data yang mulus. Integrasi ini sangat penting untuk pengembangan aplikasi yang membutuhkan pemrosesan data secara real-time dan responsivitas, terutama dalam skenario yang memerlukan waktu yang sangat sensitif (Korala dkk., 2021b, 2021a). Selain itu, ekosistem IoT didukung oleh teknologi canggih seperti komputasi awan dan komputasi tepi, yang menyediakan infrastruktur yang diperlukan untuk penyimpanan, pemrosesan, dan analitik data (Mavromatis dkk., 2019; Yousefpour dkk., 2019). Teknologi-teknologi ini tidak hanya meningkatkan efisiensi aplikasi IoT, tetapi juga mengatasi tantangan terkait dengan manajemen data dan kualitas layanan (Yousefpour dkk., 2019).

Selain itu, paradigma IoT terus berkembang dengan mengintegrasikan metodologi canggih seperti kecerdasan buatan (AI) dan pembelajaran mesin, yang meningkatkan kapabilitas sistem IoT dengan memungkinkan analitik prediktif dan pengambilan keputusan yang cerdas (Kumar dkk., 2019; Tzafestas, 2018). Sinergi antara IoT dan AI sangat terlihat dalam aplikasi-aplikasi yang terkait dengan kota pintar dan otomatisasi industri, di mana wawasan berbasis data dapat meningkatkan efisiensi operasional dan manajemen sumber daya (Kumar dkk., 2019; Peralta Abadia, 2023). Seiring dengan terus berkembangnya lanskap IoT, penting untuk mengatasi tantangan yang terkait, termasuk masalah keamanan dan interoperabilitas antar perangkat yang beragam, untuk dapat sepenuhnya mewujudkan potensi teknologi transformatif ini (Dawod, 2024; Razzaque dkk., 2016).

Kesimpulannya, Internet of Things bukan hanya sebuah kemajuan teknologi, melainkan juga perubahan fundamental dalam cara kita berinteraksi dengan dunia fisik. Dengan menghubungkan miliaran perangkat dan memungkinkan mereka untuk berkomunikasi secara mandiri, IoT diproyeksikan untuk merevolusi berbagai aspek kehidupan sehari-hari dan industri, menjadikannya area riset dan pengembangan yang sangat penting di masa depan.

### Protokol MQTT (Message Queuing Telemetry Transport)

Protokol Message Queuing Telemetry Transport (MQTT) telah muncul sebagai standar komunikasi yang sangat penting dalam perkembangan Internet of Things (IoT), terutama karena sifatnya yang ringan dan kemampuan pengiriman pesan yang efisien. MQTT beroperasi menggunakan model publish-subscribe, yang memfasilitasi komunikasi antara perangkat (klien) dan server pusat yang dikenal sebagai broker. Arsitektur ini memungkinkan pertukaran data yang dapat diskalakan dan fleksibel, menjadikannya sangat cocok untuk lingkungan dengan sumber daya terbatas yang umumnya ditemui pada aplikasi IoT (Azzedin & Alhazmi, 2023; Ihirwe dkk., 2021; Wardana dkk., 2019).

Desain MQTT dicirikan oleh kesederhanaan dan beban kerja yang rendah, yang sangat penting untuk perangkat dengan daya pemrosesan dan memori yang terbatas. Protokol ini sangat efektif dalam skenario yang membutuhkan transmisi data secara real-time, seperti dalam pemantauan kesehatan, pertanian pintar, dan otomatisasi industri (Al Hanif & Ilyas, 2024; Kurdi & Thayananthan, 2022; B. Mishra & Kertesz, 2020). Mekanisme publish-subscribe memungkinkan beberapa klien untuk menerima pesan tanpa perlu membuat koneksi langsung satu sama lain, sehingga mengurangi kemacetan jaringan dan meningkatkan efisiensi keseluruhan dalam penyebaran data (Khan dkk., 2021; Park dkk., 2018).

Meski memiliki berbagai keunggulan, MQTT juga tidak lepas dari tantangan. Model broker terpusat dapat menyebabkan bottleneck (kemacetan), terutama dalam penerapan skala besar di mana banyak perangkat berkomunikasi secara bersamaan (Sahlmann dkk., 2021; Spohn, 2022). Untuk mengatasi masalah skalabilitas ini, berbagai peningkatan telah diusulkan, termasuk pengembangan arsitektur broker terdistribusi yang dapat mengelola volume lalu lintas data yang tinggi dengan lebih baik (Longo dkk., 2020, 2022).

Selain itu, keamanan tetap menjadi perhatian penting, karena protokol ini diketahui memiliki kerentanan yang dapat dieksploitasi dalam lingkungan IoT (Chien dkk., 2020; Victor Pakpahan dkk., 2023). Para peneliti sedang aktif mengembangkan kerangka kerja untuk memperkuat keamanan komunikasi MQTT, memastikan integritas dan kerahasiaan data tetap terjaga (Azzedin & Alhazmi, 2023; Chien & Wang, 2022; Zhang dkk., 2022).

Kesimpulannya, MQTT menonjol sebagai protokol yang tangguh untuk komunikasi IoT, menawarkan perpaduan antara efisiensi dan skalabilitas. Namun, penelitian berkelanjutan sangat penting untuk mengatasi keterbatasannya, terutama terkait keamanan dan skalabilitas, untuk sepenuhnya memanfaatkan potensinya dalam berbagai aplikasi IoT.

### Unified Modelling Language (UML)

Unified Modeling Language (UML) berfungsi sebagai bahasa pemodelan terstandarisasi yang banyak diadopsi dalam rekayasa perangkat lunak, memfasilitasi visualisasi, spesifikasi, konstruksi, dan dokumentasi artefak sistem perangkat lunak. UML mencakup berbagai jenis diagram, termasuk diagram kelas, diagram sekuens, dan diagram use case, yang secara bersama-sama menyediakan kerangka kerja komprehensif untuk memodelkan aspek statis dan dinamis sistem (Kulkarni & Srinivasa, 2019; Yigitbas dkk., 2021). Keserbagunaan UML memungkinkannya digunakan dalam berbagai domain di luar rekayasa perangkat lunak tradisional, seperti rekayasa sistem dan mekatronika, di mana UML membantu dalam integrasi komponen mekanik, elektronik, dan perangkat lunak (Friedenthal & Burkhart, 2003; Valles-Barajas, 2011).

Lebih lanjut, peran UML sebagai "*lingua franca*" dalam pengembangan perangkat lunak digarisbawahi oleh kemampuannya dalam meningkatkan komunikasi antar pemangku kepentingan, termasuk pengembang, insinyur, dan otoritas sertifikasi (Yigitbas dkk., 2021; Zoughbi dkk., 2011). Hal ini terutama terlihat dalam domain yang kritis terhadap keselamatan, di mana profil UML seperti Safe-UML telah dikembangkan untuk memodelkan konsep terkait keselamatan sesuai dengan standar seperti RTCA DO-178B (Zoughbi dkk., 2011). Evolusi berkelanjutan UML, termasuk ekstensinya untuk aplikasi spesifik seperti SysML untuk rekayasa sistem, menunjukkan adaptabilitas dan relevansinya dalam menangani kompleksitas pengembangan perangkat lunak dan sistem modern (Hause & Thom, 2008).

Selanjutnya, integrasi UML dengan metode formal dan teknik verifikasi, seperti Petri nets, menyoroti kemampuannya untuk mendukung analisis dan validasi yang ketat terhadap persyaratan sistem (Gómez-Martínez dkk., 2018; Kerkouche dkk., 2010). Seiring rekayasa perangkat lunak yang semakin bergeser ke pendekatan berbasis model, UML tetap menjadi alat penting yang tidak hanya membantu dalam desain dan dokumentasi sistem tetapi juga meningkatkan kualitas dan keandalan produk perangkat lunak secara keseluruhan (Lange dkk., 2007).

Singkatnya, kerangka kerja UML yang komprehensif, kemampuan adaptasi di berbagai domain, dan integrasinya dengan metode verifikasi formal memposisikannya sebagai alat yang esensial dalam praktik rekayasa perangkat lunak dan sistem kontemporer.

### Use Case Diagram

Diagram Use Case (UCD) adalah komponen penting dalam bidang rekayasa perangkat lunak, khususnya dalam konteks analisis kebutuhan dan desain sistem. Diagram ini berfungsi sebagai representasi grafis dari kebutuhan fungsional sistem dari perspektif pengguna, yang mencakup interaksi antara pengguna ("aktor") dan sistem itu sendiri. Pentingnya UCD terletak pada kemampuannya untuk memberikan gambaran yang jelas dan ringkas tentang fungsionalitas sistem, yang memfasilitasi komunikasi antar pemangku kepentingan dan membantu dalam dokumentasi kebutuhan menggunakan Unified Modeling Language (UML) (Mustafa, 2010; Sabharwal dkk., 2015). Kualitas Diagram Use Case secara langsung mempengaruhi kualitas keseluruhan sistem perangkat lunak yang sedang dikembangkan, karena diagram ini membangun pemahaman dasar tentang kebutuhan pengguna dan kemampuan sistem (Mustafa, 2010; Sabharwal dkk., 2015).

Formalisasi UCD telah dieksplorasi melalui berbagai metodologi, seperti metode formal Event-B, yang meningkatkan ketelitian semantik dari diagram-diagram ini (Sousa dkk., 2017). Pendekatan formal ini tidak hanya membantu dalam memperjelas semantik UCD tetapi juga menyediakan mekanisme untuk mengidentifikasi inkonsistensi dalam diagram, sehingga meningkatkan keandalan kebutuhan yang ditangkap (Hazra & Dey, 2014a; Sousa dkk., 2017). Selain itu, integrasi UCD dengan diagram UML lainnya, seperti diagram aktivitas dan sekuens, memungkinkan pemodelan yang lebih komprehensif terhadap perilaku dan kebutuhan sistem (Khurana dkk., 2016; Y.-B. Yang dkk., 2019). Integrasi ini sangat penting untuk memastikan konsistensi di berbagai representasi sistem, yang esensial untuk pengembangan dan pemeliharaan perangkat lunak yang efektif (Hazra & Dey, 2014b; Ibrahim dkk., 2011).

Dalam konteks lini produk perangkat lunak, UCD juga memainkan peran vital dalam mengelola variabilitas dan menyempurnakan kebutuhan (Azevedo dkk., 2010). Eksplorasi hubungan seperti "extend" dalam UCD memungkinkan pengembang untuk mengakomodasi berbagai kebutuhan sambil mempertahankan model yang koheren. Kemampuan beradaptasi ini sangat penting dalam praktik pengembangan perangkat lunak modern, di mana kebutuhan pengguna dapat berkembang dengan cepat. Selanjutnya, penggunaan UCD bersamaan dengan pertimbangan keamanan telah ditekankan, menyoroti perlunya memasukkan kebutuhan non-fungsional ke dalam proses desain untuk memastikan aplikasi yang kuat dan aman (Gedam & Meshram, 2022, 2023).

Merangkap penjelasan-penjelasan sebelumnya, Diagram Use Case bukan hanya alat untuk memvisualisasikan kebutuhan; mereka merupakan bagian integral dari pendekatan sistematis dan ketat terhadap pengembangan perangkat lunak, memastikan bahwa aspek fungsional dan non-fungsional ditangani secara memadai.

### Sequence Diagram

Diagram Sekuens (SD) adalah komponen penting dari Unified Modeling Language (UML), yang berfungsi sebagai representasi visual dari interaksi antar objek dalam sistem dari waktu ke waktu. Diagram ini sangat berharga dalam rekayasa perangkat lunak untuk memodelkan perilaku dinamis dan memfasilitasi komunikasi antar pemangku kepentingan, termasuk pengembang dan klien. Fungsi utama diagram sekuens adalah untuk menggambarkan bagaimana objek berinteraksi melalui pesan, sehingga mengilustrasikan alur kontrol dan data dalam sistem (Ameedeen & Bordbar, 2020; Nanthaamornphong & Leatongkam, 2019; Ziadi dkk., 2011). Kemampuan ini membuat SD sangat diperlukan dalam berbagai fase pengembangan perangkat lunak, termasuk analisis kebutuhan, desain, dan pengujian (Pickin dkk., 2007; Ziadi dkk., 2011).

Formalisasi diagram sekuens telah menjadi topik penelitian yang ekstensif, dengan berbagai pendekatan yang diusulkan untuk meningkatkan kegunaannya. Misalnya, Küster-Filipe memperkenalkan semantik untuk diagram sekuens berdasarkan Labelled Event Structures, yang membantu dalam membuktikan kebenaran interaksi sistem (Ameedeen dkk., 2009, 2011). Selanjutnya, integrasi diagram sekuens dengan metodologi pengembangan berbasis model telah dieksplorasi untuk meningkatkan interoperabilitas dan memfasilitasi transformasi model menjadi kode yang dapat dieksekusi (Ameedeen dkk., 2011; Li, 2010). Transformasi ini penting untuk memastikan bahwa sistem yang dirancang mematuhi persyaratan yang ditentukan dan berperilaku sesuai yang dimaksud selama eksekusi.

Diagram sekuens tidak hanya berguna untuk memodelkan sistem baru tetapi juga memainkan peran penting dalam rekayasa balik perangkat lunak yang ada. Teknik untuk merekayasa balik diagram sekuens dari kode telah dikembangkan untuk membantu dalam pemahaman dan pemeliharaan program (Labiche dkk., 2013; Rountev dkk., 2005). Teknik-teknik ini memanfaatkan analisis statis dan dinamis untuk merekonstruksi interaksi yang digambarkan dalam diagram sekuens, sehingga memberikan wawasan tentang perilaku sistem dan membantu dalam identifikasi masalah potensial (Ghaleb dkk., 2018). Seiring sistem perangkat lunak yang semakin kompleks, kemampuan untuk memvisualisasikan interaksi melalui diagram sekuens menjadi semakin penting untuk pengembangan dan pemeliharaan perangkat lunak yang efektif.

Meringkas dari penjelasan yang telah disebutkan, diagram sekuens adalah alat vital dalam domain rekayasa perangkat lunak, memfasilitasi pemodelan interaksi dan perilaku dalam sistem. Formalisasi dan integrasi mereka ke dalam pendekatan berbasis model meningkatkan efektivitasnya, sementara aplikasi mereka dalam rekayasa balik mendukung upaya pemeliharaan dan pemahaman perangkat lunak yang berkelanjutan.

### Class Diagram

Diagram kelas adalah komponen mendasar dari Unified Modeling Language (UML), yang berfungsi sebagai *blueprint* untuk desain perangkat lunak berorientasi objek. Diagram ini memberikan representasi visual dari kelas-kelas dalam sistem dan hubungan di antara mereka, yang sangat penting untuk memahami struktur dan perilaku aplikasi perangkat lunak. Signifikansi diagram kelas melampaui sekadar representasi; mereka memainkan peran vital dalam meningkatkan kualitas dan kemampuan pemeliharaan perangkat lunak.

Misalnya, Mathur dan Kaushik menyoroti bahwa diagram kelas dapat secara efektif mengilustrasikan kompleksitas yang melekat dalam perangkat lunak berorientasi objek, sehingga membantu pengembang dalam meningkatkan kualitas perangkat lunak melalui praktik desain yang lebih baik (Mathur & Kaushik, 2016). Selanjutnya, Fauzan et al. menekankan pentingnya diagram kelas dalam konteks pendidikan, di mana diagram ini umumnya digunakan untuk menilai pemahaman siswa tentang prinsip-prinsip desain perangkat lunak (Fauzan dkk., 2021).

Konstruksi sistematis diagram kelas sangat penting dalam berbagai domain, termasuk pengembangan web, di mana mereka diterima secara luas sebagai cara untuk merepresentasikan model data konseptual (Cuaresma & Cavarero, 2005). Penerimaan ini menggarisbawahi keserbagunaan dan penerapannya di berbagai metodologi pengembangan perangkat lunak.

Lebih lanjut, peran diagram kelas dalam desain basis data tidak bisa diremehkan. Saringat dkk. membahas bagaimana diagram kelas dapat diterjemahkan menjadi tabel basis data relasional, dengan mempertimbangkan berbagai perilaku kelas seperti generalisasi dan asosiasi, yang penting untuk desain skema basis data yang efektif (Saringat dkk., 2010). Proses penerjemahan ini sangat penting untuk memastikan bahwa basis data secara akurat mencerminkan struktur sistem perangkat lunak yang dimaksud.

Selain representasi strukturalnya, diagram kelas juga memfasilitasi penalaran tentang sistem perangkat lunak. Berardi dkk. mengeksplorasi bagaimana sifat-sifat formal diagram kelas UML dapat dideteksi secara otomatis, yang membantu dalam mengidentifikasi inkonsistensi dan redundansi dalam desain (Berardi dkk., 2005). Kemampuan ini sangat berharga dalam sistem kompleks di mana menjaga konsistensi sangat penting.

Selain itu, integrasi diagram kelas dengan teknik pemodelan lainnya, seperti pemetaan objek, meningkatkan proses pengembangan perangkat lunak secara keseluruhan dengan memastikan bahwa transisi dari desain ke implementasi berjalan mulus dan koheren (Abdulmonim dkk., 2019). Dengan demikian, diagram kelas tidak hanya berfungsi sebagai elemen dasar dalam desain perangkat lunak tetapi juga berkontribusi secara signifikan terhadap kualitas dan kemampuan pemeliharaan sistem perangkat lunak secara keseluruhan.

### Activity Diagram

Diagram aktivitas merupakan komponen penting dalam Unified Modeling Language (UML), yang berfungsi sebagai representasi visual dari alur kerja dan proses dalam suatu sistem. Diagram ini sangat berguna untuk memodelkan aspek dinamis dari sistem, menggambarkan aliran kontrol dan data antar berbagai aktivitas. Seperti yang diungkapkan oleh Grönninger dkk., diagram aktivitas dapat diperkuat dengan titik variasi semantik, yang memungkinkan representasi yang lebih mendalam terhadap proses yang kompleks tanpa membuat notasi menjadi terlalu rumit (Grönninger dkk., 2010). Fleksibilitas ini sangat penting agar diagram tetap mudah dipahami namun tetap dapat menggambarkan detail perilaku sistem yang diperlukan.

Selain itu, penerapan diagram aktivitas UML tidak hanya terbatas pada representasi semata; diagram ini juga penting dalam rekayasa kebutuhan dan manajemen alur kerja. Gross dan Doerr menyoroti keefektifan diagram aktivitas UML dibandingkan dengan notasi lainnya, seperti Event-driven Process Chains (EPC), dalam konteks rekayasa kebutuhan, yang menunjukkan kemampuannya dalam menangkap kebutuhan sistem dan proses (Gross & Doerr, 2009). Selanjutnya, Bendoukha dkk. menekankan pentingnya penyempurnaan diagram aktivitas UML untuk pemetaan ke dalam spesifikasi Business Process Execution Language (BPEL), yang mendukung komposisi alur kerja berbasis layanan. Hal ini menunjukkan kemampuan diagram aktivitas untuk beradaptasi dalam berbagai konteks, khususnya dalam arsitektur berbasis layanan (Bendoukha dkk., 2012).

Signifikansi diagram aktivitas UML juga ditekankan dalam pengujian dan verifikasi perangkat lunak. Sebagai contoh, Ahmad dkk. membahas peran diagram aktivitas dalam pengujian berbasis model, di mana diagram ini menjadi dasar bagi pendekatan pengujian yang sistematis (Ahmad dkk., 2019). Selain itu, Lima dan Tavares membahas tantangan dalam memverifikasi deadlock dan non-determinisme pada sistem konkuren menggunakan diagram aktivitas, yang menunjukkan pentingnya diagram ini dalam memastikan perilaku sistem yang dapat diandalkan (Lima & Tavares, 2019). Aspek ini sangat penting dalam pengembangan sistem perangkat lunak yang kokoh, di mana pemodelan aktivitas yang akurat dapat mencegah masalah potensial saat runtime.

Sehingga dapat disimpulkan bahwa diagram aktivitas UML adalah alat yang serbaguna dalam pemodelan sistem, menawarkan pendekatan terstruktur untuk memvisualisasikan alur kerja, memperkuat pengumpulan kebutuhan, dan memfasilitasi proses pengujian dan verifikasi. Kemampuannya untuk beradaptasi dengan berbagai konteks dan kebutuhan menjadikannya bagian yang tak terpisahkan dari toolkit pemodelan dalam rekayasa perangkat lunak.

### REST API

REST (Representational State Transfer) adalah gaya arsitektur yang banyak digunakan untuk merancang aplikasi jaringan, terutama API web. Gaya ini memanfaatkan protokol HTTP standar untuk memfasilitasi komunikasi antara klien dan server, dengan mengutamakan interaksi tanpa status dan arsitektur berbasis sumber daya. Kesederhanaan dan kemampuan skalabilitas dari API RESTful menjadikannya pilihan utama di berbagai bidang, termasuk komputasi awan dan arsitektur mikroservis (Brabra dkk., 2019). API RESTful menggunakan pengenal sumber daya seragam (URI) dan metode HTTP standar (GET, POST, PUT, DELETE) untuk melakukan operasi pada sumber daya, yang meningkatkan interoperabilitas dan kemudahan penggunaannya (Corradini dkk., 2022; Pautasso, 2013).

Prinsip desain REST menekankan ketidakberadaan status, di mana setiap permintaan dari klien memuat seluruh informasi yang diperlukan untuk memproses permintaan tersebut, sehingga mengurangi beban pada server dan meningkatkan kinerja (Palma dkk., 2014; Pautasso, 2013). Status *statelessness* ini sangat penting untuk skalabilitas, terutama dalam lingkungan komputasi awan di mana sumber daya dialokasikan secara dinamis (Palma dkk., 2014; Pautasso, 2013). Selain itu, API RESTful sering kali didokumentasikan menggunakan spesifikasi seperti OpenAPI, yang membantu dalam menghasilkan pustaka klien dan dokumentasi API secara otomatis, mempermudah proses pengembangan (Malakhov dkk., 2018; J. Yang dkk., 2018).

Namun, meskipun memiliki berbagai keuntungan, API RESTful menghadapi tantangan, khususnya dalam pengujian dan keamanan. Kompleksitas interaksi dan kebutuhan akan metodologi pengujian yang kuat telah mendorong pengembangan berbagai alat pengujian otomatis yang fokus pada strategi pengujian *black box* (Corradini dkk., 2022; Ehsan dkk., 2022). Alat-alat ini bertujuan untuk mengidentifikasi kerentanannya dan memastikan keandalan layanan RESTful, yang penting untuk menjaga integritas aplikasi yang bergantung pada API ini (Godefroid dkk., 2020; Sanjana dkk., 2022). Selain itu, integrasi mekanisme otentikasi sangat penting, karena banyak API RESTful memerlukan protokol akses yang aman untuk melindungi data sensitif (Verborgh dkk., 2012).

Kesimpulannya, API RESTful merupakan kemajuan signifikan dalam desain layanan web, menyediakan sarana komunikasi yang fleksibel dan efisien antar sistem terdistribusi. Prinsip-prinsipnya, seperti ketidakberadaan status, identifikasi sumber daya, dan standarisasi melalui HTTP, telah menjadikannya sebagai dasar arsitektur web modern, meskipun tantangan dalam pengujian dan keamanan masih perlu diatasi.

### Software Development Lifecycle

Siklus Pengembangan Perangkat Lunak (SDLC) adalah pendekatan terstruktur yang mencakup semua fase dalam pengembangan perangkat lunak, mulai dari pengumpulan kebutuhan awal hingga penerapan dan pemeliharaan. Setiap fase dalam SDLC sangat penting karena masing-masing fase memiliki tugas dan tanggung jawab spesifik yang berkontribusi pada kualitas dan keberhasilan produk perangkat lunak secara keseluruhan. Fase-fase tersebut biasanya meliputi analisis kebutuhan, desain, implementasi, pengujian, penerapan, dan pemeliharaan, yang masing-masing memerlukan keterampilan dan metodologi yang berbeda (Klünder dkk., 2021; Ruparelia, 2010). Misalnya, fase analisis kebutuhan sangat krusial untuk memahami kebutuhan pengguna, dan mengabaikannya dapat menyebabkan biaya yang lebih tinggi dan ketidakpuasan pelanggan di kemudian hari (Shah dkk., 2018; *Software Practitioners Challenges in the Requirement Engineering Phase of Software Development*, 2019).

Selain itu, penerapan praktik desain berpusat pada pengguna (UCD) dalam SDLC terbukti dapat meningkatkan kegunaan dan efektivitas produk perangkat lunak. Namun, UCD sering kali dianggap sebagai kegiatan terpisah daripada diintegrasikan sepenuhnya dalam setiap fase, yang dapat menyebabkan tantangan dalam implementasinya (Achimugu dkk., 2010; Switzky, 2012). Penerapan metodologi agile juga telah mengubah model SDLC tradisional, dengan menekankan pengembangan iteratif dan umpan balik terus-menerus, yang dapat meningkatkan responsivitas terhadap perubahan kebutuhan (Gokarna, 2021; Rashid dkk., 2021).

Pertimbangan keamanan semakin diakui sebagai aspek penting dalam seluruh SDLC. Secara historis, keamanan sering kali dipandang sebagai hal yang dilakukan di tahap akhir, seperti pengujian atau pemeliharaan. Namun, praktik kontemporer menganjurkan agar langkah-langkah keamanan dimulai sejak awal, memastikan bahwa kerentanannya dapat diidentifikasi dan diminimalkan selama proses pengembangan (Alenezi dkk., 2022; Byers & Shahmehri, 2007). Pendekatan holistik ini tidak hanya meningkatkan posisi keamanan perangkat lunak, tetapi juga selaras dengan praktik terbaik dalam rekayasa perangkat lunak, yang pada gilirannya meningkatkan hasil proyek secara keseluruhan (Aljohani & Jameel Qureshi, 2016; Gupta dkk., 2014).

Singkatnya, SDLC adalah kerangka kerja komprehensif yang memandu pengembangan perangkat lunak melalui fase-fase yang jelas, masing-masing dengan tantangan dan praktik terbaiknya sendiri. Integrasi metodologi seperti UCD dan langkah-langkah keamanan selama siklus pengembangan sangat penting untuk menghasilkan perangkat lunak berkualitas tinggi yang memenuhi harapan pengguna dan tahan terhadap ancaman potensial.

### Software Development Method

Metode pengembangan perangkat lunak merupakan kerangka penting yang memandu perencanaan, pelaksanaan, dan pengelolaan proyek perangkat lunak. Beberapa metodologi yang paling dikenal antara lain model Waterfall, Agile, Iterative Incremental, Prototype, dan V-Model. Masing-masing metodologi ini memiliki karakteristik, kelebihan, dan tantangan yang berbeda, yang dapat memengaruhi keberhasilan proyek pengembangan perangkat lunak secara signifikan.

Model Waterfall adalah salah satu metodologi pengembangan perangkat lunak yang paling awal dan tradisional. Model ini ditandai dengan pendekatan linier dan berurutan, di mana setiap tahap harus diselesaikan sebelum tahap berikutnya dimulai. Model ini menekankan dokumentasi yang mendalam dan struktur yang jelas, sehingga cocok untuk proyek dengan kebutuhan yang sudah terdefinisi dengan baik. Namun, kekakuannya dapat menimbulkan tantangan ketika terjadi perubahan persyaratan, karena model ini tidak memiliki fleksibilitas untuk mengakomodasi perubahan tersebut tanpa adanya pekerjaan ulang yang signifikan (Licorish dkk., 2016). Model Waterfall sering dikritik karena ketidakmampuannya untuk beradaptasi dengan kebutuhan proyek yang berkembang, yang mengakibatkan banyaknya keterlambatan dan kegagalan proyek (Licorish dkk., 2016).

Sebaliknya, metodologi Agile muncul sebagai respons terhadap keterbatasan metode tradisional seperti Waterfall. Agile menekankan pengembangan secara iteratif, di mana persyaratan dan solusi berkembang melalui kolaborasi antara tim lintas fungsi yang terorganisir sendiri. Pendekatan ini memungkinkan fleksibilitas yang lebih besar dan responsivitas terhadap perubahan, menjadikannya sangat efektif dalam lingkungan yang diharapkan mengalami perubahan persyaratan (D. Mishra & Mishra, 2011; Nyembe, 2023). Metodologi Agile, seperti Scrum dan Extreme Programming (XP), mempromosikan umpan balik yang berkelanjutan dan siklus iteratif, memungkinkan tim untuk menyampaikan perangkat lunak yang fungsional secara bertahap (Rizky & Sugiarti, 2022). Pendekatan Agile telah diterima secara luas karena kemampuannya menghasilkan perangkat lunak berkualitas tinggi yang lebih efektif memenuhi kebutuhan pengguna dibandingkan dengan metode tradisional (D. Mishra & Mishra, 2011; Nyembe, 2023).

Model Iterative Incremental menggabungkan elemen-elemen dari pengembangan iteratif dan incremental. Pendekatan ini memungkinkan penyempurnaan perangkat lunak secara bertahap melalui siklus berulang (iterasi) dan penambahan fitur baru (inkremen) dari waktu ke waktu. Setiap iterasi menghasilkan versi perangkat lunak yang dapat diuji dan dievaluasi sebelum melanjutkan ke siklus berikutnya. Metode ini memberikan peluang untuk deteksi masalah lebih awal dan umpan balik dari pengguna, yang dapat meningkatkan kualitas perangkat lunak (Alam dkk., 2022; Diansyah, 2023). Model Iterative Incremental sangat bermanfaat dalam proyek di mana persyaratan tidak sepenuhnya dipahami sejak awal, karena memungkinkan penyesuaian berdasarkan input pengguna dan hasil pengujian (Alam dkk., 2022).

Prototyping adalah metode pengembangan perangkat lunak lain yang berharga yang fokus pada pembuatan model perangkat lunak yang dapat berfungsi sejak awal proses pengembangan. Model ini, atau prototipe, berfungsi sebagai representasi nyata dari perangkat lunak, memungkinkan pemangku kepentingan untuk memvisualisasikan dan berinteraksi dengan sistem sebelum implementasi final. Prototyping dapat membantu memperjelas persyaratan dan mengidentifikasi potensi masalah lebih awal, mengurangi risiko perubahan mahal di kemudian hari dalam siklus pengembangan (Licorish dkk., 2016). Namun, hal ini juga bisa menyebabkan scope creep jika pemangku kepentingan terlalu terikat pada prototipe, sehingga meminta fitur tambahan yang tidak termasuk dalam rencana awal (Licorish dkk., 2016).

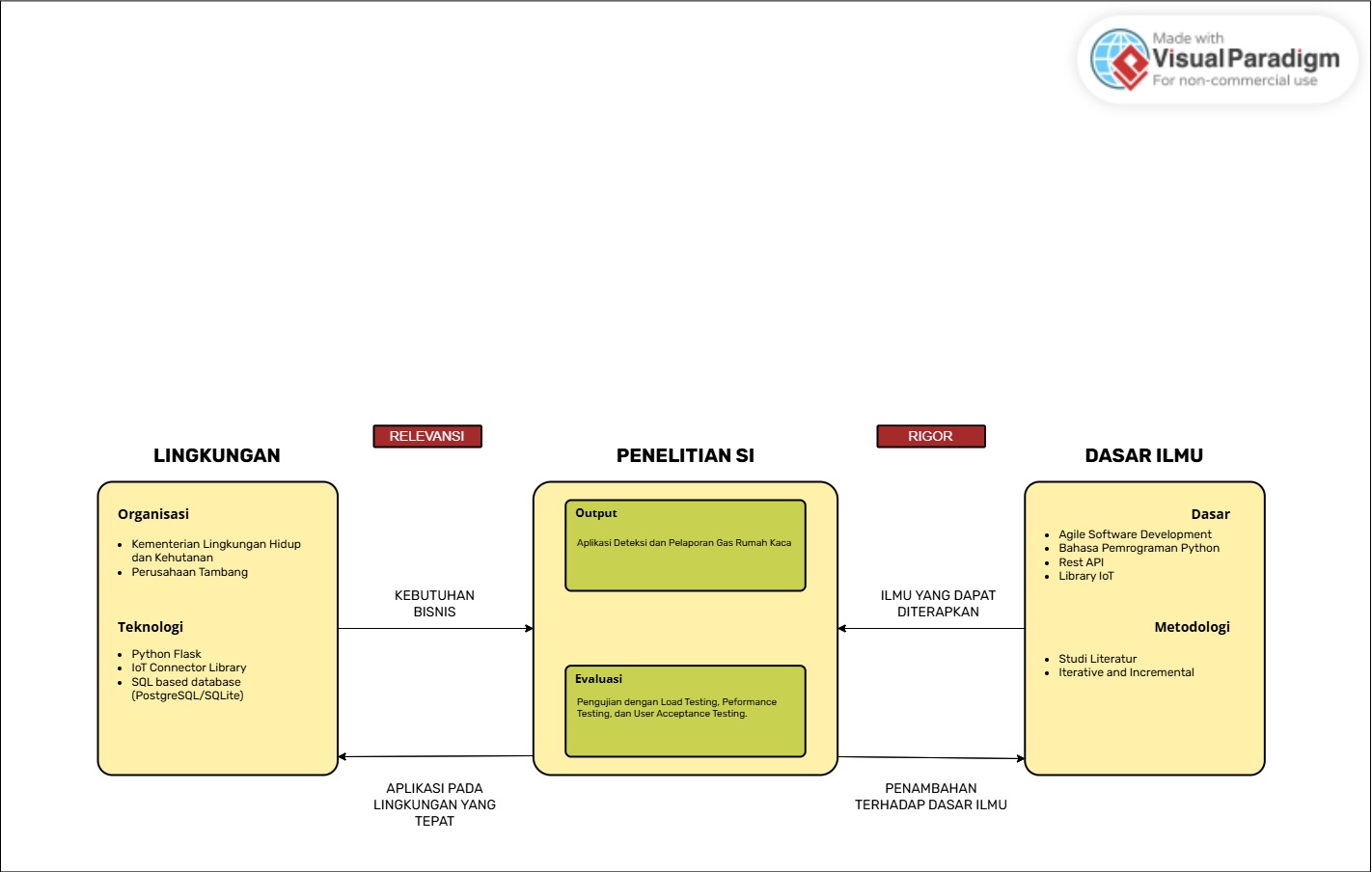
V-Model, atau model Verifikasi dan Validasi, adalah pengembangan dari model Waterfall yang menekankan pentingnya pengujian di setiap tahap pengembangan. Dalam model ini, setiap tahap pengembangan dihubungkan dengan tahap pengujian yang sesuai, memastikan bahwa verifikasi dan validasi berlangsung bersamaan dengan aktivitas pengembangan. Pendekatan ini membantu mengidentifikasi cacat lebih awal dan memastikan bahwa produk akhir memenuhi persyaratan pengguna dan standar kualitas (Alam dkk., 2022). Namun, seperti model Waterfall, V-Model bisa kurang fleksibel dalam menghadapi perubahan persyaratan, menjadikannya kurang cocok untuk proyek dengan kebutuhan yang dinamis (Alam dkk., 2022; Licorish dkk., 2016).

Sehingga maka dari itu, disimpulkan bahwa pemilihan metode pengembangan perangkat lunak sangat memengaruhi keberhasilan suatu proyek. Meskipun metodologi tradisional seperti Waterfall dan V-Model menyediakan struktur dan dokumentasi yang jelas, keduanya bisa kesulitan dengan fleksibilitas dan kemampuan beradaptasi. Di sisi lain, pendekatan Agile, Iterative Incremental, dan Prototyping menawarkan responsivitas yang lebih besar terhadap perubahan dan umpan balik pengguna, menjadikannya lebih cocok untuk lingkungan pengembangan perangkat lunak modern. Pemilihan metode yang tepat harus mempertimbangkan kebutuhan spesifik proyek, dinamika tim, dan konteks organisasi untuk mengoptimalkan hasil dan meningkatkan kualitas perangkat lunak.

# Metodologi Penelitian

## Kerangka Berpikir

Peneliti mengadopsi kerangka *design science research* (Hevner dkk, 2004) sebagai paradigma/kerangka pemecahan masalah. Tahapan ini menjelaskan analisa dan alasan pemilihan metodologi/metode/ kerangka kerja yang akan digunakan pada penelitian ini.



Gambar . Model Konseptual

Pemilihan metodologi penelitian yang tepat sangat penting dalam pengembangan aplikasi deteksi dan pelaporan gas rumah kaca. Studi literatur dan pendekatan iterative and incremental yang dipilih memiliki relevansi yang kuat dengan karakteristik penelitian sistem informasi ini, terutama dalam menghubungkan kebutuhan praktis dan aspek akademis.

Studi literatur memberikan dasar yang kokoh bagi penelitian, memungkinkan pemahaman yang lebih mendalam tentang konsep gas rumah kaca, teknologi IoT, dan pengembangan aplikasi. Metodologi ini membantu peneliti dalam mengidentifikasi praktik terbaik, standar industri, serta pembelajaran dari implementasi serupa yang telah dilakukan sebelumnya, yang sangat penting mengingat kompleksitas integrasi teknologi seperti Python Flask, IoT Connector Library, dan sistem database dalam pemantauan lingkungan.

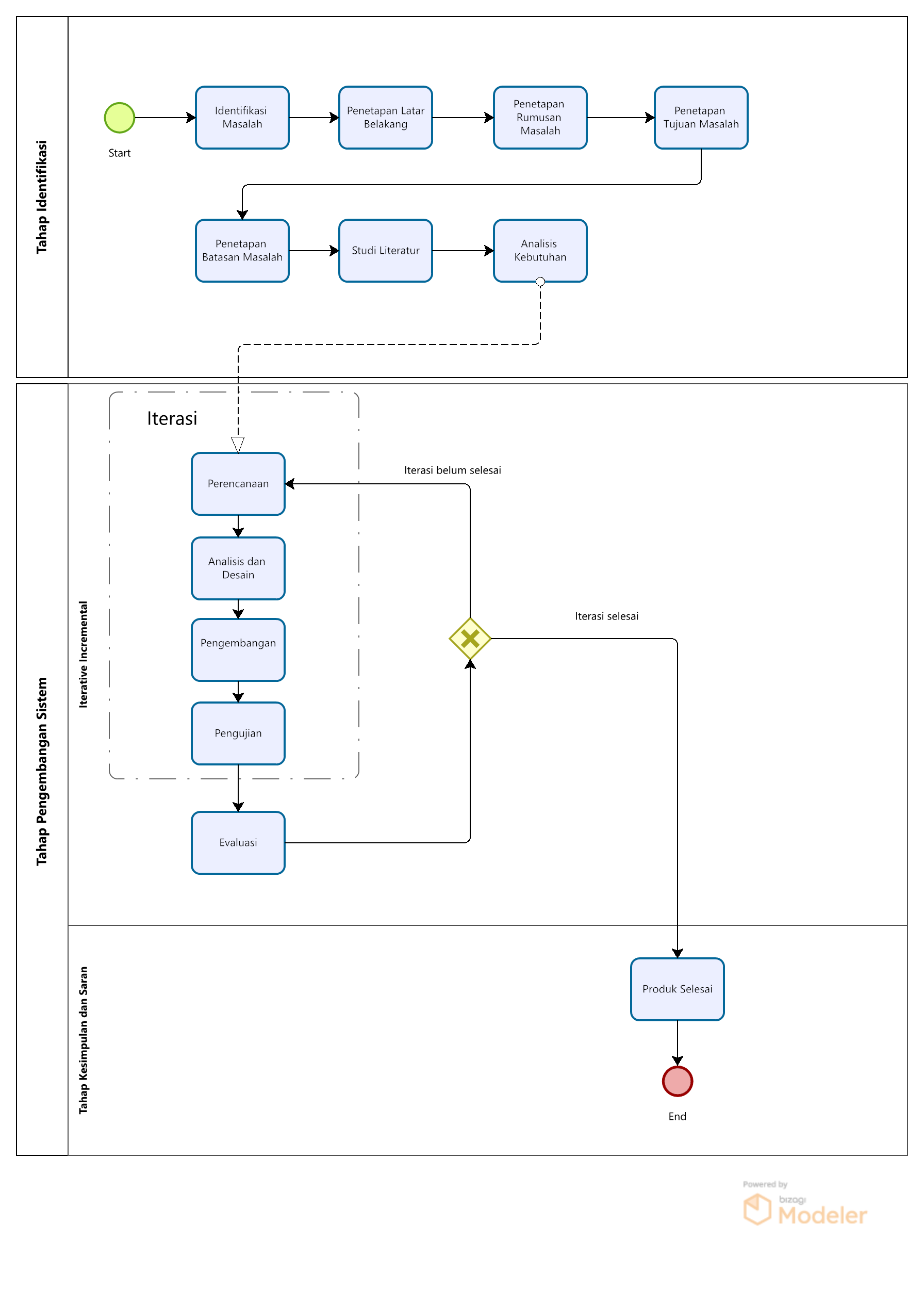
Di sisi lain, pendekatan iterative and incremental sangat sesuai dengan pengembangan sistem yang membutuhkan adaptasi yang terus-menerus. Metodologi ini memungkinkan pengembangan bertahap yang selaras dengan prinsip Agile Software Development, memfasilitasi umpan balik yang berkelanjutan antara tim pengembang dan pemangku kepentingan, seperti Kementerian Lingkungan Hidup dan perusahaan tambang. Pendekatan ini juga mendukung evaluasi menyeluruh melalui berbagai jenis pengujian, mulai dari load testing hingga user acceptance testing.

Kombinasi kedua metodologi ini menghasilkan kerangka kerja yang kuat untuk memastikan bahwa hasil penelitian tidak hanya memenuhi kebutuhan praktis organisasi, tetapi juga berkontribusi pada pengembangan ilmu pengetahuan di bidang sistem informasi. Hal ini terlihat dari bagaimana metodologi yang dipilih memungkinkan penelitian untuk memenuhi aspek relevansi dalam konteks aplikasi praktis, sambil mempertahankan ketelitian akademis melalui pendekatan yang terstruktur dan sistematis.

Dengan demikian, pemilihan metodologi ini mendukung pencapaian tujuan ganda: menghasilkan solusi teknologi yang dapat diimplementasikan secara efektif untuk pemantauan gas rumah kaca, serta memberikan kontribusi penting pada pengembangan ilmu pengetahuan dalam bidang sistem informasi. Pendekatan ini memastikan penelitian menghasilkan artefak teknologi yang berguna sekaligus pemahaman dan wawasan yang dapat diterapkan untuk pengembangan sistem serupa di masa depan.

## Sistematika Penyelesaian Masalah

Penyelesaian masalah secara sistematis adalah pendekatan terstruktur yang melibatkan beberapa tahap. Proses ini dimulai dengan identifikasi untuk mendalami pemahaman tentang masalah, dilanjutkan dengan pengembangan aplikasi menggunakan pendekatan iteratif, dan diakhiri dengan evaluasi guna memastikan penelitian serta aplikasi telah memenuhi kebutuhan yang ditentukan. Untuk penjelasan lebih lanjut, bisa dilihat untuk gambar dibawah.



Gambar . Sistematika Penyelesaian Masalah

### ****Identifikasi****

Tahap ini adalah langkah awal dalam Sistematika Penyelesaian Masalah yang bertujuan untuk mengidentifikasi permasalahan pada objek penelitian berdasarkan fakta yang tersedia, yaitu pada perusahaan tambang. Proses identifikasi ini berfokus pada pengamatan terhadap kadar emisi GRK yang dihasilkan selama aktivitas operasional.

#### Penetapan Latar Belakang

Tahap Penetapan Latar Belakang bertujuan untuk menyajikan dasar informasi yang mendukung penelitian. Informasi ini diperoleh dari hasil Identifikasi sebelumnya, yang kemudian dilengkapi dengan data tambahan dari berbagai sumber, seperti portal berita, laporan perusahaan, dan referensi lainnya. Dengan latar belakang yang kaya dan terperinci, alasan pelaksanaan penelitian menjadi lebih jelas dan terarah.

#### Penetapan Rumusan Masalah

Tahap Penetapan Rumusan Masalah bertujuan untuk merumuskan pertanyaan-pertanyaan yang spesifik dan terarah terkait penelitian yang akan dilakukan. Pada tahap ini, peneliti merancang sejumlah pertanyaan, antara lain:

1. Bagaimana cara merancang serta membangun sistem informasi guna pelaporan emisi gas rumah kaca dalam lingkup Scope 1 dengan menggunakan teknologi Internet of Things( IoT)?
2. Bagaimana cara mempersiapkan dashboard serta mengotomatisasi pelaporan periodik secara terjadwal?
3. Bagaimana auditor dari pemerintah serta internal industri bisa memantau emisi gas rumah kaca secara otomatis dari entitas yang diawasi?

#### Penetapan Tujuan Masalah

Tahap ini dilakukan untuk menjawab pertanyaan dari Rumusan Masalah dengan memperjelas informasi alur pengembangan.

#### Studi Literatur

Mengkaji referensi ilmiah terkait teknologi IoT dan metode Machine Learning dalam pelaporan emisi GRK.

#### Analisis Kebutuhan

Mengidentifikasi kebutuhan sistem, baik dari sisi fungsional (pengumpulan data, visualisasi) maupun non-fungsional (kecepatan akses, akurasi).

### Tahap Pengembangan Sistem

Tahap ini menggunakan pendekatan Iterative Incremental untuk memastikan setiap iterasi menghasilkan peningkatan kualitas sistem. Langkah-langkahnya meliputi:

#### Perencanaan

Menentukan rencana kerja untuk iterasi dimulai dengan merinci setiap tahapan yang perlu dilakukan dalam setiap siklus, termasuk pengumpulan data, pengembangan fitur, dan pengujian. Jadwal ditetapkan untuk memastikan setiap siklus selesai tepat waktu, dengan mempertimbangkan waktu yang dibutuhkan untuk setiap aktivitas, serta potensi hambatan yang mungkin muncul. Alokasi sumber daya juga dipertimbangkan dengan cermat, melibatkan pembagian tugas antar tim, pengaturan anggaran, dan penggunaan perangkat keras serta perangkat lunak yang diperlukan untuk mendukung kelancaran proses iterasi.

#### Analisis dan Desain

Merancang sistem dimulai dengan menyusun arsitektur yang mencakup struktur dasar aplikasi, komunikasi antara komponen, dan pengelolaan data yang efisien. Selanjutnya, flow diagram dibuat untuk menggambarkan alur proses dari setiap fitur dan interaksi pengguna dalam sistem, memastikan semua tahapan berjalan sesuai dengan rencana. Tampilan antarmuka dirancang dengan fokus pada pengalaman pengguna yang intuitif dan mudah digunakan, menggunakan prinsip desain yang responsif dan fungsional agar aplikasi dapat diakses dengan nyaman di berbagai perangkat.

#### Pengembangan

Melaksanakan proses pemrograman dimulai dengan mengimplementasikan kode berdasarkan desain yang telah disusun, mengikuti struktur arsitektur dan alur yang telah ditentukan. Setiap fitur dikembangkan secara bertahap, mengacu pada spesifikasi yang telah dirancang sebelumnya, untuk memastikan konsistensi dan integrasi yang baik antara komponen sistem. Pengujian dan debugging dilakukan secara berkala selama proses pemrograman untuk mendeteksi dan memperbaiki potensi kesalahan sebelum aplikasi diuji secara menyeluruh.

#### Pengujian

Melaksanakan proses coding dimulai dengan mengimplementasikan kode berdasarkan desain yang telah disusun, mengikuti alur dan struktur sistem yang telah direncanakan. Pengembangan fitur dilakukan dengan mengacu pada spesifikasi teknis yang jelas, memastikan setiap bagian sistem bekerja sesuai dengan tujuannya. Selama proses coding, pengujian unit dilakukan untuk memastikan fungsionalitas setiap komponen dan mengidentifikasi serta memperbaiki kesalahan sebelum aplikasi diuji secara keseluruhan.

#### Evaluasi

Meninjau hasil iterasi dilakukan dengan mengevaluasi pencapaian setiap fitur dan fungsi yang telah dikembangkan, serta mengidentifikasi area yang memerlukan perbaikan atau penyempurnaan. Kekurangan yang ditemukan selama tinjauan akan diperbaiki dengan melakukan perubahan pada kode atau desain untuk meningkatkan kinerja dan fungsionalitas aplikasi. Setelah perbaikan selesai, persiapan iterasi berikutnya dilakukan dengan merencanakan langkah-langkah selanjutnya, memastikan semua sumber daya dan jadwal disesuaikan untuk menjaga kelancaran proses pengembangan.

### Tahap Kesimpulan dan Saran

Tahap terakhir adalah menyimpulkan hasil penelitian dan memberikan rekomendasi untuk pengembangan lebih lanjut.

## Alasan Pemilihan Metode

Metode Iterative Incremental dipilih oleh peneliti untuk pengembangan aplikasi ini karena keunggulannya yang sangat fleksibel. Metode ini memungkinkan sistem untuk mengalami perubahan di setiap tahap pengembangan, sehingga proses adaptasi dapat dilakukan dengan mudah. Selain itu, pendekatan ini dianggap efisien dalam memanfaatkan tenaga kerja, karena mampu merespons masukan pengguna secara cepat dan efektif. Hal ini memungkinkan tim pengembang untuk segera mengimplementasikan pembaruan atau perbaikan fitur berdasarkan umpan balik yang diterima selama pengembangan berlangsung.

Dalam kerangka System Development Life Cycle (SDLC), metode Iterative Incremental memberikan kesempatan bagi pengguna untuk ikut terlibat dalam pemantauan proses pengembangan aplikasi. Partisipasi aktif ini berkontribusi pada peningkatan kualitas di setiap tahapan. Dengan kemampuannya untuk beradaptasi dan menanggapi perubahan selama proses pengembangan, metode ini tidak hanya memberikan kepuasan kepada pengguna, tetapi juga memastikan bahwa hasil akhir sesuai dengan standar yang telah ditetapkan.

Peneliti telah melakukan perbandingan antara metode Iterative Incremental yang dipilih dengan sejumlah metode pengembangan lainnya yang dirangkum dalam Tabel III.1. Perbandingan ini dilakukan untuk mengidentifikasi keunggulan utama Iterative Incremental dibandingkan metode lain, seperti Agile, Waterfall, Prototype, dan V-Model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Faktor** | **Iteratif Inkremental** | **Agile** | **Waterfall** | **Prototype** | **V-Model** |
| Pendekatan | Sistem dikembangkan secara bertahap (inkremental) dengan evaluasi di setiap iterasi. | Pengembangan iteratif cepat dengan perubahan prioritas berbasis umpan balik pengguna. | Proses linier dengan tahap-tahap berurutan, dimulai dari perencanaan hingga pemeliharaan. | Sistem dibuat dalam bentuk prototipe untuk diuji dan dikembangkan berdasarkan umpan balik pengguna. | Versi Waterfall dengan penekanan kuat pada pengujian di setiap tahap. |
| Kelebihan | - Risiko dapat dikurangi lebih awal.  - Iterasi memungkinkan evaluasi dan perbaikan terus-menerus. | - Sangat adaptif terhadap perubahan. - Proses pengembangan lebih cepat dengan iterasi pendek. | - Dokumentasi lengkap. - Struktur pengembangan jelas dan terorganisir. | - Memungkinkan umpan balik awal dari pengguna. - Cocok untuk eksplorasi ide baru. | - Pengujian terintegrasi di setiap fase. - Dokumentasi sangat lengkap. |
| Kekurangan | - Membutuhkan pengelolaan iterasi yang baik.  - Perubahan besar di awal dapat memperpanjang iterasi. | - Membutuhkan koordinasi intensif dengan pengguna. - Tidak cocok untuk tim kecil dengan sumber daya terbatas. | - Tidak fleksibel: Sulit mengakomodasi perubahan di tengah jalan. - Risiko tinggi jika kesalahan ditemukan di tahap akhir. | - Dapat menyebabkan scope creep (cakupan meluas). - Membutuhkan revisi besar jika prototipe tidak mencerminkan kebutuhan akhir pengguna. | - Tidak fleksibel terhadap perubahan. - Tidak cocok untuk kebutuhan yang dinamis. |
| Jumlah Orang Terlibat | Sedang (dapat bekerja dalam tim kecil). | Tinggi (tim kolaboratif). | Rendah hingga sedang. | Sedang (developer dan pengguna aktif terlibat). | Rendah hingga sedang. |
| Fleksibilitas | Tinggi: Perubahan dapat diterapkan di setiap iterasi. | Sangat tinggi: Fokus pada perubahan cepat. | Rendah: Perubahan sulit dilakukan setelah tahap dimulai. | Tinggi: Prototipe dapat diubah dengan cepat. | Rendah: Fokus pada tahapan linier. |
| Keterlibatan Pengguna | Rendah: Pengguna tidak selalu terlibat di tiap iterasi karena iterasi lebih teknis. | Tinggi: Pengguna terlibat aktif di tiap iterasi. | Sangat rendah: Keterlibatan pengguna hanya di awal dan akhir. | Tinggi: Pengguna terlibat langsung dalam evaluasi prototipe. | Rendah: Pengguna terlibat hanya di tahap awal dan akhir. |
| Jenis Proyek | Proyek kompleks dan skala sedang hingga besar. | Proyek yang memerlukan perubahan cepat. | Proyek dengan spesifikasi tetap dari awal. | Proyek dengan kebutuhan yang belum jelas. | Proyek kritis seperti perangkat medis |

Tabel . Tabel Perbandingan Metode

Melalui analisis tersebut, peneliti menegaskan bahwa Iterative Incremental menawarkan fleksibilitas yang lebih tinggi dan efisiensi dalam mengelola sumber daya dibandingkan metode lainnya. Metode ini juga memungkinkan pengembang untuk mengintegrasikan perubahan secara bertahap pada setiap iterasi, sehingga dapat memenuhi kebutuhan pengguna dengan lebih baik sepanjang siklus pengembangan.

Selain itu, keterlibatan pengguna dalam memantau proses pengembangan memberikan dampak positif terhadap kualitas hasil akhir. Oleh karena itu, metode Iterative Incremental dipilih sebagai pendekatan yang paling sesuai untuk proyek ini, dengan mempertimbangkan kompleksitas aplikasi yang akan dikembangkan serta kebutuhan untuk memastikan kualitas dan kepuasan pengguna.

## Batasan dan Asumsi Penelitian

Proses pengembangan aplikasi ini dilakukan menggunakan metode iterative incremental, dengan minimal dua siklus iterasi untuk memastikan perbaikan dan penyempurnaan secara bertahap. Aplikasi dirancang menggunakan bahasa pemrograman Python, yang berfungsi untuk menerima dan memproses data dari perangkat IoT, sementara antarmuka frontend aplikasi dibuat dengan memanfaatkan Javascript untuk menghadirkan tampilan yang interaktif dan ramah pengguna. Aplikasi ini ditujukan bagi perusahaan tambang serta pemerintah, khususnya Kementerian Lingkungan Hidup dan Kehutanan (KLHK), sebagai alat untuk mengelola dan memantau laporan terkait emisi gas rumah kaca (GRK).

## Rencana Jadwal Kegiatan

Pada Tabel III.2. dijelaskan tahap-tahap dalam penelitian ini dan disesuaikan jadwal yang telah ditentukan. Jadwal penelitian dimulai dari bulan Oktober 2024 hingga Juni 2025 yang mencakup tujuh tahap utama, yaitu tahap identifikasi, perencanaan, analisis & design, pengembangan, pengujian, deployment, serta kesimpulan dan saran. Setiap tahap telah dialokasikan waktu yang sesuai dengan kompleksitas dan kebutuhan masing-masing aktivitas, dimana tahap identifikasi dan analisis & design mendapat porsi waktu yang lebih besar mengingat pentingnya fase perencanaan dalam menentukan keberhasilan penelitian.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **2024** | | | **2025** | | | | | |
| **No** | **Kegiatan** | 10 | 11 | 12 | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Tahap Identifikasi |  |  |  |  |  |  |  |  |  |
|  | Mengidentifikasi Masalah | ✓ | ✓ |  |  |  |  |  |  |  |
|  | Penetapan Latar Belakang | ✓ | ✓ |  |  |  |  |  |  |  |
|  | Penetapan Rumusan Masalah | ✓ | ✓ |  |  |  |  |  |  |  |
|  | Penetapan Tujuan Penelitian | ✓ | ✓ |  |  |  |  |  |  |  |
|  | Penetapan Batasan Masalah |  | ✓ | ✓ |  |  |  |  |  |  |
|  | Studi Literatur |  | ✓ | ✓ |  |  |  |  |  |  |
|  | Analisis Kebutuhan |  |  | ✓ | ✓ |  |  |  |  |  |
| 2 | Tahap Perencanaan |  |  |  |  |  |  |  |  |  |
|  | Menetapkan Fitur-Fitur |  |  |  | ✓ | ✓ |  |  |  |  |
|  | Menetapkan Tahap Pengembangan |  |  |  | ✓ | ✓ |  |  |  |  |
| 3 | Tahap Analisis & Design |  |  |  |  |  |  |  |  |  |
|  | Membuat Use Case Diagram |  |  |  |  | ✓ | ✓ |  |  |  |
|  | Membuat Activity Diagram |  |  |  |  | ✓ | ✓ |  |  |  |
|  | Membuat Class Diagram |  |  |  |  | ✓ | ✓ |  |  |  |
|  | Membuat Sequence Diagram |  |  |  |  | ✓ | ✓ |  |  |  |
|  | Membuat Deployment Diagram |  |  |  |  | ✓ | ✓ |  |  |  |
|  | Membuat Entity Relationship Diagram |  |  |  |  | ✓ | ✓ |  |  |  |
| 4 | Tahap Pengembangan |  |  |  |  |  |  |  |  |  |
|  | Setup Development Environment |  |  |  |  |  | ✓ | ✓ |  |  |
|  | Coding & Debugging |  |  |  |  |  | ✓ | ✓ |  |  |
| 5 | Tahap Pengujian |  |  |  |  |  |  |  |  |  |
|  | Pengujian Aplikasi |  |  |  |  |  |  | ✓ | ✓ |  |
|  | Evaluasi Aplikasi |  |  |  |  |  |  |  | ✓ | ✓ |
| 6 | Tahap Deployment |  |  |  |  |  |  |  |  |  |
|  | Peluncuran Aplikasi |  |  |  |  |  |  |  | ✓ | ✓ |
| 7 | Tahap Kesimpulan |  |  |  |  |  |  |  |  |  |
|  | Mengerjakan Kesimpulan dan Saran |  |  |  |  |  |  |  |  | ✓ |

Tabel . Tabel Rencana Jadwal Kegiatan

# Analisis dan Perancangan

Pada bab ini, disajikan hasil analisis proses bisnis terhadap sistem informasi akademik, pemodelan kebutuhan perangkat lunak dan spesifikasi kebutuhan teknologi sebagai solusi permasalahan.

## Analisis Proses Bisnis

Subbab ini menyajikan hasil analisis proses bisnis di Fakultas Rekayasa Industri, Universitas Telkom.

## Desain Perangkat Lunak

Subbab ini menyajikan hasil desain / perancangan sistem informasi.

## Spesifikasi Kebutuhan Teknologi

Subbab ini menyajikan spesifikasi teknologi berdasarkan pengumpulan dan pengolahan data yang telah dilakukan.

# Implementasi dan Pengujian

## Hasil Implementasi Perangkat Lunak

Pada bab ini dilakukan analisis hasil dari bab sebelumnya. Apabila bab sebelumnya adalah mengenai pengumpulan dan pengolahan data, maka bab ini membahas mengenai apa yang dapat diketahui dan dihasilkan dari hasil pengumpulan dan pengolahan data. Secara keseluruhan bab ini membahas secara mendetail mengenai hasil dari penelitian dan refleksinya terhadap tujuan penelitian. Analisis harus dilakukan secara sistematis sesuai dengan urutan pada perumusan masalah dan tujuan penelitian.

Pada bab ini dilakukan analisis hasil dari bab sebelumnya. Hasil pengolahan data yang telah dirumuskan pada Bab II akan diverifikasi dan/atau divalidasi apakah telah benar-benar menyelesaikan masalah atau menurunkan gap antara kondisi eksisting dan target yang akan dicapai. Interpretasi hasil harus dijelaskan pada bab ini. Analisis sensitivitas juga dapat digunakan di bab ini untuk lebih mengetahui bagaimana penerapan hasil penelitian.

## Proses dan Hasil Pengujian Perangkat Lunak

Hasil pengujian dari pengujian unit hingga pengujian sistem.

## Evaluasi Akhir Solusi Sistem Informasi

Hasil kuesioner dan *expert judgment* terhadap perangkat lunak yang telah dikembangkan. Dalam evaluasi akhir, dapat dibuktikan bahwa solusi yang dikembangkan mampu menjawab rumusan permasalahan.

# Kesimpulan dan Saran

## Kesimpulan

Kesimpulan merupakan jawaban dari tujuan penelitian yang dibuat di BAB I, dengan acuan apabila terdapat dua tujuan penelitian maka kesimpulan yang dihasilkan juga harus minimal berjumlah dua. Penulisan kesimpulan harus berdasarkan hasil dari penelitian terutama setelah hasil analisis atau implementasi telah dievaluasi. Hindari penggunaan nomor jika kesimpulan penelitian hanya satu simpulan saja.

## Saran

Saran harus didasarkan pada temuan dan kesimpulan dari penelitian yang dilakukan. Rekomendasi dapat spesifik atau umum atau untuk perusahaan , termasuk saran untuk studi lebih lanjut.

# Daftar Pustaka

Abdulmonim, D. A., Muhamad, Z. H., & Alathari, B. (2019). Using the Object Mapping Approach From Analysis to Implementation for Developing Student Registration System. *Indonesian Journal of Electrical Engineering and Computer Science*, *14*(2), 1030. https://doi.org/10.11591/ijeecs.v14.i2.pp1030-1038

Achimugu, P., Afolabi, B., Adeniran, O., Gambo, I., & Oluwagbemi, O. (2010). Software Architecture and Methodology as a Tool for Efficient Software Engineering Process: A Critical Appraisal. *Journal of Software Engineering and Applications*. https://doi.org/10.4236/jsea.2010.310110

Adi Sucipto, Aulia Brilliantina, Elok Kurnia Novita Sari, Rizza Wijaya, Dimas Triardianto, & Adhima Adhamatika. (2023). Rancang Bangun Alat Deteksi dan Pengukur Gas Emisi Karbondioksida (CO₂) dan Gas Emisi Metana (CH₄) Berbasis Mikrokontroler. *JUSTER: Jurnal Sains dan Terapan* , *2*(1), 122–126.

Ahmad, T., Iqbal, J., Ashraf, A., Truşcan, D., & Porres, I. (2019). Model-Based Testing Using UML Activity Diagrams: A Systematic Mapping Study. *Computer Science Review*. https://doi.org/10.1016/j.cosrev.2019.07.001

Al Hanif, A., & Ilyas, M. (2024). Effective Feature Engineering Framework for Securing MQTT Protocol in IoT Environments. *Sensors*, *24*(6). https://doi.org/10.3390/s24061782

Alam, I., Sarwar, N., & Noreen, I. (2022). Statistical Analysis of Software Development Models by Six-Pointed Star Framework. *Plos One*. https://doi.org/10.1371/journal.pone.0264420

Alenezi, M., Basit, H. A., Beg, M. A., & Shaukat, M. S. (2022). Synthesizing Secure Software Development Activities for Linear and Agile Lifecycle Models. *Software Practice and Experience*. https://doi.org/10.1002/spe.3072

Aljohani, M., & Jameel Qureshi, M. R. (2016). Management of Changes in Software Requirements During Development Phases. *International Journal of Education and Management Engineering*. https://doi.org/10.5815/ijeme.2016.06.02

Ameedeen, M. A., & Bordbar, B. (2020). A Model Driven Analysis of the 802.11 CSMA/CA Protocol Through SD2PN. *International Journal of Simulation Systems Science & Technology*. https://doi.org/10.5013/ijssst.a.15.01.06

Ameedeen, M. A., Bordbar, B., & Anane, R. (2009). *A Model Driven Approach to the Analysis of Timeliness Properties*. https://doi.org/10.1007/978-3-642-02674-4\_16

Ameedeen, M. A., Bordbar, B., & Anane, R. (2011). Model Interoperability via Model Driven Development. *Journal of Computer and System Sciences*. https://doi.org/10.1016/j.jcss.2010.01.011

Anifah, E. M., Wahyu Rini, I. D., Hidayat, R., & Ridho, M. (2021). Estimasi Emisi Gas Rumah Kaca (Grk) Kegiatan Pengelolaan Sampah Di Kelurahan Karang Joang, Balikpapan. *Jurnal Sains &Teknologi Lingkungan*. https://doi.org/10.20885/jstl.vol13.iss1.art2

Ardisty Sitogasa, P. S. (2022a). Penerapan Solid Waste Management Tool (SWMT) Sebagai Analisis Gas Rumah Kaca Hasil Penanganan Sampah Di Kota Kediri. *Jurnal Ilmiah Sistem Informasi Dan Ilmu Komputer*. https://doi.org/10.55606/juisik.v2i2.166

Ardisty Sitogasa, P. S. (2022b). Penerapan Solid Waste Management Tool (SWMT) Sebagai Analisis Gas Rumah Kaca Hasil Penanganan Sampah Di Kota Kediri. *Jurnal Ilmiah Sistem Informasi Dan Ilmu Komputer*. https://doi.org/10.55606/juisik.v2i2.166

Ari Wibowo. (2013). Kajian Penurunan Emisi Gas Rumah Kaca Sektor Kehutanan untuk Mendukung Kebijakan Perpres No. 61/2011. *Jurnal Analisis Kebijakan Kehutanan*, *10*(3), 236–254.

Arifiyanto, B., & Sindu, R. M. (2020). Pengurangan Emisi Gas Rumah Kaca dengan Penerapan E-Reporting System di Pertambangan PT Bukit Asam. *Seminar Nasional Lahan Suboptimal ke-8*.

Azadi, M., Northey, S. A., Ali, S. H., & Edraki, M. (2020). Transparency on greenhouse gas emissions from mining to enable climate change mitigation. *Nature Geoscience*, *13*(2). https://doi.org/10.1038/s41561-020-0531-3

Azevedo, S., Machado, R. J., Bragança, A., & Ribeiro, H. (2010). *Support for Variability in Use Case Modeling With Refinement*. https://doi.org/10.1145/1865875.1865876

Azzedin, F., & Alhazmi, T. (2023). Secure Data Distribution Architecture in IoT Using MQTT. *Applied Sciences (Switzerland)*, *13*(4). https://doi.org/10.3390/app13042515

Bendoukha, H., Slimani, Y., & Benyettou, A. (2012). *UML Refinement for Mapping UML Activity Diagrams Into BPEL Specifications to Compose Service-Oriented Workflows*. https://doi.org/10.1007/978-3-642-30567-2\_44

Berardi, D., Calvanese, D., & Giacomo, G. D. (2005). Reasoning on UML Class Diagrams. *Artificial Intelligence*, *168*(1–2), 70–118. https://doi.org/10.1016/j.artint.2005.05.003

Biswas, N., Mondal, A., Kusumastuti, A., Saha, S., & Mondal, K. C. (2022). Automated Credit Assessment Framework Using ETL Process and Machine Learning. *Innovations in Systems and Software Engineering*. https://doi.org/10.1007/s11334-022-00522-x

Biswas, N., Sarkar, A., & Mondal, K. C. (2019). Efficient Incremental Loading in ETL Processing for Real-Time Data Integration. *Innovations in Systems and Software Engineering*. https://doi.org/10.1007/s11334-019-00344-4

Brabra, H., Mtibaa, A., Petrillo, F., Merle, P., Sliman, L., Moha, N., Gaaloul, W., Guéhéneuc, Y., Benatallah, B., & Gargouri, F. (2019). On Semantic Detection of Cloud API (Anti)patterns. *Information and Software Technology*. https://doi.org/10.1016/j.infsof.2018.10.012

Byers, D., & Shahmehri, N. (2007). *Design of a Process for Software Security*. https://doi.org/10.1109/ares.2007.67

Chaerul, M., Febrianto, A., & Tomo, H. S. (2020a). Peningkatan Kualitas Penghitungan Emisi Gas Rumah Kaca Dari Sektor Pengelolaan Sampah Dengan Metode IPCC 2006 (Studi Kasus: Kota Cilacap). *Jurnal Ilmu Lingkungan*. https://doi.org/10.14710/jil.18.1.153-161

Chaerul, M., Febrianto, A., & Tomo, H. S. (2020b). Peningkatan Kualitas Penghitungan Emisi Gas Rumah Kaca Dari Sektor Pengelolaan Sampah Dengan Metode IPCC 2006 (Studi Kasus: Kota Cilacap). *Jurnal Ilmu Lingkungan*. https://doi.org/10.14710/jil.18.1.153-161

Chien, H. Y., Chen, Y. J., Qiu, G. H., Liao, J. F., Hung, R. W., Lin, P. C., Kou, X. A., Chiang, M. L., & Su, C. (2020). A MQTT-API-compatible IoT security-enhanced platform. *International Journal of Sensor Networks*, *32*(1). https://doi.org/10.1504/IJSNET.2020.104463

Chien, H. Y., & Wang, N. Z. (2022). A Novel MQTT 5.0-Based Over-the-Air Updating Architecture Facilitating Stronger Security. *Electronics (Switzerland)*, *11*(23). https://doi.org/10.3390/electronics11233899

Corradini, D., Zampieri, A., Pasqua, M., Viglianisi, E., Dallago, M., & Ceccato, M. (2022). Automated Black‐box Testing of Nominal and Error Scenarios in RESTful APIs. *Software Testing Verification and Reliability*. https://doi.org/10.1002/stvr.1808

Cuaresma, M. J. E., & Cavarero, J.-L. (2005). *Building Class Diagrams Systematically*. https://doi.org/10.5220/0002511103780381

Dawod, A. (2024). A Survey of Techniques for Discovering, Using, and Paying for Third-Party IoT Sensors. *Sensors*. https://doi.org/10.3390/s24082539

Dawod, A., Georgakopoulos, D., Jayaraman, P. P., Nirmalathas, A., & Parampalli, U. (2022). IoT Device Integration and Payment via an Autonomic Blockchain-Based Service for IoT Device Sharing. *Sensors*. https://doi.org/10.3390/s22041344

Developing a Sales Dashboard With Power BI – A Case Study in a Pharmaceutical Company. (2024). *Decision Making Advances*. https://doi.org/10.31181/dma21202438

Diansyah, A. F. (2023). Comparative Analysis of Software Development Lifecycle Methods in Software Development: A Systematic Literature Review. *International Journal of Advances in Data and Information Systems*. https://doi.org/10.25008/ijadis.v4i2.1295

Ditjen PPI KLHK. (2017). Strategi Implementasi NDC (Nationally Determined Contribution). *Kementerian Lingkungan Hidup dan Kehutanan*.

Durmuşoğlu, S. S., Atuahene‐Gima, K., & Calantone, R. J. (2022). Marketing Strategy Decision Making in New Product Development: Direct Effects and Moderation by Market Information Time Sensitivity and Analyzability. *European Journal of Innovation Management*. https://doi.org/10.1108/ejim-11-2021-0575

Ehsan, A., M. Abuhaliqa, M. A., Çatal, Ç., & Mishra, D. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences*. https://doi.org/10.3390/app12094369

Fauzan, R., Siahaan, D., Rochimah, S., & Triandini, E. (2021). Automated Class Diagram Assessment Using Semantic and Structural Similarities. *International Journal of Intelligent Engineering and Systems*, *14*(2), 52–66. https://doi.org/10.22266/ijies2021.0430.06

Friedenthal, S. A., & Burkhart, R. (2003). 3.6.1 Extending UMLTM from Software to Systems. *INCOSE International Symposium*, *13*(1). https://doi.org/10.1002/j.2334-5837.2003.tb02665.x

Gedam, M. N., & Meshram, B. B. (2022). Proposed Secure 3-Use Case Diagram. *International Journal of Systems and Software Security and Protection*. https://doi.org/10.4018/ijsssp.293237

Gedam, M. N., & Meshram, B. B. (2023). Proposed Secure Activity Diagram for Software Development. *International Journal of Advanced Computer Science and Applications*. https://doi.org/10.14569/ijacsa.2023.0140671

Ghaleb, T. A., Alturki, M. A., & Aljasser, K. (2018). Program Comprehension Through Reverse‐engineered Sequence Diagrams: A Systematic Review. *Journal of Software Evolution and Process*. https://doi.org/10.1002/smr.1965

Godefroid, P., Huang, B.-Y., & Polishchuk, M. (2020). *Intelligent REST API Data Fuzzing*. https://doi.org/10.1145/3368089.3409719

Gokarna, M. (2021). *DevOps Phases Across Software Development Lifecycle*. https://doi.org/10.36227/techrxiv.13207796.v2

Goldin, E., Feldman, D., Georgoulas, G., Arranz, M. C., & Nikolakopoulos, G. (2017). *Cloud Computing for Big Data Analytics in the Process Control Industry*. https://doi.org/10.1109/med.2017.7984310

Gómez-Martínez, E., Rodríguez, R. J., Benac-Earle, C., Etxeberria, L., & Illarramendi, M. (2018). A methodology for model-based verification of safety contracts and performance requirements. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, *232*(3). https://doi.org/10.1177/1748006X16667328

Grönninger, H., Reiß, D., & Rumpe, B. (2010). *Towards a Semantics of Activity Diagrams With Semantic Variation Points*. https://doi.org/10.1007/978-3-642-16145-2\_23

Gross, A., & Doerr, J. (2009). *EPC vs. UML Activity Diagram - Two Experiments Examining Their Usefulness for Requirements Engineering*. https://doi.org/10.1109/re.2009.30

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2013.01.010

Gupta, M. K., Govil, M. C., & Singh, G. (2014). *Static Analysis Approaches to Detect SQL Injection and Cross Site Scripting Vulnerabilities in Web Applications: A Survey*. https://doi.org/10.1109/icraie.2014.6909173

Haleem, M. S., Ekuban, A., Antonini, A., Pagliara, S. M., Pecchia, L., & Allocca, C. (2023). Deep-Learning-Driven Techniques for Real-Time Multimodal Health and Physical Data Synthesis. *Electronics*. https://doi.org/10.3390/electronics12091989

Hammoodi, M. S., Stahl, F., & Badii, A. (2018). Real-Time Feature Selection Technique With Concept Drift Detection Using Adaptive Micro-Clusters for Data Stream Mining. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2018.08.007

Hause, M., & Thom, F. (2008). Building bridges between systems and software with SysML and UML. *18th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2008*, *5*. https://doi.org/10.1002/j.2334-5837.2008.tb00843.x

Hazra, R., & Dey, S. (2014a). Consistency between Use Case, Sequence and Timing Diagram for Real Time Software Systems. *International Journal of Computer Applications*, *85*(16). https://doi.org/10.5120/14924-3444

Hazra, R., & Dey, S. (2014b). Consistency Between Use Case, Sequence and Timing Diagram for Real Time Software Systems. *International Journal of Computer Applications*. https://doi.org/10.5120/14924-3444

Ibrahim, N., Ibrahim, R., Saringat, M. Z., Mansor, D., & Herawan, T. (2011). *Definition of Consistency Rules Between UML Use Case and Activity Diagram*. https://doi.org/10.1007/978-3-642-20998-7\_58

Ihirwe, F., Iovino, G., & Di Ruscio, D. (2021). Towards an MQTT5 geo-location extension for location-aware applications. *2021 44th International Conference on Telecommunications and Signal Processing, TSP 2021*. https://doi.org/10.1109/TSP52935.2021.9522590

Ihsan, R. N. (2023a). United Nations Economic and Social Council (UNESC): Pemenuhan Hak Asasi Manusia Melalui Penerapan Pajak Karbon Di Indonesia. *Padjadjaran Journal of International Relations*. https://doi.org/10.24198/padjir.v5i2.47088

Ihsan, R. N. (2023b). United Nations Economic and Social Council (UNESC): Pemenuhan Hak Asasi Manusia Melalui Penerapan Pajak Karbon Di Indonesia. *Padjadjaran Journal of International Relations*. https://doi.org/10.24198/padjir.v5i2.47088

Kamburugamuve, S., Christiansen, L., & Fox, G. (2015). A Framework for Real Time Processing of Sensor Data in the Cloud. *Journal of Sensors*. https://doi.org/10.1155/2015/468047

Kerkouche, E., Chaoui, A., Bourennane, E. B., & Labbani, O. (2010). A UML and colored petri nets integrated modeling and analysis approach using graph transformation. *Journal of Object Technology*, *9*(4). https://doi.org/10.5381/jot.2010.9.4.a2

Khan, M. A., Khan, M. A., Jan, S. U., Ahmad, J., Jamal, S. S., Shah, A. A., Pitropakis, N., & Buchanan, W. J. (2021). A deep learning-based intrusion detection system for mqtt enabled iot. Dalam *Sensors* (Vol. 21, Nomor 21). https://doi.org/10.3390/s21217016

Khurana, N., Chhillar, R. S., & Chhillar, U. (2016). A Novel Technique for Generation and Optimization of Test Cases Using Use Case, Sequence, Activity Diagram and Genetic Algorithm. *Journal of Software*. https://doi.org/10.17706/jsw.11.3.242-250

Klünder, J., Busch, M., Dehn, N., & Karras, O. (2021). *Towards Shaping the Software Lifecycle With Methods and Practices*. https://doi.org/10.48550/arxiv.2103.10104

Korala, H., Georgakopoulos, D., Jayaraman, P. P., & Yavari, A. (2021a). *A Time-Sensitive IoT Data Analysis Framework*. https://doi.org/10.24251/hicss.2021.865

Korala, H., Georgakopoulos, D., Jayaraman, P. P., & Yavari, A. (2021b). Managing Time-Sensitive IoT Applications via Dynamic Application Task Distribution and Adaptation. *Remote Sensing*. https://doi.org/10.3390/rs13204148

Kulkarni, R. N., & Srinivasa, C. K. (2019). Ameliorated methodology to represent UML use case diagram into table format. *International Journal of Engineering and Advanced Technology*, *9*(1). https://doi.org/10.35940/ijeat.A1329.109119

Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things Is a Revolutionary Approach for Future Technology Enhancement: A Review. *Journal of Big Data*. https://doi.org/10.1186/s40537-019-0268-2

Kurdi, H., & Thayananthan, V. (2022). A Multi-Tier MQTT Architecture with Multiple Brokers Based on Fog Computing for Securing Industrial IoT. *Applied Sciences (Switzerland)*, *12*(14). https://doi.org/10.3390/app12147173

Labiche, Y., Kolbah, B., & Mehrfard, H. (2013). *Combining Static and Dynamic Analyses to Reverse-Engineer Scenario Diagrams*. https://doi.org/10.1109/icsm.2013.24

Lange, C. F. J., Wijns, M. A. M., & Chaudron, M. R. V. (2007). A visualization framework for task-oriented modeling using UML. *Proceedings of the Annual Hawaii International Conference on System Sciences*. https://doi.org/10.1109/HICSS.2007.44

Li, D. (2010). *QVT Based Model Transformation From Sequence Diagram to CSP*. https://doi.org/10.1109/iceccs.2010.47

Licorish, S. A., Holvitie, J., Hyrynsalmi, S., Leppänen, V., Spínola, R. O., Mendes, T. S., MacDonell, S. G., & Buchan, J. (2016). *Adoption and Suitability of Software Development Methods and Practices*. https://doi.org/10.1109/apsec.2016.062

Lima, L., & Tavares, A. (2019). *Verifying Deadlock and Nondeterminism in Activity Diagrams*. https://doi.org/10.1109/models-c.2019.00119

Longo, E., Redondi, A. E. C., Cesana, M., Arcia-Moret, A., & Manzoni, P. (2020). MQTT-ST: A Spanning Tree Protocol for Distributed MQTT Brokers. *IEEE International Conference on Communications*, *2020-June*. https://doi.org/10.1109/ICC40277.2020.9149046

Longo, E., Redondi, A. E. C., Cesana, M., & Manzoni, P. (2022). BORDER: a Benchmarking Framework for Distributed MQTT Brokers. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2022.3155872

Ma, K., & Yang, B. (2017). Stream‐based Live Data Replication Approach of In‐memory Cache. *Concurrency and Computation Practice and Experience*. https://doi.org/10.1002/cpe.4052

Malakhov, K., Kurgaev, A. F., & Velychko, V. (2018). Modern Restful Api DLS and Frameworks for Restful Web Services Api Schema Modeling, Documenting, Visualizing. *Problems in Programming*. https://doi.org/10.15407/pp2018.04.059

Malihah, L. (2022). Tantangan Dalam Upaya Mengatasi Dampak Perubahan Iklim Dan Mendukung Pembangunan Ekonomi Berkelanjutan: Sebuah Tinjauan. *Jurnal Kebijakan Pembangunan*, *17*(2). https://doi.org/10.47441/jkp.v17i2.272

Maske, M., & Prasad, P. S. (2015). An Introduction to Real Time Processing and Streaming of Wireless Network Data. *Ijarcce*. https://doi.org/10.17148/ijarcce.2015.4150

Mathur, B., & Kaushik, M. P. (2016). Empirical Analysis of Metrics Using UML Class Diagram. *International Journal of Advanced Computer Science and Applications*, *7*(5). https://doi.org/10.14569/ijacsa.2016.070506

Mavromatis, A., Gunner, S., Tryfonas, T., & Simeonidou, D. (2019). *Dynamic Cloud Service Management for Scalable Internet of Things Applications*. https://doi.org/10.1109/smartworld-uic-atc-scalcom-iop-sci.2019.00345

Mikhaylov, A., Moiseev, N., Aleshin, K., & Burkhardt, T. (2020). Global climate change and greenhouse effect. *Entrepreneurship and Sustainability Issues*, *7*(4). https://doi.org/10.9770/jesi.2020.7.4(21)

Mishra, B., & Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2020.3035849

Mishra, D., & Mishra, A. (2011). Complex Software Project Development: Agile Methods Adoption. *Journal of Software Maintenance and Evolution Research and Practice*. https://doi.org/10.1002/smr.528

Mohd Zainuddin, Z. Q., Yahya, F., & Yahya, A. F. (2022). *Visual Design as the Key Factor for Effective Environmental Analytics Dashboard*. https://doi.org/10.3390/proceedings2022082021

MR, S. Pratama. (2023a). Pengungkapan Emisi Gas Rumah Kaca Pada Laporan Keberlanjutan Perusahaan Di Indonesia. *Akuntansiku*. https://doi.org/10.54957/akuntansiku.v2i4.549

MR, S. Pratama. (2023b). Pengungkapan Emisi Gas Rumah Kaca Pada Laporan Keberlanjutan Perusahaan Di Indonesia. *Akuntansiku*. https://doi.org/10.54957/akuntansiku.v2i4.549

Mustafa, B. A. (2010). Comparing the effect of use case format on end user understanding of system requirements. *Journal of Information Technology Research*, *3*(4). https://doi.org/10.4018/jitr.2010100101

Mustikaningrum, D., Kristiawan, K., & Suprayitno, S. (2021). Emisi Gas Rumah Kaca Sektor Pertanian Di Kabupaten Tuban: Inventarisasi Dan Potensi Aksi Mitigasi. *Jurnal Wilayah Dan Lingkungan*. https://doi.org/10.14710/jwl.9.2.155-171

Najem, G. (2023). Building a Business Intelligence Dashboard for a Lebanese Company. *Revista Inovação Projetos E Tecnologias*. https://doi.org/10.5585/iptec.v11i1.24603

Nanthaamornphong, A., & Leatongkam, A. (2019). Extended ForUML for Automatic Generation of UML Sequence Diagrams From Object-Oriented Fortran. *Scientific Programming*. https://doi.org/10.1155/2019/2542686

Nugrahayu, Q., Nurjannah, N. K., & Hakim, L. (2017). Estimasi Emisi Karbondioksida Dari Sektor Permukiman Di Kota Yogyakarta Menggunakan Ipcc Guidelines. *Jurnal Sains &Teknologi Lingkungan*. https://doi.org/10.20885/jstl.vol9.iss1.art3

Nyembe, F. H. (2023). Formal Methods for an Agile Scrum Software Development Methodology. *Icat*. https://doi.org/10.58190/icat.2023.35

Ounacer, S., Talhaoui, M. A., Ardchir, S., Daif, A., & Azouazi, M. (2017). A New Architecture for Real Time Data Stream Processing. *International Journal of Advanced Computer Science and Applications*. https://doi.org/10.14569/ijacsa.2017.081106

Palma, F., Dubois, J., Moha, N., & Guéhéneuc, Y. (2014). *Detection of REST Patterns and Antipatterns: A Heuristics-Based Approach*. https://doi.org/10.1007/978-3-662-45391-9\_16

Pappas, L. M., & Whitman, L. (2011). *Riding the Technology Wave: Effective Dashboard Data Visualization*. https://doi.org/10.1007/978-3-642-21793-7\_29

Park, J. H., Kim, H. S., & Kim, W. T. (2018). DM-MQTT: An efficient MQTT based on SDN multicast for massive IoT communications. *Sensors (Switzerland)*, *18*(9). https://doi.org/10.3390/s18093071

Pautasso, C. (2013). *RESTful Web Services: Principles, Patterns, Emerging Technologies*. https://doi.org/10.1007/978-1-4614-7518-7\_2

Penman, J., Gytarsky, M., Hiraishi, T., Irving, W., & Krug, T. (2006). 2006 IPCC - Guidelines for National Greenhouse Gas Inventories. Dalam *Directrices para los inventarios nacionales GEI*.

Peralta Abadia, J. J. (2023). *Trends and Recommendations for IoT-Based Smart City Applications*. https://doi.org/10.1007/978-3-031-32515-1\_1

Pickin, S., Jard, C., Jéron, T., Jézéquel, J.-M., & Traon, Y. Le. (2007). Test Synthesis From UML Models of Distributed Software. *Ieee Transactions on Software Engineering*. https://doi.org/10.1109/tse.2007.39

Pratama. MR, S., Zaman, A. N., & Firmansyah, A. (2023). Pengungkapan Emisi Gas Rumah Kaca Pada Laporan Keberlanjutan Perusahaan Di Indonesia. *Akuntansiku*, *2*(4). https://doi.org/10.54957/akuntansiku.v2i4.549

Rabiei, R., & Almasi, S. (2022). Requirements and Challenges of Hospital Dashboards: A Systematic Literature Review. *BMC Medical Informatics and Decision Making*. https://doi.org/10.1186/s12911-022-02037-8

Rachansa, K. H. (2024). Integrated Multi-Income Stream Performance Dashboard: A Japanese Corporate Banking Case. *International Journal of Advances in Data and Information Systems*. https://doi.org/10.59395/ijadis.v5i1.1313

Rahayu, S. K., & Anggadini, S. D. (2014). Analisis Budaya Organisasi Pada Pengembangan Sistem Informasi Di Unikom. *Majalah Ilmiah Unikom*. https://doi.org/10.34010/miu.v12i2.27

Rahmasari, D. (2024a). Carbon Tax Study as a Social Engineering Tool in Realizing the Agenda of Sustainable Development Goals (SDGs) in Indonesia. *Ikatan Penulis Mahasiswa Hukum Indonesia Law Journal*. https://doi.org/10.15294/ipmhi.v4i1.71582

Rahmasari, D. (2024b). Carbon Tax Study as a Social Engineering Tool in Realizing the Agenda of Sustainable Development Goals (SDGs) in Indonesia. *Ikatan Penulis Mahasiswa Hukum Indonesia Law Journal*. https://doi.org/10.15294/ipmhi.v4i1.71582

Rashid, N., Khan, S. U., Khan, H. U., & Ilyas, M. (2021). Green-Agile Maturity Model: An Evaluation Framework for Global Software Development Vendors. *Ieee Access*. https://doi.org/10.1109/access.2021.3079194

Raza, E., Sabaruddin, L. O., & Komala, A. L. (2020). Manfaat dan Dampak Digitalisasi Logistik di Era Industri 4.0. *Jurnal Logistik Indonesia*, *4*(1). https://doi.org/10.31334/logistik.v4i1.873

Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Clarke, S. (2016). Middleware for Internet of Things: A Survey. *Ieee Internet of Things Journal*. https://doi.org/10.1109/jiot.2015.2498900

Real-Time Data Stream Processing - Challenges and Perspectives. (2017). *International Journal of Computer Science Issues*. https://doi.org/10.20943/01201705.612

Rizky, M., & Sugiarti, Y. (2022). Pengunaan Metode Scrum Dalam Pengembangan Perangkat Lunak: Literature Review. *Journal of Computer Science and Engineering (Jcse)*. https://doi.org/10.36596/jcse.v3i1.353

Rountev, A., Volgin, O., & Reddoch, M. (2005). Static Control-Flow Analysis for Reverse Engineering of UML Sequence Diagrams. *Acm Sigsoft Software Engineering Notes*. https://doi.org/10.1145/1108768.1108816

Ruparelia, N. B. (2010). Software Development Lifecycle Models. *Acm Sigsoft Software Engineering Notes*. https://doi.org/10.1145/1764810.1764814

Sabharwal, S., Sibal, R., & Kaur, P. (2015). Deriving Complexity Metric based on Use Case Diagram and its validation. *2014 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2014*. https://doi.org/10.1109/ISSPIT.2014.7300571

Sahlmann, K., Clemens, V., Nowak, M., & Schnor, B. (2021). Mup: Simplifying secure over-the-air update with mqtt for constrained iot devices. *Sensors (Switzerland)*, *21*(1). https://doi.org/10.3390/s21010010

Sallaby, A. F., & Kanedi, I. (2020). Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter. *Jurnal Media Infotama*. https://doi.org/10.37676/jmi.v16i1.1121

Sanjana, A., Anusha, M., Pravallika, G., & Radhika, Mrs. S. (2022). REST APIs Cloud Service Security Checks. *International Journal for Research in Applied Science and Engineering Technology*. https://doi.org/10.22214/ijraset.2022.45147

Saringat, M. Z., Herawan, T., & Ibrahim, R. (2010). A Proposal for Constructing Relational Database From Class Diagram. *Computer and Information Science*, *3*(2). https://doi.org/10.5539/cis.v3n2p38

Sattar, Mohd. A., Anwaruddin, M., & Ali, M. A. (2017). A Review on Internet of Things - Protocols, Issues. *Ijireeice*. https://doi.org/10.17148/ijireeice.2017.5217

Sekarini, L. A., & Setiadi, I. (2022). PENGARUH LEVERAGE, PROFITABILITAS, UKURAN PERUSAHAAN DAN KINERJA LINGKUNGAN TERHADAP PENGUNGKAPAN EMISI KARBON PERUSAHAAN (STUSI EMPIRIS PADA PERUSAHAAN PERTAMBANGAN YANG TERDAFTAR DI BURSA EFEK INDONESIA PERIODE 2014-2018). *Kompartemen : Jurnal Ilmiah Akuntansi*, *19*(2). https://doi.org/10.30595/kompartemen.v19i2.8627

Serhalawan, R. P. (2023). Penerapan Metode Hot-Fit Dalam Mengevaluasi Sistem Informasi Manajemen Rumah Sakit (Literature Review). *Comserva Jurnal Penelitian Dan Pengabdian Masyarakat*. https://doi.org/10.59141/comserva.v3i08.1106

Shah, J., Kama, N., & Abu Bakar, N. A. (2018). A Novel Effort Estimation Model for Software Requirement Changes During Software Development Phase. *International Journal of Software Engineering & Applications*. https://doi.org/10.5121/ijsea.2018.9602

Shahrivari, S. (2014). Beyond Batch Processing: Towards Real-Time and Streaming Big Data. *Computers*. https://doi.org/10.3390/computers3040117

Shi, P.-P., & Li, L. (2017). Design of Network Analysis System Based on Stream Computing. *Journal of Computational and Theoretical Nanoscience*. https://doi.org/10.1166/jctn.2017.6125

Sihotang, E. T., & Yutanto, H. (2021). Tata Kelola Organisasi Mahasiswa Melalui Pengembangan  Sistem Informasi. *Matrik Jurnal Manajemen Teknik Informatika Dan Rekayasa Komputer*. https://doi.org/10.30812/matrik.v21i1.1391

*Software Practitioners Challenges in the Requirement Engineering Phase of Software Development*. (2019). https://doi.org/10.17758/eirai5.f0619103

Sousa, T. de J., Kelvin, L., & Neto, C. D. (2017). A Formal Semantics for Use Case Diagram Via Event-B. *Journal of Software (2017)*, *12*(4), 189–200.

Spohn, M. A. (2022). On MQTT Scalability in the Internet of Things: Issues, Solutions, and Future Directions. *Journal of Electronics and Electrical Engineering*. https://doi.org/10.37256/jeee.1120221687

Supriyono, S. (2020). Evaluasi Sistem Informasi Manajemen Rumah Sakit Dengan Metode Hot Fit Di Rumah Sakit Umum Daerah Raden Mattaher Jambi. *Journal of Information Systems for Public Health*. https://doi.org/10.22146/jisph.17142

Suryawan, M. B., & Prihandoko, P. (2018). Evaluasi Penerapan SIAKAD Politeknik Negeri Madiun Menggunakan Pendekatan TAM Dan EUCS. *Creative Information Technology Journal*. https://doi.org/10.24076/citec.2017v4i3.113

Switzky, A. (2012). *Incorporating UCD Into the Software Development Lifecycle*. https://doi.org/10.1145/2212776.2212823

Syarif Hidayatullah, D. A. (2023). Rancang Bangun Sistem Informasi Desa Berbasis Website Menggunakan Metode Scrum. *Jurnal Teknologi Sistem Informasi*. https://doi.org/10.35957/jtsi.v4i2.5313

Tri Putra, M. R. (2019). *Sistem Informasi Manajemen Di Bidang Teknologi Dan Kesehatan*. https://doi.org/10.31219/osf.io/7qdtj

Tumara, D. (2015). *Metodologija izračuna emisije ugljičnog dioksida* [Technical Sciences, University of Zagreb]. https://repozitorij.fkit.unizg.hr/islandora/object/fkit:213

Tzafestas, S. G. (2018). Synergy of IoT and AI in Modern Society: The Robotics and Automation Case. *Robotics & Automation Engineering Journal*. https://doi.org/10.19080/raej.2018.03.555621

Ufresti Praptyanti, D. A., Barchi, M. F., Utama, S. P., Suharyanto, S., & Yansen, Y. (2022a). Strategi Mitigasi Emisi Gas Rumah Kaca (Grk) Sektor Persampahan Di Kabupaten Lebong (Studi Kasus Kecamatan Amen). *Naturalis Jurnal Penelitian Pengelolaan Sumber Daya Alam Dan Lingkungan*. https://doi.org/10.31186/naturalis.11.1.20828

Ufresti Praptyanti, D. A., Barchi, M. F., Utama, S. P., Suharyanto, S., & Yansen, Y. (2022b). Strategi Mitigasi Emisi Gas Rumah Kaca (Grk) Sektor Persampahan Di Kabupaten Lebong (Studi Kasus Kecamatan Amen). *Naturalis Jurnal Penelitian Pengelolaan Sumber Daya Alam Dan Lingkungan*. https://doi.org/10.31186/naturalis.11.1.20828

Ullah, M., J. Nardelli, P. H., Wolff, A., & Smolander, K. (2020). Twenty-One Key Factors to Choose an IoT Platform: Theoretical Framework and Its Applications. *Ieee Internet of Things Journal*. https://doi.org/10.1109/jiot.2020.3000056

Valles-Barajas, F. (2011). A survey of UML applications in mechatronic systems. *Innovations in Systems and Software Engineering*, *7*(1). https://doi.org/10.1007/s11334-011-0143-6

Verborgh, R., Steiner, T., Deursen, D. Van, Coppens, S., Vallès, J. G., & de Walle, R. Van. (2012). *Functional Descriptions as the Bridge Between Hypermedia APIs and the Semantic Web*. https://doi.org/10.1145/2307819.2307828

Victor Pakpahan, A., Cory Sakti Triwangsa, M., & Fakhrurroja, H. (2023). Analisis Keamanan Protokol Komunikasi Message Queuing Telemetry Transport (Studi Kasus Smart Greenhouse). *Smart Comp: Jurnalnya Orang Pintar Komputer*, *12*(4). https://doi.org/10.30591/smartcomp.v12i4.4681

Wardana, A. A., Rakhmatsyah, A., Minarno, A. E., & Anbiya, D. R. (2019). Internet of Things Platform for Manage Multiple Message Queuing Telemetry Transport Broker Server. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*. https://doi.org/10.22219/kinetik.v4i3.841

Widiarta, I. M. (2024). Rancang Bangun Sistem Informasi Manajemen Organisasi We SAVE Indonesia Terintegrasi Berbasis Web. *Digital Transformation Technology*. https://doi.org/10.47709/digitech.v3i2.3426

Wilbanks, B. A., & Langford, P. A. (2014). A Review of Dashboards for Data Analytics in Nursing. *Cin Computers Informatics Nursing*. https://doi.org/10.1097/cin.0000000000000106

Wulandari, S. (2020). Rancang Bangun Sistem Informasi Budgeting Ormawa (Organisasi Mahasiswa) Universitas Muhammadiyah Semarang Berbasis Web. *Media Elektrika*. https://doi.org/10.26714/me.13.1.2020.43-53

Xuan Phan, L. T., Zhang, Z., Zheng, Q., Loo, B. T., & Lee, I. (2011). *An Empirical Analysis of Scheduling Techniques for Real-Time Cloud-Based Data Processing*. https://doi.org/10.1109/soca.2011.6166240

Yang, J., Wittern, E., T. Ying, A. T., Dolby, J., & Tan, L. (2018). *Towards Extracting Web API Specifications From Documentation*. https://doi.org/10.1145/3196398.3196411

Yang, Y.-B., Ke, W., Yang, J., & Li, X. (2019). Integrating UML With Service Refinement for Requirements Modeling and Analysis. *Ieee Access*. https://doi.org/10.1109/access.2019.2892082

Yigitbas, E., Gorissen, S., Weidmann, N., & Engels, G. (2021). Collaborative Software Modeling in Virtual Reality. *Proceedings - 24th International Conference on Model-Driven Engineering Languages and Systems, MODELS 2021*. https://doi.org/10.1109/MODELS50736.2021.00034

Yousefpour, A., Patil, A., Ishigaki, G., Kim, I., Wang, X., Cankaya, H. C., Zhang, Q., Xie, W., & Jue, J. P. (2019). FOGPLAN: A Lightweight QoS-Aware Dynamic Fog Service Provisioning Framework. *Ieee Internet of Things Journal*. https://doi.org/10.1109/jiot.2019.2896311

Zaitsev, S. (2023). Optimizing SME Performance Through KPI Utilization. *Journal of Innovations and Sustainability*. https://doi.org/10.51599/is.2023.07.04.09

Zhang, H., Zhang, H., Wang, Z., Zhou, Z., Wang, Q., Xu, G., Yang, J., & Gan, Z. (2022). Delay-reliability-aware protocol adaption and quality of service guarantee for message queuing telemetry transport-empowered electric Internet of things. *International Journal of Distributed Sensor Networks*, *18*(5). https://doi.org/10.1177/15501329221097815

Ziadi, T., Almeida Silva, M. A., Hillah, L. M., & Ziane, M. (2011). *A Fully Dynamic Approach to the Reverse Engineering of UML Sequence Diagrams*. https://doi.org/10.1109/iceccs.2011.18

Zoughbi, G., Briand, L., & Labiche, Y. (2011). Modeling safety and airworthiness (RTCA DO-178B) information: Conceptual model and UML profile. *Software and Systems Modeling*, *10*(3). https://doi.org/10.1007/s10270-010-0164-x

**LAMPIRAN A – Daftar Pertanyaan Wawancara**

**[Beri nama lampiran anda**

**Contoh : Data *Demand*]**