PENGEMBANGAN SISTEM KONFIGURASI MODEL PROSES BISNIS BERBASIS WEBSITE MENGGUNAKAN *OPENROUTER* API

SKRIPSI

Oleh :
AIRLANGGA WISNU ZUIZZA
NIM. 200605110131

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026

PENGEMBANGAN SISTEM KONFIGURASI MODEL PROSES BISNIS BERBASIS WEBSITE MENGGUNAKAN *OPENROUTER* API

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
AIRLANGGA WISNU ZUIZZA
NIM. 200605110131

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026

MOTTO

... Sebaik-baik manusia adalah manusia yang bermanfaat bagi manusia lain ...

HALAMAN PERSEMBAHAN

Saya persembahkan karya ini kepada: kedua orang tua saya, bapak Abdul Khoris Yudin, beliau adalah orang yang selalu mendoakan saya dan mendukung saya hingga bisa sampai dititik ini.

Ibu Zuliatin, beliau adalah seseorang yang selalu mengingatkan saya bagaimana cara menghadapi kehidupan, yang menyebut nama saya pada tiap doanya, beliau jauh lebih tinggi dari aneka macam surga. Terimakasih atas segala doa, dukungan dan semangat bagi saya sampai pada titik ini.

Mufida Naima Rohmah, orang yang selalu mendukung dan membantu saya, yang selalu ada ketika saya dalam kondisi susah dan senang saat menyelesaikan karya ini. Terimakasih atas segala doa, dukungan dan semangat bagi saya.

Sahabat-sahabat saya, Osamma Abdurrahman Amein, Ilhammudin, Fitrah Tirto Widodo, Rifki Aliansyah, Shofey M. Yusuef Ali, Zaky Fathurrahman, Kholid Fahrudin, Rokhim Nur Rifa'i, dan yang lainnya yang tidak bisa saya sebut satu persatu karena banyak sekali yang membantu dan mensupport saya untuk menyelesaikan karya ini, terima kasih semuanya, doa baik akan selalu menyertai kalian.

Terima kasih telah berbagi suka, duka, cerita, cinta dan pengalaman selama masa perkuliahan. Semoga kita semua selalu dikelilingi kabar baik, pertolongan, dan kemudahan oleh Allah SWT.

KATA PENGANTAR

Bismillahirrohmanirrohim

Alhamdulillahi Robbil 'Aalamiin. Segala puji hanya milik Allah Subhanahu Wa Ta'ala atas segala nikmat dan kasih sayang-Nya yang telah memudahkan penulis untuk menyelesaikan skripsi yang berjudul "Pengembangan Sistem Konfigurasi Model Proses Bisnis Berbasis Website Menggunakan *OpenRouter* API". Semoga shalawat dan salam senantiasa terlimpah kepada baginda kita, Nabi Muhammad *Sallalahu 'Alaihi wa Sallam*. Dan semoga kita semua mendapat syafaatnya di hari kiamat nanti, Aamiin.

Penulis mengucapkan rasa syukur dan terima kasih yang tak terhingga kepada semua pihak-pihak yang selalu memberikan bantuan dan motivasi kepada penulis untuk dapat menyelesaikan skripsi ini. Ucapan ini penulis sampaikan kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP. selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Supriyono, M.Kom, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. M. Ainul Yaqin, M.Kom dan Dr. Ir. Supriyono, M.Kom selaku dosen pembimbing yang dengan penuh kesabaran dan kebijaksanaan telah membimbing saya dalam menyelesaikan skripsi ini. Bimbingan, nasihat, dan

pengarahan yang diberikan sangat berarti bagi kemajuan dan kelancaran penelitian ini.

5. Fajar Rohman Hariri, M.Kom_dan Ashri Shabrinah Afrah, M.T selaku dosen penguji yang dengan sabar telah memberi arahan dan saran dalam proses penelitian ini.
6. Segenap civitas akademika Program Studi Teknik Informatika, terutama seluruh dosen. Terimakasih atas ilmu dan bimbingan yang telah diberikan selama masa perkuliahan ini.
7. Semua pihak yang telah membantu penulis dalam proses perkuliahan hingga menyelesaikan skripsi ini, baik secara langsung maupun tidak langsung yang tidak dapat disebutkan satu per satu.

Akhir kata, penulis menyadari bahwa penulisan pada skripsi ini masih jauh dari kata sempurna. Saya berdoa semoga skripsi ini dapat diterima sebagai amal ibadah yang ikhlas dan bermanfaat disisi Allah SWT. Semoga karya ini menjadi bentuk kontribusi dalam rangka memperkokoh dan mengembangkan ilmu pengetahuan dan menjalankan tugas sebagai hamba Allah SWT yang bertanggung jawab.

Malang, 16 November 2026

Airlangga Wisnu Zuizza

DAFTAR ISI

HALAMAN PERSETUJUAN iii
HALAMAN PENGESAHAN iv
PERNYATAAN KEASLIAN TULISAN v
MOTTO vi
HALAMAN PERSEMBAHAN vii
KATA PENGANTAR viii
DAFTAR ISI x
DAFTAR GAMBAR xiii
DAFTAR TABEL xiv
ABSTRAK xv
ABSTRACT xvi
البحث مستخلص xvii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 7
1.3 Batasan Masalah 7
1.4 Tujuan Penelitian 8
1.5 Manfaat Penelitian 8
1.6 Sistematika Penulisan 9
BAB II STUDI PUSTAKA 11
2.1 Penelitian Terkait 11
2.2 *Business Process Model* (BPM) 16
2.2.1 *Business Process Model and Notation* (BPMN) 16
2.2.2 XML *Process Definition Language* (XPDL) 19
2.3 *Artificial Intelligence* (AI) dan *Large Language Models* (LLM) 21
2.3.1 Implementasi LLM untuk Pemrosesan Teks ke Data Terstruktur 22
2.3.2 *OpenRouter* API sebagai Platform *Agregator* LLM 23
2.3.3 *Gpt-4o-mini* 24
2.3.4 *Gemini 3.5 Flash Lite* 27
2.3.5 *Claude Haiku 4.5* 29

2.4 Website 32
2.5 Teknik *Prompt Engineering* 33
2.5.1 Jenis-Jenis Teknik *Prompting* 34
2.6 Bizagi Modeler 42
2.7 *Sentence Complexity* 43
2.7.1 *Word* Per *Sentence* 46
BAB III METODOLOGI PENELITIAN 49
3.1 Data 49
3.1.1 Data Masukan (*Textual Requirements)* 49
3.1.2 Data Acauan Pembanding (*Ground Truth)* 53
3.1.3 Pengukuran Kompleksitas Kalimat *Textual Requirements* 53
3.2 Tahap Penlitian 54
3.2.1 *Input Textual Requirements* 55
3.2.2 Memilih Model AI dan Teknik *Prompting* 55
3.2.3 *Generate* XPDL 56
3.2.4 *File* XPDL 56
3.2.5 Diagram BPMN 56
3.2.6 Evaluasi Hasil 57
3.3 Desain Eksperimen 57
3.4 Perancangan *Use Case* Diagram 59
3.5 Penjelasan Fitur Website 61
BAB IV HASIL DAN PEMBAHASAN 63
4.1 Hasil 63
4.1.1 Hasil *Generate* XPDL 63
4.1.2 Hasil *Generate* BPMN 66
4.1.2 Hasil Validitas *Generate File* XPDL 66
4.1.4 Hasil Perbandingan BPMN dengan Metode *Jaccard Similarity* 68
4.1.5 Perhitungan Standar Deviasi untuk Skor *Jaccard Similarity* 73
4.2 Pembahasan 77
4.2.1 Pengaruh Teknik *Prompting* 78
4.2.2 Pengaruh Model AI 79
4.2.3 Pengaruh Struktur Kalimat 80

4.2.4 Perbandingan BPMN dengan Metode *Jaccard Similarity* 81
4.2.5 *File* XPDL Invalid 84
4.3 Keterkaitan Integrasi Islam 85
BAB V PENUTUP 89
5.1 Kesimpulan 89
5.2 Saran 90
DAFTAR PUSTAKA 93
LAMPIRAN-LAMPIRAN 96
Lampiran 1 96
Lampiran 2 99

DAFTAR GAMBAR

Gambar 3.1 Tahap Penelitian 54
Gambar 3.2 *Use Case* Diagram 59
Gambar 4.1 *Generate File* XPDL *Valid* 64
Gambar 4.2 *File* XPDL yang sudah diunduh 65
Gambar 4.3 *Generate File* XPDL *Invalid* 66
Gambar 4.4 Model BPMN A (*Ground Truth*) 68
Gambar 4.5 Model BPMN B (Hasil *Import* dari Bizagi) 69

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait 14
Tabel 2.2 Notasi BPMN 18
Tabel 3.1 Data Textual Requirements 49
Tabel 3.2 Hasil Pengukuran Kompleksitas Kalimat 54
Tabel 4.1 Validitas *Generate File* XPDL 67
Tabel 4.2 Rincian Himpunan Data *Jaccard Similarity* 70
Tabel 4.3 Standar Deviasi Teknik *Chain Of Thought* 74
Tabel 4.4 Standar Deviasi Teknik *Few Shot* 74
Tabel 4.5 Standar Deviasi Teknik *Zero Shot* 75
Tabel 4.6 Standar Deviasi Teknik *Tree Of Thoughts* 75

ABSTRAK

Zuizza, Airlangga Wisnu. 2026. **Pengembangan Sistem Konfigurasi Model Proses Bisnis Berbasis Website Menggunakan *OpenRouter* API.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. M. Ainul Yaqin, M.Kom (II) Dr. Ir. Supriyono., S.Kom., M.Kom.

.

Kata kunci: Model Proses Bisnis, *XPDL, Large Language Model, OpenRouter API, prompt engineering, Jaccard Similarity*

Konfigurasi model proses bisnis secara manual masih menjadi tugas yang rumit dan memakan waktu karena tingginya ketergantungan antaraktivitas serta kompleksitas kebutuhan pemangku kepentingan. *Large Language Model* (LLM) menawarkan potensi otomasi konversi persyaratan tekstual (*textual requirement*) menjadi berkas model proses bisnis berformat XML *Process Definition Language* (XPDL), namun tingkat keberhasilan dan akurasinya masih perlu diuji secara sistematis. Penelitian ini bertujuan membangun sistem berbasis website yang mengonversi *textual requirement* menjadi *file* XPDL menggunakan *OpenRouter* API, serta menganalisis pengaruh empat teknik prompt engineering (*Zero-Shot, Few-Shot, Chain of Thought,* dan *Tree of Thought*) *file* XPDL yang dihasilkan oleh tiga model AI (*GPT-4o-mini, Claude Haiku 4.5, dan Gemini 3.5 Flash Lite*). Sistem diuji menggunakan 10 *textual requirement* , kemudian dibandingkan dengan *Ground Truth* yang ada di penelitian tersebut melalui perhitungan *Jaccard Similarity*. Hasil pengujian validitas menunjukkan *GPT-4o-mini* mencatatkan tingkat keberhasilan tertinggi dengan 38 *file* XPDL valid, *Gemini 3.5 Flash Lite* bervariasi tergantung kompleksitas kalimat, sedangkan *Claude Haiku 4.5* gagal pada seluruh skenario karena menyertakan teks naratif di luar struktur XML murni. Teknik *Chain of Thought* dan *Tree of Thought* terbukti lebih stabil dibandingkan *Zero-Shot* dalam menjaga validitas maupun akurasi struktur, dengan skor *Jaccard Similarity* tertinggi mencapai 1,00 pada dataset sederhana dan 0,92 pada dataset paling kompleks (kombinasi *Gemini 3.5 Flash Lite* dan *Tree of Thought*). Temuan ini mengindikasikan bahwa pemilihan model AI dan teknik *Prompting* mempengaruhi keandalan otomasi konversi teks menjadi model proses bisnis, sehingga dapat menjadi acuan pengembangan sistem BPM berbasis AI yang lebih andal.

ABSTRACT

Zuizza, Airlangga Wisnu. 2026. **Development of a Web-Based Business Process Model Configuration System Using OpenRouter API**. Undergraduate Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Dr. M. Ainul Yaqin, M.Kom (II) Dr. Ir. Supriyono., S.Kom., M.Kom.

Keywords: Business Process Model, XPDL, Large Language Model, OpenRouter API, Prompt Engineering, Jaccard Similarity

Manual configuration of business process models remains a complex and time-consuming task due to high interdependencies among activities and the complexity of stakeholder requirements. Large Language Models (LLMs) offer the potential to automate the conversion of textual requirements into business process model files formatted in XML Process Definition Language (XPDL); however, their success rate and accuracy still require systematic evaluation. This study aims to develop a web-based system that converts textual requirements into XPDL files using the OpenRouter API, as well as to analyze the impact of four prompt engineering techniques (Zero-Shot, Few-Shot, Chain of Thought, and Tree of Thought) on XPDL files generated by three AI models (GPT-4o-mini, Claude Haiku 4.5, and Gemini 3.5 Flash Lite). The system was evaluated using 10 textual requirements and subsequently compared against the study's Ground Truth through Jaccard Similarity calculations. Validity testing results show that GPT-4o-mini achieved the highest success rate with 38 valid XPDL files, Gemini 3.5 Flash Lite varied depending on sentence complexity, whereas Claude Haiku 4.5 failed across all scenarios due to including narrative text outside the pure XML structure. Chain of Thought and Tree of Thought techniques proved more stable than Zero-Shot in maintaining structural validity and accuracy, with the highest Jaccard Similarity scores reaching 1.00 on simple datasets and 0.92 on the most complex dataset (achieved by combining Gemini 3.5 Flash Lite and Tree of Thought). These findings indicate that the selection of AI models and prompting techniques significantly influences the reliability of automated text-to-business-process-model conversion, serving as a reference for developing more reliable AI-based Business Process Management (BPM) systems.

البحث مستخلص

زويزا، أيرلانغا ويسنو. ٢٠٢٦. **تطوير نظام تكوين نماذج العمليات التجارية القائم على الويب باستخدام واجهة برمجة تطبيقات *OpenRouter* API**. بحث جامعي (أطروحة بكالوريوس). قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفان: (١) الدكتور محمد عين اليقين، الماجستير في الحاسوب، (٢) الدكتور المهندس سوبريونو، بكالوريوس الحاسوب، الماجستير في الحاسوب.

الكلمات المفتاحية: نموذج العمليات التجارية، XPDL، النماذج اللغوية الكبيرة (*Large Language Model*)، واجهة برمجة تطبيقات *OpenRouter* API، هندسة الأوامر (*Prompt Engineering*)، تشابه جاكارد (*Jaccard Similarity*).

لا يزال تكوين نماذج العمليات التجارية يدويًا مهمةً معقدةً وتستغرق وقتًا طويلاً نظرًا لارتفاع مستوى التبعية بين الأنشطة وتعقيد متطلبات أصحاب المصلحة. وتوفر النماذج اللغوية الكبيرة (LLM) إمكانية أتمتة تحويل المتطلبات النصية (*Textual Requirements*) إلى ملفات نماذج عمليات تجارية بصيغة لغة تعريف العمليات (XPDL) XML، إلا أن معدل نجاحها ودقتها لا يزالان بحاجة إلى اختبار منهجي. يهدف هذا البحث إلى بناء نظام قائم على الويب يقوم بتحويل المتطلبات النصية إلى ملفات XPDL باستخدام واجهة برمجة تطبيقات *OpenRouter* API، بالإضافة إلى تحليل تأثير أربع تقنيات لهندسة الأوامر (*Zero-Shot*، و*Few-Shot*، و*Chain of Thought*، و*Tree of Thought*) على ملفات XPDL التي تُنتجها ثلاثة نماذج للذكاء الاصطناعي (*GPT-4o-mini*، و*Claude Haiku 4.5*، و*Gemini 3.5 Flash Lite*). تم اختبار النظام باستخدام ١٠ متطلبات نصية مستمدة من دراسة، ثم قُورنت بالبيانات المرجعية الحقيقية (*Ground Truth*) الواردة في تلك الدراسة من خلال حساب معامل تشابه جاكارد (*Jaccard Similarity*). أظهرت نتائج اختبار الصلاحية أن نموذج *GPT-4o-mini* سجل أعلى معدل نجاح بواقع ٣٨ ملف XPDL صالحًا، بينما تباين أداء نموذج *Gemini 3.5 Flash Lite* تبعًا لتعقيد الجمل، في حين فشل نموذج *Claude Haiku 4.5*. في جميع السيناريوهات لتضمينه نصوصًا سردية خارج بنية XML الخالصة. وأثبتت تقنيتا *Chain of Thought* و*Tree of Thought* استقرارًا أكبر مقارنة بتقنية *Zero-Shot* في الحفاظ على صلاحية البنية ودقتها، حيث بلغ أعلى معدل لتشابه جاكارد ١.٠٠ في مجموعة البيانات البسيطة و٠.٩٢ في مجموعة البيانات الأكثر تعقيدًا (وذلك عبر الدمج بين نموذج *Gemini 3.5 Flash Lite* وتقنية *Tree of Thought*). وتشير هذه النتائج إلى أن اختيار نموذج الذكاء الاصطناعي وتقنية هندسة الأوامر يؤثران بشكل مباشر على موثوقية أتمتة تحويل النصوص إلى نماذج عمليات تجارية، مما يجعلها مرجعًا لتطوير أنظمة إدارة العمليات التجارية (BPM) القائمة على الذكاء الاصطناعي بشكل أكثر موثوقية.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Organisasi dituntut untuk meningkatkan efektivitas dan efisiensi operasional agar mampu beradaptasi terhadap perkembangan teknologi dan dinamika lingkungan bisnis. Salah satu pendekatan yang dapat diterapkan untuk mendukung pencapaian tujuan tersebut adalah Manajemen Proses Bisnis (*Business Process Management* atau BPM). BPM merupakan pendekatan sistematis yang digunakan untuk mengidentifikasi, mendokumentasikan, menganalisis, dan meningkatkan proses bisnis dalam suatu organisasi. Proses bisnis dapat direpresentasikan melalui model proses bisnis, yaitu representasi aktivitas dan hubungan antarkegiatan yang dilakukan untuk mencapai tujuan tertentu. Salah satu notasi yang digunakan dalam pemodelan proses bisnis adalah *Business Process Model and Notation* (BPMN), yang memungkinkan proses bisnis direpresentasikan secara visual melalui elemen aktivitas, alur urutan, dan keputusan (Aqimuddin et al, 2023). Pemodelan tersebut dapat membantu organisasi memahami pelaksanaan proses bisnis serta mengidentifikasi bagian yang memerlukan perbaikan.

Model proses bisnis yang telah disusun perlu disesuaikan dengan kebutuhan dan kondisi operasional organisasi. Penyesuaian tersebut dapat dilakukan ketika terdapat perubahan kebijakan, kebutuhan pemangku kepentingan, ketentuan regulasi, maupun lingkungan bisnis. Dalam konteks ini, konfigurasi model proses bisnis berkaitan dengan penyesuaian elemen dan alur proses agar model yang digunakan sesuai dengan kebutuhan yang telah ditentukan. Namun, penyusunan

dan penyesuaian model proses bisnis dapat menghadapi kendala ketika kebutuhan yang diberikan melibatkan banyak aktivitas, hubungan antarkegiatan, percabangan keputusan, dan pengulangan proses. Kebutuhan tersebut harus diterjemahkan menjadi representasi model yang memiliki struktur dan alur logis. Proses penerjemahan ini memerlukan ketelitian agar elemen yang dibentuk sesuai dengan kebutuhan yang dinyatakan dalam deskripsi tekstual.

Perkembangan kecerdasan buatan (*Artificial Intelligence* atau AI) memberikan peluang untuk mengotomatisasi sebagian tahapan pemodelan proses bisnis. (Gomes et al, 2022), melalui tinjauan literatur sistematis, mengkaji penerapan metode berbasis AI dalam pengelolaan proses bisnis. Selanjutnya, (Weinzierl et al, 2024) meninjau penerapan machine learning pada berbagai tahapan siklus hidup BPM. Penelitian-penelitian tersebut menunjukkan bahwa AI berpotensi mendukung otomatisasi berbagai aktivitas dalam pengelolaan proses bisnis. Salah satu penerapan yang relevan adalah pemanfaatan model bahasa besar *(Large Language Models* atau LLM) untuk mengolah deskripsi tekstual menjadi representasi model proses bisnis. Meskipun demikian, penerapan teknologi tersebut tetap memerlukan pengujian untuk mengetahui kesesuaian hasil yang dihasilkan dengan kebutuhan dan struktur model yang diharapkan.

Penelitian mengenai pembangkitan model proses bisnis dari deskripsi tekstual telah dilakukan dengan berbagai pendekatan. (Busch et al, 2023) membahas penerapan prompt engineering dalam BPM untuk mendukung pelaksanaan tugas pengelolaan proses bisnis menggunakan model bahasa besar. Penelitian (Wenger et al, 2025) juga mengkaji penggunaan LLM untuk membantu

pembangkitan model BPMN melalui antarmuka yang dapat digunakan oleh pengguna. Penelitian-penelitian tersebut menunjukkan adanya peluang pemanfaatan LLM dalam mendukung pemodelan proses bisnis. Namun, kemampuan menghasilkan model secara otomatis tidak dengan sendirinya menjamin bahwa seluruh elemen, hubungan, dan struktur model telah sesuai dengan kebutuhan. Oleh karena itu, kualitas keluaran perlu dievaluasi menggunakan pendekatan yang sesuai dengan karakteristik model yang dihasilkan.

Permasalahan tersebut menjadi dasar pentingnya pengembangan sistem yang tidak hanya memfasilitasi pembangkitan model proses bisnis dari deskripsi tekstual, tetapi juga memungkinkan hasilnya dievaluasi secara terukur. Dalam proses pembangkitan otomatis, keluaran yang tidak sesuai dengan struktur yang diharapkan dapat memengaruhi pemanfaatan model pada perangkat pemodelan proses bisnis. Ketidaksesuaian tersebut dapat berupa elemen yang tidak lengkap, hubungan antarelemen yang tidak tepat, maupun struktur berkas yang tidak sesuai dengan format yang diperlukan. Dengan demikian, diperlukan penelitian yang mengkaji penerapan teknologi AI dalam pembangkitan model proses bisnis sekaligus mengevaluasi kesesuaian hasilnya terhadap model acuan.

Salah satu teknologi yang dapat digunakan untuk mendukung pengembangan sistem tersebut adalah *Application Programming Interface* (API). API memungkinkan aplikasi berkomunikasi dengan layanan perangkat lunak lain melalui antarmuka yang telah ditentukan. *OpenRouter* API menyediakan akses ke berbagai model AI melalui satu antarmuka, sehingga pengembang dapat memanfaatkan model yang tersedia tanpa harus membangun model bahasa besar

secara mandiri. Pemanfaatan *OpenRouter* API dalam sistem berbasis website memungkinkan proses pengolahan kebutuhan tekstual dan pembangkitan keluaran diintegrasikan dalam satu aplikasi. Melalui pendekatan tersebut, pengguna dapat memasukkan kebutuhan proses bisnis dan memperoleh hasil pembangkitan dalam format berkas yang dapat digunakan untuk pemodelan lebih lanjut.

Selain pemilihan model AI, teknik penyusunan instruksi atau prompting merupakan aspek yang perlu dikaji dalam pembangkitan model proses bisnis. *Teknik Zero-Shot, Few-Shot, Chain-of-Thought* (CoT), dan *Tree-of-Thoughts* (ToT) memiliki pendekatan yang berbeda dalam memberikan instruksi kepada model. Perbedaan tersebut memungkinkan model menerima konteks, contoh, atau arahan pemrosesan yang berbeda ketika menghasilkan keluaran. Dalam penelitian (Busch et al, 2023) membahas relevansi prompt engineering dalam penerapan LLM pada BPM. Meskipun demikian, efektivitas setiap teknik prompting perlu diuji dalam konteks tugas yang diteliti karena hasilnya dapat dipengaruhi oleh model AI, karakteristik masukan, dan struktur keluaran yang diharapkan.

Berdasarkan uraian tersebut, penelitian ini berfokus pada pengembangan sistem konfigurasi model proses bisnis berbasis website dengan memanfaatkan *OpenRouter* API. Sistem menggunakan sepuluh *textual requirement* yang bersumber dari penelitian (Sholiq et al, 2022) sebagai data pengujian. Dalam proses pembangkitan model, diterapkan empat teknik prompting, yaitu *Zero-Shot, Few-Shot, Chain-of-Thought* (CoT), dan *Tree-of-Thoughts* (ToT). Hasil pembangkitan berupa berkas *XML Process Definition Language* (XPDL), yang selanjutnya dibandingkan dengan model acuan yang dibuat menggunakan Bizagi Modeler.

Penggunaan model acuan tersebut bertujuan menyediakan pembanding untuk mengevaluasi kesesuaian elemen model yang dihasilkan oleh sistem.

Evaluasi hasil pembangkitan dilakukan menggunakan *Jaccard Similarity* berdasarkan himpunan elemen model yang meliputi aktivitas, *gateway*, dan *sequence flow*. Pengukuran ini digunakan untuk mengetahui tingkat kemiripan elemen antara model hasil generasi dan model acuan. Penggunaan metrik tersebut memberikan dasar kuantitatif untuk membandingkan hasil pembangkitan berdasarkan teknik *prompting* yang diterapkan. Namun, *Jaccard Similarity* berbasis himpunan elemen sehingga tidak secara langsung mengukur kesesuaian urutan aktivitas maupun keseluruhan topologi model. Oleh karena itu, hasil pengukuran diinterpretasikan sebagai kemiripan berdasarkan elemen yang dianalisis, bukan sebagai ukuran menyeluruh atas seluruh aspek kualitas model proses bisnis.

Urgensi penelitian ini didasarkan pada kebutuhan untuk meningkatkan efektivitas pembangkitan model proses bisnis dari deskripsi tekstual sekaligus memastikan kualitas model yang dihasilkan. (Kourani et al, 2026) menunjukkan bahwa kemampuan berbagai Large Language Model (LLM) dalam pemodelan proses bisnis memiliki perbedaan kinerja yang signifikan, sehingga diperlukan pendekatan evaluasi yang sistematis untuk menilai kualitas model yang dihasilkan. Selain itu, penelitian (Licardo et al, 2024) menunjukkan bahwa pembangkitan model proses bisnis dari deskripsi tekstual masih dapat menghadapi ketidakakuratan keluaran dan dipengaruhi oleh kualitas teks masukan. Temuan tersebut menunjukkan bahwa penerapan AI dalam pemodelan proses bisnis tidak cukup hanya berorientasi pada kemampuan menghasilkan model, tetapi juga perlu

memperhatikan kesesuaian hasil terhadap kebutuhan proses bisnis yang diberikan. Oleh karena itu, penelitian ini mengembangkan sistem konfigurasi model proses bisnis berbasis website dengan memanfaatkan *OpenRouter* API untuk menghasilkan model dalam format XPDL serta membandingkan hasil pembangkitan menggunakan teknik *prompting Zero-Shot*, *Few-Shot*, *Chain-of-Thought* (CoT), dan *Tree-of-Thoughts* (ToT). Evaluasi dilakukan menggunakan Jaccard Similarity dengan membandingkan elemen model yang dihasilkan terhadap model referensi. Dengan demikian, penelitian ini diharapkan dapat memberikan gambaran empiris mengenai kinerja kombinasi model AI dan teknik prompting dalam menghasilkan model proses bisnis berdasarkan kebutuhan tekstual. Hal tersebut dapat memberikan kontribusi positif terhadap kemampuan perusahaan dalam mengadaptasi dan mengelola perubahan dalam lingkungan bisnis yang dinamis seperti dalam Al-Quran yang diperintahkan untuk saling tolong menolong sesama manusia, lebih tepatnya dalam surah Al-Maidah ayat 2 yang berbunyi :

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْا لَا تُحِلُّوْا شَعَاۤىِٕرَ اللّٰهِ وَلَا الشَّهْرَ الْحَرَامَ وَلَا الْهَدْيَ وَلَا الْقَلَاۤىِٕدَ وَلَآ اٰۤمِّيْنَ الْبَيْتَ الْحَرَامَ يَبْتَغُوْنَ فَضْلًا مِّنْ رَّبِّهِمْ وَرِضْوَانًا ۗ وَاِذَا حَلَلْتُمْ فَاصْطَادُوْا ۗ وَلَا يَجْرِمَنَّكُمْ شَنَاٰنُ قَوْمٍ اَنْ صَدُّوْكُمْ عَنِ الْمَسْجِدِ الْحَرَامِ اَنْ تَعْتَدُوْا ۘ وَتَعَاوَنُوْا عَلَى الْبِرِّ وَالتَّقْوٰى ۖ وَلَا تَعَاوَنُوْا عَلَى الْاِثْمِ وَالْعُدْوَانِ ۚ وَاتَّقُوا اللّٰهَ ۗ اِنَّ اللّٰهَ شَدِيْدُ الْعِقَابِ ۝٢

Arti:Hai orang-orang yang beriman, janganlah kamu melanggar syi'ar-syi'ar Allah, dan jangan melanggar kehormatan bulan-bulan haram, jangan (mengganggu) binatang-binatang had-ya, dan binatang-binatang qalaa-id, dan jangan (pula) mengganggu orang-orang yang mengunjungi Baitullah sedang mereka mencari kurnia dan keridhaan dari Tuhannya dan apabila kamu telah menyelesaikan ibadah haji, maka bolehlah berburu. Dan janganlah sekali-kali kebencian(mu) kepada sesuatu kaum karena mereka menghalang-halangi kamu dari Masjidilharam, mendorongmu berbuat aniaya (kepada mereka). Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya.(Q.S. Al-Maidah: 2)

Dalam tafsir kemenag dijelaskan untuk saling tolong-menolonglah kamu dalam mengerjakan kebajikan, melakukan yang diperintahkan Allah SWT, dan takwa, takut kepada larangannya, dan jangan tolong-menolong dalam berbuat dosa, melakukan maksiat dan permusuhan, sebab yang demikian itu melanggar hukum-hukum Allah SWT. Bertakwalah kepada Allah SWT, takut kepada Allah SWT dengan melakukan perintah-Nya dan meninggalkan larangan-Nya, karena sungguh Allah SWT sangat berat siksaan-Nya kepada orang-orang yang tidak taat kepada-Nya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah :

a. Bagaimana merancang dan mengimplementasikan sistem berbasis website yang mampu mengonversi *textual requirement* menjadi *file* XPDL menggunakan *OpenRouter* API?
b. Bagaimana pengaruh penggunaan 4 jenis teknik *Prompting* yang berbeda terhadap kualitas *file* XPDL yang dihasilkan oleh model AI?
c. Bagaimana hasil perbandingan (komparasi) antara *file* XPDL yang dihasilkan oleh sistem dengan *file* XPDL acuan dari Bizagi Modeler?

1.3 Batasan Masalah

Batasan masalah ini dicantumkan agar penelitian ini tidak keluar dari pembahasan, berikut beberapa poin dari batasan masalah :

a. Data uji yang digunakan terbatas pada 10 skenario *textual requirement* berbahasa Inggris yang diadopsi dari penelitian (Sholiq et al, 2022), tanpa penambahan skenario domain bisnis lain.
b. Model AI yang diuji dibatasi pada 3 model yang tersedia melalui *OpenRouter* API pada saat penelitian dilakukan, yaitu *GPT-4o-mini, Claude Haiku 4.5,* dan *Gemini 3.5 Flash Lite*.
c. Eksperimen Prompting terbatas pada 4 jenis teknik Prompting yang telah ditentukan yaitu: *chain of thougt, few shot, zero shot,* dan *tree of thoughts*.
d. Evaluasi akhir dilakukan dengan membandingkan jumlah elemen BPMN hasil impor dari *file* XPDL hasil sistem terhadap jumlah elemen BPMN acuan yang ada dibab 3.

1.4 Tujuan Penelitian

Penelitian ini dilakukan untuk mencapai beberapa tujuan yang diinginkan yaitu :

a. Membangun sistem konfigurasi model proses bisnis berbasis website menggunakan *OpenRouter* API.
b. Menganalisis pengaruh 4 jenis teknik *prompting* terhadap performa dan struktur *file* XPDL yang dihasilkan oleh model AI.
c. Mengevaluasi tingkat akurasi dan kesesuaian *file* XPDL buatan sistem melalui perbandingan komparatif dengan Bizagi Modeler.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini, diharapkan dapat memberikan manfaat diantaranya yaitu :

a. Bagi Akademisi: Memberikan referensi serta kontribusi pengembangan ilmu pengetahuan, khususnya di bidang rekayasa perangkat lunak dan integrasi kecerdasan buatan (*Generative AI*) dengan standar pemodelan proses bisnis industri (*Business Process Management*).
b. Bagi Praktisi dan Organisasi: Memberikan solusi teknologi berbasis website yang dapat membantu pengembang maupun analis sistem dalam meningkatkan efisiensi perancangan, mempercepat konversi dokumen *textual requirement* ke dalam format XPDL, serta mengurangi kompleksitas pemodelan manual.
c. Bagi Peneliti Lain (*Developer*):Memberikan gambaran serta acuan empiris mengenai tingkat efektivitas penggunaan *OpenRouter* API dan perbandingan 4 jenis teknik *Prompting* terhadap kualitas struktur *file* XPDL, yang dapat dijadikan landasan untuk pengembangan penelitian lanjutan di masa mendatang.

1.6 Sistematika Penulisan

Dalam penelitian ini, struktur penulisan yang dipakai tercantum sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah yang dibahas dan sistematika penulisan.

BAB II STUDI PUSTAKA

Bab ini menjelaskan mengenai teori-teori yang berhubungan dengan penelitian yang dilakukan. Adapun teori yang berhubungan dengan penelitian ini meliputi *Business Process Model*, *Artificial Intelligence*, *Large Languange Model*, Website, Teknik *Prompt Engineering*, Bizagi Modeler, dan *Sentence Complexity*.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan dalam penelitian, desain penelitian, Studi kasus dan skenario pengujian sistem.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil perancangan sistem serta pembahasan dari penelitian yang dilakukan.

BAB V PENUTUP

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan , dan saran perbaikan untuk pengembangan aplikasi.

BAB II
STUDI PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian (Sholiq et al, 2022) melakukan pengembangan metode untuk menghasilkan diagram BPMN dari persyaratan tekstual untuk mengatasi keterbatasan metode sebelumnya dalam mengubah input *Natural Language* (NL) menjadi BPMN yang lebih lengkap. Metode yang digunakan dalam penelitian ini memiliki dua tahap utama: analisis persyaratan tekstual menggunakan pemrosesan bahasa alami dan pembuatan diagram BPMN berdasarkan hasil analisis tersebut. Penelitian ini dilakukan untuk menangani kalimat kompleks dan majemuk kompleks pada input NL yang termasuk kelemahan dari metode sebelumnya. Dengan menggunakan pendekatan yang terintegrasi dari berbagai macam metode NLP, definisi SBVR, dan spreadsheet diharapkan dapat menghasilkan BPMN yang lebih lengkap dan akurat. Dari penelitian yang telah dilakukan diperoleh hasil yang menunjukkan bahwa metode yang digunakan mampu menghasilkan diagram BPMN dari persyaratan tekstual dengan akurasi yang lebih tinggi dibandingkan metode lain.

Dalam penlitian (Zamhari, 2019) dilakukan untuk menyediakan panduan konfigurasi yang memungkinkan untuk karakterisasi secara formal semua konfigurasi proses yang benar. Pendekatan ini didasarkan pada sintesis mitra, yang memungkinkan untuk mengidentifikasi dan memperbaiki kesalahan dalam model proses yang dikonfigurasi. Ketika proses konfigurasi dianggap sebagai “layanan

eksternal" dan kemudian disintesis oleh mitra "paling toleran" untuk memastikan kebenaran perilaku model proses konfigurasi, penelitian ini memperkenalkan metode verifikasi baru. Metode ini memungkinkan untuk menemukan dan memperbaiki masalah perilaku seperti deadlock dan livelock dalam model proses konfigurasi. Studi ini juga membahas alat bantu yang mendukung metode ini, serta keuntungan metode baru dibandingkan metode lama.

Dalam penelitian (Rosa et al, 2006) menghasilkan beberapa poin penting. Pertama, pendekatan kuesioner digunakan untuk mencari rekomendasi model proses bisnis berdasarkan jawaban yang diberikan oleh pengguna. Kedua, untuk mengukur tingkat kemiripan antara model proses bisnis, dilakukan perhitungan menggunakan algoritma Levenshtein distance terhadap aktivitas yang terdapat dalam model tersebut.

Dalam penelitian (Zhao et al, 2026) melakukan survei komprehensif mengenai pemanfaatan *Large Language Models* (LLM) untuk menerjemahkan teks bebas menjadi data terstruktur melalui teknik instruksi. Melalui analisis pengujian sintaksis pada berbagai arsitektur LLM, penelitian ini mengukur respons model terhadap batasan instruksi yang diberikan. Hasil penelitian mengungkapkan bahwa kejelasan konteks dan batasan format pada *input prompt* sangat menentukan akurasi keluaran, di mana instruksi yang terstruktur mampu meminimalkan *syntax error* secara signifikan. Penelitian ini memberikan kontribusi teoritis bagi skripsi ini dalam merancang aturan instruksi agar LLM mampu menghasilkan berkas XPDL yang valid.

Dalam penelitian (Vidgof et al, 2023) mengkaji peluang dan tantangan penerapan *Large Language Models* dalam disiplin *Business Process Management* (BPM). Menggunakan analisis kerangka kerja *Generative AI*, penelitian ini menguraikan bagaimana LLM dapat dimanfaatkan untuk melakukan ekstraksi alur kerja, analisis proses bisnis, dan pembuatan model otomatis. Hasil penelitian menegaskan bahwa penggunaan LLM berbasis API terbukti sangat efisien untuk mempercepat tahap *process discovery*, meskipun tetap memerlukan mekanisme verifikasi terhadap skema standar agar model yang dihasilkan dapat dieksekusi. Penelitian ini menjadi rujukan utama dalam membenarkan penggunaan *OpenRouter* API sebagai jembatan integrasi model AI untuk tugas-tugas BPM.

Dalam penelitian (Setiawan, 2021), mengembangkan sistem *Data Flow Management System* (DFMS) untuk melakukan otomasi konfigurasi sistem informasi dengan mengekstraksi dan mengeksekusi aliran data dari diagram DFD. Hasil dari penelitian tersebut berhahsil memparsing komponen-komponen DFD yang disimpan dalam *file* yang berformat XML, memetakan komponen-komponen yang ada di dalam DFD dengan *web service.*

Pada penelitian (Caterina, 2020) berfokus pada pengelompokan dan pengukuran tingkat kemiripan model proses bisnis pendaftaran, input barang, serta transaksi pada *e-marketplace* Tokopedia, Bukalapak, dan Shopee untuk meminimalkan duplikasi proses. Penelitian ini mengombinasikan analisis *semantic similarity* menggunakan algoritma *Path* dengan dataset *sinonimkata.com* (berdasarkan kelas kata benda dan kata kerja) serta *structural similarity* menggunakan algoritma *Jaccard Coefficient Similarity* untuk membandingkan

elemen *task*, *connector*, dan percabangan. Melalui proses *clustering* dengan *threshold* 0,27 dan ekstraksi *common fragment*, penelitian ini berhasil menghasilkan satu model proses bisnis acuan yang umum, menyimpulkan alur *e-marketplace* Shopee sebagai yang paling mencakup keseluruhan proses, serta mencatatkan tingkat akurasi pengujian sistem sebesar 88,8%.

Tabel 2.1 Penelitian Terkait

Referensi	**Input**	**Output**	**Metode**
(Sholiq et al, 2022)	Persyaratan tekstual berupa *Natural Languange* (NL)	Model proses bisnis dengan notasi Diagram BPMN	1. Analisis persyaratan tekstual 2. Pembuatan diagram BPMN
(Rosa et al, 2006)	Kuesioner	Model proses bisnis dengan notasi EPC	Menggunakan *Questionare-Driven*
(Zamhari 2019)	Kuesioner	Diagram BPMN	Pengukuran kemiripan model proses bisnis dengan algortitam *leventhstein distance*
(Zhao et al, 2026)	Teks bebas (*free text*) dan instruksi *prompt* terstruktur	Data terstruktur yang valid (meminimalkan *syntax error*)	Survei komprehensif dan analisis pengujian sintaksis arsitektur LLM via teknik instruksi (*prompt engineering*)
(Vidgof et al, 2023)	Teks deskripsi alur kerja / data proses bisnis	Ekstraksi alur kerja dan model proses bisnis otomatis	Kerangka kerja *Generative AI* menggunakan LLM berbasis API dengan mekanisme verifikasi skema standar
(Setiawan, 2021)	Diagram DFD visual / berkas DFD berformat XML	Hasil *parsing* komponen DFD dan pemetaan alur data ke *web service*	*Data Flow Management System* (DFMS) dengan ekstraksi XML, *text preprocessing*, dan

Referensi	Input	Output	Metode
			pemetaan semantik *web service*
(Caterina, 2020)	Model proses bisnis *e-marketplace* (Tokopedia, Bukalapak, Shopee) pada alur pendaftaran, input barang, dan transaksi	Nilai tingkat kemiripan model, pengelompokan (*cluster*), serta 1 model proses bisnis acuan umum (*common fragment*)	Kombinasi *Semantic Similarity* (Algoritma Path + *sinonimkata.com*) dan *Structural Similarity* (Jaccard Coefficient Similarity) + *Clustering* & ekstraksi *Common Fragment*

Dari beberapa penelitian diatas yang sudah dijelaskan telah dihasilkan beberapa model proses bisnis dengan berbagai macam notasi. Akan tetapi masih ada kekurangan yang ditemukan yaitu pada penelitian sebelumnya harus menggunakan requirement fungsional yang terstruktur untuk mengenerate menjadi model proses bisnis. Dengan adanya penelitian ini diharapkan dapat mengenerate requirement fungsional yang tidak terstruktur dengan memanfaatkan *OpenRouter* API, serta memperlengkap penelitian sebelumnya sehingga dapat menghasilkan model proses bisnis dengan cepat dan akurat dengan menggunakan website yang memanfaatkan *OpenRouter* API.

Kebaruan (*novelty*) dari penelitian ini terletak pada integrasi penggunaan *Large Language Models* (LLM) melalui *OpenRouter* API dengan penerapan empat variasi teknik *prompt engineering* untuk mengotomatisasi konversi persyaratan tekstual (*textual requirements*) menjadi berkas berformat XPDL berstandar BPMN. Berbeda dengan penelitian sebelumnya yang mengandalkan pemrosesan NLP berbasis aturan (*rule-based*) kaku (Sholiq et al, 2022) maupun ekstraksi dari diagram DFD visual konvensional (Setiawan, 2021), penelitian ini

mentransformasikan teks bahasa alami secara langsung menjadi struktur alur proses bisnis yang dapat dieksekusi dan diimpor. Selain itu, jika studi terdahulu mengenai LLM dan BPM masih terbatas pada kerangka teoritis (Vidgof et al, 2023; Zhao et al, 2026), penelitian ini mengimplementasikan kerangka kerja tersebut ke dalam sistem berbasis web, serta melakukan validasi kuantitatif secara objektif dengan mengukur tingkat kemiripan berkas XPDL hasil sistem terhadap model acuan (*Ground Truth*) Bizagi Modeler menggunakan algoritma *Jaccard Similarity.*

2.2 *Business Process Model* (BPM)

Model proses bisnis merupakan suatu representasi dari semua fungsi yang berkaitan dengan operasi bisnis yang diwakili secara terurut. Model proses bisnis dapat digambarkan dalam berbagai bentuk yaitu UML(*Unified Modeling Language*), EPC(*Event-Driven Process Chain*), Petri Net(Sintaks XML yang menggambarkan berbagai aktivitas), dan BPMN(Yaqin et al, 2019).

2.2.1 *Business Process Model and Notation* (BPMN)

BPMN merupakan metode penggambaran diagram alur yang dikembangkan oleh BPMI dan digunakan untuk menggambarkan proses bisnis dalam notasi grafis dalam desain proses bisnis. Pelaku bisnis dapat menggunakan BPMN untuk memahami notasi model proses bisnis. Notasi ini mencakup bisnis analis untuk memperbaiki konsep dasar proses, meningkatkan pemanfaatan teknologi untuk mendukung pelaksanaan proses, dan pelaksana bisnis yang akan mengatur dan mengawasi proses(Zamhari, 2019).

Business Process Model and Notation (BPMN) adalah standar pemodelan grafis yang dikembangkan oleh *Object Management Group* (OMG) untuk

menggambarkan alur kerja dan proses bisnis secara intuitif namun tetap memiliki kejelasan akademis serta teknis (OMG, 2020). BPMN menjembatani jurang komunikasi antara analis bisnis (*business analysts*) yang merancang alur kerja dan pengembang perangkat lunak (*software developers*) yang mengimplementasikan sistem.

Secara umum, notasi BPMN dibagi ke dalam empat kategori elemen dasar (Geiger et al, 2017):

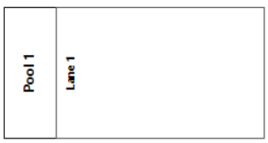
1. Objek Alur (*Flow Objects*): Elemen grafik utama yang mendefinisikan perilaku proses bisnis, terdiri dari:
   a. Peristiwa *(Events*): Menandai sesuatu yang terjadi dalam alur proses (diwakili ikon lingkaran), seperti *Start Event, Intermediate Event, dan End Event*.
   b. Aktivitas (*Activities/Tasks*): Menunjukkan pekerjaan atau tindakan yang harus dilakukan (diwakili persegi panjang bersudut tumpul).
   c. Gerbang Komputasi (*Gateways*): Mengontrol percabangan, penggabungan, dan keputusan dalam alur kerja berdasarkan kondisi tertentu (diwakili bentuk belah ketupat), seperti *Exclusive Gateway* (XOR) dan *Parallel Gateway* (AND).
2. Objek Penghubung (*Connecting Objects*): Garis notasi yang menghubungkan elemen-elemen *Flow Objects*, seperti *Sequence Flow* (menunjukkan urutan pelaksanaan aktivitas) dan *Message Flow* (menunjukkan pertukaran pesan antaraktor).
3. Jalur Pemisah (*Swimlanes*): Digunakan untuk mengelompokkan elemen proses berdasarkan subjek atau peran operasional, mencakup *pool* (mewakili

organisasi/sistem utama) dan *lane* (mewakili unit kerja atau aktor spesifik di dalam *pool*).

4. Artefak (*Artifacts*): Elemen pelengkap untuk memberikan informasi tambahan tanpa memengaruhi alur eksekusi, seperti *Data Object, Group, dan Annotation.*

Dari penjelasan diatas gambar notasi dari BPMN disajikan dalam table berikut :

Tabel 2.2 Notasi BPMN

Elemen	Gambar Notasi
Start Event	
Intermediate Event	
End Event	
Activities/Tasks	
Gateways	
Sequence Flow	
Message Flow	
Pool	
Lane	
Data Object	

Elemen	Gambar Notasi
Group	
Annotation	

2.2.2 XML *Process Definition Language* (XPDL)

Sementara BPMN berfokus pada penampilan visual diagram, *XML Process Definition Language* (XPDL) adalah format berkas berbasis *Extensible Markup Language* (XML) yang dirancang oleh *Workflow Management Coalition* (WfMC) khusus untuk menyimpan dan mempertukarkan model proses bisnis antaraplikasi yang berbeda (WfMC, 2019).

Keunggulan utama XPDL dibandingkan format XML generik lainnya adalah kemampuannya untuk menyimpan metadata grafis (tata letak/koordinat diagram) sekaligus elemen logika proses secara bersamaan (Chinosi and Trombetta, 2012). Hal ini membuat *file* XPDL yang dihasilkan oleh satu sistem (seperti generator kecerdasan buatan) dapat dibuka, dirender kembali secara visual, dan diedit di perangkat lunak pemodelan standar industri lainnya tanpa kehilangan struktur alurnya.

XML Process Definition Language (XPDL) merupakan format berbasis XML yang digunakan untuk merepresentasikan dan mempertukarkan definisi proses bisnis antarperangkat lunak pemodelan proses bisnis. XPDL mendukung penyimpanan informasi mengenai elemen proses serta representasi grafis diagram,

sehingga model proses bisnis dapat dipindahkan antarperangkat lunak dengan mempertahankan informasi yang diperlukan. Penggunaan format XPDL dalam penelitian terkini ditunjukkan oleh (Sholiq and Ainul, 2024), yang menyediakan diagram BPMN beserta berkas XPDL untuk mendukung pertukaran model proses bisnis melalui perangkat lunak terkait. Selain itu, (Niculescu, 2023) membahas XPDL sebagai salah satu format representasi proses bisnis dalam konteks otomatisasi proses menggunakan teknologi *Robotic Process Automation* (RPA). Secara struktural, XPDL memiliki elemen-elemen yang digunakan untuk mendefinisikan paket proses, aktivitas, alur kerja, partisipan, dan data yang berkaitan dengan proses bisnis. Mengacu pada spesifikasi XPDL yang diterbitkan oleh *Workflow Management Coalition* (WfMC), beberapa elemen yang digunakan dalam definisi proses bisnis meliputi:

1. <*Package*>, berfungsi sebagai wadah yang mengorganisasikan definisi proses bisnis beserta informasi pendukungnya.
2. <*WorkflowProcess*>, digunakan untuk mendefinisikan suatu proses alur kerja yang terdiri atas aktivitas dan hubungan alur proses.
3. <*ActivitySet*>, digunakan untuk mengelompokkan aktivitas beserta transisi yang menghubungkan aktivitas tersebut, misalnya dalam pemodelan bagian proses tertentu.
4. <*BlockActivity*>, merupakan jenis aktivitas yang memungkinkan sekumpulan aktivitas dalam suatu *ActivitySet* dijalankan sebagai satu kesatuan.

5. <*Participant*>, digunakan untuk mendefinisikan partisipan yang terlibat dalam proses alur kerja, seperti pihak atau entitas yang berhubungan dengan pelaksanaan aktivitas.
6. <*DataType*> dan <*DataField*>, digunakan untuk mendefinisikan tipe data dan variabel atau bidang data yang dibutuhkan dalam proses alur kerja.

Dengan demikian, XPDL dapat dimanfaatkan sebagai format pertukaran dan penyimpanan model proses bisnis dalam sistem yang dikembangkan. Dalam penelitian ini, XPDL digunakan sebagai format keluaran model proses bisnis yang dihasilkan melalui website, sehingga berkas tersebut dapat diuji dan dibandingkan dengan model referensi menggunakan perangkat lunak pemodelan proses bisnis yang mendukung format terkait.

2.3 *Artificial Intelligence* (AI) dan *Large Language Models* (LLM)

Artificial Intelligence (AI) telah mengalami pergeseran paradigma yang signifikan, dari sistem berbasis aturan (*rule-based systems*) menuju sistem generatif (*Generative AI*) berbasis pembelajaran mendalam (*deep learning*). Salah satu cabang utama AI yang melandasi pemrosesan bahasa manusia oleh komputer adalah *Natural Language Processing* (NLP). NLP bertujuan untuk memungkinkan komputer memahami, menginterpretasi, dan menghasilkan bahasa alami manusia dengan cara yang bernalar serta bermakna (Bannert et al, 2023)

Lompatan terbesar dalam domain NLP terjadi sejak diperkenalkannya arsitektur *Transformer* oleh (Vaswani, 2017). Arsitektur ini memanfaatkan mekanisme self-attention yang memungkinkan model komputer memproses seluruh urutan kata dalam suatu teks secara paralel, serta memahami konteks

semantik antarkata yang saling berjauhan secara presisi. Arsitektur *Transforme*r inilah yang menjadi fondasi utama lahirnya model pemrosesan bahasa berskala masif yang dikenal sebagai *Large Language Models* (LLM).

2.3.1 Implementasi LLM untuk Pemrosesan Teks ke Data Terstruktur

Large Language Models (LLM) adalah model AI yang dilatih menggunakan miliaran hingga triliunan parameter data teks dari berbagai domain pengetahuan (Zhao et al. 2026). Model seperti GPT-4, Claude, maupun Llama memiliki kemampuan tidak hanya untuk memahami teks bebas (*unstructured text*), melainkan juga menerjemahkan instruksi bahasa alami tersebut menjadi format data terstruktur (*structured output*) seperti JSON, kode pemrograman, maupun berkas XML/XPDL.

Dalam konteks pemodelan proses bisnis, LLM bertindak sebagai penterjemah logis yang membaca kebutuhan dokumen tekstual (*textual requirement*) untuk mengidentifikasi elemen-elemen seperti:

a. Aktor atau Peran(*Lanes/Pools*)
b. Peristiwa(*Tasks/Activites*)
c. Gerbang Komputasi (*Gateways*)
d. Alur Penghubung (*Sequence Flow*s)

Meskipun memiliki kapasitas pemrosesan yang sangat baik, sifat dasar LLM yang probabilistik dan non-deterministik membawa tantangan tersendiri (Brown et al, 2020). LLM berpotensi menghasilkan variasi jawaban yang berbeda dari instruksi yang sama, atau mengalami kesalahan sintaksis. Oleh karena itu, diperlukan perancangan *Prompting* yang cermat serta mekanisme validasi untuk

memastikan bahwa kode terstruktur (seperti sintaks XPDL) yang dihasilkan oleh LLM bersifat valid dan patuh terhadap skema standar.

2.3.2 *OpenRouter* API sebagai Platform *Agregator* LLM

Dalam pengembangan aplikasi berbasis AI, mengintegrasikan satu per satu *Application Programming Interface* (API) dari berbagai penyedia model (provider)—seperti *OpenAI, Anthropic, Google*, atau *Meta* memerlukan kompleksitas struktur kode dan pengelolaan kredensial yang memakan waktu. Untuk mengatasi keterbatasan ini, hadir layanan *OpenRouter* API sebagai platform agregator API terpusat.

OpenRouter API berfungsi sebagai gateway tunggal yang menyediakan antarmuka terintegrasi (*unified interface*) untuk mengakses puluhan model LLM terkemuka secara dinamis (*OpenRouter* , 2024). Penggunaan *OpenRouter* API memberikan sejumlah keuntungan metodologis dalam penelitian rekayasa perangkat lunak;

a. Interoperabilitas dan Fleksibilitas: Memungkinkan pengembang mengganti atau membandingkan berbagai model AI tanpa harus merombak basis kode sistem berbasis web yang dibangun.
b. Standarisasi Format *Request/Response*: Menyediakan format pengiriman data berstandar yang mempermudah pengiriman instruksi prompt dan penerimaan hasil sintaksis XPDL secara konsisten.
c. Efisien dalam Pengujian Sistem: Memfasilitasi proses eksperimen pengujian 10 *textual requirement* dengan berbagai teknik *prompting* secara lebih stabil dan terukur.

2.3.3 *Gpt-4o-mini*

GPT-4o-mini merupakan salah satu model *Large Language Model* (LLM) yang dikembangkan oleh OpenAI sebagai varian yang lebih efisien dari *GPT-4*. Model ini termasuk dalam keluarga *GPT-4o* yang dirancang untuk mengoptimalkan keseimbangan antara kemampuan pemrosesan bahasa dan efisiensi komputasi. Penamaan "Mini" menunjukkan bahwa model ini memiliki ukuran yang lebih ringkas serta dioptimalkan untuk menghasilkan waktu respons (*latency*) yang lebih rendah dan biaya penggunaan per token yang lebih ekonomis dibandingkan dengan model *GPT-4o* berukuran lebih besar.

Secara arsitektural, model *GPT-4o* menggunakan pendekatan *Transformer* yang dilengkapi dengan mekanisme multi-head attention. Mekanisme tersebut memungkinkan model untuk mengidentifikasi hubungan antarkata dalam suatu konteks secara lebih komprehensif sehingga dapat mendukung pemahaman dan pemrosesan bahasa alami. Model ini dikembangkan melalui proses pelatihan menggunakan data berskala besar yang mencakup berbagai domain informasi(Vaswani, 2017; Brown et al. 2020).

a. Karakteristik Teknis

GPT-4o-mini merupakan model kecerdasan buatan berbasis arsitektur Transformer yang memanfaatkan mekanisme attention untuk memproses dan memahami hubungan antarelemen dalam teks. Model ini dikembangkan dengan pendekatan pelatihan yang mencakup *Reinforcement Learning from Human Feedback*(RLHF) untuk menyelaraskan respons dengan preferensi manusia. Kapasitas konteks yang besar memungkinkan model memproses instruksi dan

dokumen dengan jumlah token yang relatif tinggi, sehingga dapat dimanfaatkan untuk menangani kebutuhan yang memiliki banyak informasi. Dari segi kemampuan, *GPT-4o-mini* mendukung penalaran terhadap permasalahan yang kompleks, pembuatan kode program, pemahaman bahasa, serta pelaksanaan instruksi secara terperinci. Model ini juga dapat menghasilkan keluaran dengan format tertentu apabila diberikan instruksi yang jelas. Dalam aspek implementasi, *GPT-4o-mini* dapat diakses melalui *OpenAI* API maupun *OpenRouter* API, sehingga pengguna dapat mengintegrasikannya ke dalam aplikasi dan memanfaatkan berbagai model melalui satu antarmuka layanan. Selain itu, dukungan terhadap respons secara bertahap (*streaming responses*) serta mekanisme pengaturan batas penggunaan (*rate limiting*) memungkinkan pengembang mengelola proses komunikasi antara aplikasi dan layanan model.

b. Keunggulan

GPT-4o-mini memiliki sejumlah keunggulan yang mendukung penggunaannya dalam pengembangan sistem berbasis kecerdasan buatan. Salah satu keunggulannya adalah kemampuannya menghasilkan keluaran terstruktur, seperti JSON dan XML, dengan mengikuti format yang ditentukan melalui instruksi. Kemampuan tersebut dapat dimanfaatkan dalam pembangkitan berkas XPDL, meskipun struktur dan validitas berkas tetap perlu diperiksa menggunakan mekanisme validasi. Selain itu, *GPT-4o-mini* mampu mengikuti instruksi yang terperinci, sehingga dapat digunakan untuk menerapkan teknik prompt engineering dalam mengarahkan proses pembangkitan model proses bisnis. Kemampuan memahami hubungan logis dan ketergantungan antaraktivitas juga berpotensi

membantu model dalam menginterpretasikan kebutuhan fungsional yang kompleks. Dari sisi implementasi, ketersediaan API dan dokumentasi resmi dapat mempermudah integrasi model ke dalam aplikasi berbasis website. Dukungan dokumentasi tersebut juga membantu pengembang memahami parameter penggunaan, mengelola permintaan API, serta melakukan penyesuaian terhadap proses generasi keluaran sesuai kebutuhan sistem.

c. Keterbatasan

Meskipun memiliki kemampuan dalam memahami instruksi dan menghasilkan keluaran terstruktur, *GPT-4o-mini* masih memiliki beberapa keterbatasan. Dalam menangani kebutuhan fungsional yang kompleks, model berpotensi tidak menghasilkan seluruh aktivitas atau elemen proses yang seharusnya direpresentasikan. Kondisi tersebut dapat menyebabkan berkas XPDL yang dihasilkan tidak lengkap atau tidak sepenuhnya sesuai dengan kebutuhan awal. Selain itu, model dapat menghasilkan informasi yang tidak didukung oleh masukan atau membuat asumsi yang kurang tepat, terutama ketika persyaratan yang diberikan bersifat ambigu. Keluaran dalam format XML juga tidak selalu menjamin kesesuaian struktur dengan spesifikasi XPDL, sehingga diperlukan pemeriksaan sintaksis, struktur elemen, dan validitas berkas sebelum digunakan pada perangkat lunak pemodelan proses bisnis. Dari segi penggunaan, biaya pemrosesan API perlu dipertimbangkan, khususnya ketika sistem melakukan permintaan dalam jumlah besar atau memproses masukan yang panjang. Waktu respons juga dapat meningkat ketika jumlah token masukan dan keluaran bertambah, bergantung pada kondisi layanan dan kompleksitas permintaan. Oleh karena itu, penerapan GPT-4o-mini

dalam sistem konfigurasi model proses bisnis perlu disertai validasi hasil dan pengujian untuk memastikan bahwa keluaran yang dihasilkan sesuai dengan kebutuhan fungsional serta standar format yang digunakan.

2.3.4 *Gemini 3.5 Flash Lite*

Gemini adalah salah satu model ai yang dikembangkan oleh *Google* ,sebelumnya dikenal sebagai *Bard*. *Gemini models* tersedia dalam beberapa macam yang dapat digunakan untuk berbagai macam kasus yaitu; *Gemini Ultra* (*largest, most capable*), *Gemini Pro* (*balanced*), dan *Gemini Flash Lite* (*smallest, fastest*).

Gemini 3.5 Flash Lite merupakan varian model kecerdasan buatan berukuran ringan yang dirancang untuk mengoptimalkan efisiensi komputasi dan kecepatan pemrosesan. Model ini ditujukan untuk mendukung implementasi pada perangkat dengan sumber daya komputasi terbatas serta kebutuhan yang memerlukan waktu respons cepat. Istilah *Flash* menunjukkan kemampuan model dalam memproses permintaan secara cepat, sedangkan *Lite* mengacu pada desain model yang lebih ringan dengan konsekuensi adanya keterbatasan pada beberapa aspek kemampuan dibandingkan dengan model yang lebih besar.

a. Karakteristik Teknis

Gemini 3.5 Flash Lite memiliki arsitektur multimodal yang memungkinkan pemrosesan berbagai jenis data, seperti teks, gambar, audio, dan video, meskipun penelitian ini berfokus pada pemrosesan teks. Model ini dirancang untuk mendukung inferensi yang efisien, termasuk pada perangkat dengan sumber daya terbatas, serta memiliki kapasitas konteks yang dapat dimanfaatkan untuk memproses dokumen panjang. Data pelatihannya berasal dari beragam sumber

pengetahuan *Google*. Dalam aspek kemampuan, model ini memiliki kemampuan pemahaman bahasa secara umum, pengenalan pola dalam teks, serta dukungan terhadap berbagai bahasa. Kemampuan mengikuti instruksi dapat bervariasi bergantung pada kejelasan dan ketepatan prompt yang diberikan. Model ini juga memiliki potensi untuk diadaptasi pada domain tertentu melalui proses fine-tuning. Dari segi karakteristik respons, *Gemini 3.5 Flash Lite* cenderung menghasilkan jawaban yang langsung dan ringkas, tetapi konsistensi format keluaran sangat dipengaruhi oleh ketepatan instruksi. Model ini juga berpotensi menghasilkan keluaran yang kreatif dan beragam, meskipun tetap memiliki risiko menghasilkan informasi atau struktur yang tidak akurat.

b. Keunggulan

Gemini 3.5 Flash Lite menunjukkan performa yang relatif konsisten dalam menangani kebutuhan dengan tingkat kompleksitas yang beragam, mulai dari kebutuhan sederhana hingga kompleks. Model ini mampu menyesuaikan hasil generasinya berdasarkan karakteristik dan kompleksitas kebutuhan yang diberikan. Berdasarkan hasil penelitian ini, penggunaan teknik prompting tingkat lanjut, khususnya *Tree of Thought*, menunjukkan peningkatan performa dalam menangani kebutuhan yang sangat kompleks, yaitu kebutuhan yang memiliki lebih dari sepuluh elemen, dengan nilai *Jaccard Similarity* mencapai 0,92. Selain itu, model ini menawarkan keseimbangan antara kemampuan pemrosesan, latensi, dan biaya penggunaan. *Gemini 3.5 Flash Lite* juga berpotensi menghasilkan keluaran terstruktur dengan baik apabila diberikan instruksi yang jelas dan sistematis. Penggunaan teknik prompt engineering, seperti *Chain of Thought* dan *Tree of*

Thought, dapat membantu meningkatkan kualitas keluaran yang dihasilkan. Dukungan infrastruktur *Google* juga menjadi salah satu faktor yang dapat menunjang ketersediaan dan pengoperasian layanan model ini.

c. Keterbatasan

Meskipun memiliki sejumlah keunggulan, *Gemini 3.5 Flash Lite* masih memiliki beberapa keterbatasan. Penggunaan teknik *Zero-Shot Prompting* dapat menghasilkan performa yang bervariasi, terutama ketika kebutuhan yang diberikan memiliki karakteristik tertentu atau instruksi yang kurang spesifik. Model ini juga berpotensi menghasilkan ketidakkonsistenan format pada keluaran XML, seperti tag yang tidak tertutup atau hierarki elemen yang tidak sesuai, sehingga menyebabkan kegagalan validasi berkas. Selain itu, penjelasan mengenai proses penalaran yang dihasilkan model belum tentu cukup untuk membantu mengidentifikasi penyebab kesalahan pada keluaran. Sebagai model yang bersifat umum, *Gemini 3.5 Flash Lite* juga tidak secara khusus dioptimalkan untuk domain BPMN dan XPDL, sehingga masih berpotensi menghasilkan model proses bisnis yang kurang sesuai dengan kebutuhan teknis penelitian. Di samping itu, tingkat kematangan penggunaannya dalam lingkungan produksi dan ketersediaan pengalaman implementasi jangka panjang perlu dievaluasi berdasarkan versi model yang digunakan.

2.3.5 *Claude Haiku 4.5*

Claude merupakan salah satu *large language models* (LLM) yang dikembangkan oleh *Anthropic*, sebuah perusahaan yang berfokus pada penelitian keselamatan kecerdasan buatan (*AI safety)* serta penyelarasan kemampuan

kecerdasan buatan dengan nilai-nilai manusia (*AI alignment*). *Claude Haiku 4.5* merupakan salah satu varian model dalam keluarga *Claude 4* yang dirancang untuk mengoptimalkan efisiensi komputasi dan kecepatan pemrosesan. Namun, optimalisasi tersebut dapat disertai dengan keterbatasan pada beberapa aspek kemampuan penalaran tingkat lanjut dibandingkan dengan model yang memiliki kapasitas lebih besar. Penamaan Haiku mencerminkan karakteristik model yang mengutamakan keringkasan dan efisiensi. Model ini ditujukan untuk mendukung berbagai kebutuhan pemrosesan yang memerlukan waktu respons singkat dan latensi rendah, khususnya pada skenario pemrosesan permintaan dalam jumlah besar (*high-volume processing*).

a. Karakteristik Teknis

Claude Haiku merupakan model kecerdasan buatan yang dikembangkan oleh *Anthropic* dengan arsitektur berbasis *Transformer* yang dirancang untuk mendukung pemrosesan secara efisien. Pengembangan model *Claude* menggunakan pendekatan *Constitutional AI* (CAI), yaitu pendekatan pelatihan yang bertujuan meningkatkan keselarasan respons model dengan prinsip keselamatan dan perilaku yang bertanggung jawab. Model ini memiliki kemampuan untuk memahami bahasa alami, mengikuti instruksi, serta memproses teks dalam jumlah yang relatif besar sesuai dengan kapasitas konteks versi model yang digunakan. Dari segi kemampuan, *Claude Haiku* dapat digunakan untuk berbagai tugas pemrosesan bahasa, memahami kode program, dan menghasilkan respons berdasarkan instruksi yang diberikan. Model ini juga menekankan prinsip respons yang membantu, aman, dan jujur. Dalam menghasilkan keluaran, *Claude*

Haiku cenderung memberikan penjelasan tambahan atau narasi untuk memperjelas responsnya. Namun, karakteristik tersebut dapat menjadi tantangan ketika model diharapkan menghasilkan keluaran dengan format yang ketat, seperti XML atau XPDL, karena teks tambahan di luar struktur yang diminta dapat memengaruhi proses pengolahan keluaran oleh sistem.

b. Keunggulan

Claude Haiku memiliki keunggulan dalam efisiensi pemrosesan dan kecepatan menghasilkan respons, sehingga berpotensi digunakan pada aplikasi yang membutuhkan waktu tanggap relatif singkat. Selain itu, pendekatan pengembangan model yang menekankan keselamatan dan keselarasan dapat membantu mengurangi risiko keluaran yang tidak sesuai dengan prinsip penggunaan kecerdasan buatan secara bertanggung jawab. Dari segi biaya, beberapa versi *Claude Haiku* ditawarkan dengan harga yang kompetitif, sehingga dapat menjadi alternatif untuk aplikasi yang membutuhkan pemrosesan dalam jumlah besar. Model ini juga memiliki kemampuan pemahaman bahasa secara umum yang dapat dimanfaatkan untuk menginterpretasikan kebutuhan fungsional dalam bentuk teks. Di samping itu, gaya komunikasi yang relatif konsisten dapat membantu menghasilkan respons yang mudah dipahami. Berbagai karakteristik tersebut menjadikan *Claude Haiku* sebagai salah satu alternatif model bahasa besar yang dapat dievaluasi dalam pengembangan sistem pembangkitan model proses bisnis berbasis website.

c. Keterbatasan

Claude Haiku memiliki beberapa keterbatasan ketika digunakan untuk menghasilkan model proses bisnis dalam format XPDL. Salah satu permasalahan yang ditemukan dalam penelitian ini adalah kemungkinan munculnya teks naratif atau penjelasan tambahan di luar struktur XML yang diharapkan. Sebagai contoh, model dapat menambahkan kalimat pengantar sebelum elemen XML atau memberikan penjelasan setelah elemen penutup dokumen. Apabila keluaran tersebut langsung diproses menggunakan parser XML tanpa mekanisme pembersihan, proses parsing dapat mengalami kegagalan karena keseluruhan keluaran tidak lagi memenuhi struktur XML yang valid. Selain itu, kemampuan penalaran model dalam menangani kebutuhan dengan hubungan antar aktivitas yang kompleks perlu dievaluasi lebih lanjut, terutama ketika suatu proses melibatkan banyak aktivitas, percabangan, dan ketergantungan logis. Model juga berpotensi menyederhanakan kebutuhan fungsional secara berlebihan sehingga beberapa detail penting tidak tercakup dalam model proses yang dihasilkan. Pada tugas khusus seperti pembangkitan XPDL, konsistensi keluaran juga perlu diuji untuk mengetahui sejauh mana model mampu mempertahankan kelengkapan aktivitas dan ketepatan struktur dokumen. Oleh karena itu, penerapan *Claude Haiku* dalam penelitian ini memerlukan mekanisme validasi XML dan evaluasi kesesuaian model yang dihasilkan terhadap kebutuhan fungsional serta model referensi.

2.4 Website

Menurut Robiansyah dalam (Saputro 2021), website dapat didefinisikan sebagai sekumpulan halaman yang menyajikan berbagai jenis informasi, seperti teks, gambar statis maupun bergerak, animasi, audio, video, serta kombinasi dari

seluruh elemen tersebut, yang tersusun dalam suatu struktur saling terhubung melalui tautan. Berdasarkan sifatnya, website dibedakan menjadi dua jenis, yaitu statis dan dinamis. Website statis memiliki konten yang relatif tetap, jarang mengalami perubahan, serta aliran informasi yang bersifat satu arah dari pemilik kepada pengguna. Sebaliknya, website dinamis ditandai dengan konten yang dapat berubah secara berkala dan memungkinkan interaksi dua arah antara pemilik dan pengguna. Dari segi pengembangannya, website statis umumnya hanya dapat diperbarui oleh pemilik, sedangkan website dinamis memungkinkan pembaruan dilakukan baik oleh pemilik maupun pengguna.

Fungsi utama website sebagai media penyampaian informasi. Melalui ketersediaan informasi tersebut, website dimanfaatkan sebagai sarana untuk mengonversi pengunjung menjadi prospek. Dalam hal ini, pengelola website dapat menyediakan formulir yang memungkinkan pengunjung memberikan alamat email serta data lainnya, sehingga mereka dapat diidentifikasi sebagai prospek potensial(Pratiwi et al. 2025).

2.5 Teknik *Prompt Engineering*

Prompt engineering adalah disiplin dan praktik merancang, memformulasi, serta mengoptimalkan teks instruksi (*input prompt*) yang diberikan kepada *Large Language Models* (LLM) agar dapat menghasilkan respon (*output*) yang presisi, relevan, dan sesuai dengan skema format yang diharapkan (Liu et al, 2023). Dalam konteks pemrosesan bahasa alami, prompt berfungsi sebagai konteks awal dan pengarah logika bagi model AI untuk membatasi ruang pencarian probabilistiknya.

Urgensi *prompt engineering* sangat krusial ketika LLM diperintahkan untuk menghasilkan keluaran berformat terstruktur seperti berkas *XML Process Definition Language* (XPDL). Karena berkas XPDL membutuhkan kepatuhan sintaksis yang sangat ketat terhadap hierarki tag (seperti <*WorkflowProcess*>, <*Activity*>, <*Gateway*>, dan <*Transition*>), instruksi yang samar dapat menyebabkan syntax error atau hallucination. Melalui prompt engineering yang sistematis, struktur instruksi dirancang sedemikian rupa guna meminimalkan kesalahan logika maupun kegagalan parsing XML pada sistem (White et al, 2023).

2.5.1 Jenis-Jenis Teknik *Prompting*

Untuk mengevaluasi efektivitas instruksi terhadap kualitas berkas XPDL yang dihasilkan oleh *OpenRouter* API, penelitian ini mengimplementasikan dan menguji 4 jenis teknik *Prompting* yang berbeda pada antarmuka sistem berbasis web.

1. Zero-Shot Prompting

Zero-Shot Prompting merupakan teknik pemberian instruksi kepada model bahasa besar (*Large Language Model*) untuk menyelesaikan tugas tanpa menyertakan contoh demonstrasi input dan output dalam prompt. Model memanfaatkan pengetahuan yang diperoleh selama proses pelatihan untuk memahami instruksi dan menghasilkan respon yang sesuai dengan permintaan pengguna.(Brown et al, 2020) membahas kemampuan model bahasa dalam menyelesaikan tugas melalui instruksi tanpa pelatihan khusus pada tugas tersebut, serta membandingkannya dengan pendekatan *few-shot*.

Dalam penelitian ini, *Zero-Shot Prompting* digunakan sebagai konfigurasi dasar untuk mengetahui kemampuan model AI dalam menerjemahkan textual requirement menjadi file XPDL tanpa diberikan contoh konversi sebelumnya. Instruksi dalam prompt menjelaskan peran model, format masukan, elemen proses bisnis yang perlu diidentifikasi, serta format keluaran yang diharapkan. Penerapan teknik ini dapat dilakukan melalui beberapa tahapan berikut.

a. Pemberian instruksi utama. Model diminta berperan sebagai pembangkit model proses bisnis yang memahami elemen BPMN dan struktur XPDL.
b. Pemberian textual requirement. Sistem memasukkan deskripsi proses bisnis, misalnya proses pengajuan pinjaman anggota koperasi.
c. Interpretasi kebutuhan. Model mengidentifikasi aktor, aktivitas, urutan proses, kondisi, dan kemungkinan percabangan berdasarkan teks.
d. Pembangkitan XPDL. Model menghasilkan struktur XML yang merepresentasikan proses bisnis sesuai instruksi dan skema XPDL yang diminta.
e. Pemeriksaan keluaran. Sistem memeriksa apakah respons berisi XPDL dan apakah struktur berkas dapat diproses lebih lanjut.

Sebagai contoh, apabila *textual requirement* menyatakan bahwa petugas menerima permohonan anggota dan memeriksa kelengkapan dokumen, prompt *Zero-Shot* dapat langsung meminta model mengubah deskripsi tersebut menjadi XPDL yang memuat aktivitas penerimaan permohonan dan pemeriksaan dokumen, tanpa memberikan contoh XPDL sebelumnya.

Keunggulan teknik ini terletak pada kesederhanaan implementasi, kebutuhan prompt yang relatif singkat, serta kemampuannya menjadi pembanding dasar bagi teknik lainnya. Namun, karena tidak diberikan contoh pemetaan teks ke XPDL, model harus mengandalkan instruksi dan pengetahuan yang dimilikinya. Oleh sebab itu, konsistensi struktur XML, ketepatan pemetaan aktivitas, dan kelengkapan elemen proses perlu dievaluasi melalui pengujian.

2. Few-Shot Prompting

Pemberian contoh ini membantu model AI mengenali pola pemetaan elemen teks ke dalam tag XPDL secara lebih konsisten. *Few-Shot Prompting* merupakan teknik yang menyertakan sejumlah contoh demonstrasi dalam prompt agar model dapat mengenali pola tugas yang diharapkan.(Brown et al, 2020) menjelaskan kemampuan model bahasa menyelesaikan tugas dengan menggunakan beberapa contoh dalam konteks masukan, tanpa melakukan pembaruan parameter model melalui pelatihan khusus.

Pada penelitian ini, *Few-Shot Prompting* digunakan dengan memberikan contoh pasangan textual requirement dan representasi XPDL yang sesuai. Contoh tersebut berfungsi sebagai acuan bagi model untuk memahami bagaimana deskripsi aktivitas, aktor, kondisi, dan alur proses diterjemahkan menjadi elemen-elemen XPDL. Tahapan penerapannya adalah sebagai berikut.

a. Menyiapkan contoh demonstrasi. Peneliti menyiapkan beberapa textual requirement beserta XPDL yang sudah diperiksa validitas dan kesesuaiannya.

b. Memasukkan contoh ke dalam *prompt*. Setiap contoh menampilkan deskripsi kebutuhan dan representasi XPDL yang diharapkan.
c. Memberikan requirement baru. Model menerima deskripsi proses bisnis yang belum memiliki contoh keluaran langsung.
d. Menerapkan pola yang dipelajari dari konteks. Model menggunakan contoh yang diberikan untuk memandu pemetaan aktivitas, alur, dan elemen XPDL.
e. Menghasilkan dan mengevaluasi XPDL. Hasilnya diperiksa berdasarkan validitas struktur, kesesuaian dengan requirement, dan kesamaan terhadap model referensi.

Keunggulan *Few-Shot Prompting* adalah memberikan contoh konkret tentang format dan pola pemetaan yang diinginkan. Teknik ini berpotensi meningkatkan konsistensi keluaran, khususnya dalam penggunaan nama elemen, struktur XML, dan hubungan antarkegiatan. Meskipun demikian, kualitas hasil bergantung pada relevansi, jumlah, dan validitas contoh. Contoh yang keliru atau terlalu berbeda dari requirement yang diuji dapat menyebabkan model mengikuti pola yang tidak sesuai.

3. *Chain-of-Thought* (COT)

Chain-of-Thought Prompting merupakan teknik yang mendorong model menghasilkan rangkaian langkah perantara untuk membantu menyelesaikan masalah yang memerlukan penalaran bertahap.(Wei et al, 2022)menunjukkan bahwa pemberian contoh penalaran bertahap dapat meningkatkan kinerja model pada sejumlah tugas penalaran, termasuk tugas aritmetika, penalaran umum, dan penalaran simbolis.

Dalam konteks penelitian ini, CoT diarahkan untuk menguraikan proses bisnis menjadi bagian-bagian yang lebih terstruktur sebelum membangkitkan berkas XPDL. Alih-alih langsung mengubah teks menjadi XML, instruksi meminta model mengidentifikasi komponen proses secara berurutan. Tahapan penerapannya meliputi:

a. Analisis requirement. Model mengidentifikasi tujuan proses bisnis, aktor yang terlibat, masukan, dan keluaran proses.
b. Identifikasi aktivitas. Model menguraikan tindakan yang dilakukan oleh setiap aktor, seperti menerima dokumen, memeriksa data, memperbarui basis data, dan mengirimkan hasil.
c. Penentuan urutan proses. Model menentukan hubungan urutan antarkegiatan yang direpresentasikan melalui *Sequence Flow*.
d. Identifikasi kondisi dan percabangan. Model mengidentifikasi kondisi keputusan, misalnya dokumen lengkap atau tidak lengkap, dan menentukan kebutuhan *gateway* yang sesuai.
e. Penyusunan representasi proses. Hasil analisis diterjemahkan menjadi elemen BPMN dan struktur XPDL.
f. Pemeriksaan konsistensi. Model diminta memeriksa kesesuaian aktivitas, alur, dan kondisi terhadap requirement sebelum menghasilkan keluaran akhir.

Sebagai contoh, pada proses pengajuan pinjaman anggota, instruksi CoT dapat meminta model mengidentifikasi aktor, aktivitas pemeriksaan dokumen, kondisi kelengkapan dokumen, jalur pengembalian dokumen, dan pemeriksaan

status pinjaman. Setelah struktur proses ditentukan, sistem meminta model menghasilkan XPDL yang sesuai.

Keunggulan CoT dalam konteks ini adalah membantu menguraikan requirement yang memuat beberapa tindakan atau kondisi. Pemisahan tahap analisis dari tahap pembangkitan dapat memudahkan identifikasi ketidaksesuaian antara teks dan struktur proses. Namun, penalaran bertahap tidak menjamin bahwa hasil akhir selalu benar. Kesalahan pada tahap identifikasi aktivitas atau kondisi tetap dapat menghasilkan XPDL yang tidak sesuai.

4. *Tree of Thoughts* (TOT)

Tree of Thoughts merupakan kerangka pemecahan masalah yang memungkinkan model mengeksplorasi beberapa jalur pemikiran, mengevaluasi alternatif, serta menentukan jalur yang paling menjanjikan untuk dilanjutkan.(Yao et al, 2023) memperkenalkan ToT sebagai pengembangan dari pendekatan CoT yang memungkinkan eksplorasi beberapa rangkaian pemikiran dan evaluasi terhadap pilihan-pilihan tersebut.

Berbeda dengan CoT yang umumnya disusun sebagai rangkaian langkah, ToT memungkinkan beberapa alternatif dipertimbangkan pada tahap tertentu. Dalam pembangkitan XPDL, pendekatan ini dapat diterapkan untuk mengevaluasi alternatif representasi proses ketika suatu requirement memiliki kondisi, pengulangan, atau hubungan alur yang belum jelas. ToT pada penelitian ini dapat dijabarkan melalui empat komponen.

a. Dekomposisi masalah

Requirement diuraikan menjadi bagian-bagian yang lebih kecil, seperti aktor, aktivitas, kondisi, jalur alternatif, dan keluaran proses. Tujuannya adalah membuat masalah yang kompleks lebih mudah dianalisis.

Contohnya, *requirement* pengajuan pinjaman diuraikan menjadi penerimaan dokumen, pemeriksaan kelengkapan, pemeriksaan simpanan wajib, pemeriksaan pinjaman berjalan, dan penentuan status pengajuan.

b. Generasi alternatif

Model menghasilkan beberapa alternatif representasi untuk bagian proses yang membutuhkan keputusan. Misalnya, model mempertimbangkan bagaimana memodelkan pengembalian dokumen yang tidak lengkap: sebagai alur kembali ke pemeriksaan awal atau sebagai jalur menuju aktivitas pelengkapan dokumen.

Alternatif tersebut harus tetap didasarkan pada isi requirement. Model tidak seharusnya menambahkan aktivitas baru yang tidak didukung oleh deskripsi kebutuhan.

c. Evaluasi alternatif

Setiap alternatif dinilai berdasarkan kriteria yang ditetapkan. Dalam penelitian ini, kriteria yang dapat digunakan mencakup:

- Kesesuaian dengan textual requirement.
- Kelengkapan aktivitas dan kondisi.
- Konsistensi hubungan alur.
- Kesesuaian penggunaan elemen BPMN.
- Kemungkinan merepresentasikan proses dalam struktur XPDL yang valid.

- Evaluasi ini sebaiknya memiliki aturan yang jelas. Apabila seluruh penilaian hanya diserahkan kepada model yang sama tanpa pemeriksaan tambahan, alternatif yang dipilih belum tentu benar.

d. Pencarian alternatif

Kerangka ToT dapat menggunakan strategi pencarian seperti *Breadth-First Search* (BFS) atau *Depth-First Searc*h (DFS) untuk mengeksplorasi cabang pemikiran. Strategi yang dipilih memengaruhi bagaimana alternatif dikembangkan dan dievaluasi.

- BFS: mengeksplorasi alternatif pada tingkat yang sama sebelum melanjutkan ke tingkat berikutnya.
- DFS: menelusuri satu cabang lebih jauh sebelum kembali mengeksplorasi cabang lainnya.

Pada pembangkitan XPDL, strategi pencarian tersebut dapat digunakan untuk menguji alternatif struktur alur sebelum menentukan representasi akhir. Namun, penggunaannya memerlukan mekanisme yang benar-benar menghasilkan beberapa kandidat, mengevaluasinya, dan memilih kandidat berdasarkan kriteria tertentu.

Oleh karena itu, *prompt* yang hanya meminta model mempertimbangkan beberapa alternatif belum tentu merupakan implementasi ToT secara penuh. Implementasi ToT yang lebih sesuai dengan kerangka asli memerlukan eksplorasi dan evaluasi beberapa jalur pemikiran, serta mekanisme pencarian atau pemilihan kandidat.

2.6 Bizagi Modeler

Bizagi Modeler adalah salah satu perangkat lunak pemodelan proses bisnis (*business process modeling tool*) yang paling luas digunakan di tingkat industri maupun akademis untuk merancang, memetakan, mendokumentasikan, dan mengoordinasikan alur kerja organisasi (Chang et al, 2023). Dikembangkan oleh Bizagi, perangkat lunak ini menyediakan lingkungan grafis yang intuitif dan mematuhi secara penuh standar notasi Business Process Model and Notation (BPMN 2.0) yang ditetapkan oleh *Object Management Group* (OMG).

Dalam praktik Business Process Management (BPM), Bizagi Modeler bertindak sebagai jembatan antara kebutuhan bisnis abstrak dengan spesifikasi teknis eksekusi sistem. Keunggulan utama perangkat lunak ini terletak pada ketaatannya yang ketat terhadap sintaksis notasi BPMN, kemudahan navigasi elemen (*pools, lanes, gateways, dan events*), serta skalabilitasnya untuk pemodelan proses bisnis berskala kompleks (Ivančić, 2019).

Salah satu fitur paling krusial dari Bizagi Modeler adalah kapabilitas interoperabilitasnya melalui dukungan format *XML Process Definition Language* (XPDL) versi 2.2 yang distandarisasi oleh *Workflow Management Coalition* (WfMC) (Bizagi, 2023). Format XPDL pada Bizagi Modeler tidak hanya menyimpan informasi atribut dan logika proses bisnis, melainkan juga menyimpan metadata grafis seperti koordinat posisi setiap elemen notasi pada diagram.

Fitur ekspor dan impor XPDL pada Bizagi Modeler memungkinkan dokumen proses bisnis ditukarkan antar-platform yang berbeda tanpa kehilangan struktur atau informasi visualnya (Fairuzia et al, 2020). Ketika sebuah berkas XPDL

diimpor ke dalam Bizagi Modeler, perangkat lunak akan melakukan parsing sintaksis XML dan merekonstruksi kembali diagram proses bisnis secara visual. Jika terdapat kesalahan sintaksis atau ketidaksesuaian struktur skema pada berkas XPDL, Bizagi Modeler akan menolak proses impor atau menampilkan diagram yang cacat/tidak *valid*.

2.7 *Sentence Complexity*

Sentence complexity atau kompleksitas kalimat merupakan konsep yang menggambarkan tingkat kerumitan suatu kalimat berdasarkan karakteristik linguistik dan usaha yang diperlukan untuk memahami maknanya. Kompleksitas kalimat tidak hanya dipengaruhi oleh panjang kalimat, tetapi juga oleh struktur sintaksis, hubungan antarklausa, penggunaan kata, dan konteks yang menyertai kalimat tersebut.(Iavarone et al, 2021) menjelaskan bahwa penilaian kompleksitas kalimat dapat dipengaruhi oleh konteks di sekitarnya. Oleh karena itu, kompleksitas kalimat dapat dipandang sebagai suatu gradasi, bukan hanya sebagai pembagian antara kalimat sederhana dan kompleks.

Dalam penelitian (Iavarone et al, 2021) mengkaji pengaruh konteks terhadap persepsi manusia dalam menilai kompleksitas kalimat. Penelitian tersebut menggunakan penilaian kompleksitas yang diberikan oleh manusia, menganalisis hubungan antara skor kompleksitas dan berbagai fitur linguistik, serta melatih model regresi untuk memprediksi nilai kompleksitas. Fitur yang dianalisis mencakup karakteristik permukaan, seperti panjang kalimat dan panjang kata, informasi morfosintaksis, serta struktur sintaksis yang lebih kompleks, seperti

kedalaman pohon sintaksis, panjang hubungan dependensi, dan penggunaan subordinasi.

Pengukuran kompleksitas kalimat melalui fitur linguistik eksplisit dilakukan dengan mengekstraksi karakteristik bahasa yang dapat dihitung atau dianalisis dari suatu kalimat. Panjang kalimat dapat digunakan untuk mengetahui jumlah kata yang digunakan, sedangkan jumlah klausa dapat memberikan gambaran mengenai banyaknya bagian informasi yang disampaikan dalam satu kalimat. Selain itu, karakteristik sintaksis dapat digunakan untuk menggambarkan hubungan gramatikal yang lebih kompleks. Meskipun demikian, masing-masing fitur hanya menggambarkan aspek tertentu sehingga tidak dapat digunakan secara tunggal untuk mewakili seluruh dimensi kompleksitas kalimat.

Selain menggunakan fitur linguistik, kompleksitas kalimat dapat dianalisis melalui penilaian manusia. Dalam pendekatan ini, penilai memberikan skor terhadap kalimat berdasarkan tingkat kompleksitas yang dipersepsikan. Skor tersebut dapat dianalisis menggunakan nilai rata-rata untuk menggambarkan tingkat kompleksitas dan simpangan baku untuk mengetahui variasi penilaian antarpenilai. Pendekatan ini memungkinkan peneliti mempertimbangkan persepsi pembaca yang tidak selalu dapat dijelaskan hanya melalui karakteristik linguistik.

Pendekatan lainnya adalah menggunakan model regresi untuk memperkirakan skor kompleksitas berdasarkan fitur linguistik yang diekstraksi dari kalimat. Model dilatih menggunakan data kalimat yang telah memiliki skor kompleksitas sebagai acuan. (Iavarone et al, 2021) menunjukkan bahwa fitur

linguistik eksplisit yang mencakup karakteristik morfosintaksis dan sintaksis dapat memberikan kontribusi penting dalam memprediksi penilaian kompleksitas. Namun, penerapan model regresi memerlukan data berlabel yang memadai agar hasil prediksi dapat dievaluasi secara objektif.

Kompleksitas kalimat juga dapat dikaji melalui analisis keterbacaan (*readability analysis*), yaitu pendekatan yang memperkirakan kemudahan teks untuk dibaca berdasarkan karakteristik linguistik tertentu. Beberapa formula keterbacaan, seperti *Flesch–Kincaid Grade Level* dan *Gunning Fog Index*, mempertimbangkan panjang kalimat serta karakteristik kata. Formula tersebut dapat digunakan sebagai indikator pendukung untuk menganalisis keterbacaan textual requirement, tetapi tidak secara langsung mengukur kompleksitas struktur proses bisnis.

Dalam penelitian ini, konsep *sentence complexity* digunakan sebagai dasar untuk menganalisis karakteristik tekstual dari textual requirement yang menjadi masukan sistem pembangkitan model proses bisnis. Analisis karakteristik bahasa dapat membantu mengidentifikasi *requirement* yang memiliki kalimat panjang atau struktur linguistik yang lebih rumit. Namun, karena kompleksitas kalimat berbeda dari kompleksitas proses bisnis, analisis tersebut perlu dibedakan dari pemeriksaan jumlah aktivitas, kondisi, percabangan, dan perulangan yang terdapat dalam requirement. Dengan demikian, pengukuran karakteristik linguistik digunakan untuk mendeskripsikan kompleksitas teks, sedangkan analisis struktural digunakan untuk memahami tingkat kerumitan proses bisnis yang harus direpresentasikan dalam model BPMN dan XPDL.

2.7.1 *Word* Per *Sentence*

Word per *Sentence* (WPS) merupakan ukuran yang digunakan untuk mengetahui rata-rata jumlah kata dalam satu kalimat pada suatu teks. Pengukuran ini dilakukan dengan membandingkan jumlah keseluruhan kata dengan jumlah kalimat yang terdapat dalam teks. Nilai WPS dapat memberikan gambaran mengenai panjang kalimat yang digunakan dalam menyampaikan suatu informasi atau kebutuhan sistem. Semakin tinggi nilai WPS, semakin banyak kata yang digunakan dalam satu kalimat secara rata-rata. Sebaliknya, semakin rendah nilai WPS, semakin sedikit kata yang digunakan dalam satu kalimat secara rata-rata.

Panjang kalimat merupakan salah satu karakteristik linguistik yang dapat digunakan dalam kajian kompleksitas kalimat. (Maat, 2017) mengkaji hubungan antara panjang kalimat dan berbagai faktor kompleksitas struktural, seperti penggunaan klausa subordinatif, modifikator, koordinasi, serta panjang dependensi sintaksis. Penelitian tersebut menunjukkan bahwa panjang kalimat berkaitan dengan sejumlah faktor kompleksitas, tetapi panjang kalimat saja tidak selalu dapat memprediksi kompleksitas struktural secara akurat. Oleh karena itu, panjang kalimat dapat digunakan sebagai indikator linguistik, tetapi tidak dapat sepenuhnya menggambarkan kompleksitas struktur suatu kalimat.

Penelitian (Iavarone et al, 2021) juga membahas penilaian kompleksitas kalimat dengan mempertimbangkan berbagai fitur linguistik. Fitur yang dikaji meliputi panjang kalimat, informasi morfosintaksis, dan karakteristik sintaksis. Penelitian tersebut menunjukkan bahwa penilaian kompleksitas kalimat dapat dipengaruhi oleh lebih dari satu karakteristik linguistik. Dengan demikian,

penggunaan jumlah kata per kalimat dalam penelitian ini diposisikan sebagai pengukuran panjang kalimat, bukan sebagai representasi menyeluruh dari kompleksitas linguistik.

Dalam penelitian ini, WPS digunakan untuk menganalisis karakteristik tekstual *textual requirement* yang menjadi masukan bagi sistem konfigurasi model proses bisnis berbasis website dengan memanfaatkan *OpenRouter* API. Pengukuran dilakukan untuk mengetahui rata-rata jumlah kata dalam setiap kalimat pada masing-masing *textual requirement*. Informasi tersebut digunakan sebagai indikator awal untuk membandingkan panjang teks kebutuhan yang digunakan dalam proses pembangkitan model proses bisnis dalam format XPDL. Nilai WPS dihitung menggunakan persamaan berikut.

$$WPS = \frac{W}{S}$$

Keterangan:

WPS = rata-rata jumlah kata per kalimat (*Words per Sentence*).

W = jumlah keseluruhan kata dalam *textual requirement*.

S = jumlah kalimat dalam *textual requirement*.

Sebagai contoh, apabila suatu *textual requirement* terdiri atas 40 kata dan 4 kalimat, nilai WPS yang diperoleh adalah 10. Artinya, teks tersebut memiliki rata-rata 10 kata dalam setiap kalimat. Nilai ini menunjukkan panjang rata-rata kalimat,

bukan secara langsung menunjukkan jumlah aktivitas, percabangan, atau kompleksitas alur proses bisnis yang terkandung di dalamnya.

Hasil pengukuran WPS selanjutnya dapat digunakan untuk mendeskripsikan variasi panjang kalimat pada dataset *textual requirement*. Apabila penelitian mengelompokkan nilai WPS ke dalam kategori rendah, sedang, dan tinggi, batas setiap kategori perlu ditetapkan dan dijelaskan secara konsisten berdasarkan tujuan penelitian atau dasar empiris yang relevan. Batas kategori tersebut tidak dapat dianggap sebagai standar universal kompleksitas kalimat tanpa dukungan referensi yang sesuai.

Dengan demikian, penggunaan WPS dalam penelitian ini bertujuan untuk memberikan gambaran kuantitatif mengenai panjang kalimat pada *textual requirement* yang digunakan sebagai masukan sistem. Nilai WPS dianalisis sebagai salah satu karakteristik teks dan tidak digunakan sebagai satu-satunya dasar untuk menyimpulkan kompleksitas proses bisnis maupun kualitas model BPMN/XPDL yang dihasilkan. Evaluasi kualitas hasil pembangkitan model proses bisnis tetap dilakukan melalui metode evaluasi model yang ditetapkan dalam penelitian.

BAB III
METODOLOGI PENELITIAN

3.1 Data

Data yang digunakan dalam penelitian ini dibagi menjadi dua yaitu data masukan berupa dokumen persyaratan tekstual (*textual requirements*) dan data acuan pembanding (*Ground Truth*) yang diambil dari jurnal (Sholiq et al, 2022).

3.1.1 Data Masukan (*Textual Requirements)*

Data masukan dalam penelitian ini merupakan deskripsi fungsional atau narasi alur proses bisnis tekstual yang merepresentasikan skenario operasional sistem tertentu (misalnya alur pendaftaran, input data barang, hingga transaksi). Dokumen persyaratan tekstual ini dirancang dengan variasi struktur kalimat mulai dari kalimat sederhana hingga kalimat majemuk kompleks guna menguji fleksibilitas dan tingkat ketepatan model AI dalam mengekstraksi elemen proses bisnis. Seluruh rincian data masukan (*textual requirements*) yang akan diuji oleh sistem disajikan pada tabel dibawah ini.

Tabel 3.1 Data *Textual Requirements*

No.	***Requirement***	***Textual Input***	***Sentence Structure***
1.	*Making a cash transaction*	*An officer submits evidence of a transaction. An accountant records the cash transaction to a cash database, including transaction type, date, proof number, value for money, and description.*	*- Simple Sentence*

No.	*Requirement*	*Textual Input*	*Sentence Structure*
2.	*Withdraw mandatory savings*	*Management wants to withdraw the mandatory savings by providing information about the members. An officer displays the mandatory savings based on the member ID, so the officer processes the withdrawal of the mandatory savings.*	- *Simple Sentence* - *Compound Sentence*
3.	*Resign a member*	*Management asks a member to be removed from the cooperative membership. An officer accepts the management's request. The officer reads information about members from the member database, updates the member's status to resign to the member database, then informs the management about the member's status.*	- *Simple Sentence* - *Sequential Verbs* - *Keyword Verb for Datastore*
4.	*Making a cash outflow report*	*Management orders a cash-out report which of the year and month is given by management, that an accountant filters out cash transactions from the cash database. If the query is not empty, the accountant prints a cash-out report, gives it to management; otherwise, there is no cash-out.*	- *Complex Sentence with Adjective Clauses* - *Passive Voice* - *Keyword Verb for Datastore* - *Complex Sentence with If-then-otherwise* - *Sequential Verbs* - *Intransitive Clause*
5.	*Display a member*	*A member needs to display detailed information about a member by providing a member ID. An officer accepts member requests. The officer reads a member's information from the member database. If the member exists, then the officer reads the*	- *Simple Sentence* - *Keyword Verb for Datastore* - *Complex sentence with if-then-else* - *Intransitive clause* - *Sequential clauses*

No.	*Requirement*	*Textual Input*	*Sentence Structure*
		work unit data from the member database, the officer provides the detailed member information to the member, else the officer confirms no information.	
6.	*Making a bill report*	*A member needs a monthly billing statement by providing his member ID and billing month. An officer reads a member's billing data from the billing database based on member ID and month. If the monthly bill exists, the officer prints the member's bill and informs the member, otherwise the bill does not exist.*	- *Simple sentence* - *Keyword verb for datastore* - *Complex sentenced with if-then-otherwise* - *The conjunction 'and'* - *Intransitive clause*
7.	*Changing a work unit*	*A member submits changes to the work unit by filling out a form, so an officer accepts member's requests. The officer reads member and the works unit information from the member and the works unit databases. The officer updates member information based on the member database.*	- *Compound sentence* - *Keyword verb for datastore* - *Simple sentence*
8.	*Realizing a member loan*	*Management wants to realize member loans. An officer receives orders for member loan realization. The officer opens the member loan realization menu, sets the loan type, enters the member ID. The officer sets the instalment amount, principal instalment, and repay ment period. The officer realizes member loans,*	- *Simple sentence* - *The conjunction 'and' on object* - *Sequential verbs* - *Keyword verb for datastore*

No.	*Requirement*	*Textual Input*	*Sentence Structure*
		saves loan realization information to the loan database.	
9.	*Making a new member registration*	*A prospective member applies a requirements document to become a cooperative member that an officer accepts. Once the document has been lightly checked by the officer, the officer sends the document to the management for further processing. Management fully reviews the document based on the cooperative master book. Management assigns a status to the document based on the review results, so management sends the document back to the officer. If the status is accepted, then the officer creates a new member, adds the new member to the member database, and prints the cooperative card. Otherwise, the officer returns the requirements document to the prospective member to complete the requirements. Then, the prospective member receives the requirements document to be completed.*	- *Complex sentence with adjective clause* - *Complex sentence with if-then-else* - *Simple sentence* - *Passive voice* - *Adverb for manners* - *Keyword verb for data object* - *Keyword verb for datastore* - *Intransitive clause* - *Sequential verbs*
10.	*Selecting a member loan request*	*A member submits a loan request by attaching a credit requirements document. The credit section accepts the member's loan request. The credit section checks the credit requirements document; if the document is incomplete, the*	- *Looping process* - *Simple sentence* - *Compound-complex sentence* - *Complex sentence with adjective clause* - *Complex sentence with if-then-else*

No.	*Requirement*	*Textual Input*	*Sentence Structure*
		credit section returns it to the member to complete the required documents (go to step 1). *The credit section checks the mandatory deposit from the deposit database and checks the current loan from the loan database. If the mandatory deposit has not been paid or a current loan exists, then the submission status is rejected; otherwise, the submission status is okay. The credit section sends the submission document to the officer for further processing, and the officer receives the document.*	- *Keyword verb for data object* - *Keyword verb for datastore* - *The conjunction 'and'* - *The disjunction 'or'* - *Intransitive clause*

3.1.2 Data Acauan Pembanding (*Ground Truth)*

Selain data masukan berupa teks, penelitian ini juga menggunakan data acuan pembanding atau *gold standard* yang berfungsi sebagai model standar validasi. Data acuan ini berupa BPMN yang diperoleh melalui permodelan yang dilakukan oleh *expert judgement* di dalam jurnal (Sholiq, Sarno, and Astuti, 2022) dari *textual requirements* yang terdapat di point 3.1.1, rincian data acuan pembanding dapat dilihat dalam lampiran 1.

3.1.3 Pengukuran Kompleksitas Kalimat *Textual Requirements*

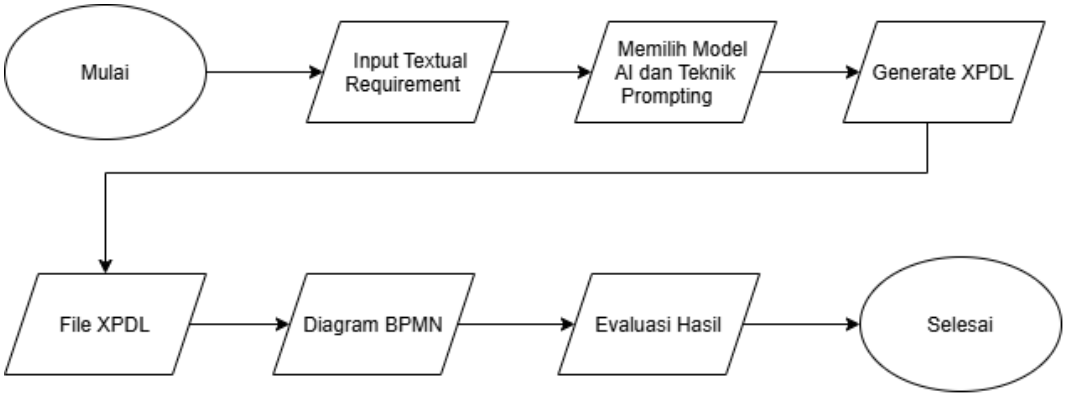
Kompleksitas setiap *textual requirements* diukur menggunakan pendekatan *words per sentence* (WPS) yang dikombinasikan dengan jumlah kata, kalimat, dan klausa. Pengukuran dilakukan terhadap 10 *textual requirements* pada Tabel 3.1.

Jumlah kata (W) dihitung dengan memisahkan teks berdasarkan spasi, dan jumlah kalimat (S) dihitung berdasarkan tanda titik sebagai pemisah kalimat. Jumlah klausa (K) dihitung dengan menghitung setiap pasangan subjek dan predikat dalam kalimat, termasuk klausa utama dan anak kalimat.

Tabel 3.2 Hasil Pengukuran Kompleksitas Kalimat

No.	Textual Requirement	Kata (W)	Kalimat (S)	WPS
1	*Making a cash transaction*	28	2	14,00
2	*Withdraw mandatory savings*	34	2	17,00
3	*Resign a member*	45	3	15,00
4	*Making a cash outflow report*	47	2	23,50
5	*Display a member*	62	4	15,50
6	*Making a bill report*	53	3	17,67
7	*Changing a work unit*	46	3	15,33
8	*Realizing a member loan*	55	5	11,00
9	*Making a new member registration*	120	7	17,14
10	*Selecting a member loan request*	114	6	19,00

3.2 Tahap Penlitian



Gambar 3.1 Tahap Penelitian

Penelitian ini dilaksanakan melalui 6 tahapan yang berurutan, sebagaimana ditunjukkan pada Gambar 3.1. Tahapan tersebut dimulai dari penyiapan input textual requirement hingga evaluasi hasil file XPDL yang diimpor ke Bizagi Modeler menjadi BPMN. Tahap 2 sampai tahap 7 dijalankan secara berulang untuk

seluruh kombinasi model AI, teknik *prompting*, dan *textual requirements* yang diuji.

3.2.1 *Input Textual Requirements*

Pada tahap ini dilakukan penyiapan data uji berupa 10 *textual requirement* berbahasa Inggris yang diadopsi dari penelitian (Sholiq et al, 2022). *Requirement* tersebut memiliki tingkat kompleksitas yang bervariasi, mulai dari kalimat sederhana hingga kalimat majemuk. Setiap requirement disimpan dalam bentuk teks tanpa perubahan dan digunakan secara identik pada seluruh kombinasi eksperimen. Dengan demikian, perbedaan hasil yang diperoleh dapat dikaitkan dengan model dan teknik prompting, bukan dengan perbedaan masukan. Keluaran dari tahap ini adalah 10 *requirement* yang siap diproses.

3.2.2 Memilih Model AI dan Teknik *Prompting*

Pada tahap ini ditentukan konfigurasi eksperimen yang akan dijalankan. Tiga model bahasa besar digunakan, yaitu *GPT-4o-mini* dari *OpenAI*, *Claude Haiku 4.5* dari *Anthropic*, dan *Gemini 3.5 Flash Lite* dari *Google*. Ketiga model tersebut diakses melalui *OpenRouter* API. Selanjutnya, empat teknik prompting diterapkan, yaitu *Zero-Shot*, yang memberikan instruksi tanpa contoh; *Few-Shot*, yang menyertakan beberapa contoh pasangan requirement dan XPDL; *Chain of Thought*, yang meminta model menjelaskan penalaran secara bertahap; dan *Tree of Thought*, yang meminta model mengeksplorasi beberapa alternatif penalaran sebelum memilih hasil yang paling konsisten. Kombinasi tiga model, empat teknik, dan sepuluh requirement menghasilkan 120 eksekusi. Parameter pemanggilan API,

seperti temperatur dan jumlah maksimum token, ditetapkan sama untuk seluruh model.

3.2.3 *Generate* XPDL

Pada tahap ini, prompt yang telah disusun sesuai teknik yang dipilih dikirim ke model melalui *OpenRouter* API. Respons yang dikembalikan oleh model disimpan sebagaimana diterima. Penyimpanan respons tanpa pembersihan awal dilakukan agar kegagalan format, seperti adanya kalimat penjelasan di luar struktur XML, dapat terdokumentasi secara utuh.

3.2.4 *File* XPDL

Respons yang berhasil diterima kemudian disimpan sebagai berkas dengan ekstensi .xpdl. Penamaan *file* memuat informasi model, teknik, dan nomor requirement, misalnya gpt4omini_cot_req03.xpdl, sehingga setiap berkas dapat ditelusuri kembali pada tahap analisis. *File* XPDL inilah yang menjadi keluaran utama sistem dan masukan untuk tahap berikutnya.

3.2.5 Diagram BPMN

Pada tahap ini, setiap berkas XPDL diimpor ke dalam Bizagi Modeler untuk ditampilkan sebagai diagram BPMN. Keberhasilan impor dan tampilnya diagram digunakan sebagai indikator validitas berkas. *File* yang berhasil diimpor dicatat sebagai *valid*, sedangkan berkas yang gagal diimpor dicatat sebagai tidak valid beserta jenis kegagalannya. Pada tahap ini BPMN yang dihasilkan dapat dilihat di lampiran

3.2.6 Evaluasi Hasil

Pada tahap evaluasi hasil, diagram BPMN hasil impor dari Bizagi Modeler dibandingkan dengan diagram BPMN *ground truth* menggunakan metrik *Jaccard Similarity*. Metrik ini dihitung dengan rumus sebagai berikut.

$$Jaccard\ similarity\ = \frac{|A \cap B|}{|A \cup B|} \qquad (3.1)$$

Elemen A merupakan himpunan elemen diagram *ground truth* dan B merupakan himpunan elemen diagram hasil model. Nilai yang diperoleh berada pada rentang 0 hingga 1, dengan nilai mendekati 1 menunjukkan tingkat kemiripan yang semakin tinggi. Skor *Jaccard Similarity* hanya dihitung untuk berkas yang dinyatakan valid pada tahap sebelumnya.

3.3 Desain Eksperimen

Desain eksperimen dirancang untuk menguji dan mengukur sejauh mana sistem mampu mereproduksi model proses bisnis yang akurat dibandingkan dengan model standar.

1. Variabel Eksperimen
    a. Variabel Masukan (*Independent Variables*): Variasi teknik prompt engineering dan variasi tingkat kompleksitas kalimat pada *textual requirement*s.
    b. Variabel Keluaran (*Dependent Variables*): Berkas XPDL hasil generasi sistem dan nilai indeks kemiripan.
2. Skenario Pengujian

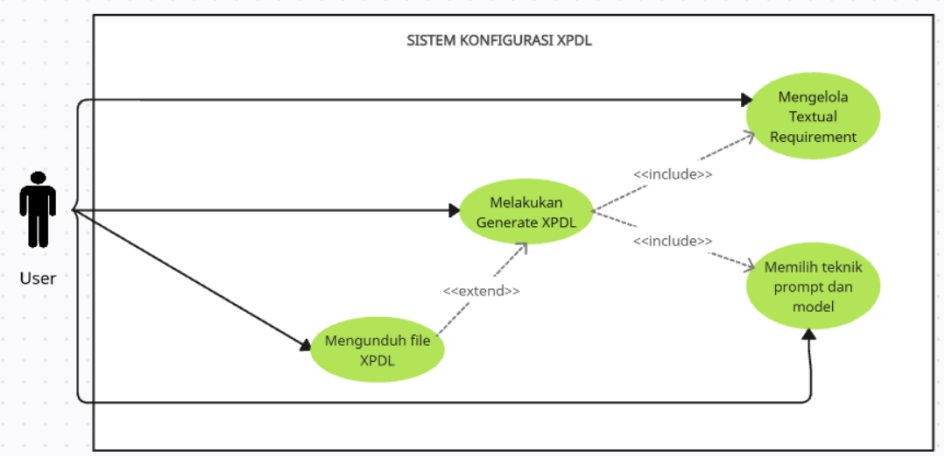
Eksperimen dilakukan dengan menguji sejumlah skenario kasus proses bisnis. Setiap skenario teks akan dieksekusi menggunakan variasi prompt, kemudian hasilnya dievaluasi berdasarkan tiga komponen utama dalam struktur BPMN:

a. Kemiripan Aktivitas (*Task Similarity*): Mengukur ketepatan pelabelan nama aktivitas atau task antara hasil sistem dan Bizagi Modeler.
b. Kemiripan Percabangan *(Gateway Similarity*): Mengukur kesesuaian jenis logika kontrol (seperti *Exclusive/XOR Gateway* atau *Parallel/AND Gateway*).
c. Kemiripan Alur Koneksi (*Sequence Flow Similarity*): Mengukur ketepatan hubungan antar-node berdasarkan titik asal dan tujuan.

3. Metode Pengukuran Performa

Untuk mendapatkan nilai akurasi yang objektif, penelitian ini menggunakan persamaan *Jaccard Similarity* guna membandingkan himpunan elemen pada berkas XPDL sistem (B) dengan berkas XPDL *Ground Truth* Bizagi Modeler (A).

3.4 Perancangan *Use Case* Diagram



Gambar 3.2 *Use Case* Diagram

Use case diagram digunakan untuk memodelkan interaksi antara pengguna dengan sistem website konfigurasi model proses bisnis yang dibangun. Diagram ini menggambarkan fungsi-fungsi yang dapat dilakukan pengguna serta hubungan antarfungsi, sehingga alur penggunaan sistem dalam penelitian dapat dipahami secara jelas sebelum tahap implementasi. Diagram *use case* dapat dilihat pada Gambar 3.2.

Diagram ini memuat satu aktor dan empat *use case*. Relasi antaraktor dengan use case berupa asosiasi, sedangkan relasi antarusecase menggunakan dua jenis, yaitu <<*include*>> dan <<*extend*>>. Relasi <<*include*>> menunjukkan bahwa suatu *use case* selalu memanggil *use case* lain sebagai bagian dari alurnya, sedangkan relasi <<*extend*>> menunjukkan bahwa suatu *use case* dapat menambahkan perilaku pada *use case* lain secara opsional.

1. Aktor yang Terlibat

Aktor dalam sistem adalah Pengguna atau Peneliti, yaitu pihak yang mengakses website untuk memasukkan *teks requirement*, menjalankan proses

generate XPDL, dan mengunduh hasil keluaran. Aktor tidak memiliki hak akses khusus karena sistem dirancang untuk digunakan oleh satu peran dalam penelitian ini.

2. Daftar *Use Case* (Skenario Fungsional)

Sistem memiliki 4 *use case* sebagai berikut.

a. UC-01 Mengelola Textual Requirement. Pengguna dapat memasukkan, mengubah, atau menghapus teks requirement yang akan diuji. *Use case* ini disertakan oleh UC-03 melalui relasi <<*include*>>, karena setiap proses *generate* memerlukan *teks requirement* sebagai masukan.

b. UC-02 Memilih Teknik *Prompt* dan Model LLM. Pengguna menentukan model AI (*GPT-4o-mini, Claude Haiku 4.5,* atau *Gemini 3.5 Flash Lite*) dan teknik *prompt engineering* (*Zero-Shot, Few-Shot, Chain of Thought,* atau *Tree of Thought)* sebelum permintaan dikirim ke *OpenRouter* API. *Use case* ini juga disertakan oleh UC-03 melalui relasi <<*include*>>, karena konfigurasi model dan teknik wajib ditentukan sebelum proses generate dijalankan.

c. UC-03 Melakukan *Generate* XPDL. Sistem memproses permintaan dengan mengirimkan *teks requirement* beserta konfigurasi prompt ke *OpenRouter* API, kemudian menerima respons model dan mengubahnya menjadi berkas XPDL. *Use cas*e ini menjadi inti interaksi pengguna dengan sistem dan menyertakan UC-01 dan UC-02.

d. UC-04 Mengunduh Berkas XPDL. Pengguna dapat menyimpan berkas XPDL hasil keluaran sistem ke perangkat lokal. *Use case* ini bersifat opsional dan

memperluas UC-03 melalui relasi <<*extend*>>, karena pengunduhan hanya dilakukan setelah berkas XPDL berhasil dihasilkan dan tidak selalu diperlukan oleh pengguna.

Dengan demikian, alur utama penggunaan sistem dimulai dari pengelolaan *teks requirement* dan pemilihan model serta teknik *prompting*, dilanjutkan dengan proses *generate* XPDL, dan dapat diakhiri dengan pengunduhan hasil *file*.

3.5 Penjelasan Fitur Website

Berdasarkan sistem perancangan website konfigurasi model proses bisnis yang dibangun, terdapat beberapa fitur utama yang disediakan untuk mendukung jalannya penelitian yaitu.

1. Fitur Input *Textual requirement*

   Digunakan oleh pengguna untuk memasukkan atau mengetikkan teks deskripsi persyaratan alur proses bisnis (*textual requirements*) yang akan diuji.

2. Fitur Pengaturan Prompt dan Integrasi *OpenRouter* API

   Fitur yang memungkinkan sistem untuk menghubungkan teks masukan ke *Large Language Model* (LLM) melalui *OpenRouter* API. Pengguna juga dapat memilih atau mengatur variasi teknik *prompt engineering* yang diuji dalam eksperimen.

3. Fitur Konversi dan Generate Berkas XPDL

   Fitur inti pemrosesan di mana sistem secara otomatis menerjemahkan respons teks dari LLM menjadi struktur berkas berformat XPDL (*Xml Process*

Definition Language) yang sesuai dengan standar BPMN 2.0. Pengguna dapat langsung mengunduh *(download*) berkas XPDL tersebut.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil

Subbab ini memaparkan hasil pengujian yang diperoleh dari 120 eksekusi, yaitu kombinasi tiga model AI, 4 teknik *prompting*, dan 10 *textual requirement*. Pemaparan mengikuti alur penelitian, dimulai dari hasil *generate file* XPDL, hasil generate diagram BPMN beserta nilai *Jaccard Similarity* terhadap *ground truth*, hasil *file* XPDL, hingga proses import *file* XPDL menjadi BPMN di Bizagi Modeler. Subbab ini hanya menyajikan data dan temuan apa adanya, sedangkan penjelasan mengenai penyebabnya dibahas pada subbab 4.2.

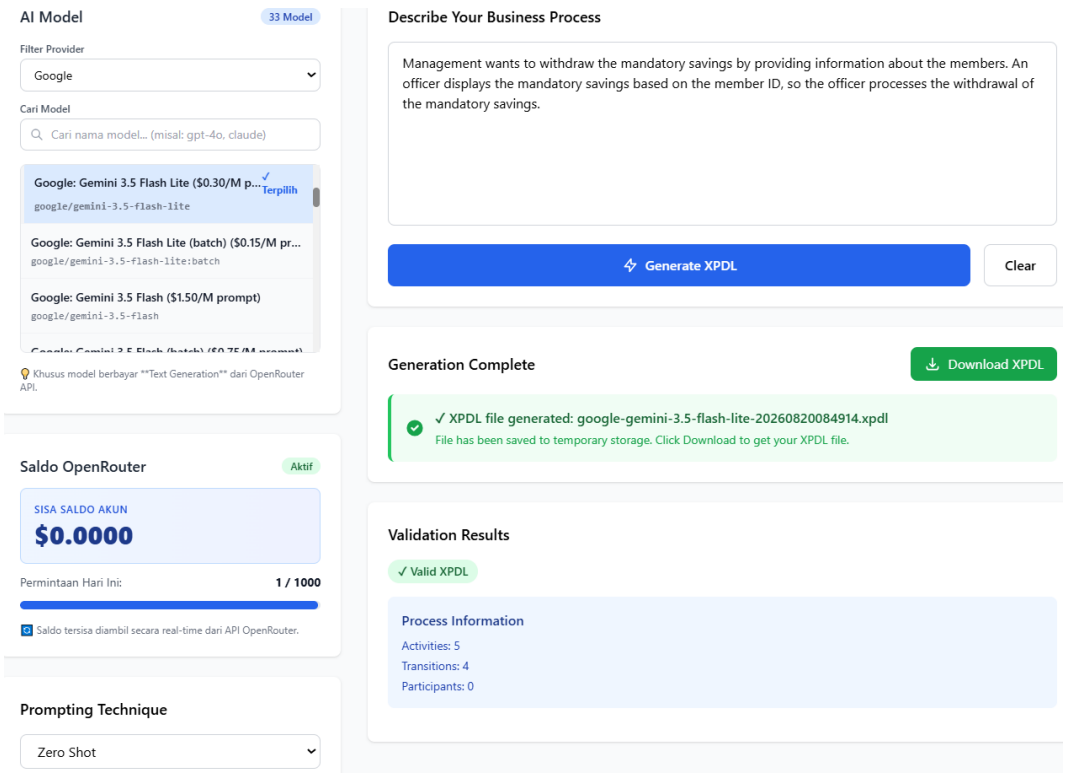
4.1.1 Hasil *Generate* XPDL

Tahap *generate* XPDL dijalankan melalui website sistem konfigurasi model proses bisnis yang terhubung dengan *OpenRouter* API. Setiap *textual requirement* dikirim ke model AI menggunakan *prompt* sesuai teknik yang dipilih. Pengujian melibatkan 3 model (*GPT-4o-mini, Claude Haiku 4.5,* dan *Gemini 3.5 Flash Lite*), 4 teknik *prompting* (*Zero-Shot, Few-Shot, Chain of Thought,* dan *Tree of Thought*), serta 10 textual requirement (Data 1 sampai Data 10). Kombinasi tersebut menghasilkan 120 eksekusi.

Respon dari model disimpan sebagaimana diterima, tanpa pembersihan awal, lalu disimpan sebagai berkas berekstensi .xpdl. Nama berkas memuat model,

teknik, dan nomor requirement, misalnya gpt4omini_cot_req03.xpdl, sehingga setiap hasil dapat ditelusuri kembali.

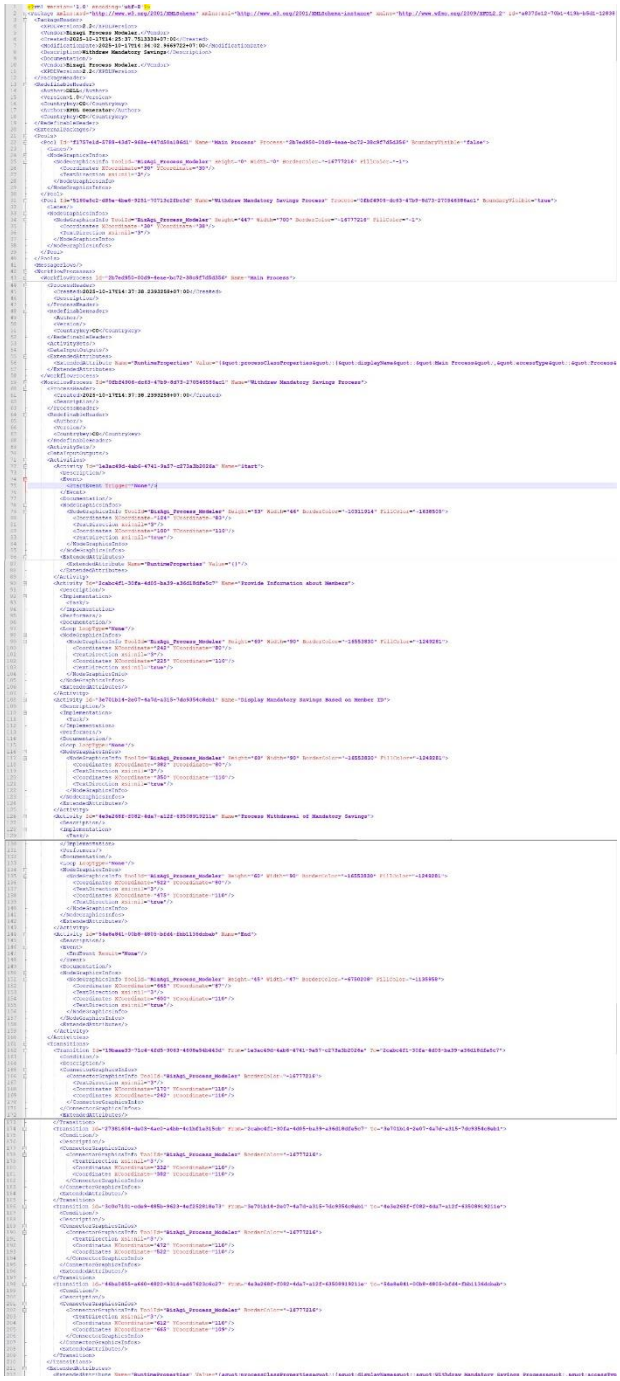
Dari 120 eksekusi yang dilakukan dihasilkan *file* XPDL *valid* dan *file* XPDL *invalid*. *File* XPDL yang valid akan diimpor menjadi BPMN menggunakan Bizagi Modeler. Berikut akan disajikan tabel 4.1 yang menjelaskan validitas *File* XPDL yang di *generate*.



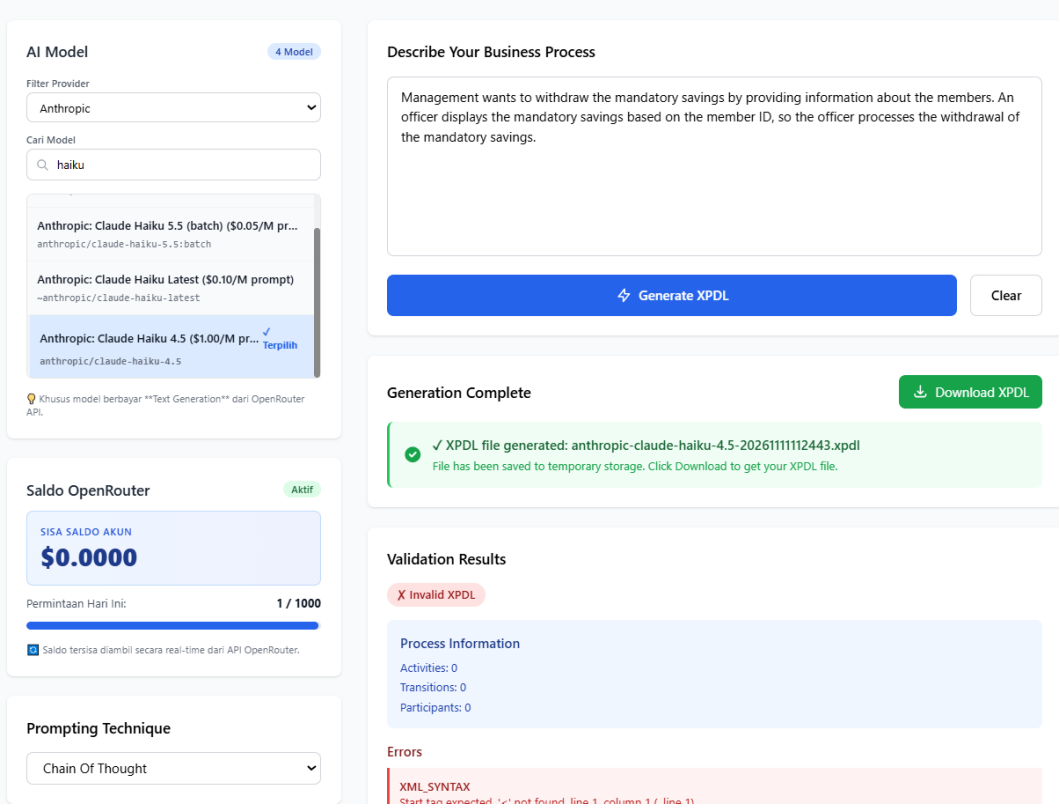
Gambar 4.1 *Generate File* XPDL *Valid*

Pada Gambar 4.1 ditunjukkan bagaimana proses *generate File* XPDL dari *requirement* data 2 dan bagaimana hasil generate nya, dimana pada gambar tersebut menggunakan Model AI *Gemini 3.5 Flash Lite* dan menggunakan teknik prompting *zero shot*. *File* XPDL yang digenerate memiliki status *valid* XPDL, kemudian File

XPDL yang sudah di *generate* akan diunduh. *File* XPDL yang sudah diunduh dicontohkan pada gambar 4.2 dibawah ini. Sedangkan hasil *generate file* XPDL yang memiliki status *invalid* XPDL dapat dilihat pada gambar 4.3.



Gambar 4.2 *File* XPDL yang sudah diunduh



Gambar 4.3 *Generate File* XPDL *Invalid*

4.1.2 Hasil *Generate* BPMN

File XPDL yang *valid* akan diimpor ke Bizagi Modeler ditampilkan sebagai diagram BPMN. Diagram hasil *generate* ini kemudian dibandingkan dengan diagram ground truth yang ada di lampiran 1 menggunakan *Jaccard Similarity*. *File* XPDL yang berhasil diimpor menjadi BPMN akan disajikan pada lampiran 2. Untuk data 9 dikarenakan tidak ada *ground truth* nya maka tidak dilakukan perbandingan terhadap data tersebut.

4.1.2 Hasil Validitas *Generate File* XPDL

Sebelum melakukan evaluasi tingkat kemiripan jumlah elemen BPMN menggunakan algoritma *Jaccard Similarity*, tahap awal pengujian sistem dilakukan

untuk menguji tingkat keberhasilan (*validity*) dari *file* XPDL yang dihasilkan oleh ketiga model AI (*GPT-4o-mini, Claude Haiku 4.5*, dan *Gemini 3.5 Flash Lite*) dengan menerapkan empat variasi teknik prompt engineering (*Chain of Thought*, *Few-Shot*, *Zero-Shot*, dan *Tree of Thought*) terhadap 10 skenario dataset (Data 1 hingga Data 10). Pengujian ini bertujuan untuk memastikan apakah teks keluaran dari model AI berhasil diterjemahkan ke dalam format XPDL yang sesuai tanpa mengalami kegagalan sintaksis. Hasil pengujian validitas *file* xdpl disajikan pada tabel dibawah ini.

Tabel 4.1 Validitas *Generate File* XPDL

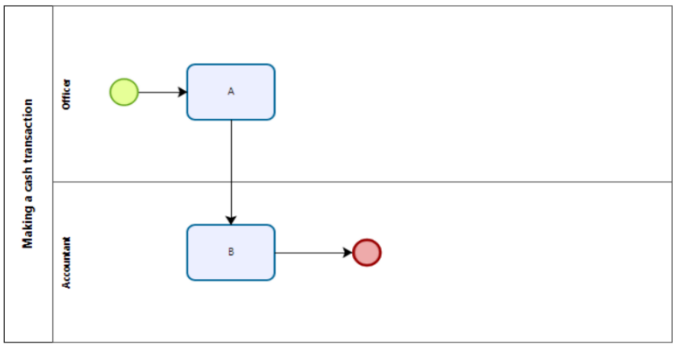
Model AI	**Teknik *Prompting***	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**D10**
GPT-4o-mini	*Chain of Thought*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Few-Shot	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
	Zero-Shot	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
	Tree of Thought	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Claude Haiku 4.5	*Chain of Thought*	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Few-Shot	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Zero-Shot	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Tree of Thought	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Gemini 3.5 Flash Lite	*Chain of Thought*	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗
	Few-Shot	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓
	Zero-Shot	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗
	Tree of Thought	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓

4.1.4 Hasil Perbandingan BPMN dengan Metode *Jaccard Similarity*

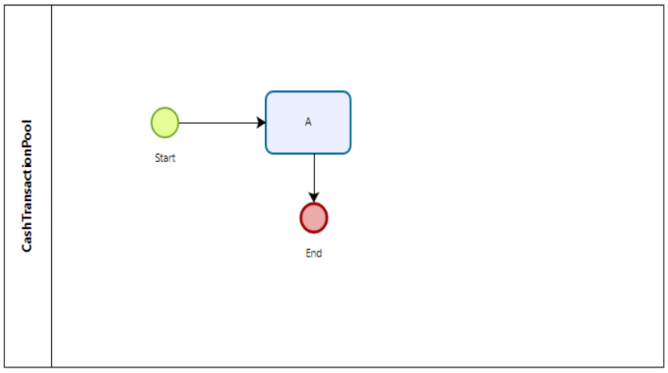
Perbandingan BPMN dilakukan menggunakan algoritma *Jaccard Similarity* untuk membandingkan tingkat kesesuaian jumlah elemen BPMN hasil *file* XPDL valid yang diimpor menggunakan Bizagi Modeler terhadap BPMN acuan (*Ground Truth*). Evaluasi kuantitatif ini difokuskan pada tiga komponen utama proses bisnis, yaitu kemiripan aktivitas (*Task Similarity*), kemiripan gerbang percabangan (*Gateway Similarity*), dan kemiripan alur koneksi (*Sequence Flow Similarity*). Proses perhitungan *Jaccard Similarity* dilakukan dengan mengekstrak komponen semantik utama dari *file* XPDL, yaitu Elemen Aktivitas (`<*Activity*>`), Elemen Percabangan (`<*Gateway*>`), dan Elemen Alur (`<*Transition*>`). Seluruh elemen tersebut dibersihkan dari sintaks XML mentah, lalu dimasukkan ke dalam bentuk himpunan token untuk dibandingkan menggunakan *Jaccard Similarity*.

$$Jaccard\ similarity\ = \frac{|A \cap B|}{|A \cup B|} \qquad (4.1)$$

Berikut cara menghitung *Jaccard Similarity* nya, sebagai contoh disini akan dicoba menghitung berdasarkan data 1.



Gambar 4.4 Model BPMN A (*Ground Truth)*



Gambar 4.5 Model BPMN B (Hasil *Import* dari Bizagi)

Berdasarkan gambar dari model A dan model B, model 1 memiliki elemen yaitu, (*start*, A, *end*) dan model 2 memiliki elemen yaitu, (*start*, A, B, *end*). Dari kedua model dihitung menggunakan *Jaccard Similarity* sebagai berikut.

$$Jaccard\ similarity\ = \frac{|A \cap B|}{|A \cup B|} = \frac{|3|}{|4|} = 0.75$$

Berdasarkan hasil pengujian pada *file-file* yang dinyatakan berstatus *valid* (✓), diperoleh rekapitulasi skor rata-rata *Jaccard Similarity* untuk setiap kombinasi model AI dan teknik *prompt engineering* sebagai berikut:

Tabel 4.2 Rincian Himpunan Data *Jaccard Similarity*

Dataset	**Model AI**	**Teknik Prompt**	**Total Elemen *Ground Truth* (A)**	**Total Elemen Hasil Generate (B)**	**Elemen Irisan A dan B ($A \cap B$)**	**Elemen Gabungan A dan B ($A \cup B$)**	**Skor *Jaccard Similarity***
Data 1	*GPT-4o-mini*	COT	4	4	4	4	1
Data 1	*GPT-4o-mini*	*Few-Shot*	4	4	4	4	1
Data 1	*GPT-4o-mini*	*Zero-Shot*	4	4	4	4	1
Data 1	*GPT-4o-mini*	TOT	4	4	4	4	1
Data 1	*Gemini 3.5 Flash Lite*	COT	4	4	4	4	1
Data 1	*Gemini 3.5 Flash Lite*	*Few-Shot*	4	4	4	4	1
Data 1	*Gemini 3.5 Flash Lite*	*Zero-Shot*	4	4	4	4	1
Data 1	*Gemini 3.5 Flash Lite*	TOT	4	4	4	4	1
Data 2	*GPT-4o-mini*	COT	5	5	5	5	1
Data 2	*GPT-4o-mini*	*Few-Shot*	5	4	4	5	0.8
Data 2	*GPT-4o-mini*	*Zero-Shot*	5	4	4	5	0.8
Data 2	*GPT-4o-mini*	TOT	5	4	4	5	0.8
Data 2	*Gemini 3.5 Flash Lite*	COT	5	5	5	5	1
Data 2	*Gemini 3.5 Flash Lite*	*Few-Shot*	5	5	5	5	1
Data 2	*Gemini 3.5 Flash Lite*	*Zero-Shot*	5	5	5	5	1

Dataset	**Model AI**	**Teknik Prompt**	**Total Elemen *Ground Truth* (A)**	**Total Elemen Hasil Generate (B)**	**Elemen Irisan A dan B ($A \cap B$)**	**Elemen Gabungan A dan B ($A \cup B$)**	**Skor *Jaccard Similarity***
Data 2	*Gemini 3.5 Flash Lite*	TOT	5	5	5	5	1
Data 3	*GPT-4o-mini*	COT	7	6	6	7	0.85
Data 3	*GPT-4o-mini*	*Few-Shot*	7	6	6	7	0.85
Data 3	*GPT-4o-mini*	*Zero-Shot*	7	6	6	7	0.85
Data 3	*GPT-4o-mini*	TOT	7	6	6	7	0.85
Data 3	*Gemini 3.5 Flash Lite*	*Zero-Shot*	7	6	6	7	0.85
Data 3	*Gemini 3.5 Flash Lite*	TOT	7	7	7	7	1
Data 4	*GPT-4o-mini*	COT	9	4	4	9	0.44
Data 4	*GPT-4o-mini*	*Few-Shot*	9	5	5	9	0.55
Data 4	*GPT-4o-mini*	*Zero-Shot*	9	4	4	9	0.44
Data 4	*GPT-4o-mini*	TOT	9	5	5	9	0.55
Data 4	*Gemini 3.5 Flash Lite*	*Few-Shot*	9	6	6	9	0.67
Data 4	*Gemini 3.5 Flash Lite*	TOT	9	7	7	9	0.78
Data 5	*GPT-4o-mini*	COT	10	6	6	10	0.6
Data 5	*GPT-4o-mini*	*Zero-Shot*	10	6	6	10	0.6

Dataset	**Model AI**	**Teknik Prompt**	**Total Elemen *Ground Truth* (A)**	**Total Elemen Hasil Generate (B)**	**Elemen Irisan A dan B ($A \cap B$)**	**Elemen Gabungan A dan B ($A \cup B$)**	**Skor *Jaccard Similarity***
Data 5	*GPT-4o-mini*	TOT	10	6	6	10	0.6
Data 5	*Gemini 3.5 Flash Lite*	*Few-Shot*	10	9	9	10	0.9
Data 6	*GPT-4o-mini*	COT	11	5	5	11	0.45
Data 6	*GPT-4o-mini*	*Few-Shot*	11	6	6	11	0.54
Data 6	*GPT-4o-mini*	*Zero-Shot*	11	5	5	11	0.45
Data 6	*GPT-4o-mini*	TOT	11	6	6	11	0.54
Data 6	*Gemini 3.5 Flash Lite*	TOT	11	8	8	11	0.72
Data 7	*GPT-4o-mini*	COT	6	5	5	6	0.83
Data 7	*GPT-4o-mini*	*Few-Shot*	6	5	5	6	0.83
Data 7	*GPT-4o-mini*	*Zero-Shot*	6	5	5	6	0.83
Data 7	*GPT-4o-mini*	TOT	6	5	5	6	0.83
Data 7	*Gemini 3.5 Flash Lite*	COT	6	6	6	6	1
Data 7	*Gemini 3.5 Flash Lite*	*Few-Shot*	6	6	6	6	1
Data 7	*Gemini 3.5 Flash Lite*	*Zero-Shot*	6	6	6	6	1
Data 7	*Gemini 3.5 Flash Lite*	TOT	6	6	6	6	1

Dataset	**Model AI**	**Teknik Prompt**	**Total Elemen *Ground Truth* (A)**	**Total Elemen Hasil Generate (B)**	**Elemen Irisan A dan B ($A \cap B$)**	**Elemen Gabungan A dan B ($A \cup B$)**	**Skor *Jaccard Similarity***
Data 8	*GPT-4o-mini*	COT	10	7	7	10	0.7
Data 8	*GPT-4o-mini*	*Few-Shot*	10	7	7	10	0.7
Data 8	*GPT-4o-mini*	TOT	10	7	7	10	0.7
Data 8	*Gemini 3.5 Flash Lite*	COT	10	8	8	10	0.8
Data 8	*Gemini 3.5 Flash Lite*	*Few-Shot*	10	7	7	10	0.7
Data 10	*GPT-4o-mini*	COT	14	6	6	14	0.42
Data 10	*GPT-4o-mini*	*Few-Shot*	14	8	8	14	0.57
Data 10	*GPT-4o-mini*	*Zero-Shot*	14	7	7	14	0.5
Data 10	*GPT-4o-mini*	TOT	14	7	7	14	0.5
Data 10	*Gemini 3.5 Flash Lite*	*Few-Shot*	14	12	12	14	0.85
Data 10	*Gemini 3.5 Flash Lite*	TOT	14	13	13	14	0.92

4.1.5 Perhitungan Standar Deviasi untuk Skor *Jaccard Similarity*

Selain nilai rata-rata, sebaran skor *jaccard similarity* pada setiap teknik *prompting* dihitung menggunakan standar deviasi. Standar deviasi menunjukkan seberapa jauh skor menyimpang dari rata-ratanya. Semakin kecil nilainya, semakin konsisten skor yang dihasilkan suatu teknik pada berbagai dataset. Karena data

yang digunakan hanya 10 *requirement* yang mewakili sebagian kemungkinan skenario, perhitungan memakai standar deviasi sampel (fungsi STDEV.S pada Microsoft Excel) dengan rumus berikut.

$$s = \sqrt{[\Sigma(x_i - \bar{x})^2 / (n - 1)]} \qquad (4.2)$$

Dengan s adalah standar deviasi sampel, x_i adalah skor *Jaccard Similarity* ke-i, $\bar{x}$ adalah rata-rata skor, dan n adalah jumlah skor. Perhitungan hanya menggunakan skor dari *file* yang *valid,* sehingga *Claude Haiku 4.5* tidak diikutkan karena seluruh *file* yang dihasilkan tidak *valid.* Standar deviasi tiap *prompting* akan ditampilkan pada tabel dibawah ini.

Tabel 4.3 Standar Deviasi Teknik *Chain Of Thought*

No	**Data**	**Model ai**	**Nilai *Jaccard Similarity***
1	data 1	*Gpt-4o-mini*	1
2	data 1	*Gemini 3.5 flash lite*	1
3	data 2	*Gpt-4o-mini*	1
4	data 2	*Gemini 3.5 flash lite*	1
5	data 3	*Gpt-4o-mini*	0,85
6	data 4	*Gpt-4o-mini*	0,44
7	data 5	*Gpt-4o-mini*	0,6
8	data 6	*Gpt-4o-mini*	0,45
9	data 7	*Gpt-4o-mini*	0,83
10	data 7	*Gemini 3.5 flash lite*	1
11	data 8	*Gpt-4o-mini*	0,7
12	data 8	*Gemini 3.5 flash lite*	0,8
13	data 10	*Gpt-4o-mini*	0,42
Nilai standar deviasi			0,23

Tabel 4.4 Standar Deviasi Teknik *Few Shot*

No	**Data**	**Model AI**	**Nilai jaccard similarity**
1	data 1	*Gpt-4o-mini*	1
2	data 1	*Gemini 3.5 flash lite*	1
3	data 2	*Gpt-4o-mini*	0,8

No	Data	Model AI	Nilai jaccard similarity
4	data 2	*Gemini 3.5 flash lite*	1
5	data 3	*Gpt-4o-mini*	0,85
6	data 4	*Gpt-4o-mini*	0,55
7	data 4	*Gemini 3.5 flash lite*	0,6
8	data 5	*Gemini 3.5 flash lite*	0,9
9	data 6	*Gpt-4o-mini*	0,54
10	data 7	*Gpt-4o-mini*	0,83
11	data 7	*Gemini 3.5 flash lite*	1
12	data 8	*Gpt-4o-mini*	0,7
13	data 8	*Gemini 3.5 flash lite*	0,7
14	data 10	*Gpt-4o-mini*	0,57
15	daata 10	*Gemini 3.5 flash lite*	0,85
Nilai standar deviasi			0,17

Tabel 4.5 Standar Deviasi Teknik *Zero Shot*

No	Data	Model AI	Nilai Jaccard Similarity
1	data 1	*Gpt-4o-mini*	1
2	data 1	*Gemini 3.5 flash lite*	1
3	data 2	*Gpt-4o-mini*	0,8
4	data 2	*Gemini 3.5 flash lite*	1
5	data 3	*Gpt-4o-mini*	0,85
6	data 3	*Gemini 3.5 flash lite*	0,85
7	data 4	*Gpt-4o-mini*	0,44
8	data 5	*Gpt-4o-mini*	0,6
9	data 6	*Gpt-4o-mini*	0,45
10	data 7	*Gpt-4o-mini*	0,83
11	data 7	*Gemini 3.5 flash lite*	1
12	data 10	*Gpt-4o-mini*	0,5
Nilai standar deviasi			0,22

Tabel 4.6 Standar Deviasi Teknik *Tree Of Thoughts*

No	Data	Model AI	Nilai Jaccard Similarity
1	data 1	*Gpt-4o-mini*	1
2	data 1	*Gemini 3.5 flash lite*	1
3	data 2	*Gpt-4o-mini*	0,8
4	data 2	*Gemini 3.5 flash lite*	1
5	data 3	*Gpt-4o-mini*	0,85
6	data 3	*Gemini 3.5 flash lite*	1
7	data 4	*Gpt-4o-mini*	0,55
8	data 4	*Gemini 3.5 flash lite*	0,78

No	Data	Model AI	Nilai Jaccard Similarity
9	data 5	*Gpt-4o-mini*	0,6
10	data 6	*Gpt-4o-mini*	0,54
11	data 6	*Gemini 3.5 flash lite*	0,72
12	data 7	*Gpt-4o-mini*	0,83
13	data 7	*Gemini 3.5 flash lite*	1
14	data 8	*Gpt-4o-mini*	0,7
15	data 10	*Gpt-4o-mini*	0,5
16	daata 10	*Gemini 3.5 flash lite*	0,92
Nilai standar deviasi set			0,18

Teknik *Chain of Thought* memiliki standar deviasi tertinggi, yaitu 0,23, dengan rata-rata skor 0,78. Skor teknik ini tersebar dari 0,42 hingga 1,00, atau rentang 0,58, yang merupakan rentang terlebar di antara keempat teknik. Skor sempurna (1,00) diperoleh pada Data 1, Data 2, dan Data 7. Skor yang rendah muncul pada Data 4 (0,44), Data 6 (0,45), dan Data 10 (0,42) pada GPT-4o-mini. Hal ini menunjukkan bahwa hasil *Chain of Thought* sangat bergantung pada kompleksitas dataset, sehingga kualitasnya kurang konsisten.

Teknik *Few Shot* memiliki standar deviasi terendah, yaitu 0,17, dengan rata-rata skor 0,80. Skor terendah teknik ini adalah 0,54 (Data 6, *GPT-4o-mini*), jauh lebih tinggi dibanding *Chain of Thought* dan *Zero Shot*, sehingga rentang skornya paling sempit (0,54 sampai 1,00). Contoh struktur yang diberikan pada prompt tampaknya membantu model menghasilkan elemen yang lebih seragam, bahkan pada dataset yang kompleks.

Teknik *Zero Shot* memiliki standar deviasi 0,22 dengan rata-rata 0,78. Skornya berkisar dari 0,44 hingga 1,00. Nilai ini mendekati *Chain of Thought*, dengan skor rendah pada Data 4 (0,44), Data 6 (0,45), dan Data 10 (0,50). Tanpa

contoh maupun arahan penalaran, hasil model lebih bervariasi mengikuti tingkat kesulitan dataset.

Teknik *Tree of Thought* memiliki standar deviasi 0,18 dengan rata-rata 0,80. Skor terendahnya 0,50 (Data 10, *GPT-4o-mini*), dan skor tertingginya 1,00. Nilai ini hanya terpaut 0,01 dari Few-Shot, sehingga kedua teknik tersebut termasuk yang paling konsisten.

Secara keseluruhan, urutan konsistensi dari yang terbaik adalah *Few Shot* (0,17), *Tree of Thought* (0,18), *Zero Shot* (0,22), dan *Chain of Thought* (0,23). Teknik yang memberikan struktur eksplisit, yaitu contoh pada *Few Shot* dan eksplorasi alternatif pada *Tree of Thought*, menghasilkan skor lebih stabil. Selisih rata-rata antarteknik sangat kecil (0,78 sampai 0,80), sehingga perbedaan keempat teknik terutama tampak pada konsistensi, bukan pada tingkat akurasi.

4.2 Pembahasan

Subbab ini membahas dan menafsirkan hasil pada subbab 4.1 untuk menjawab rumusan masalah penelitian. Pembahasan dilakukan dari empat sudut, yaitu pengaruh teknik *prompting*, pengaruh model AI, dan pengaruh struktur kalimat terhadap validitas dan akurasi *file* XPDL, serta penyebab *file* XPDL tidak valid. Hasil disertai angka dari Tabel 4.1 sampai Tabel 4.3 agar setiap temuan dapat ditelusuri, dan dibahas dengan mempertimbangkan keterbatasan masing-masing model dan metrik yang digunakan.

4.2.1 Pengaruh Teknik *Prompting*

Selain faktor Model AI, variasi teknik *prompt* yang diterapkan memberikan pengaruh terhadap tingkat keberhasilan model AI dalam menyusun struktur berkas XPDL secara akurat.

a. *Chain of Thought*

Teknik ini terbukti sangat stabil dalam memandu model AI karena menerapkan alur penalaran secara bertahap (*step-by-step reasoning*). Dengan menguraikan proses penulisan kode XML secara runtut, model mampu menerjemahkan urutan aktivitas teks ke dalam tag elemen XPDL dengan konsistensi tingkat validitas yang tinggi.

b. *Tree of Thought*

Teknik ini menunjukkan keunggulan tersendiri, terutama dalam menangani skenario dataset yang memiliki kompleksitas percabangan alur. Dengan mengevaluasi beberapa alternatif jalur pemikiran sebelum merumuskan bentuk struktur akhir, TOT membantu model meminimalkan kesalahan logika sintaksis pada berkas keluaran.

c. *Few-Shot*

Teknik ini memberikan hasil yang cukup baik dalam membantu model memahami pola struktur XPDL melalui contoh data yang diberikan di awal. Meskipun demikian, pada beberapa model tertentu, efektivitasnya masih sedikit di bawah COT dan TOT ketika dihadapkan pada skenario dataset yang kompleks.

d. *Zero-Shot*

Teknik *Zero-Shot* (tanpa contoh atau panduan tahapan berpikir) mencatatkan tingkat kegagalan yang paling sering. Tanpa adanya contoh struktur sintaksis atau arahan penalaran yang jelas, model AI memiliki risiko lebih tinggi dalam melakukan kesalahan penulisan format tag XML, sehingga sering menghasilkan berkas yang berstatus invalid

4.2.2 Pengaruh Model AI

Berdasarkan hasil pengujian terhadap 10 skenario dataset, tingkat keberhasilan penerjemahan teks menjadi file XPDL yang valid sangat dipengaruhi oleh arsitektur dan kapabilitas bawaan dari masing-masing model provider AI:

a. OpenAI (*GPT-4o-mini*)

Model ini menunjukkan tingkat stabilitas, konsistensi, dan kepatuhan instruksi yang paling tinggi di antara seluruh model yang diuji. Hampir seluruh dataset berhasil digenerate secara *valid* tanpa kendala sintaksis yang berarti, membuktikan bahwa *GPT-4o-mini* sangat handal dalam merumuskan struktur tag XML/XPDL yang terstandarisasi.

b. Google (*Gemini 3.5 Flash Lite*)

Model ini menunjukkan tingkat keberhasilan yang variatif. Pada beberapa dataset awal dan menengah, *Gemini* mampu menghasilkan berkas valid dengan baik. Namun, pada dataset dengan kompleksitas alur yang lebih tinggi, beberapa skenario mengalami kegagalan (*invalid*) akibat adanya ketidakonsistenan dalam penutupan tag elemen XML.

c. Anthropic (*Claude Haiku 4.5*)

Model ini mencatatkan tingkat keberhasilan paling rendah, di mana seluruh skenario pengujian menghasilkan status tidak *valid* (✗). Berdasarkan penelusuran sistem, hal ini disebabkan oleh karakteristik respons *Claude* yang sering menyertakan teks naratif atau penjelasan percabangan di luar struktur murni kode XML, sehingga gagal diproses oleh fungsi parser XPDL pada sistem website.

4.2.3 Pengaruh Struktur Kalimat

Pada *GPT-4o-mini*, validitas tetap tinggi di seluruh kelompok. Dua kegagalan dari 40 eksekusi, yaitu *Few-Shot* pada Data 5 dan *Zero-Shot* pada Data 9, berada pada kelompok kalimat kompleks, tetapi keduanya tidak mengikuti pola struktur tertentu. Pada *Gemini 3.5 Flash Lite*, pola yang paling jelas muncul. Kalimat majemuk diproses *valid* pada seluruh teknik, sedangkan kalimat kompleks hanya menghasilkan 5 dari 16 *file valid*. Kalimat majemuk kompleks pada Data 10 berada di antara keduanya, dengan 2 dari 4 *file valid*. Menariknya, kalimat sederhana justru lebih sulit bagi *Gemini* dibanding kalimat majemuk (66,7%). Kegagalan pada kelompok ini terkonsentrasi pada Data 3 dan Data 8, yang sama-sama memuat urutan kata kerja berurutan (*sequential verbs*) dan kata kunci *verb* untuk *datastore*. Data 1, yang tidak memuat fitur tersebut, berhasil diproses pada seluruh teknik. Hal ini menunjukkan bahwa label kelompok tidak sepenuhnya menentukan tingkat kesulitan, dan jenis operasi yang dijelaskan dalam kalimat juga berpengaruh. *Claude Haiku 4.5* gagal pada seluruh dataset di semua kelompok, sehingga kegagalannya tidak bergantung pada struktur kalimat, melainkan pada format respons model.

4.2.4 Perbandingan BPMN dengan Metode *Jaccard Similarity*

Berdasarakan tabel 4.2 dilakukan analisis hasil pengujian akurasi *Jaccard Similarity* secara mendetail untuk masing-masing dataset:

1. Analisis Dataset 1 (Kompleksitas Rendah: 4 Elemen *Ground Truth*)

Pada dataset pertama yang memiliki tingkat kompleksitas paling rendah dengan Total 4 elemen acuan, baik model *GPT-4o-mini* maupun *Gemini 3.5 Flash Lite* menunjukkan performa yang sangat optimal. Seluruh kombinasi teknik prompt (COT, *Few-Shot*, *Zero-Shot*, dan TOT) berhasil mencatatkan skor akurasi sempurna 1.00 (100%) dengan jumlah irisan elemen yang identik (4 elemen). Hal ini membuktikan bahwa pada struktur proses bisnis yang sederhana, kedua model AI mampu mengekstrak seluruh entitas tanpa mengalami kesalahan maupun kekurangan elemen.

2. Analisis Dataset 2 (Kompleksitas Rendah-Menengah: 5 Elemen *Ground Truth*)

Pada dataset kedua dengan 5 elemen acuan, *Gemini 3.5 Flash Lite* kembali mempertahankan konsistensi sempurna dengan skor 1.00 pada seluruh teknik prompt. Sementara itu, *GPT-4o-mini* memperoleh skor sempurna 1.00 hanya ketika menggunakan teknik *Chain of Thought* (COT). Namun, pada teknik *Few-Shot*, *Zero-Shot*, dan TOT, skor *GPT-4o-mini* sedikit menurun menjadi 0.80 akibat hanya mengekstrak 4 dari 5 elemen yang tersedia. Hal ini menunjukkan bahwa penalaran bertahap (COT) membantu *GPT-4o-mini* menangkap seluruh detail elemen secara lebih utuh.

3. Analisis Skenario Dataset 3 (Kompleksitas Menengah: 7 Elemen *Ground Truth*)

Memasuki dataset ketiga dengan 7 elemen acuan, *GPT-4o-mini* menunjukkan hasil yang seragam dan stabil di angka 0.85 (berhasil mengekstrak 6 dari 7 elemen) di seluruh variasi teknik prompt. Di sisi lain, *Gemini 3.5 Flash Lite* memperoleh skor 0.85 pada teknik *Zero-Shot*, dan meningkat mencapai tingkat kesempurnaan 1.00 saat dikombinasikan dengan teknik *Tree of Thought* (TOT). Kombinasi Gemini dengan TOT terbukti sangat efektif dalam memetakan seluruh elemen secara presisi pada tingkat kerumitan menengah ini.

4. Analisis Dataset 4 (Kompleksitas Menengah-Tinggi: 9 Elemen *Ground Truth*)

Pada dataset keempat dengan 9 elemen acuan, mulai terlihat adanya penurunan performa yang cukup signifikan, khususnya pada model *GPT-4o-mini* yang mencatatkan skor 0.44 (COT & *Zero-Shot*) hingga 0.55 (*Few-Shot* & TOT), di mana model hanya mampu memproduksi 4 sampai 5 elemen saja (under-generation). Sebaliknya, *Gemini 3.5 Flash Lite* menunjukkan ketahanan yang lebih baik dengan mencatatkan skor 0.67 (*Few-Shot*) dan 0.78 (TOT, menghasilkan 7 dari 9 elemen).

5. Analisis Dataset 5 (Kompleksitas Menengah-Tinggi: 10 Elemen *Ground Truth*)

Pada dataset kelima dengan 10 elemen acuan, *GPT-4o-mini* konsisten menghasilkan 6 dari 10 elemen di berbagai teknik prompt (COT, *Zero-Shot*, TOT), sehingga mencatatkan skor stabil di angka 0.60. Sebaliknya, *Gemini 3.5 Flash Lite*

yang diuji menggunakan teknik *Few-Shot* mampu mendongkrak pencapaian skor hingga 0.90 (berhasil mengekstrak 9 dari 10 elemen acuan dengan sangat baik).

6. Analisis Dataset 6 (Kompleksitas Tinggi: 11 Elemen *Ground Truth*)

Dataset keenam memiliki tingkat kerumitan yang tinggi dengan 11 elemen *Ground Truth*. *GPT-4o-mini* mengalami keterbatasan dalam merangkum teks yang panjang, sehingga hanya memperoleh skor berkisar antara 0.45 hingga 0.54 (hanya mampu menghasilkan 5 hingga 6 elemen). Sementara itu, *Gemini 3.5 Flash Lite* dengan teknik TOT berhasil unggul dan mencatatkan skor yang lebih tinggi, yaitu 0.72 (berhasil mengekstrak 8 dari 11 elemen secara akurat).

7. Analisis Dataset 7 (Kompleksitas Menengah: 6 Elemen *Ground Truth*)

Meskipun memiliki alur yang bercabang, dataset ketujuh dengan 6 elemen acuan kembali menunjukkan stabilitas yang sangat tinggi bagi *Gemini 3.5 Flash Lite*, di mana model ini sukses meraih skor sempurna 1.00 di seluruh variasi teknik prompt. Di sisi lain, *GPT-4o-mini* mencatatkan skor stabil di angka 0.83 (berhasil mendeteksi 5 dari 6 elemen) pada semua teknik prompt, menunjukkan selisih konsistensi yang tipis namun stabil.

8. Analisis Dataset 8 (Kompleksitas Tinggi: 10 Elemen *Ground Truth*)

Pada dataset kedelapan dengan 10 elemen acuan, *GPT-4o-mini* menunjukkan pola keluaran yang seragam dengan skor 0.70 (menghasilkan 7 elemen yang valid) pada teknik COT, *Few-Shot*, dan TOT. Adapun *Gemini 3.5 Flash Lite* menunjukkan performa yang sedikit lebih unggul pada penggunaan

teknik COT dengan perolehan skor 0.80 (8 elemen valid), dan skor 0.70 pada teknik *Few-Shot*.

9. Analisis Dataset 10 (Kompleksitas Kompleks / 14 Elemen *Ground Truth*)

Dataset kesepuluh merupakan skenario pengujian dengan tingkat kompleksitas tertinggi, yakni memuat 14 elemen *Ground Truth*. Pada dataset ini, batasan kemampuan model terlihat sangat jelas. *GPT-4o-mini* mengalami penurunan akurasi yang tajam dengan skor berkisar antara 0.42 hingga 0.57 (hanya mampu mengekstrak 6 sampai 8 elemen). Sebaliknya, *Gemini 3.5 Flash Lite* tampil sangat superior; teknik *Few-Shot* menghasilkan skor 0.85 (12 elemen), dan puncaknya pada kombinasi teknik *Tree of Thought* (TOT) yang berhasil mencatatkan skor akurasi mendekati sempurna sebesar 0.92 (13 dari 14 elemen berhasil diekstrak dengan presisi tinggi).

4.2.5 *File* XPDL Invalid

Selain mengevaluasi berkas yang berhasil, analisis mendalam juga dilakukan terhadap berkas-berkas yang mencatatkan status tidak valid selama proses generation. Kegagalan status validitas ini menunjukkan adanya kendala pada struktur sintaksis XML atau format keluaran model AI yang tidak sesuai dengan standar parser pada sistem website. Berdasarkan rekapitulasi pengujian, analisis terhadap data yang invalid dapat diuraikan sebagai berikut:

a. Pengaruh Karakteristik Respons Model AI:

Seluruh skenario pengujian pada model *Claude Haiku 4.5* menghasilkan status invalid. Hal ini disebabkan oleh kecenderungan model tersebut dalam menyertakan teks naratif, kalimat pengantar, atau penjelasan percakapan di luar

blok kode XML murni. Ketika teks non-XML tersebut disertakan, fungsi parser pada sistem gagal membaca struktur dokumen XPDL secara utuh, sehingga memicu kesalahan sintaksis (*syntax error*).

b. Kompleksitas Skenario dan Keterbatasan Model Ringan:

Pada beberapa model seperti *Gemini 3.5 Flash Lite* dan *GPT-4o-mini* menggunakan teknik *Zero-Shot*, kegagalan validitas umumnya ditemukan pada dataset dengan tingkat kerumitan alur yang tinggi. Tanpa adanya panduan struktur contoh (*Few-Shot*) atau kerangka berpikir logis (COT/TOT), model rentan mengalami kesalahan berupa tag XML yang tidak tertutup atau ketidaksesuaian hierarki elemen.

c. Efektivitas Teknik *Prompt* Terhadap Konsistensi Format:

Teknik *Zero-Shot* tercatat menghasilkan jumlah data *invalid* yang lebih banyak dibanding teknik berbasis penalaran bertahap. Hal ini membuktikan bahwa instruksi tekstual mentah tanpa panduan struktur atau contoh sintaksis yang jelas membuat model memiliki risiko lebih tinggi untuk melenceng dari aturan tata bahasa format *file* XPDL yang ketat

4.3 Keterkaitan Integrasi Islam

Hasil penelitian ini memiliki makna yang sejalan dengan beberapa nilai dalam Al-Qur'an, yaitu verifikasi informasi, proses bertahap, keadilan dalam pengukuran, dan kerendahan hati terhadap keterbatasan ilmu.

1. Verifikasi Informasi (*tabayyun*): QS. Al-Hujurat ayat 6

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِنْ جَاۤءَكُمْ فَاسِقٌۢ بِنَبَاٍ فَتَبَيَّنُوْٓا اَنْ تُصِيْبُوْا قَوْمًاۢ بِجَهَالَةٍ فَتُصْبِحُوْا عَلٰى مَا فَعَلْتُمْ نٰدِمِيْنَ ۝٦

Artinya: "Wahai orang-orang yang beriman! Jika seseorang yang fasik datang kepadamu membawa berita, maka telitilah kebenarannya, agar kamu tidak mencelakakan suatu kaum karena kebodohan (kecerobohan), yang akhirnya kamu menyesali perbuatanmu itu." (QS. Al-Hujurat: 6)

Ayat ini memerintahkan *tabayyun*, yaitu memeriksa kebenaran berita sebelum menerimanya .Hasil pengujian validitas menunjukkan bahwa keluaran model AI tidak boleh diterima begitu saja. Gemini menghasilkan *file* tidak valid pada banyak skenario, dan seluruh keluaran *Claude* tidak dapat diproses. Validasi XPDL, pembandingan dengan *ground truth* Bizagi, dan penghitungan *Jaccard Similarity* dalam penelitian ini dapat dipandang sebagai bentuk *tabayyun* terhadap keluaran AI.

2. Proses bertahap dalam prompting: QS. Al-Furqan ayat 32

وَقَالَ الَّذِيْنَ كَفَرُوْا لَوْلَا نُزِّلَ عَلَيْهِ الْقُرْاٰنُ جُمْلَةً وَّاحِدَةً ۛ كَذٰلِكَ ۛ لِنُثَبِّتَ بِهٖ ۖ فُؤَادَكَ وَرَتَّلْنٰهُ تَرْتِيْلًا ۝٣٢

Artinya: "Dan orang-orang kafir berkata, 'Mengapa Al-Qur'an itu tidak diturunkan kepadanya sekaligus?' Demikianlah, agar Kami memperteguh hatimu (Muhammad) dengan itu dan Kami membacakannya secara tartil (berangsur-angsur, benar, dan baik)." (QS. Al-Furqan: 32)

Ayat ini menunjukkan hikmah penyampaian secara bertahap dan tertata agar mudah dipahami dan dipegang teguh. Sebagai ibrah, pendekatan ini sejalan dengan teknik *Chain of Thought, Tree of Thought,* dan *Few-Shot*. Ketiganya memberi model arahan yang runtut dan contoh yang jelas, sedangkan *Zero-Shot* tanpa arahan memiliki tingkat kegagalan paling tinggi. Perlu dicatat bahwa ayat ini berbicara

tentang pewahyuan, sehingga kaitannya dengan *prompting* adalah analogi hikmah, bukan penafsiran langsung.

3. Keadilan dan ketelitian dalam pengukuran: QS. Ar-Rahman ayat 8-9

اَلَّا تَطْغَوْا فِى الْمِيْزَانِ ۝٨ وَاَقِيْمُوا الْوَزْنَ بِالْقِسْطِ وَلَا تُخْسِرُوا الْمِيْزَانَ ۝٩

Artinya: "Agar kamu jangan merusak keseimbangan itu. Dan tegakkanlah timbangan itu dengan adil dan janganlah kamu mengurangi neraca itu." (QS. Ar-Rahman: 8-9)

Ayat ini memerintahkan penimbangan yang adil dan tidak mengurangi neraca. Dalam penelitian ini, *Jaccard Similarity* berperan sebagai timbangan yang bertugas untuk menilai kesesuaian hasil generate dengan *ground truth*. Penilaian dilakukan dengan ukuran yang sama bagi semua model dan teknik, termasuk pelaporan hasil yang kurang baik seperti under-generation pada *GPT-4o-mini*, tanpa melebih-lebihkan atau mengurangi hasil.

4. Keterbatasan ilmu dan kerendahan hati: QS. Yusuf ayat 76

وَفَوْقَ كُلِّ ذِيْ عِلْمٍ عَلِيْمٌ ۝٧٦

Artinya: "...dan di atas setiap orang yang berpengetahuan ada Yang Maha Mengetahui." (QS. Yusuf: 76)

Ayat ini mengingatkan bahwa pengetahuan makhluk terbatas dan di atas setiap yang berilmu ada Yang Maha Mengetahui. Hal ini relevan dengan temuan bahwa tidak ada model yang sempurna. *GPT-4o-mini* unggul dalam keandalan tetapi lemah pada dataset kompleks. Gemini unggul pada kualitas struktur tetapi kurang andal, dan *Claude Haiku 4.5* terkendala format respons. Hasil ini juga

mengingatkan bahwa AI adalah alat bantu yang tetap memerlukan peran manusia, dalam hal ini *business analyst*, untuk memeriksa dan menyempurnakan hasilnya.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengembangan sistem, pengujian validitas, dan analisis terhadap akurasi struktur berkas XPDL, diperoleh beberapa kesimpulan sebagai berikut.

1. Sistem konfigurasi model proses bisnis berbasis website berhasil dibangun dan mampu mengonversi *textual requirement* menjadi *file* XPDL secara otomatis melalui integrasi dengan *OpenRouter* API. Sistem menyediakan fitur input teks persyaratan, pengaturan model AI dan teknik *Prompting*, proses *generate* XPDL, serta unduh *file* keluaran, sehingga menjawab rumusan masalah pertama penelitian ini.
2. Teknik *prompt engineering* yang diterapkan memberikan pengaruh terhadap validitas dan kualitas struktur *file* XPDL yang dihasilkan. Teknik *Few-Shot* dan *Tree of Thought (TOT)* secara konsisten menghasilkan tingkat validitas dan skor *Jaccard Similarity* yang lebih stabil dibandingkan teknik *Chain of Thought (COT)*, dan khususnya *Zero-Shot*, yang mencatatkan tingkat kegagalan sintaksis paling tinggi terutama pada dataset dengan kompleksitas kalimat tinggi. Hal ini menjawab rumusan masalah kedua penelitian ini.
3. Hasil perbandingan antara BPMN *ground truth* dan BPMN hasil impor dari *file* hasil sistem menggunakan Bizagi Modeler menunjukkan tingkat

kemiripan yang bervariasi, dengan skor *Jaccard Similarity* antara 0,42 hingga 1,00. *GPT-4o-mini* mencatatkan tingkat validitas keberhasilan generate tertinggi namun mengalami penurunan akurasi struktur (under-generation elemen) pada dataset kompleks; *Gemini 3.5 Flash Lite*, khususnya saat dikombinasikan dengan teknik *Tree of Thought*, menunjukkan ketahanan terbaik pada dataset kompleks dengan skor tertinggi 0,92; sedangkan *Claude Haiku 4.5* gagal pada seluruh skenario pengujian akibat menyertakan teks naratif non-XML dalam responsnya, sehingga tidak dapat diproses oleh parser sistem. Temuan ini menjawab rumusan masalah ketiga penelitian ini.

Secara umum, penelitian ini membuktikan bahwa keberhasilan otomasi konversi *textual requirement* menjadi model proses bisnis berformat XPDL menggunakan LLM sangat bergantung pada kombinasi pemilihan model AI dan teknik prompt engineering, serta pada tingkat kompleksitas kalimat input yang diberikan

5.2 Saran

Saran dibuat berdasarkan pengalaman dan pertimbangan penulis. Saran ditujukan kepada para peneliti dalam bidang sejenis yang ingin melanjutkan atau mengembangkan penelitian yang sudah diselesaikan, atau kepada pengguna hasil penelitian dalam rangka pemanfaatan penelitian. Berdasarkan kesimpulan dan keterbatasan penelitian, beberapa saran dapat disampaikan sebagai bahan pertimbangan untuk pengembangan penelitian selanjutnya serta peningkatan pemanfaatan sistem secara praktis.

1. Bagi peneliti selanjutnya, disarankan untuk memperluas jumlah dan variasi dataset *textual requirement* dari berbagai domain bisnis, serta melakukan replikasi (*multiple runs*) pada setiap kombinasi model dan teknik *Prompting* guna mengukur konsistensi dan variabilitas hasil secara statistik.
2. Disarankan menambahkan metrik evaluasi pelengkap yang memperhitungkan struktur graf proses bisnis, seperti *graph edit distance* atau *precision-recall* per jenis elemen, untuk melengkapi keterbatasan *Jaccard Similarity* dalam menangkap urutan dan relasi antarelemen.
3. Bagi pengembang sistem, perlu ditambahkan mekanisme *post-processing* atau validasi otomatis yang mampu membersihkan teks naratif di luar struktur XML sebelum proses parsing, mengingat karakteristik respons sebagian model AI (seperti *Claude Haiku 4.5*) yang cenderung menyertakan penjelasan tambahan di luar kode.
4. Berdasarkan hasil penelitian, teknik *Chain of Thought* dan *Tree of Thought* direkomendasikan sebagai pilihan utama teknik *Prompting* bagi pengembang maupun praktisi yang hendak mengimplementasikan otomasi konversi persyaratan tekstual menjadi model proses bisnis berbasis LLM, khususnya untuk skenario dengan kompleksitas kalimat tinggi.
5. Penelitian lanjutan juga disarankan untuk menguji sistem pada studi kasus nyata di organisasi atau perusahaan, disertai evaluasi *usability* dari

sisi pengguna (*business analyst*), guna mengukur dampak praktis sistem secara lebih komprehensif.

DAFTAR PUSTAKA

Aqimuddin, Ibrohim M, Seta M Pamungkas, Chandra Gunawan, and Muhammad Ainul Yaqin. 2023. "Analisis Kemiripan Model Proses Bisnis PMBoK Dan Scrum Menggunakan Metode Jaccard Coefficient Similarity Dan Semantic Similarity" 5 (2): 53–64.

Bannert, Maria, Enkelejda Kasneci, Kathrin Sessler, K Stefan, Frank Fischer, Urs Gasser, Georg Groh, et al. 2023. "ChatGPT for Good ? On Opportunities and Challenges of Large Language Models for Education," 1–13.

Brown, Tom B, Gretchen Krueger, Benjamin Mann, Amanda Askell, Ariel Herbert-voss, Clemens Winter, Daniel M Ziegler, Alec Radford, and Sam Mccandlish. 2020. "Language Models Are Few-Shot Learners," no. NeurIPS.

Busch, Kiran, Alexander Rochlitzer, Diana Sola, and Henrik Leopold. 2023. "Business Process Management," 1–8.

Caterina, Yessy. 2020. "PENGUKURAN KEMIRIPAN MODEL PROSES BISNIS PADA BEBERAPA E-MARKETPLACE DI INDONESIA MENGGUNAKAN PATH DAN JACCARD COEFFICIENT SIMILARITY."

Chang, Victor, Yian Chen, and Qianwen Ariel Xu. 2023. "Evaluation and Comparison of Various Business Process Management Tools," 2–33.

Chinosi, Michele, and Alberto Trombetta. 2012. "Computer Standards & Interfaces BPMN : An Introduction to the Standard" 34: 124–34. https://doi.org/10.1016/j.csi.2011.06.002.

Fairuzia, Arifah, Muhammad Ainul Yaqin, and Maulana Muhammad El-sulthan. 2020. "Pengembangan Model Proses Bisnis Berbasis Aliran Proses Dan Aliran Data," 1–11.

Geiger, Matthias, Simon Harrer, and Guido Wirtz. 2017. "BPMN 2.0: The State of Support and Implementation." https://doi.org/10.1016/j.future.2017.01.006.

Gomes, Poliana, Luiz Verçosa, Fagner Melo, Vinícius Silva, Carmelo Bastos Filho, and Byron Bezerra. 2022. "Artificial Intelligence-Based Methods for Business Processes: A Systematic Literature Review."

Iavarone, Benedetta, Dominique Brunato, and Felice Dell Orletta. 2021. "Kompleksitas Kalimat Dalam Konteks 1 Perkenalan," 186–99.

Ivančić, L. 2019. "BUSINESS PROCESS MODELLING VS. PROCESS MINING

A COMPARATIVE STUDY OF METHODS AND SOFTWARE TOOLS."

Kourani, Humam, Alessandro Berti, Daniel Schuster, and Wil M P Van Der Aalst. 2026. *Evaluating Large Language Models on Business Process Modeling : Framework , Benchmark , and Self-Improvement Analysis*. *Software and Systems Modeling*. Vol. 25. Springer Berlin Heidelberg. https://doi.org/10.1007/s10270-025-01318-w.

Licardo, Josip Tomo, Nikola Tanković, Darko Etinger, Josip Tomo Licardo, Nikola Tanković, and Darko Etinger. 2024. "ScienceDirect ScienceDirect Health and Social Care Information Systems and Technologies 2023 A Method for Extracting BPMN Models from Textual Descriptions A Method for Extracting BPMN Models Processing from Textual Descriptions Using Natural Language Using Natural Language Processing." *Procedia Computer Science* 239 (2023): 483–90. https://doi.org/10.1016/j.procs.2024.06.196.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. "Pre-Train , Prompt , and Predict : A Systematic Survey" 55 (9).

Maat, Henk Pander. 2017. "Zinslengte En Zinscomplexiteit" 39 (3): 297–328.

Niculescu, Virginia. 2023. "Business Process Representation Analysis in the RPA Context," 1600–1606. https://doi.org/10.1109/COMPSAC57700.2023.00247.

Pratiwi, eka hana, Moch. Sahri, Friska Ayu, muslikah nourma Rhomadhoni, and fitri rahma Dewi. 2025. "Inovasi Pembuatan Sistem Inspeksi Alat Pemadam Api Ringan Berbasis Website Di PT . Semen Indonesia Logistik" 7 (2): 290–97.

Rosa, La, Arthur H M Questionnaire-driven Configuration, Marcello La Rosa, Johannes Lux, Stefan Seidel, Marlon Dumas, and Arthur H M Hofstede. 2006. "Cover Sheet."

Saputro, Guntur Adi. 2021. "Perancangan Website Sebagai Media Informasi Dan Promosi Pada Fm Karaoke Salatiga" 1 (2): 12–21.

Setiawan, Pandu. 2021. "OTOMASI KONFIGURASI SISTEM INFORMASI BERDASARKAN ALIRAN DATA."

Sholiq, Sholiq, and Muhammad Ainul. 2024. "BPMN Diagram Dataset : Comprehensive Collection of Functional Requirements." *Data in Brief* 57: 110882. https://doi.org/10.1016/j.dib.2024.110882.

Sholiq, Sholiq, Riyanarto Sarno, and Endang Siti Astuti. 2022. "Jurnal Universitas King Saud – Ilmu Komputer Dan Informasi Menghasilkan Diagram BPMN

Dari Persyaratan Tekstual" 34: 10079–93. https://doi.org/10.1016/j.jksuci.2022.10.007.

Vaswani, Ashish. 2017. "Attention Is All You Need," no. Nips.

Vidgof, Maxim, Bachhofner Stefan, and Jan Mendling. 2023. "Large Language Models for Business Process Management: Opportunities and Challenges," no. 957402: 1–16.

Wei, Jason, Dale Schuurmans, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," no. NeurIPS: 1–14.

Weinzierl, Sven, Sandra Zilker, Sebastian Dunzer, and Martin Matzner. 2024. "Machine Learning in Business Process Management : A Systematic Literature Review." *Expert Systems With Applications* 253 (April): 124181. https://doi.org/10.1016/j.eswa.2024.124181.

Wenger, S., Spahic-Bogdanovic, M., & Martin, A. 2024. Large language models for democratizing business process modeling: BPMN model generation and style guide adherence. In Southern African Conference for Artificial Intelligence Research (pp. 372-389). Cham: Springer Nature Switzerland.

White, Jules, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-smith, and Douglas C Schmidt. 2023. "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT."

Yao, Shunyu, Dian Yu, Jeffrey Zhao, and Thomas L Griffiths. 2023. "Tree of Thoughts : Deliberate Problem Solving with Large Language Models," no. 1: 1–14.

Yaqin, Muhammad Ainul, Muqtadirul Majid, Fikri Fiuca Fradana, and Muhammad Rizal Mustofa. 2019. "Pertumbuhan Model Proses Bisnis Pada Permainan Hay Day Menggunakan Metode Regresi," 42–49.

Zamhari, Makhfud. 2019. "QUESTIONNAIRE DRIVEN UNTUK MENENTUKAN KONFIGURASI MODEL PROSES BISNIS SKRIPSI Oleh : MAKHFUD ZAMHARI."

Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, et al. 2026. "A Survey of Large Language Models," 1–144.