

BAB II

LANDASAN TEORI

2.1 Start Of The art

Tabel 2. 1 Perbandingan Penelitian Sebelumnya (*State of the Art*)

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
1	(Rahayu et al., 2024) DOI 10.31849/z n.v6i2.2053 8 URL https://journal.unilak.ac.id/index.php/zn/article/view/20538 ISSN 2656-7407	Implementasi Metode Waterfall pada Pengembangan Sistem Informasi Mobile E- Disarpus	Waterfall	Aplikasi mobile sistem informasi kearsipan dan perpustakaan (e-Disarpus) untuk Pemerintah Kabupaten Karanganyar	Penelitian ini mengimplementasikan metode Waterfall untuk mengembangkan aplikasi mobile bernama e-Disarpus guna mengatasi permasalahan keterbatasan akses, koleksi, jam buka layanan, fasilitas, ruang penyimpanan, dan risiko kerusakan arsip di lingkungan Pemerintah Kabupaten Karanganyar. Metode Waterfall dipilih karena pendekatannya yang sistematis dan berurutan, mencakup tahapan analisis kebutuhan,

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					desain sistem, implementasi, pengujian, dan pemeliharaan. Aplikasi dikembangkan menggunakan bahasa pemrograman Dart dengan kerangka kerja Flutter. Hasil penelitian menunjukkan bahwa aplikasi berhasil meningkatkan kualitas pelayanan kearsipan dan perpustakaan secara digital.
2	(Perdana et al., 2025) DOI 10.33395/sinkron.v9i4.15337 URL https://jurnal.polgan.ac.id/index.php/sinkron/article/view/15337 ISSN 2541-2019	Comparative Analysis of SDLC and R&D Methods in System Development: A Case Study of Integrity Zone Management System	SDLC Waterfall dan R&D (Studi Komparatif dengan Implementasi Langsung)	Sistem Informasi Manajemen Zona Integritas di lingkungan Universitas Negeri Medan	Penelitian ini menerapkan dan membandingkan secara langsung metodologi SDLC (model Waterfall) dan Research and Development (R&D) dalam membangun Sistem Informasi Manajemen Zona Integritas. Kedua metode diimplementasikan untuk mengembangk

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					an sistem nyata, kemudian dievaluasi berdasarkan lima dimensi utama: fleksibilitas, manajemen risiko, potensi inovasi, kebutuhan dokumentasi, dan keterlibatan pemangku kepentingan. Temuan utama menunjukkan bahwa metodologi SDLC memberikan struktur dan prediktabilitas yang sangat baik untuk proyek dengan kebutuhan yang terdefinisi dengan jelas, sedangkan pendekatan hibrida yang menggabungkan kekuatan kedua metodologi berhasil meningkatkan kepuasan pemangku

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					kepentingan hingga 94%.
3	(Sembiring et al., 2024) DOI 10.55537/gabdimas.v2i2.920 URL https://journal.aira.or.id/gabdimas/article/view/920 ISSN 3047-1230	Pengembangan Aplikasi Mobile Sistem Informasi Pengaduan Masyarakat dengan Pendekatan Waterfall dan Pengujian Black Box	Waterfall dan Black Box Testing	Aplikasi mobile sistem informasi pengaduan masyarakat	Penelitian ini mengembangkan aplikasi mobile sistem informasi pengaduan masyarakat menggunakan metode Waterfall. Perancangan sistem dilakukan melalui tahapan analisis kebutuhan, desain sistem, implementasi, dan pengujian. Penerapan teknologi sistem informasi pengaduan masyarakat bertujuan untuk meminimalisir keterbatasan pelayanan sehingga masyarakat dapat dengan mudah mengakses dan menyampaikan informasi dalam waktu singkat dari mana pun mereka berada. Pengujian sistem dilakukan menggunakan metode Black Box Testing yang membuktikan

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					bahwa seluruh fungsi sistem berjalan dengan baik
4	(Jannah, 2023) DOI 10.59134/jsk.v8i01.239 URL https://ojs-teknik.usni.ac.id/index.php/jsk/article/view/239 ISSN 2528-5033	Analisis Pengembangan Sistem Informasi Batik Madura Menggunakan Metode Waterfall dan Black Box Testing	Waterfall dan Black Box Testing	Aplikasi database batik Madura berbasis web menggunakan Framework Laravel	Aplikasi database batik Madura berbasis web menggunakan Framework Laravel Penelitian ini mengembangkan aplikasi sistem informasi batik Madura berbasis web menggunakan metode Waterfall yang terdiri dari lima tahapan: analisis kebutuhan, desain sistem, implementasi atau penulisan kode, pengujian, serta evaluasi dan pemeliharaan. Aplikasi dirancang untuk memudahkan pengelolaan data batik Madura, pelestarian, dan penyebaran informasi tentang motif batik Madura menggunakan Framework Laravel. Pengujian sistem dilakukan menggunakan

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					metode Black Box Testing dan seluruh fungsi aplikasi 100% berjalan sesuai perencanaan. Aplikasi yang dihasilkan bersifat user-friendly dengan tampilan antarmuka yang jelas dan intuitif, di mana pengguna dapat mencari dan melihat informasi tentang batik Madura sementara admin memiliki akses penuh untuk mengelola data.
5	(Halawa et al., 2024) DOI 10.62712/ju ktisi.v3i1.2 19 URL <a href="https://ejurnal.lkpkarya.prima.id/index.php/juk
tisi/article/view/219">https://ejurnal.lkpkarya.prima.id/index.php/juk tisi/article/view/219 ISSN 2963- 7104, 2962- 3022	Rancang Bangun Web Interaktif Pemasaran Penginapan Villa di Sibolangit Menggunakan Metode Waterfall	Waterfall	Website interaktif pemasaran penginapan villa di Sibolangit berbasis WordPress	Penelitian ini mengembangkan website interaktif untuk media promosi penginapan villa di Sibolangit menggunakan metode Waterfall yang meliputi tahapan analisis kebutuhan, perancangan, pengkodean, dan pengujian. Pada tahap analisis

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					kebutuhan, ditemukan bahwa calon pelanggan membutuhkan akses informasi yang cepat dan mudah mengenai fasilitas, harga, dan ketersediaan penginapan, serta integrasi dengan media sosial. Tahap perancangan menghasilkan desain antarmuka pengguna yang intuitif dan responsif serta desain database yang efisien. Implementasi dilakukan menggunakan <i>platform</i> WordPress dengan tambahan plugin untuk fitur reservasi online dan galeri foto. Hasil pengujian menunjukkan semua fitur utama berfungsi

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					dengan baik dan website responsif di berbagai perangkat seperti komputer, tablet, dan smartphone.
6	(Saravanos & Curinga, 2023) DOI 10.3390/asi6060108 URL https://www.mdpi.com/2571-5577/6/6/108 ISSN 2571-5577	Simulating the Software Development Lifecycle: The Waterfall Model	Waterfall	Tahapan pengembangan perangkat lunak sekuensial (analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan)	Penelitian ini mensimulasikan seluruh tahapan model Waterfall secara sekuensial yang terdiri dari analisis kebutuhan, perancangan sistem, implementasi kode, pengujian, dan pemeliharaan. Studi ini menganalisis alokasi sumber daya, manajemen waktu, serta dinamika tim pada setiap fase untuk membuktikan efektivitas pendekatan Waterfall dalam pengelolaan proyek perangkat lunak secara terstruktur.

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					Hasil simulasi mengonfirmasi bahwa model Waterfall sangat efektif untuk proyek dengan spesifikasi kebutuhan yang stabil dan terdefinisi sejak awal, serta menghasilkan dokumentasi yang komprehensif di setiap tahapan.
7	(Arizka Anggraini et al., 2024) DOI 10.61132/neptunus.v2i2.92 URL https://jurnal.artei.or.id/index.php/Neptunus/article/view/92 ISSN 3031-898X, 3031-8998	Pemodelan Aplikasi Pengadaan Barang Menggunakan Metode Waterfall	Waterfall	Aplikasi pemodelan pengadaan barang pada perusahaan	Penelitian ini mengembangkan model aplikasi pengadaan barang yang lebih efisien dan efektif menggunakan metodologi Waterfall. Latar belakang penelitian adalah proses pengadaan barang tradisional yang sering dianggap kurang efisien, memakan waktu lama, berbiaya tinggi, dan kurang

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					transparan sehingga menimbulkan risiko penyalahgunaan. Metode penelitian menggunakan observasi untuk mempelajari proses pengadaan yang berjalan saat ini dan wawancara untuk memperoleh masukan dari pemangku kepentingan mengenai kebutuhan aplikasi. Hasil penelitian menunjukkan bahwa model aplikasi pengadaan barang yang dikembangkan dapat membantu perusahaan menghemat waktu dan biaya, serta meningkatkan transparansi dan akuntabilitas dalam proses pengadaan.

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
8	(Mintarsih, 2023) DOI 10.47233/jt eksis.v5i1.7 27 URL http://jurnal.unidha.ac.id/index.php/jteksis/article/view/727 ISSN 2655-8238, 2964-2132	Pengujian Black Box Dengan Teknik Transition Pada Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Waterfall Pada SMC Foundation	Waterfall dan Black Box Testing (teknik State Transition)	Sistem informasi perpustakaan berbasis web pada SMC Foundation	Penelitian ini mengembangk an sistem informasi perpustakaan berbasis web pada SMC Foundation menggunakan metode Waterfall dan melakukan pengujian menggunakan metode Black Box Testing dengan teknik State Transition. Teknik State Transition dipilih karena mampu mengidentifika si kesalahan pada fungsi, antarmuka, model informasi, dan akses ke sumber informasi luar. Pengujian dilakukan dengan melihat kesesuaian alur dari satu pipeline ke pipeline berikutnya untuk memastikan bahwa perangkat

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					lunak yang diuji telah melewati tahapan pengujian secara menyeluruh. Hasil pengujian membuktikan bahwa seluruh alur sistem berjalan sesuai dengan fungsi yang diharapkan, memvalidasi bahwa pendekatan Waterfall yang digunakan dalam proses pengembangan berhasil menghasilkan sistem yang fungsional dan handal.
9	(Meriana Malo et al., 2025) DOI 10.62951/router.v3i3.640 URL https://jurnal.aptii.or.id/index.php/Router/article/view/640 ISSN 3032-3312, 3026-3611	Pengembangan Aplikasi E-Learning Berbasis Web Menggunakan Model Waterfall di Sekolah Dasar Masehi Tanggaba	Waterfall dan Black Box Testing	Aplikasi e-learning berbasis web untuk Sekolah Dasar Masehi Tanggaba	Penelitian ini mengembangkan aplikasi e-learning berbasis web di Sekolah Dasar Masehi Tanggaba menggunakan model Waterfall. Model Waterfall dipilih karena menyediakan alur kerja

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					sistematis melalui tahapan analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Aplikasi yang dihasilkan memiliki fitur utama berupa pengelolaan materi pembelajaran, kuis interaktif, forum diskusi, dan penilaian otomatis. Guru dapat mengunggah materi dan tugas secara terstruktur, sementara siswa dapat mengakses konten pembelajaran dan mengerjakan latihan secara lebih fleksibel. Pengujian sistem dilakukan menggunakan metode Black Box Testing untuk memastikan setiap fungsi berjalan sesuai

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					kebutuhan pengguna. Hasil pengujian membuktikan bahwa aplikasi berfungsi dengan baik, praktis digunakan, dan efektif dalam mendukung proses pembelajaran.
10	(Sugiyanto et al., 2025) DOI 10.33005/santika.v5i1.538 URL https://santika.upnjatim.ac.id/submissions/index.php/santika/article/view/538 ISSN 2747-0563	Rancang Bangun Aplikasi Web Profil untuk UMKM Aquavita II Menggunakan Model Waterfall	Waterfall dan Black Box Testing	Aplikasi web profil untuk UMKM Aquavita II	Penelitian ini merancang dan mengimplementasikan aplikasi web profil untuk UMKM Aquavita II menggunakan model pengembangan perangkat lunak Waterfall. Model ini dipilih karena kemampuannya dalam memastikan analisis kebutuhan dan desain yang komprehensif sebelum tahap implementasi, melalui fase sistematis yang mencakup analisis kebutuhan,

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Ringkasan Isi Jurnal
					desain, pengkodean, pengujian, dan penerapan program pemeliharaan. Hasil implementasi menghasilkan aplikasi web profil dengan dua fitur utama: menyediakan akses mudah bagi konsumen terhadap informasi produk, promosi, lokasi, dan kontak bisnis, serta memungkinkan pemilik usaha untuk mempromosikan penawaran secara efektif. Pengujian fungsional dengan metode Black Box Testing mengonfirmasi bahwa semua fitur beroperasi sesuai dengan spesifikasi yang ditetapkan.

Tabel 2. 2 Tabel Posisi Peniltian

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Posisi Penelitian
1	(Reza Riyadhusolihin, 2026)	Pengembangan Aplikasi Desktop Otomatisasi Metadata Aset Digital pada <i>Platform</i> Microstock menggunakan Metode Waterfall di Studio RZ Digital ID	Waterfall	Aplikasi desktop otomatisasi metadata microstock multi- <i>platform</i> berbasis AI	Penelitian ini mengambil posisi sebagai sistem terintegrasi yang menerapkan <i>Multimodal Large Language Model</i> (MLLM) berarsitektur <i>Transformer</i> untuk otomatisasi analisis visual (image/video-to-text). Secara fungsional, posisi sistem ini difokuskan untuk menghasilkan metadata (judul, kata kunci, kategori) secara massal yang hasil keluarannya langsung disesuaikan secara dinamis dengan standar 4 <i>platform</i>

No	Peneliti, (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Posisi Penelitian
					microstock berbeda (Adobe Stock, Shutterstock, Freepik, dan Vecteezy).

Posisi penelitian ini berada pada pengembangan aplikasi desktop yang mengintegrasikan teknologi *Multimodal Large Language Model* (MLLM) untuk mengotomatisasi pembuatan metadata aset digital berupa *title*, *keyword*, dan *category*. Pengembangan sistem secara umum, penelitian ini memanfaatkan kemampuan MLLM dalam memahami konten visual untuk menghasilkan metadata secara otomatis. Selain itu, penelitian ini mengimplementasikan penyesuaian format metadata sesuai kebutuhan beberapa *platform* microstock, yaitu Adobe Stock, Shutterstock, Freepik, dan Vecteezy, dalam satu sistem yang terintegrasi.

Tabel 2. 3 Perbandingan Metode Pengembangan Sistem

Aspek	Waterfall	Agile	Prototyping
Pendekatan	Linear dan bertahap (sekuensial). Setiap tahap harus selesai sebelum tahap berikutnya dimulai.	Iteratif, fleksibel, dan kolaboratif. Terbagi dalam siklus-siklus pendek (sprints).	Berbasis prototipe. Pembuatan rancangan awal untuk dievaluasi oleh pengguna sebelum sistem sebenarnya dibangun.
Perubahan	Sulit menerima perubahan setelah tahap awal (analisis kebutuhan) disetujui dan diselesaikan.	Sangat mudah beradaptasi terhadap perubahan kebutuhan pengguna di tengah proses pengembangan.	Cukup adaptif karena perubahan dilakukan berdasarkan feedback pengguna terhadap prototipe.
Dokumentasi	Sangat terstruktur, komprehensif, dan rinci di setiap	Minimal, lebih fokus pada komunikasi langsung dan	Dokumentasi sering kali terabaikan karena fokus pada

Aspek	Waterfall	Agile	Prototyping
	tahapan pengembangan.	produk perangkat lunak yang berfungsi (kode).	perbaikan prototipe secara cepat.
Keterlibatan Pengguna	Terutama di awal (pengumpulan kebutuhan) dan di akhir (pengujian)	Terlibat secara aktif dan intensif di setiap iterasi/siklus.	Terlibat penuh selama proses evaluasi prototipe.
Cocok untuk	Proyek dengan spesifikasi kebutuhan yang jelas, stabil, dan terdefinisi dengan baik sejak awal.	Proyek dinamis yang kebutuhannya sering berubah atau belum jelas di awal.	Proyek di mana pengguna kesulitan mendefinisikan kebutuhan sistem di awal secara rinci.

Berdasarkan perbandingan metode pengembangan sistem pada tabel di atas, terlihat bahwa setiap metode memiliki keunggulan dan karakteristiknya masing-masing. Metode Agile dan Prototyping menawarkan fleksibilitas yang tinggi terhadap perubahan, namun membutuhkan keterlibatan pengguna yang intensif serta memiliki fokus dokumentasi yang lebih minim.

Dalam penelitian ini, metode Waterfall dipilih sebagai pendekatan pengembangan sistem. Hal ini dikarenakan aplikasi otomatisasi metadata aset digital yang dibangun memiliki spesifikasi dan alur kerja (pemrosesan gambar, pembangkitan metadata dengan *API Transformer*, hingga modifikasi basis data SQLite) yang sudah terdefinisi secara jelas, stabil, dan spesifik sejak tahap awal. Metode Waterfall menawarkan pendekatan yang sistematis dan linear, memastikan bahwa setiap tahapan seperti analisis kebutuhan, perancangan antarmuka, hingga implementasi logika program diselesaikan secara matang sebelum melangkah ke tahap pengujian (*Black Box Testing*). Selain itu, dokumentasi yang terstruktur dan komprehensif pada metode Waterfall sangat penting dalam penyusunan penelitian akademis ini, sehingga setiap fase pengembangan dapat dilacak dan dipertanggung jawabkan dengan baik.

2.2 Kajian Teori

Kajian teori pada penelitian ini mencakup berbagai konsep dasar dan landasan ilmiah yang berkaitan dengan otomasi metadata, teknologi kecerdasan buatan, serta metodologi pengembangan perangkat lunak yang digunakan.

2.2.1 Industri Microstock

Microstock merupakan sebuah model bisnis dalam industri kreatif yang mengelola penjualan lisensi aset digital seperti foto, vektor, ilustrasi, dan video melalui *platform* online dengan harga yang relatif terjangkau namun dalam volume penjualan yang tinggi (Afriansyah & Saputra, 2025). Menurut *Adobe Inc.* (2024), *platform* microstock bertindak sebagai perantara antara kreator (kontributor) dengan konsumen global yang membutuhkan aset visual untuk keperluan komersial maupun editorial. Keberhasilan seorang kontributor dalam industri ini sangat bergantung pada kuantitas dan kualitas aset, serta kemudahan aset tersebut untuk ditemukan oleh calon pembeli di dalam pangkalan data *platform* yang sangat besar.

- a. Adobe Stock merupakan *platform* microstock premium yang dimiliki oleh *Adobe Inc.* dan terintegrasi secara langsung dengan ekosistem perangkat lunak *Adobe Creative Cloud* seperti *Photoshop*, *Illustrator*, dan *Premiere Pro*. Integrasi ini memberikan keunggulan bagi kontributor karena karya yang diunggah dapat dicari dan dibeli langsung oleh jutaan profesional dari dalam aplikasi kerja mereka. Untuk mendukung integrasi tersebut, Adobe Stock menerapkan standar metadata CSV menggunakan pemisah koma (,) dengan pengodean karakter UTF-8 dengan BOM (*Byte Order Mark*) untuk menjamin karakter teks tidak mengalami kerusakan saat diimpor. Struktur kolom berkas CSV Adobe Stock wajib terdiri dari *Filename* untuk nama berkas tanpa kutipan, *Title* untuk judul aset yang diapit tanda kutip dengan panjang maksimal 200 karakter, *Keywords* berisi maksimal 49 kata kunci yang diapit tanda kutip, serta kolom *Category*

berupa kode angka untuk mencocokkan aset dengan sistem kategorisasi internal Adobe Stock.

- b. Shutterstock didirikan pada tahun 2003 dan dikenal sebagai salah satu *platform* agensi microstock terbesar di dunia dengan basis pelanggan global yang mencakup agensi periklanan, media massa, hingga korporasi besar. Karena skalanya yang besar, Shutterstock menerapkan sistem peninjauan (*review*) aset yang sangat ketat untuk memastikan relevansi pencarian pembeli. Berkas metadata massal yang diterima oleh Shutterstock harus berupa *CSV* berformat *UTF-8* dengan *BOM* dengan pemisah koma (.). Struktur kolomnya terdiri dari *Filename*, *Description* sebagai judul deskriptif aset yang diapit tanda kutip ganda, *Keywords* berkisar antara 7 hingga 50 kata kunci, *Categories* yang berisi maksimal dua kategori relevan dari daftar resmi Shutterstock yang dipisahkan oleh tanda koma, serta tiga parameter status hukum dan jenis aset yang secara *default* diset bernilai "no", yaitu kolom *Editorial*, *Mature content*, dan *Illustration*.
- c. Freepik adalah *platform* microstock asal *Spain* yang didirikan pada tahun 2010 dan mendominasi pasar penyedia aset grafis berbasis vektor, ilustrasi, dan foto dengan model bisnis *freemium*. Seiring dengan perkembangan teknologi kecerdasan buatan, Freepik menjadi salah satu *platform* pertama yang mewajibkan kontributor untuk mendeklarasikan karya yang dibuat menggunakan bantuan *AI* generatif secara transparan demi melindungi hak cipta dan keaslian karya. Regulasi ini berdampak pada berkas metadata Freepik yang menggunakan format *CSV* khusus dengan pemisah titik koma (;) dan pengodean *UTF-8* Standar tanpa *BOM*. Kolom-kolom di dalam berkas *CSV* Freepik terdiri dari *Filename*, *Title*, *Keywords*, serta dua kolom wajib tambahan khusus untuk karya hasil kecerdasan buatan, yaitu *Prompt* untuk mencantumkan teks perintah *AI* dan *Model* untuk mencantumkan nama mesin kecerdasan buatan yang digunakan, di mana seluruh nilai kolom tersebut wajib diapit tanda kutip ganda.

- d. Vecteezy merupakan *platform* microstock asal *Amerika Serikat* yang berfokus pada penyediaan visual berupa grafik vektor, foto, dan cuplikan video bebas royalti untuk kebutuhan kreator konten independen. Vecteezy memiliki karakteristik *parser* pengunggah CSV yang sangat sensitif terhadap format berkas, di mana penggunaan penanda *BOM* pada baris pertama dapat menyebabkan kegagalan sistem dalam mengenali kolom nama berkas. Oleh karena itu, Vecteezy menyyaratkan berkas CSV dengan pengodean *UTF-8* Standar tanpa *BOM* menggunakan pemisah koma (.). Berkas ini terdiri dari kolom *Filename*, *Title* untuk judul, *Description* untuk deskripsi yang diisi dengan nilai yang sama dengan judul, *Keywords* sebagai satu-satunya kolom yang wajib diapit tanda kutip ganda, serta *License* untuk menentukan jenis lisensi aset yang ditulis dalam huruf kecil seperti *free*, *pro*, atau *editorial*. Untuk mencegah kerusakan pembacaan kolom akibat pemisah koma, karakter koma di dalam judul dan deskripsi otomatis dibersihkan sebelum diekspor.

2.2.2 Metadata Aset Digital

Metadata secara umum didefinisikan sebagai data yang memberikan informasi mengenai data lainnya. Dalam ekosistem microstock, metadata merupakan informasi tekstual yang melekat pada sebuah file digital yang mencakup tiga elemen utama: *Adobe Inc. (2026)*

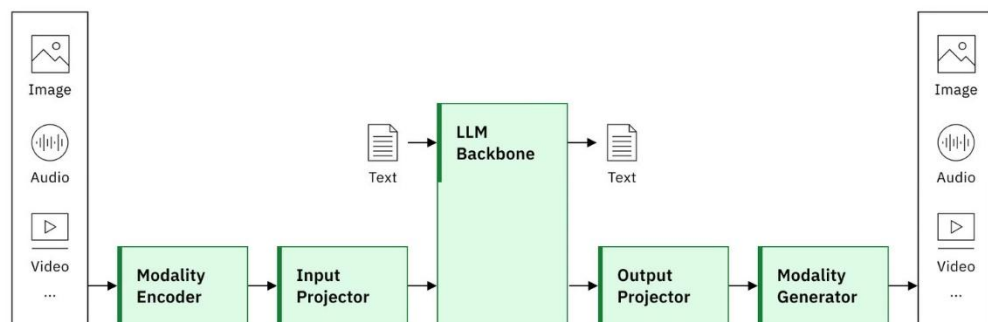
1. **Judul (*Title/Description*):** Rangkaian kata yang mendeskripsikan subjek utama aset secara ringkas.
2. **Kata Kunci (*Keywords/Tags*):** Kumpulan istilah yang digunakan oleh mesin pencari *platform* untuk mencocokkan kueri pembeli dengan aset yang tersedia.
3. **Kategori (*Category*):** Klasifikasi tematik yang berfungsi mengelompokkan aset ke dalam topik atau industri spesifik untuk mempersempit hasil pencarian calon pembeli. Penentuan kategori ini harus dilakukan secara cermat, mengingat setiap *platform* microstock

memiliki standar pengelompokan dan format kategori yang berbeda-beda.

2.2.3 Multimodal Large Language Model (MLLM) dan Algoritma Transformer

Multimodal Large Language Model (MLLM) merupakan evolusi dari *Large Language Model (LLM)* standar yang dirancang untuk menerima, memproses, dan memahami kombinasi dari berbagai jenis data (modalitas) secara bersamaan, terutama gambar visual dan teks instruksi (*prompt*) (Yin et al., 2024). Kemampuan canggih ini didukung penuh oleh fondasi Algoritma *Transformer* yang pertama kali diperkenalkan (Vaswani et al., 2017). Algoritma ini merupakan arsitektur jaringan saraf tiruan yang memproses seluruh elemen masukan secara paralel melalui mekanisme utama yang disebut *Self-Attention*.

Secara konseptual, mekanisme *Self-Attention* pada algoritma *Transformer* memungkinkan sistem untuk memindai keseluruhan gambar, mendeteksi hubungan antar-objek di dalam gambar tersebut, lalu memutuskan bagian piksel mana yang paling relevan dengan instruksi teks yang diberikan oleh pengguna. Mekanisme inilah yang menjadi mesin pendorong utama kecerdasan buatan dalam mengekstraksi informasi visual mentah menjadi data teks bahasa alami manusia.



Gambar 2.1 Proses Algoritma *Transformer*

Berdasarkan Gambar 2.1, alur pemrosesan data multimodal terbagi ke dalam beberapa tahapan dan modul utama. Dalam konteks pengembangan aplikasi RZ Autometadata, sistem bekerja memproses sisi masukan (*input modalities*) hingga menghasilkan keluaran teks (*text output*), dengan tahapan cara kerja sebagai berikut:

1. **Masukan Modalitas (*Input Modalities*)** Proses analisis dimulai ketika pengguna (kontributor microstock) mengunggah aset digital berupa *Image* (gambar) atau *Video* ke dalam sistem. Data visual mentah ini merupakan bentuk data yang belum terstruktur dan belum dapat dipahami secara komputasional oleh model bahasa.
2. ***Modality Encoder*** Modul ini bertugas menerima masukan visual tersebut dan mengubahnya menjadi sekumpulan vektor numerik (*visual features*). Pada tahap inilah algoritma *Transformer* versi visual (*Vision Transformer*) bekerja untuk mengekstrak informasi detail dari gambar, seperti komposisi warna, pencahayaan, bentuk, dan objek utama yang mendominasi piksel aset digital.
3. ***Input Projector*** Vektor fitur visual yang telah diekstrak oleh *Encoder* masih memiliki dimensi matematis yang berbeda dengan pemrosesan bahasa teks. *Input Projector* berfungsi sebagai jembatan penerjemah yang menyelaraskan fitur visual tersebut ke dalam "ruang

pemahaman" yang sama dengan teks, sehingga siap dibaca dan diolah oleh model bahasa.

4. **LLM Backbone (Tulang Punggung Utama LLM)** Modul ini adalah pusat otak pemrosesan. Pada tahap ini, fitur visual yang telah diselaraskan akan digabungkan dengan instruksi teks (*Text Prompt*) dari sistem. Pada aplikasi RZ Autometadata, instruksi teks ini berupa perintah *Prompt Engineering* yang disesuaikan dengan aturan masing-masing *platform* (contoh: "Beri judul maksimal 200 karakter dan 50 kata kunci relevan untuk *platform* Adobe Stock"). Berdasarkan penggabungan data visual dan teks perintah tersebut, *LLM Backbone* akan mengkalkulasinya melalui mekanisme *Self-Attention* dan menghasilkan keluaran akhir (*output*) berupa teks metadata (judul dan kata kunci) yang akurat dan siap diunduh dalam format CSV.

2.2.4 Bahasa Pemrograman Python

Python adalah bahasa pemrograman berorientasi objek yang berfungsi sebagai *scripting language*. Bahasa pemrograman ini dapat digunakan untuk pengembangan perangkat lunak dan dapat dijalankan di berbagai sistem operasi. Saat ini, Python juga populer di bidang data science dan analisis data. Popularitas ini didukung oleh berbagai library Python yang menyediakan fungsi-fungsi analisis data, machine learning, alat-alat untuk preprocessing data, serta visualisasi data. (Yulisa Geni et al., 2024).

2.2.5 Sistem Manajemen Pangkalan Data SQLite

SQLite merupakan sebuah sistem manajemen pangkalan data relasional yang bersifat mandiri serta tidak memerlukan proses server secara terpisah dari perangkat lunak utama. Berbeda dengan sistem pangkalan data tradisional yang menuntut konfigurasi jaringan yang rumit, SQLite menyimpan seluruh kumpulan data ke dalam satu *file* tunggal pada penyimpanan lokal komputer pengguna. Karakteristik tersebut menjadikan SQLite sebagai teknologi yang sangat ideal untuk pengembangan aplikasi

desktop yang memprioritaskan efisiensi ruang penyimpanan serta kecepatan akses data tanpa harus bergantung pada koneksi internet. Dalam arsitektur rekayasa perangkat lunak, penggunaan SQLite dapat menjamin integritas data secara konsisten, serta memudahkan proses distribusi aplikasi karena pangkalan data tersebut menyatu secara utuh dengan sistem tanpa memerlukan instalasi tambahan. Hasil evaluasi kinerja menunjukkan bahwa SQLite memiliki keunggulan signifikan dalam menangani proses transaksional seperti operasi *insert*, *update*, *delete*, dan *select* dibandingkan dengan sistem pangkalan data lainnya, sehingga sangat sesuai untuk diimplementasikan pada aplikasi yang membutuhkan responsivitas tinggi dalam pemrosesan data lokal (Nugraha & A. Susetyo, 2023). Menurut dokumentasi resmi *SQLite* (Hipp, 2024), *SQLite* menerapkan sistem *Dynamic Type Affinity* yang hanya mengenal lima kelas penyimpanan utama (*Storage Class*), yaitu *NULL*, *INTEGER*, *REAL*, *TEXT*, dan *BLOB*.

2.2.6 Antarmuka Pemrograman Aplikasi

Antarmuka Pemrograman Aplikasi atau dikenal dengan singkatan API (*Application Programming Interface*) merupakan sekumpulan protokol serta instruksi komputasi yang menjembatani komunikasi antar dua sistem perangkat lunak yang berbeda. Dalam arsitektur sistem yang terintegrasi dengan kecerdasan buatan, API berfungsi layaknya kurir data yang bertugas membawa instruksi visual dari aplikasi lokal menuju pemrosesan Multimodal *Large Language Model*, untuk kemudian membawa kembali hasil ekstraksi teks tersebut ke dalam sistem lokal. Penggunaan arsitektur berbasis API ini memungkinkan aplikasi desktop untuk melakukan komputasi analisis gambar tingkat tinggi tanpa membebani kinerja perangkat keras pada komputer milik pengguna secara berlebihan. Pengembangan API menjadi elemen krusial dalam sistem berbasis kecerdasan buatan karena mampu memfasilitasi proses integrasi, distribusi, serta pemantauan kinerja sistem secara efisien dan berkelanjutan (Bagus Syafiq Faqihuddin et al., 2025)

2.2.7 Antarmuka Pengguna Grafis

Antarmuka Pengguna Grafis atau *Graphical User Interface* adalah medium interaksi visual yang memungkinkan pengguna untuk mengoperasikan perangkat lunak melalui elemen visual seperti tombol perintah, kolom teks masukan, serta menu navigasi, tanpa harus mengingat maupun mengetikkan baris perintah sistem operasi secara manual. Pada sistem aplikasi desktop, perancangan tata letak antarmuka yang ergonomis serta intuitif menjadi sangat krusial. Desain yang terstruktur akan mempercepat alur kerja pengguna serta meminimalisasi potensi kesalahan operasional ketika kontributor microstock harus memproses pengelolaan metadata untuk ratusan aset digital secara serentak (Ginting et al., 2024)

2.2.8 Pengujian *Black-box Testing*

Pengujian *Black Box* merupakan metode pengujian perangkat lunak yang menitikberatkan pada input yang diberikan kepada suatu aplikasi dan output yang diharapkan untuk setiap nilai input. *Box-Testing* adalah untuk tidak memperhatikan bagaimana sistem bekerja secara internal. Seperti kotak hitam, pengujian menggunakan *Black Box Testing* difokuskan pada penemuan kesalahan atau ketidaksesuaian dengan spesifikasi sistem pada data awal (Wardah Gracillaria Suharyono et al., 2024).

2.2.9 Github

GitHub merupakan *platform* berbasis web yang menyediakan layanan hosting repository source code menggunakan sistem version control Git. Selain berfungsi sebagai tempat penyimpanan dan pengelolaan kode program, GitHub juga menyediakan fitur GitHub Actions yang memungkinkan pengembang untuk membangun alur kerja otomatis seperti *Continuous Integration* (CI) dan *Continuous Deployment* (CD). Menurut (Faqih et al., 2024) GitHub Actions memfasilitasi proses otomatisasi pembangunan (*build*), pengujian, dan distribusi perangkat lunak secara

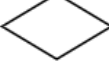
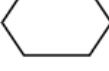
efisien melalui konfigurasi alur kerja yang terintegrasi langsung dengan repository source code.

Salah satu pemanfaatan GitHub yang relevan dalam pengembangan aplikasi desktop adalah fitur GitHub Releases API. Fitur ini memungkinkan pengembang untuk mempublikasikan versi terbaru aplikasi beserta berkas instalasi yang dapat diunduh oleh pengguna. Dengan memanfaatkan API ini, sebuah aplikasi desktop dapat memeriksa ketersediaan pembaruan secara otomatis setiap kali dijalankan, mengunduh berkas pembaruan dari server GitHub, dan menerapkan pembaruan tersebut tanpa mengharuskan pengguna mengunduh ulang aplikasi secara manual dari situs web. Mekanisme pembaruan otomatis ini penting untuk memastikan bahwa seluruh pengguna selalu menggunakan versi aplikasi yang paling mutakhir dengan perbaikan dan peningkatan fitur terbaru.

Dalam penelitian ini, GitHub digunakan untuk dua keperluan utama, yaitu sebagai alur kerja CI/CD untuk membangun berkas instalasi aplikasi secara otomatis melalui GitHub Actions, dan sebagai sistem pembaruan otomatis (*auto-updater*) yang memungkinkan aplikasi RZ Autometadata memeriksa serta mengunduh versi terbaru langsung dari GitHub Releases.

2.2.10 Flowchart

Flowchart atau Diagram alur adalah representasi simbolis dari suatu algoritma atau prosedur untuk memecahkan suatu masalah. Diagram alur memungkinkan pengguna untuk memeriksa bagian-bagian dari analisis masalah yang dilupakan oleh tim proyek.(Sidik et al., 2024)

Simbol	Nama	Keterangan
	Flow / Garis Alir	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.
	Input/Output	Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	On-Page Reference	Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.
	Manual Operation	Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	Off-Page Reference	Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.
	Document	Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	Terminator	Simbol yang menyatakan awal atau akhir suatu program.
	Predefine Process	Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	Process	Simbol yang menyatakan suatu proses yang dilakukan komputer.
	Display	Simbol yang menyatakan peralatan output yang digunakan.
	Decision	Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.
	Preparation	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

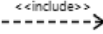
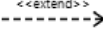
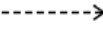
Gambar 2. 2 Flowchart

Pada penelitian ini, *Flowchart* digunakan untuk menggambarkan alur logika pemrosesan metadata secara otomatis pada aplikasi RZ Autometadata, mulai dari tahap pembukaan aplikasi, pemilihan *platform*, pengunggahan

aset, pemanggilan *API*, hingga penyimpanan hasil metadata ke dalam *database*.

2.2.11 Use Case Diagram

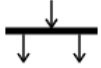
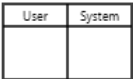
Use Case Diagram adalah salah satu Diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem yang akan dibangun. Diagram ini memvisualisasikan fungsionalitas apa saja yang disediakan oleh sistem serta siapa saja aktor yang berhak mengakses fungsi-fungsi tersebut (Siska Narulita et al., 2024). Berikut simbol-simbol dan fungsinya:

Simbol	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
	Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur bagi suatu aktor.
	System / Boundary	Menspesifikasikan paket yang menampilkan sistem secara terbatas. Membatasi antara aktor dan use case.
	Association	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
	<<include>>	Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya.
	<<extend>>	Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.
	Generalization	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah use case di mana fungsi yang satu lebih umum dari lainnya.
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya.

Gambar 2. 3 Use Case Diagram

2.2.12 Activity Diagram

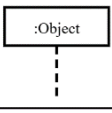
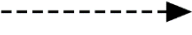
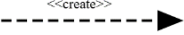
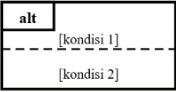
Activity Diagram adalah Diagram *UML* yang digunakan untuk memodelkan alur kerja (*workflow*) dari sebuah sistem. Menurut Afifan (2024) dalam *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, Activity Diagram digunakan pada tahap perancangan sistem untuk menunjukkan aliran kerja sistem atau alur kejadian secara visual, memodelkan alur aktivitas yang menggambarkan proses yang terjadi pada sebuah sistem, baik berupa runtutan menu maupun proses bisnis. Berikut simbol simbol dan fungsinya:

Simbol	Nama	Keterangan
	Start Point / Initial State	Lingkar hitam kecil yang digunakan untuk menandakan status awal, tindakan awal, atau titik awal aktivitas dalam sistem.
	Activity (Aktivitas)	Merupakan aktivitas yang dilakukan atau sedang terjadi dalam sistem. Biasanya diawali dengan kata kerja dari aktivitas yang dilakukan.
	Decision (Percabangan)	Suatu titik yang mengindikasikan suatu kondisi dengan kemungkinan perbedaan transisi ketika sistem memiliki beberapa kemungkinan atau alternatif alur.
	Synchronization (Fork & Join)	Fork digunakan untuk memecah behaviour menjadi beberapa aktivitas secara paralel. Join digunakan untuk menggabungkan kembali aktivitas secara paralel.
	Merge (Penggabungan)	Menggabungkan kembali beberapa alur (flow) alternatif yang sebelumnya sudah dipecah oleh suatu titik percabangan (decision).
	Swimlanes	Memecah activity diagram menjadi kolom dan baris untuk membagi tanggung jawab objek-objek atau aktor yang melakukan suatu aktivitas.
	Transition (Transisi)	Garis alir dengan anak panah yang digunakan untuk menunjukan urutan aktivitas dari aktivitas sebelumnya ke aktivitas selanjutnya.
	End State (Status Akhir)	Notasi akhir yang digunakan untuk menandakan bahwa seluruh alur proses atau aktivitas dalam sistem tersebut telah selesai dan berakhir (bull's eye).

Gambar 2. 4 Activity Diagram

2.2.13 Sequence Diagram

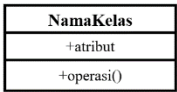
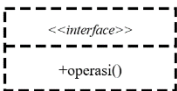
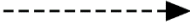
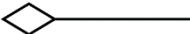
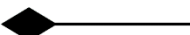
Sequence Diagram adalah Diagram yang menggambarkan interaksi antar objek. *Sequence Diagram* juga menekankan pada pengiriman pesan dalam suatu waktu tertentu dan mengilustrasikan bagaimana pesan dikirim dan diterima oleh objek serta bagaimana urutannya (Hakim et al., 2024). Berikut beberapa komponen utama yang sering digunakan yaitu:

Simbol	Nama	Keterangan
	Actor	Orang atau sistem eksternal yang berinteraksi dengan sistem. Actor memulai atau menerima pesan.
	Lifeline	Garis hidup yang merepresentasikan keberadaan objek dalam jangka waktu tertentu (garis putus-putus vertikal).
	Activation / Execution	Kotak tipis pada lifeline yang menunjukkan objek sedang aktif menjalankan suatu operasi.
	Synchronous Message	Pesan sinkron, pengirim menunggu hingga penerima selesai memproses.
	Return Message	Pesan balasan yang menunjukkan hasil kembalian dari suatu operasi.
	Self Message	Pesan yang dikirim oleh objek kepada dirinya sendiri untuk memanggil metode internal.
	Create Message	Pesan untuk membuat objek baru. Objek muncul saat pesan diterima.
	Destroy / Delete	Simbol X yang menandakan penghancuran sebuah objek. Setelah titik ini, objek tidak lagi ada.
	Combined Fragment	Kotak fragmen untuk mengelompokkan interaksi: alt (percabangan), loop (perulangan), opt (opsional).

Gambar 2. 5 Sequence Diagram

2.2.14 Class Diagram

Class Diagram merupakan salah satu jenis Diagram struktural pada *Unified Modeling Language* (UML) yang berfungsi untuk memodelkan struktur sistem secara statis. Menurut (Ramdany, 2024) *Class Diagram* adalah Diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibangun untuk merancang sebuah perangkat lunak, mencakup atribut, operasi (*methods*), serta hubungan relasi antar kelas tersebut. Diagram ini berperan penting dalam pengembangan berorientasi objek karena menjadi satu-satunya Diagram UML yang dapat dipetakan secara langsung ke dalam bahasa pemrograman, sehingga memberikan landasan konseptual (*blueprint*) bagi pengembang sebelum tahapan pengkodean dilakukan. Berikut beberapa komponen utama yang sering digunakan yaitu:

Simbol	Nama	Keterangan
	Class	Representasi dari sebuah kelas yang terdiri dari tiga bagian: nama kelas, atribut (data), dan operasi (fungsi).
	Interface	Kumpulan operasi tanpa implementasi. Mendefinisikan kontrak yang harus dipenuhi oleh kelas yang mengimplementasikan.
	Association	Hubungan statis antar kelas yang menunjukkan relasi antar objek.
	Directed Association	Relasi satu arah, satu kelas mengetahui kelas lain namun tidak sebaliknya.
	Generalization / Inheritance	Hubungan pewarisan, kelas anak mewarisi atribut dan operasi kelas induk.
	Dependency	Hubungan ketergantungan, perubahan satu kelas mempengaruhi kelas lain.
	Aggregation	Hubungan "memiliki" yang bersifat lemah. Kelas bagian dapat berdiri sendiri tanpa kelas induk.
	Composition	Hubungan "memiliki" yang bersifat kuat. Kelas bagian tidak dapat berdiri sendiri tanpa kelas induk.
	Multiplicity	Jumlah objek dalam suatu relasi. Contoh: 1 (satu), * (banyak), 0..1 (nol atau satu), 1..* (satu/lebih).

Gambar 2. 6 Class Diagram

2.2.15 Waterfall

Metode Waterfall adalah proses pengembangan *software* yang berurutan dimana prosesnya mengalir ke bawah seperti air terjun. Tahapan pada Waterfall harus diselesaikan secara berurutan satu demi satu dan tidak dapat pindah ke tahapan selanjutnya apabila tahapan sebelumnya belum selesai sepenuhnya (Nagara et al., 2023).

2.3 Penelitian Terdahulu

Penelitian pertama dilakukan oleh (Xiang et al., 2024) dalam jurnal *arXiv* dengan judul penelitian "*Text Understanding and Generation Using Transformer Models for Intelligent E-commerce Recommendations*". Penelitian tersebut mengkaji penerapan model pra-latih berbasis arsitektur *Transformer* yang difungsikan untuk mengotomatisasi pembuatan deskripsi teks pada *platform e-commerce*. Hasil penelitian dari Xiang menunjukkan bahwa algoritma *Transformer* terbukti sangat efektif dalam memahami konteks visual dan tekstual suatu produk dagang, sehingga mampu menghasilkan teks deskripsi otomatis yang akurat guna mengurangi beban kerja manual (*manual labor*). Namun demikian, fokus penelitian tersebut terbatas pada ekosistem *e-commerce* dan rekomendasi produk pelanggan, serta belum menyentuh ranah standarisasi metadata pada industri kreatif digital. Berangkat dari konsep pembuatan teks otomatis tersebut, penelitian ini mengambil langkah pembaruan dengan mengadaptasi teknologi *Transformer* secara spesifik ke dalam aplikasi desktop untuk industri microstock, di mana algoritma tersebut dimanfaatkan untuk mengotomatisasi pembuatan metadata (judul dan kata kunci) yang terstandarisasi ke dalam format berkas CSV.

Penelitian kedua dilakukan oleh Zhang dkk. (2024) dalam konferensi *ACL 2024* dengan judul *MM-LLMs: Recent Advances in MultiModal Large Language Models*. Penelitian tersebut menunjukkan bahwa model *MLLM* terkini, seperti GPT-4o dan Gemini, mampu menganalisis konten visual berupa gambar dan video serta menghasilkan deskripsi tekstual yang akurat dan kontekstual. Kajian ini menjadi landasan teoretis dalam pemanfaatan *MLLM* pada penelitian ini. Meskipun demikian, penelitian tersebut masih bersifat survei umum dan belum mengarah

pada implementasi untuk kebutuhan spesifik, seperti standarisasi metadata pada *platform* microstock. Dalam penelitian ini, kemampuan *MLLM* dimanfaatkan untuk menghasilkan judul, kata kunci, dan kategori yang sesuai dengan standar Adobe Stock, Shutterstock, Freepik, dan Vecteezy.

Penelitian Ketiga dilakukan oleh (Wasilewski, 2025) dalam jurnal *Computer Standards & Interfaces* dengan judul penelitian "*Harnessing Generative AI for Personalized E-commerce Product Descriptions: A Framework and Practical Insights*". Penelitian tersebut membahas pengembangan kerangka kerja pemanfaatan *Generative AI* berbasis *Large Language Model (LLM)* untuk mengotomatisasi pembuatan deskripsi produk *e-commerce* secara massal yang dipersonalisasi berdasarkan segmen pelanggan. Hasil kajian menunjukkan bahwa pemanfaatan *AI* yang dipadukan dengan teknik *prompt engineering* terbukti efektif dalam memecahkan masalah skalabilitas pembuatan konten tulisan yang sebelumnya tidak mungkin dilakukan secara manual. Meskipun demikian, penelitian tersebut berfokus pada ekosistem *e-commerce* dengan *input* berupa data teks pelanggan, dan belum menjangkau ranah pengolahan *input* visual (*image-to-text*) pada industri kreatif. Dari kerangka otomatisasi *AI* tersebut, penelitian ini mengembangkannya lebih jauh dengan memanfaatkan varian *Multimodal LLM (MLLM)* ke dalam sebuah aplikasi desktop untuk memecahkan masalah skalabilitas metadata (judul dan kata kunci) berdasarkan *input* visual pada empat *platform* microstock.

2.4 Profil Perusahaan

a. Gambaran Umum Objek Penelitian (RZ Digital ID)

RZ Digital ID merupakan sebuah *digital creative studio* skala kecil yang didirikan pada bulan September 2025. Studio ini didirikan untuk merespons pesatnya pertumbuhan industri kreatif digital, khususnya pasar konten visual global (microstock) dan kebutuhan akan peralatan penunjang (*digital tools*) bagi para kontributor.

Dalam menjalankan operasionalnya, RZ Digital ID memiliki dua fokus bisnis utama:

1. **Produksi Aset Visual (Microstock)**

Memproduksi dan mendistribusikan aset digital berkualitas tinggi (seperti ilustrasi vektor, desain grafis, foto, dan video) ke berbagai *platform* agen microstock global terkemuka seperti Shutterstock, Adobe Stock, Freepik, dan Vecteezy.

2. **Pengembangan Produk Digital (*Digital Product Development*)**

Merancang dan menjual aplikasi utilitas mandiri (*software utilities*) untuk membantu mengoptimalkan alur kerja (*workflow*) para kreator konten digital. Salah satu produk utama yang dikembangkan adalah RZ Autometadata, sebuah aplikasi berbasis kecerdasan buatan (*AI*) untuk otomatisasi pengisian metadata aset microstock.

Selain fokus pada kegiatan produksi dan pengembangan, RZ Digital ID juga aktif menggunakan media sosial, khususnya *platform* TikTok yang dibangun pada September 2025, sebagai media pemasaran digital, edukasi komunitas kreator, serta sarana interaksi untuk mendapatkan umpan balik (*feedback*) dari pengguna produk digital mereka.

b. Visi dan Misi RZ Digital ID

1. **Visi:** Menjadi studio kreatif digital yang inovatif dan terpercaya dalam menyediakan aset visual berkualitas tinggi serta solusi teknologi yang mempermudah alur kerja para kreator konten di era digital.
2. **Misi:** Memproduksi aset visual yang relevan dengan tren pasar global dan memenuhi standar kualitas *platform* microstock internasional dan mengembangkan aplikasi dan perangkat lunak penunjang (*tools*) yang solutif, efisien, dan mudah digunakan oleh para pelaku industri kreatif digital serta memanfaatkan media sosial secara interaktif untuk mengedukasi masyarakat mengenai potensi industri microstock serta memperkenalkan produk-produk buatan studio.

c. Struktur Organisasi dan Alur Kerja

Sebagai studio kreatif rintisan skala kecil, RZ Digital ID menerapkan struktur organisasi yang ramping (*lean structure*) namun fleksibel, yang terdiri dari peran:

1. **Creative & Content Creator:** Bertanggung jawab atas produksi aset visual microstock, riset tren pasar visual, serta pembuatan konten promosi dan edukasi di *platform* TikTok.