

# Implementasi Gemini API untuk Generatif Teks Deskripsi Karya Otomatis dalam Aplikasi Pameran Berbasis Web dengan Metode Waterfall

Hariyadi  
Program Studi Teknik Informatika  
Universitas Palangka Raya  
Palangkaraya, Indonesia  
[hariyadi@mhs.eng.upr.ac.id](mailto:hariyadi@mhs.eng.upr.ac.id)

Nova Noor Kamala Sari  
Program Studi Teknik Informatika  
Universitas Palangka Raya  
Palangkaraya, Indonesia  
[novanoorks@it.upr.ac.id](mailto:novanoorks@it.upr.ac.id)

Jordi Irawan  
Program Studi Teknik Informatika  
Universitas Palangka Raya  
Palangkaraya, Indonesia  
[jordiirawandotpribadi@mhs.eng.upr.ac.id](mailto:jordiirawandotpribadi@mhs.eng.upr.ac.id)

Viktor Handrianus Pranatawijaya  
Program Studi Teknik Informatika  
Universitas Palangka Raya  
Palangkaraya, Indonesia  
[viktorhp@it.upr.ac.id](mailto:viktorhp@it.upr.ac.id)

**Abstract** - In the face of the era of rapid technological development, software has become a major necessity in various aspects of life. The advancement of LLM as a part of artificial intelligence has been rapid, with its implementation expanding across various applications and fields. Currently, Google has launched their latest LLM named Gemini, Google opens access to its API key for application developers who want to utilize it in making applications. This research aims to find out how the implementation of Gemini API keys can be applied to applications, especially in the work exhibition application being developed, and whether the application is able to function according to the functionality of the Gemini API. In developing the works exhibition application using the Gemini API, a waterfall development methodology is used, and the works exhibition application developed is web-based. After the work exhibition application was completed and tested, it was found that the resulting work exhibition application functioned as expected and was able to utilize the functionality of the Gemini API.

**Keywords:** API, Application, Gemini, Work Exhibition

**Abstrak** - Dalam menghadapi era pesatnya perkembangan teknologi, perangkat lunak telah menjadi kebutuhan utama dalam berbagai aspek kehidupan. Kemajuan LLM sebagai bagian dari kecerdasan buatan telah melaju pesat, dengan implementasinya yang meluas di berbagai aplikasi dan bidang. Saat ini, Google telah meluncurkan LLM terbaru mereka yang diberi nama Gemini, Google membuka akses ke kunci API-nya untuk para pengembang aplikasi yang ingin memanfaatkannya dalam pembuatan aplikasi. Penelitian ini bertujuan untuk mengetahui bagaimana implementasi kunci API Gemini dapat diterapkan pada aplikasi, khususnya pada aplikasi pameran karya yang sedang dikembangkan, serta apakah aplikasi tersebut mampu berfungsi sesuai dengan fungsionalitas dari Gemini API. Dalam pengembangan aplikasi pameran karya dengan menggunakan Gemini API, menggunakan metodologi pengembangan *waterfall*, dan aplikasi pameran karya yang dikembangkan berbasis web. Setelah aplikasi pameran karya selesai dibuat dan diuji, didapatkan hasil bahwa aplikasi pameran karya yang dihasilkan berfungsi sebagaimana yang diharapkan dan mampu memanfaatkan fungsionalitas dari Gemini API.

**Kata Kunci :** API, Aplikasi, Gemini, Pameran Karya

## PENDAHULUAN

Salah satu pendorong utama pengembangan aplikasi pameran karya adalah melihat tren peningkatan jumlah proyek mahasiswa, khususnya yang berasal dari tugas-tugas kuliah. Proyek-proyek ini sering kali memiliki potensi keberlanjutan dan nilai lebih, namun sering kali tidak memiliki wadah yang sesuai untuk menampung dan mengembangkan kreativitas tersebut. Oleh karena itu, pameran karya dihadirkan sebagai platform yang efisien untuk mengatasi tantangan ini.

Teknologi kecerdasan buatan (AI), termasuk *machine learning*, *deep learning*, dan pemrosesan bahasa alami (NLP), sedang berkembang pesat dan memiliki dampak yang luas di berbagai bidang [1]. Salah satu bentuk AI yang menarik perhatian saat ini adalah LLM (*Large Language Model*) atau model bahasa besar, yang merupakan hasil dari pengembangan dalam jaringan saraf tiruan dan pemrosesan bahasa alami yang telah ada sebelumnya, dikenal sebagai LM (*Language Model*). LLM, sebagai jenis model AI, mampu memproses dan menghasilkan teks bahasa. Biasanya, model-model LLM dilatih menggunakan sejumlah besar data teks dan menerapkan teknik *deep learning* untuk memahami pola dan struktur bahasa. Ini memungkinkan LLM digunakan untuk berbagai tugas yang terkait dengan bahasa, termasuk pembuatan teks, terjemahan, ringkasan, menjawab pertanyaan, penulisan kode, dan analisis sentimen [2].

Permasalahan yang menjadi fokus penelitian ini adalah bagaimana menghasilkan teks deskripsi karya yang menarik dan informatif untuk pameran karya berbasis web dengan menggunakan teknologi generatif seperti *Large Language Model* (LLM). Pada pameran karya berbasis web, deskripsi karya sangat penting untuk memberikan konteks dan menarik minat pengunjung. Namun, penulisan deskripsi ini sering kali memakan waktu dan memerlukan keahlian khusus. Pada akhirnya penelitian ini mencoba mengintegrasikan Gemini API ke dalam aplikasi yang sedang dikembangkan, yaitu sebuah aplikasi pameran karya bernama pameran karya. Penggunaan LLM Gemini dalam pameran karya

meningkatkan kemampuannya untuk meniru perilaku manusia dalam membuat deskripsi, memahami konteks karya, dan memberikan respons yang sesuai tanpa campur tangan manusia. Kemajuan dalam pembelajaran mesin, NLP, dan kecerdasan buatan telah membuat berbagai aplikasi semakin pintar, akurat, dan berguna, dan semakin banyak digunakan di berbagai bidang seperti pendidikan, kesehatan, robotika, dan industri [3].

Penelitian terdahulu mengenai *Large Language Models* (LLM) telah menunjukkan kemajuan signifikan dalam pemrosesan bahasa alami (NLP) dan aplikasi terkait. Misalnya, penelitian oleh Brown et al. (2020) memperkenalkan GPT-3, model bahasa yang dilatih dengan 175 miliar parameter, yang menunjukkan kemampuan luar biasa dalam berbagai tugas NLP seperti terjemahan, penulisan esai, dan menjawab pertanyaan dengan hanya sedikit atau tanpa contoh pelatihan tambahan. GPT-3 memperkenalkan konsep "*few-shot learning*," yang memungkinkan model untuk menangani tugas baru dengan hanya beberapa contoh, menunjukkan fleksibilitas dan kemampuan generalisasi yang tinggi [4]. Sementara itu, Devlin et al. (2019) mengembangkan BERT, model yang menggunakan pendekatan *bidirectional* untuk pemahaman bahasa, yang meningkatkan pemahaman konteks dan makna teks secara signifikan. BERT telah menunjukkan kinerja yang luar biasa dalam tugas-tugas seperti klasifikasi teks dan pengenalan entitas [5]. Selanjutnya, Raffel et al. (2020) memperkenalkan T5, model yang mengubah semua tugas NLP menjadi format *text-to-text*, memungkinkan *transfer learning* yang lebih efektif dan efisien. T5 menyatukan berbagai tugas NLP dalam satu kerangka kerja *text-to-text*, menunjukkan adaptasi yang tinggi dalam berbagai aplikasi NLP [6]. Penelitian-penelitian ini menunjukkan potensi besar LLM dalam menghasilkan teks dengan kualitas yang mendekati manusia, membuka peluang baru dalam pengembangan aplikasi berbasis teks.

Namun, aplikasi khusus untuk generatif teks deskripsi karya seperti pada pameran karya berbasis web masih terbatas. Sehingga, penelitian ini mencoba melakukan implementasi Gemini API dalam konteks yang lebih spesifik, yaitu untuk generatif teks deskripsi karya, serta evaluasi kinerjanya dalam lingkungan pameran karya berbasis web. Dengan menggunakan Gemini API, diharapkan deskripsi karya dapat dihasilkan secara otomatis dan akurat, sehingga memudahkan pengguna dalam memamerkan karya mereka dengan deskripsi yang menarik dan informatif.

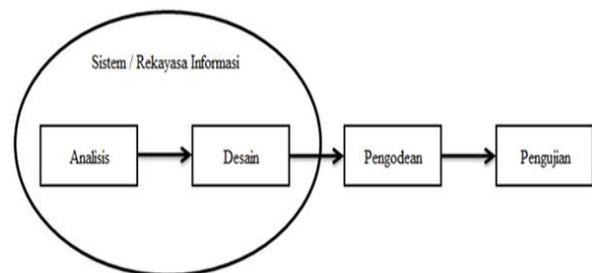
Penggunaan Gemini API dalam penelitian ini didasarkan pada beberapa keunggulan utamanya, yaitu ketersediaan layanan secara gratis serta dukungannya yang komprehensif untuk membaca teks dan gambar. Sebagai layanan yang gratis, Gemini API memungkinkan peneliti dan pengembang untuk mengimplementasikan solusi canggih tanpa harus khawatir mengenai biaya tambahan yang menjadi hambatan dalam pengembangan teknologi baru. Hal ini sangat penting untuk penelitian dan pengembangan yang memiliki anggaran terbatas. Selain itu, Gemini API menawarkan dukungan yang kuat untuk membaca dan memahami teks serta gambar. Kemampuan ini memungkinkan API untuk menghasilkan deskripsi teks yang akurat dan kaya berdasarkan analisis mendalam terhadap konten gambar. Dukungan ini tidak hanya meningkatkan kualitas deskripsi yang dihasilkan tetapi juga memastikan bahwa setiap karya mendapatkan deskripsi yang relevan dan informatif. Dengan kombinasi ini, Gemini API menjadi pilihan yang ideal untuk proyek ini, memungkinkan

otomatisasi deskripsi karya yang efisien dan efektif dalam aplikasi pameran berbasis web, sambil memanfaatkan teknologi terbaru dalam pengolahan bahasa alami dan analisis gambar.

Untuk memanfaatkan LLM Gemini dalam aplikasi pameran karya ini, diperlukan akses ke kunci API (*Application Programming Interface*) Gemini. Dimana akan digunakan kunci API dari Gemini. Google telah memberikan akses ke API Gemini kepada para pengembang perangkat lunak dan bisnis yang ingin mengintegrasikan kemampuan Gemini dalam aplikasi buatan mereka untuk berbagai skenario penggunaan.

#### METODE PENELITIAN

Metode penelitian adalah tahapan yang menggambarkan langkah-langkah yang dilakukan selama proses penelitian, dengan tujuan untuk memfasilitasi pelaksanaan penelitian [7]. Model *Waterfall* digunakan dalam penelitian ini. Model *Waterfall* menawarkan pendekatan berurutan dalam pengembangan perangkat lunak, dimulai dari tahap analisis, desain, implementasi, dan pengujian [8].



Gambar 1. Ilustrasi model *Waterfall*

1. Analisis Kebutuhan  
Dalam proses pengembangan aplikasi, diperlukan analisis kebutuhan dan evaluasi terhadap konsep pengembangan aplikasi. Tahap ini akan mengidentifikasi komponen-komponen yang diperlukan, mulai dari perangkat lunak hingga pengguna akhir dari aplikasi yang sedang dikembangkan, serta kebutuhan aplikasi dari segi *input*, proses, dan *output* [9].
2. Desain  
Proses desain merupakan tahapan di mana aplikasi direncanakan secara rinci sehingga menghasilkan spesifikasi yang tepat sesuai dengan kebutuhan yang telah ditetapkan sebelumnya. Salah satu aspek dari proses desain adalah perancangan antarmuka, yang bertujuan untuk menghasilkan spesifikasi aplikasi yang sesuai dengan kebutuhan pengguna [10].
3. Pengkodean  
Pengkodean merupakan tahap di mana desain yang telah dibuat diwujudkan menjadi aplikasi yang berfungsi sesuai dengan spesifikasi yang telah ditetapkan sebelumnya. Ini melibatkan penulisan kode, pengembangan fitur-fitur aplikasi, pengujian terintegrasi, dan penyesuaian sesuai kebutuhan yang mungkin muncul selama proses pengkodean [11].
4. Pengujian

Pengujian adalah tahap penting dalam pengembangan aplikasi di mana aplikasi yang telah diimplementasikan akan diuji untuk memastikan bahwa berbagai fitur dan fungsionalitasnya berjalan sesuai dengan yang diharapkan [12]. Proses pengujian melibatkan pengujian berbagai aspek aplikasi, seperti fungsionalitas, keamanan, kinerja, dan kegunaan, dengan menggunakan berbagai metode pengujian yang sesuai.

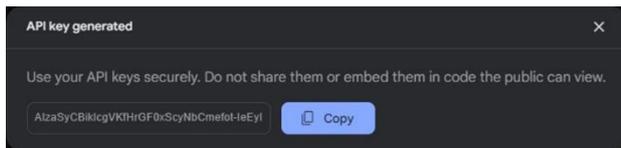
HASIL DAN PEMBAHASAN

Pembahasan terhadap hasil penelitian ini adalah implementasi Gemini API dalam generatif teks deskripsi karya otomatis pada pengembangan aplikasi pameran karya berbasis *website*. Karena aplikasi pameran karya ini akan dikembangkan sebagai aplikasi berbasis *website*, pada tahap analisis kebutuhan, dilakukan evaluasi terhadap kebutuhan perangkat lunak dan perangkat keras yang diperlukan sesuai dengan *platform* yang dipilih. Keputusan diambil untuk menggunakan Visual Studio Code sebagai lingkungan pengembangan untuk aplikasi pameran karya ini, dengan menggunakan React JS yang merupakan *framework library* untuk bahasa pemrograman JavaScript.

Pada tahap Desain, proses pembuatan aplikasi pameran karya dengan integrasi gemini API dimulai, yang terdiri dari serangkaian langkah-langkah berikut:

1. Mendapatkan Kunci API Gemini

Pentingnya untuk mempersiapkan kunci API (API Key) dari Gemini seperti terlihat pada gambar 2. di bawah. Dalam pengembangan perangkat lunak, API adalah fondasi untuk pertukaran data antara modul perangkat lunak yang berbeda, memungkinkan integrasi fitur-fitur dari API tersebut ke dalam kode aplikasi yang sedang dikembangkan. Untuk memanfaatkan layanan API dari Gemini, langkah awalnya adalah membuat kunci API melalui Google AI Studio, yang tersedia secara gratis dari Google. Saat ini, Google menawarkan model Gemini yang mendukung *input* dan *output* teks dan gambar, yang bisa diakses melalui API mereka.



Gambar 2. Mendapatkan Kunci API Gemini

2. Menambahkan Dependensi SDK ke Project

Selanjutnya, menambahkan dependensi SDK ke *project* aplikasi dengan perintah *npm*, karena *project* ini menggunakan Node.js v18+ dan *npm*. Contoh kode dependensinya dapat dilihat pada gambar 3. di bawah ini.



Gambar 3. Kode Dependensi Untuk Google Generative

3. Menginisialisasi Generative Model

Dalam melakukan panggilan API, perlu dilakukan inisialisasi objek *GenerativeModel* terlebih dahulu.

Contoh kodenya dapat dilihat pada gambar 4, 5 dan 6. di bawah ini.



Gambar 4. Kode Inisialisasi Objek Generative Gemini API



Gambar 5. Kode GenerativeModel Berdasarkan Teks dan Gambar



Gambar 6. Kode GenerativeModel Berdasarkan Teks

Untuk pengembangan aplikasi pameran karya ini, menggunakan model Gemini Pro untuk melakukan generatif teks deskripsi. *Project* ini menggunakan model gemini yaitu untuk *generate* berdasarkan teks dan gambar.

4. Implementasi Kode Generative Ke Project

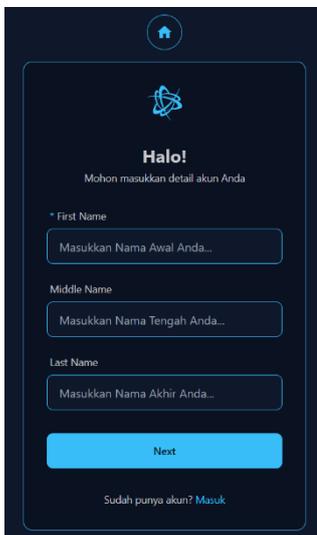
Pada *project* ini kode *generative* diimplementasikan ke dalam API *project* yang menerima data berdasarkan *website/client* yang terhubung. Implementasi kode *generative AI* yang digunakan pada pembuatan aplikasi pameran karya ini adalah sebagai berikut yang dapat dilihat pada gambar 7. dibawah ini:



Gambar 7. Implementasi Kode Generative

Gambar 7. diatas adalah potongan kode menggunakan Next.js yang memanfaatkan Google *Generative AI* untuk menghasilkan deskripsi berdasarkan input yang diberikan. Pertama, modul *GoogleGenerativeAI* dan *NextResponse* diimpor. Kemudian, objek genAI diinisialisasi dengan API key. Fungsi asinkron POST menangani permintaan POST, mengambil data dari *form*, dan menyimpannya dalam variabel *title*, *category*, dan *image*. Berdasarkan ada tidaknya gambar, model yang sesuai dipilih (*gemini-pro-vision* untuk gambar dan *gemini-pro* untuk teks saja). *Prompt* disiapkan untuk menghasilkan deskripsi, dan jika ada gambar, data gambar dikonversi menjadi base64. Model kemudian digunakan untuk menghasilkan konten berdasarkan *prompt*, dan hasilnya dikembalikan dalam format JSON yang berisi deskripsi otomatis yang dihasilkan.

Pada tahap pengkodean, kode yang telah dirancang sebelumnya diubah menjadi tampilan yang dapat berfungsi sesuai dengan kebutuhan pengguna. Di tahap ini, akan mengembangkan program berdasarkan dari hasil perancangan yang telah disiapkan sebelumnya.



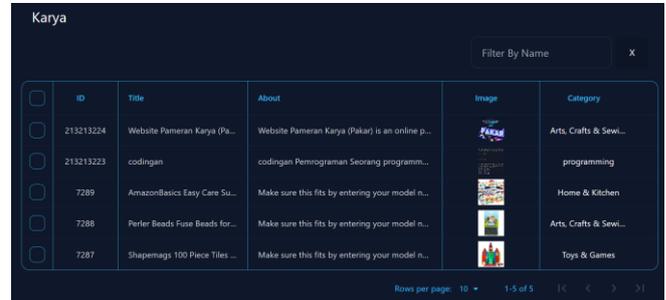
Gambar 8. Halaman Registrasi

Gambar 8. adalah tampilan yang diperuntukkan hanya untuk pengguna yang belum memiliki akun, di mana mereka dapat melakukan registrasi untuk membuat akun baru. Pada bagian halaman, terdapat instruksi singkat yang meminta pengguna untuk memasukkan detail akun mereka. Formulir registrasi terdiri dari beberapa kolom *input* yang harus diisi oleh pengguna, seperti kolom untuk nama depan, nama tengah, dan nama belakang.



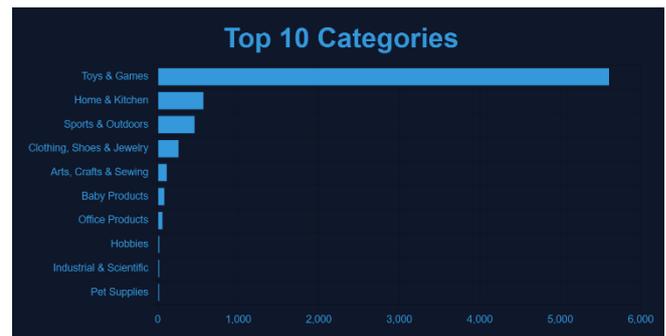
Gambar 9. Halaman Masuk

Gambar 9. adalah tampilan halaman masuk yang diperuntukkan bagi pengguna yang sudah memiliki akun. Pada bagian halaman, terdapat instruksi singkat yang meminta pengguna untuk memasukkan detail akun mereka. Formulir *login* terdiri dari dua kolom *input* utama yaitu untuk alamat email dan kata sandi. Setiap kolom dilengkapi dengan *placeholder* yang memberikan petunjuk kepada pengguna tentang informasi apa yang harus dimasukkan.



Gambar 10. Halaman Tabel Daftar Karya

Gambar 10. menampilkan daftar atau list semua karya yang telah dimasukkan pengguna dalam tabel. Tabel ini berfungsi sebagai tampilan ringkasan dari semua karya yang telah diunggah oleh pengguna ke dalam aplikasi. Setiap baris dalam tabel mewakili satu karya, dengan kolom-kolom yang mencakup informasi penting seperti ID karya, judul, deskripsi singkat, gambar, dan kategori.



Gambar 11. Halaman Grafik Daftar Karya

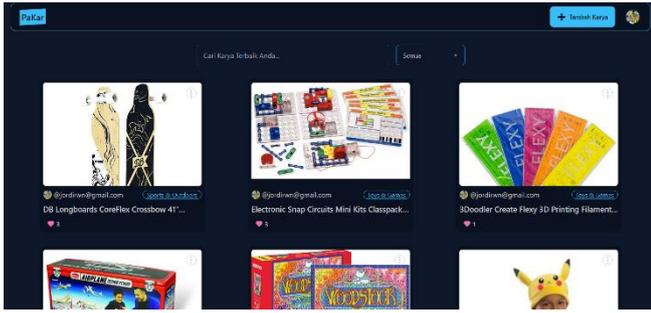
Gambar 11. menampilkan daftar semua karya yang telah diunggah oleh pengguna dalam bentuk grafik batang. Grafik ini memberikan visualisasi mengenai distribusi karya berdasarkan kategori. Setiap batang dalam grafik mewakili satu kategori, dengan panjang batang menunjukkan jumlah karya yang termasuk dalam kategori tersebut.



Gambar 12. Halaman Utama

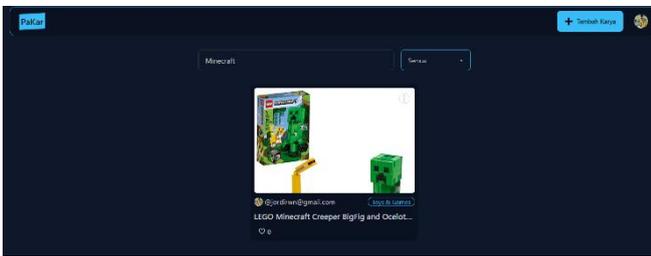
Gambar 12. menampilkan halaman utama aplikasi pameran karya yang terdapat menu navigasi yang memungkinkan pengguna untuk dengan mudah mengakses berbagai bagian situs, seperti *home*, *about*, *services*, *team*, dan

FAQ. Di sudut kanan atas, terdapat tombol *login* dan *register* yang memudahkan pengguna untuk masuk atau mendaftar akun baru.



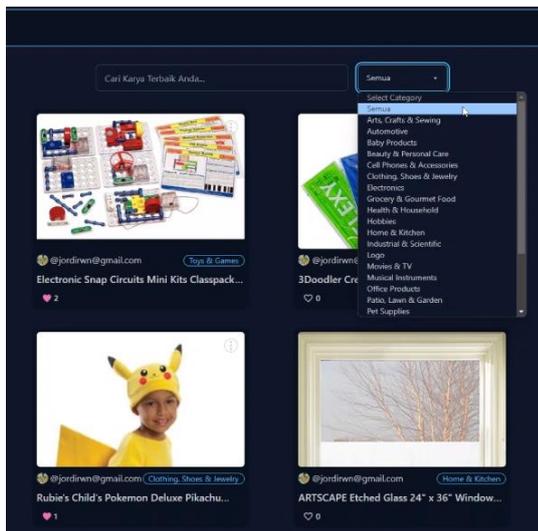
Gambar 13. Halaman Karya

Gambar 13. menampilkan halaman yang berisi semua karya yang telah diunggah oleh pengguna, dengan informasi lengkap seperti judul, deskripsi, penulis, kategori, gambar, dan jumlah suka. Halaman ini dirancang untuk memudahkan pengguna dalam menjelajahi berbagai karya yang tersedia di aplikasi pameran karya.



Gambar 14. Halaman Cari Karya

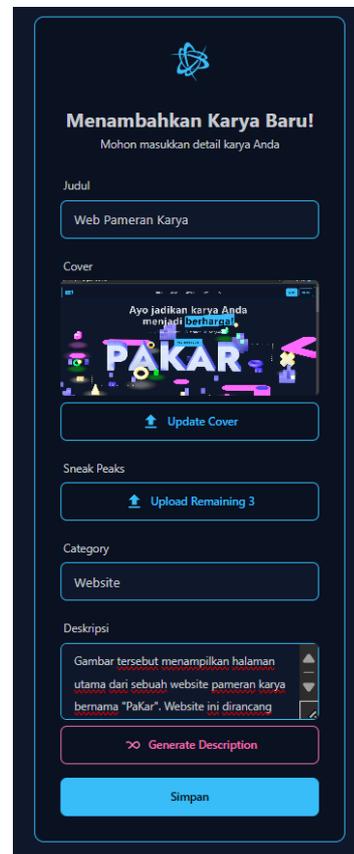
Gambar 14. menampilkan hasil pencarian karya pada aplikasi pameran karya berdasarkan kata kunci yang telah ditentukan sebelumnya. Halaman ini dirancang untuk memudahkan pengguna dalam menemukan karya yang spesifik sesuai dengan kata kunci yang mereka masukkan.



Gambar 15. Halaman Kategori Karya

Gambar 15. menampilkan halaman aplikasi pameran karya yang memungkinkan pengguna untuk menelusuri karya berdasarkan kategori yang telah ditentukan sebelumnya. Halaman ini dirancang untuk memudahkan pengguna dalam menemukan karya yang sesuai dengan minat mereka melalui fitur penyaringan kategori.

Tahapan terakhir yaitu tahap pengujian, pada tahap pengujian ini akan dilakukan percobaan untuk menguji fungsionalitas aplikasi pameran karya guna memastikan apakah integrasi Gemini API pada aplikasi dapat beroperasi sesuai yang diharapkan. Dalam uji coba ini, akan ditambahkan sebuah karya baru ke dalam aplikasi. Selanjutnya mengisi detail karya seperti judul, cover, dan kategori. Setelah semua informasi diisi, disini akan mencoba menggunakan fitur *generate description* yang memanfaatkan Gemini API untuk menghasilkan deskripsi otomatis berdasarkan judul karya dan gambar yang diunggah. Proses ini bertujuan untuk menguji apakah deskripsi yang dihasilkan oleh Gemini API sesuai dengan konteks dan kualitas yang diharapkan. Pada gambar yang ditampilkan, terlihat bahwa pengguna telah mengisi judul “Web Pameran Karya” dan mengunggah gambar cover. Setelah itu, lakukan dengan menekan tombol *generate description* untuk menghasilkan deskripsi otomatis. Hasil dari pengujian ini akan menunjukkan apakah integrasi API berjalan dengan baik dan apakah deskripsi yang dihasilkan dapat membantu pengguna dalam mendeskripsikan karya mereka secara efektif, seperti yang terlihat pada gambar 16. di bawah ini.



Gambar 16. Halaman Tambah Karya



Gambar 17. Halaman Detail Karya

Gambar 17. menampilkan halaman detail karya yang berisi informasi lengkap tentang karya yang diunggah oleh pengguna. Halaman ini dirancang untuk memberikan informasi lengkap tentang karya yang diunggah.

Penelitian ini berhasil mengimplementasikan Gemini API dalam generatif teks deskripsi karya otomatis pada aplikasi pameran karya berbasis web. Dalam prosesnya, berbagai aspek telah dievaluasi, termasuk akurasi dan kualitas teks yang dihasilkan, serta kemudahan integrasi API ke dalam sistem yang ada.

Dibandingkan penelitian sebelumnya seperti yang dilakukan oleh Brown et al. (2020) dengan model GPT-3. Gemini API menawarkan solusi yang lebih terjangkau dan mudah diimplementasikan, tanpa mengurangi kualitas teks deskripsi yang dihasilkan. Salah satu keunggulan signifikan dari Gemini API adalah kemampuannya untuk membaca dan memproses gambar, yang belum didukung oleh GPT-3. Hal ini memberikan keunggulan besar dalam konteks aplikasi pameran karya, di mana deskripsi yang dihasilkan dapat lebih relevan dan kontekstual dengan karya yang dipamerkan.

### KESIMPULAN

Aplikasi pameran karya berbasis web yang dikembangkan dengan mengintegrasikan Gemini API berhasil menghasilkan teks deskripsi karya yang menarik dan informatif secara otomatis. Implementasi Gemini API ke dalam aplikasi ini mudah dilakukan dan mampu meningkatkan efisiensi waktu serta kualitas deskripsi karya dibandingkan dengan metode manual. Hasil penelitian ini menunjukkan bahwa Gemini API memiliki potensi besar dalam aplikasi generatif teks deskripsi karya, namun, ada

beberapa hal yang perlu diperhatikan berkenaan dengan penggunaan Gemini secara umum di mana pada proses generatif otomatis penting untuk memperhatikan judul serta teknik dari *text prompting* yang digunakan karena kualitas *text prompt* yang diberikan sangat mempengaruhi kuantitas, kualitas serta kompleksitas dari *respon* yang diberikan oleh *generate description*. Penelitian ini membuka peluang untuk pengembangan lebih lanjut dalam memanfaatkan teknologi LLM seperti Gemini API dalam berbagai aplikasi berbasis teks.

### REFERENSI

- [1] S. Fosso Wamba, R. E. Bawack, C. Guthrie, M. M. Queiroz, and K. D. A. Carillo, "Are we preparing for a good AI society? A bibliometric review and research agenda," *Technol. Forecast. Soc. Change*, vol. 164, no. December 2020, 2021.
- [2] A. Latif and J. Kim, "Evaluation and Analysis of Large Language Models for Clinical Text Augmentation and Generation," *IEEE Access*, vol. 12, no. April, pp. 48987–48996, 2024.
- [3] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Mach. Learn. with Appl.*, vol. 2, no. November, p. 100006, 2020.
- [4] T. B. Brown et al., "Language models are few-shot learners," *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, 2020.
- [5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.
- [6] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 1–67, 2020.
- [7] D. Darmawan and A. Ratnasari, "Rancang Bangun Sistem Informasi Manajemen Proyek Berbasis Web Pada Pt Seatech Infosys," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 365–372, 2020.
- [8] I. R. Yunita, A. Pramono, R. Waluyo, and . S., "Implementasi Metode Waterfall Pada Perancangan Aplikasi Rekam Medis Berbasis Website dan Whatshap Gateway," *J. Informatics Inf. Syst. Softw. Eng. Appl.*, vol. 5, no. 1, pp. 8–16, 2022.
- [9] A. N. Madyanti, N. Ananda, M. W. Rini, and S. Rizal, "Design of a Web Based Information System to Manage Delivery Activities," *J. Rekayasa Sist. Ind.*, vol. 10, no. 02, p. 112, 2023.
- [10] D. Andriansyah, "Penerapan Model Waterfall Pada Sistem Informasi Layanan Jasa Laundry Berbasis Web," *Indones. J. Softw. Eng.*, vol. 4, no. 1, pp. 27–32, 2018.
- [11] H. Hermansyah, R. F. Wijaya, and R. B. Utomo, "Metode Waterfall Dalam Rancang Bangun Sistem Informasi Manajemen Kegiatan Masjid Berbasis Web," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 3, no. 5, pp. 563–571, 2023.
- [12] S. Restu Aji and W. Trisari Harsanti Putri, "Implementasi Teknologi Blockchain dalam Aplikasi E-Voting Berbasis Mobile," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 14, no. 2, pp. 219–231, 2023.