

Pengembangan Aplikasi Desain Grafis Berbasis Web dengan Framework React-Konva dan Generative AI

Widya Ardianto, Nurkhamid

Universitas Negeri Yogyakarta

Email: widyaardianto.2020@student.uny.ac.id

ABSTRAK

Penelitian ini bertujuan mengembangkan aplikasi desain grafis berbasis web menggunakan framework React-Konva dengan Generative AI serta menilai kualitasnya berdasarkan ISO25010:2023. Pengembangan aplikasi menggunakan metodologi *Research and Development* (RnD) dengan model pengembangan perangkat lunak *Feature-Driven Development*. Pengujian kualitas dilakukan pada aspek *functional suitability* melalui validasi ahli dan *performance efficiency* menggunakan Chrome Dev Tools. Hasil penelitian menunjukkan aplikasi berhasil dikembangkan dengan *reactive* dan *declarative binding* serta integrasi Generative AI. Pengujian menghasilkan skor *functional suitability* 100% dengan kategori "Sanget Baik", dan *performance efficiency* menunjukkan penggunaan CPU rata-rata 10%, memori 20,2-37,5 MB, *frame rate* 133,9 fps, dan *response time* 132ms. Aplikasi telah memenuhi standar kualitas yang baik, namun memerlukan optimasi untuk mencapai *response time* di bawah 100ms.

Kata kunci: Aplikasi Desain Grafis, Generative AI, React-Konva, web

ABSTRACT

This research aims to develop a web-based graphic design application utilising the React-Konva framework with Generative AI and assess its quality in accordance with ISO 25010:2023. The application development employed Research and Development (R&D) methodology with the Feature-Driven Development software development model. Quality testing was conducted to assess functional suitability aspects through expert validation and performance efficiency using Chrome DevTools. The results showed that the application was successfully developed with reactive and declarative binding, as well as Generative AI integration. Testing yielded a functional suitability score of 100% in the "Excellent" category, with performance efficiency demonstrating an average CPU usage of 10%, memory usage ranging from 20.2 to 37.5 MB, a frame rate of 133.9 fps, and a response time of 132ms. The application has met good quality standards but requires optimization to achieve response times below 100ms.

Keywords: Generative AI, graphic design application, React-Konva, web

PENDAHULUAN

Perkembangan teknologi *Artificial Intelligence (AI)* yang semakin masif telah membawa transformasi signifikan dalam berbagai bidang pekerjaan, terutama di sektor digital. Munculnya platform Generative AI seperti ChatGPT pada tahun 2022 telah mengakselerasi kemampuan teknologi dalam menghasilkan konten berkualitas tinggi, termasuk *text generation* dan *image*

generation, yang berpotensi mengantikan peran manusia dalam pekerjaan tertentu.

Laporan Goldman Sachs pada April 2023 memperkirakan sekitar 300 juta pekerjaan dapat digantikan oleh Generative AI (Sachs 2023). Hal ini memberikan tantangan sekaligus peluang bagi pelaku industri, khususnya Usaha Mikro, Kecil, dan Menengah (UMKM). Data Kementerian Komunikasi dan Informatika menunjukkan hanya 29% dari

64 juta unit UMKM di Indonesia yang telah mampu berbisnis secara online, mengindikasikan besarnya potensi transformasi digital yang belum dimanfaatkan (Rozza and Metekohy 2023). Pelaku industri UMKM tidak mampu membuat konten promosi digital yang dapat meningkatkan penjualan karena tidak memiliki pakar dikarenakan anggaran yang kurang (Mucharomah 2023). Berdasarkan laporan Boston Consulting Group (BCG) dan Telkom Indonesia, sebesar 57% dari sekitar 3700 UMKM lokal yang disurvei terkendala melakukan transformasi digital karena kekurangan pembiayaan (Ahdiat 2022).

PT Enamo Inovasi Rekatama memberikan solusi digitalisasi UMKM dengan memanfaatkan *Generative AI* yang diintegrasikan ke aplikasi desain grafis berbasis web. *Generative AI* adalah salah satu cabang AI yang mampu menciptakan konten baru seperti teks, gambar, atau audio yang semakin lama semakin tidak dapat dibedakan lagi dari hasil karya manusia (Feuerriegel et al. 2024).

Terdapat beberapa jenis *Generative AI* yang umum digunakan, yaitu *text generation* dan *image generation*. Untuk menggunakan generative AI, digunakan prompt sebagai teknik komunikasi. *Generative AI* memberikan kemudahan bagi pengguna untuk membuat konten, sehingga seorang siswa dapat membuat sebuah esai yang ditulis dengan baik hanya dengan prompt sederhana tanpa melakukan riset apapun (AbuMusab 2023).

Layanan *Generative AI* yang dipasarkan secara umum dilatih dan dijalankan dengan kekuatan komputasi yang sangat tinggi sehingga tidak mungkin dijalankan di perangkat pengguna. Biaya untuk melatih model

ChatGPT mencapai tiga miliar dolar dengan terdapat 350.000 server yang memuat GPU NVIDIA A100 untuk proses *inference* (Moss 2024). Oleh karena itu, perusahaan menyediakan *API* (*Application Programming Interface*) yang dapat diintegrasikan ke aplikasi pihak ketiga.

Generative AI yang diintegrasikan ke aplikasi desain grafis dapat membantu pelaku industri UMKM dalam membuat konten promosi yang lebih menarik secara mudah. *Generative AI* memudahkan individu yang kekurangan kemampuan atau sumber daya untuk memproduksi konten berkualitas tinggi dan bahkan dapat membantu mereka mendapatkan keunggulan kompetitif di pasar (Yao et al. 2024).

Untuk mengembangkan aplikasi desain grafis berbasis web, digunakan elemen *canvas* di HTML5 yang dikendalikan dengan bahasa pemrograman Javascript. *Canvas* memberikan area menggambar berbasis bitmap yang untuk *me-render* grafik, grafis permainan, dan konten visual lainnya secara *real-time* (Pilgrim 2010). *Canvas* memberikan kendali yang lebih rinci atas proses *rendering* dalam mengelola banyak objek, akan tetapi harus dibayar dengan pengelolaan setiap detail secara manual sehingga meningkatkan kompleksitas kode dalam pengaturan dasar dan animasi (Wan 2019). Kompleksitas kode ini dapat diperbaiki dengan menggunakan *framework* atau *library* yang sesuai (Wan 2019).

Konva adalah *library* manipulasi *canvas* yang dikembangkan oleh Anton Lavrenov. *Konva* mendukung berbagai fitur yang sesuai untuk mengembangkan aplikasi desain grafis seperti modifikasi teks, gambar, *vector*, *layer*, dan masih banyak lagi. *Konva* dapat diintegrasikan

ke framework React dengan *framework* React-Konva. Dukungan *reactive* dan *declarative binding* di React-Konva dapat mengurangi kompleksitas pada aplikasi yang dikembangkan. Meskipun terjadi penurunan performa jika dibandingkan dengan menggunakan *vanilla Javascript*, performa yang dihasilkan masih tergolong sangat bagus (Lavrenov 2020).

Penelitian ini membahas tentang bagaimana penerapan *framework* Konva dengan *library* React-Konva dalam pengembangan aplikasi desain grafis berbasis web yang terintegrasi dengan *Generative AI*. *Generative AI* yang digunakan berupa *API* (*Application Programming Interface*) yang menggabungkan beberapa penyedia layanan *Generative AI* berjenis *image generation* seperti *text-to-image generation* dan *image-to-image generation*.

METODE

Penelitian ini menggunakan metode *Research and Development* (R&D). Metode Research and Development digunakan apabila peneliti bermaksud menghasilkan produk tertentu, dan sekaligus menguji keefektifan produk tersebut (Sugiyono 2013). Dalam proses pengembangannya, digunakan model pengembangan perangkat lunak *Feature-Driven Development* (FDD) dengan lima tahapan utama yaitu *Develop an Overall Object Model*, *Build Feature List*, *Plan by Feature Design by Feature*, dan *Develop by Feature*. FDD adalah pendekatan yang dirancang untuk mendapatkan hasil kerja yang nyata dan dapat berfungsi secara berkala (Palmer and Felsing 2002). Tahap pengujian dilakukan menggunakan standar ISO/IEC 25010:2023 dengan menilai aspek *functional suitability* dan *performance efficiency*.

Waktu dan Tempat Penelitian

Kegiatan penelitian dilaksanakan pada bulan Agustus hingga November tahun 2024. Penelitian dilaksanakan di Universitas Negeri Yogyakarta, serta di rumah peneliti.

Subjek dan Objek Penelitian

Subjek penelitian yang menjadi penguji kualitas aplikasi desain grafis berbasis web adalah ahli di bidang teknologi informasi dan desain grafis. Sedangkan objek penelitian ini adalah aplikasi desain grafis berbasis web yang menerapkan *framework* React-Konva dan *Generative AI*.

Teknik Pengumpulan Data

Metode yang digunakan untuk mengumpulkan data pada proses penelitian ini adalah sebagai berikut.

1. Test case

Metode kuesioner pada penelitian ini dilakukan dengan membagikan angket kepada para responden yang merupakan ahli di bidang teknologi informasi dan desain grafis untuk mengetahui tingkat kelayakan aplikasi yang dikembangkan pada aspek *functional suitability*.

2. Dokumentasi

Dokumentasi dilakukan menggunakan Chrome Dev Tools untuk menghasilkan data kuantitatif terkait kinerja aplikasi. Pengukuran meliputi pencatatan penggunaan CPU, penggunaan *memory*, *response time*, dan *frame per seconds* (FPS) yang bertujuan mengevaluasi efisiensi performa aplikasi yang dikembangkan.

Instrumen Penelitian

Instrumen penelitian dirancang untuk mengevaluasi dua aspek utama: *functional suitability* dan *performance efficiency*.

efficiency. Penilaian *functional suitability*, dilakukan menggunakan metode *test case* dengan ceklis yang memuat pertanyaan dengan pilihan jawaban "Ya" dan "Tidak", yang dihitung menggunakan skala Guttman. Penilaian *performance efficiency* dilakukan dengan pengukuran melalui Chrome Dev Tools untuk menganalisis *response time*, *frame per seconds* (FPS), penggunaan CPU, dan penggunaan *memory*.

Teknik Analisis Data

Analisis data dilakukan dengan pendekatan kuantitatif untuk kedua aspek penelitian. Pada penilaian *functional suitability*, perhitungan dilakukan menggunakan rumus persentase dengan membandingkan jumlah item yang lolos terhadap total item dan responden, dengan hasil diklasifikasikan dalam lima kategori dari "Sangat Buruk" hingga "Sangat Baik". Sedangkan penilaian *performance efficiency* dilakukan evaluasi dengan target sebesar 60 *frame per seconds* (Li and Bao 2014), *response time* kurang dari 100ms (Nielsen 1993), serta analisis penggunaan sumber daya komputasi yaitu CPU dan *memory* dengan tujuan mengukur kinerja dan responsivitas aplikasi secara komprehensif.

HASIL DAN PEMBAHASAN

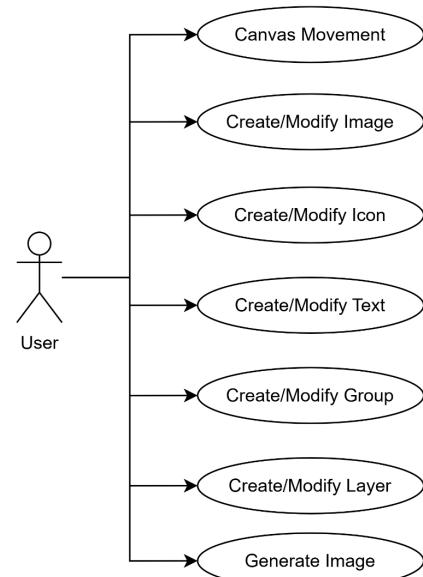
1. Pengembangan Aplikasi

a. Develop an Overall Object Model

Pada tahap awal ini, seluruh anggota pengembang memikirkan, merancang, dan mengajukan apa saja yang diperlukan dalam pengembangan aplikasi. Hasil pemikiran ini kemudian disatukan menjadi gambaran garis besar keseluruhan sistem yang akan dikembangkan. Pada beberapa kasus, terdapat tim atau pimpinan yang sudah

memiliki gambaran keseluruhan sistem yang akan dibuat. Jika demikian, pengembang dapat mengikuti dengan memberikan masukan dan tanggapan untuk menyempurnakan gagasan tersebut (Palmer and Felsing 2002).

Pada penelitian ini, tim desain dari PT Enamo Inovasi Rekatama telah membuat gambaran keseluruhan sistem yang akan dikembangkan. Tim desain telah melakukan analisa terhadap kebutuhan client dan dengan mempertimbangkan kelebihan dan kekurangan dari aplikasi kompetitor seperti Canva, Claid.ai, dan lain-lain. Hasil dari tahap ini adalah *use case diagram* fitur aplikasi secara umum seperti pada Gambar 1.



Gambar 1. Use Case Diagram Overall Object Model

Pada Gambar 1, terdapat fitur umum atau *major feature sets* aplikasi yang akan dikembangkan. Daftar fitur tersebut telah sejalan dengan (Alfie, 2023), seorang tutor IB berpengalaman di forum TutorChase yang menjelaskan fitur standar pada sebuah aplikasi desain grafis yang mencakup *Colour Palettes*,

Typography Tools, Layers, Image Editing, dan Vector Graphics Editing.

b. Build Feature Lists

Tahap ini adalah pemecahan fungsional dari gambaran besar aplikasi dengan berfokus pada fungsional bisnis yang dapat dipahami oleh *client*. Identifikasi interaksi antar fitur dilakukan untuk mengetahui *dependencies* pada setiap fitur (Palmer and Felsing 2002). Pada setiap *major feature sets*, terdapat *feature sets* yang merujuk pada beberapa aktivitas bisnis, sedangkan *features* adalah langkah dari aktivitas bisnis tersebut. Berikut ini adalah hasil penjabaran fitur aplikasi yang akan dikembangkan.

Tabel 1. Feature List Aplikasi

No.	Major Feature Sets	Feature Name
1	Canvas	Drag canvas to other position
2	Canvas	Zoom canvas to other scale
3	Canvas	Store data to automatically rerender when data is changing
4	Image	Drag an image to the canvas
5	Image	Modify size for an image
6	Image	Modify rotation for an image
7	Image	Modify border radius for an image
8	Image	Modify fill for an image
9	Image	Modify border for an image
10	Icon	Drag an icon to the canvas
11	Icon	Modify size for an icon
12	Icon	Modify rotation for an icon
13	Icon	Modify border radius for an icon
14	Icon	Modify fill for an icon
15	Icon	Modify border for an icon
16	Text	Drag a text to the canvas
17	Text	Modify size for a text
18	Text	Modify rotation for a text
19	Text	Modify border radius for a text
20	Text	Modify properties for a text
21	Text	Modify fill for a text
22	Text	Modify border for a text
23	Group	Create a group for some nodes
24	Group	Modify size for a group
25	Group	Modify rotation for a group
26	Layer	Move the layer of a node
27	Generative AI	Generate image from a text (depends on Image)
28	Generative AI	Remove background from an image (depends on Image)
29	Generative AI	Replace background from an image (depends on Image)

c. Plan by Feature

Setiap fitur ditugaskan ke masing-masing penanggungjawab. Selain itu, dilakukan penghitungan estimasi waktu peluncuran tiap fitur. Penentuan

dependencies yang telah dilakukan pada tahap sebelumnya dapat mempermudah proses di tahap ini setiap fitur telah diurutkan. Sebagai contoh, *feature list* pada *major feature set Generative AI* memiliki *dependencies Image*. Oleh karena itu, *Generative AI* hanya dapat dikerjakan setelah *major feature set Image* selesai. Dengan demikian, pengerjaan setiap fitur dapat dilakukan secara beriringan. Berikut ini hasil perencanaan estimasi waktu pengerjaan tiap fitur.

d. *Design by Feature*

Desain user interface telah disediakan oleh tim desain dari PT Enamo Inovasi Rekatama. Analisa dilakukan untuk menentukan komponen React-Konva yang digunakan pada masing-masing *major feature sets* dan struktur data yang digunakan. Se lanjutnya dibuat desain alur kerja setiap komponen berupa *flowchart* atau *sequence diagram*.

1) Canvas

Aplikasi desain grafis yang dikembangkan memiliki beberapa bagian. Sidebar kiri yang memuat pengaturan *canvas*, gudang aset, dan menu pembuatan teks. Sidebar kanan memuat pengaturan *Generative AI*, properti desain dari setiap item, dan pengaturan ekspor. Di bagian bawah terdapat alat untuk melakukan *undo*, *redo*, *zoom*, dan mode kurSOR. Terakhir, di bagian tengah terdapat *canvas* sebagai area kerja utama.

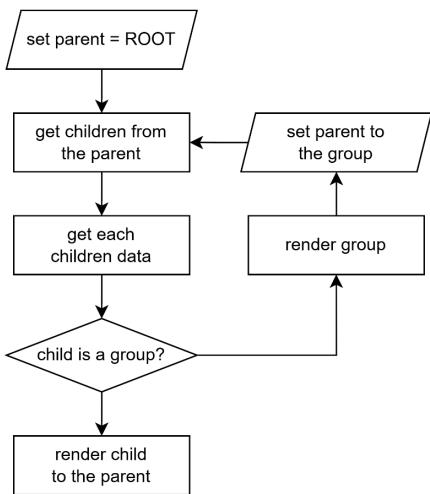
Aplikasi desain grafis memiliki bidang gambar yang tidak terbatas. Oleh karena itu, diperlukan fitur *drag* dan *zoom* untuk dapat melihat seluruh bagian *canvas* pada layar yang memiliki ukuran terbatas. *Drag* adalah menyeret *canvas* untuk melihat area lain dengan menahan tombol pada *scroll wheel* di *mouse* kemudian menggeser posisi *mouse* berlawanan

dengan arah area yang diinginkan. *Drag* yang dilakukan dengan menghitung pergerakan *mouse* saat pengguna menekan *scroll wheel*. Sedangkan *zoom* adalah memperluas atau mempersempit bidang penglihatan untuk melihat ke area luas secara umum atau melihat area sempit secara detail. *Zoom* dilakukan dengan memutar scroll wheel untuk menambah atau mengurangi skala tampilan *canvas*.

Canvas memuat semua gambar dan teks yang ditambahkan oleh pengguna. Semua item tersebut disimpan ke dalam penyimpanan data yang mendukung *reactive state management* di *framework* React. Setiap terjadi perubahan pada data yang disimpan, maka React akan melakukan *rerender* untuk memperbarui tampilan *canvas* sesuai data terbaru. Hal ini dimungkinkan berkat *framework* React-Konva yang menjadi jembatan sebagai interpretasi class yang disediakan Konva menjadi komponen React.

Setiap komponen memiliki identitas berupa atribut *id* yang bersifat unik. Untuk menerapkan hirarki pada proses *render* setiap komponen, maka ditambahkan atribut *parentId* yang berisi *id* dari *parent* suatu komponen. Atribut ini akan digunakan pada oleh komponen *Group* untuk mengetahui *children* yang dimiliki komponen *Group* tersebut. Selain itu, setiap komponen memiliki posisi, ukuran, rotasi, *zIndex*, dan atribut lain sesuai kebutuhan.

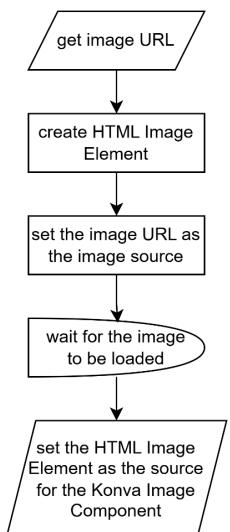
Untuk menerapkan fitur *group*, dibuat komponen *Group* yang akan mengikat beberapa komponen sekaligus secara bersarang. Komponen ini menciptakan hirarki untuk memudahkan pengguna dalam memodifikasi beberapa komponen secara bersamaan. Alur kerja proses *render* hirarki dapat dilihat pada Gambar 2.



Gambar 2. Flowchart Render Hirarki

2) Image

Komponen *Image* digunakan untuk menampilkan gambar dari URL yang disediakan. Komponen *Image* memiliki atribut *imageUrl*, *rotation*, *border radius*, *fill*, dan *border*. Atribut yang berkaitan dengan *border* dan *fill* diterapkan ke bentuk *rectangle* pada komponen. Komponen *Image* pada React-Konva menggunakan elemen *Image* dari HTML sebagai sumber. Elemen ini dapat menerima parameter URL gambar atau base64. Alur *render* komponen *Image* dapat dilihat pada Gambar 3.



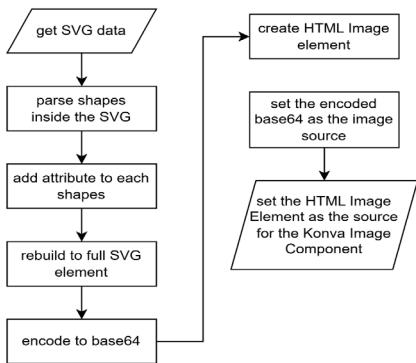
Gambar 3. Flowchart Render Komponen *Image*

3) Icon

Komponen *Icon* digunakan untuk menampilkan *vector* yang berasal dari format elemen SVG pada HTML. Komponen *Icon* memiliki atribut *imageUrl*, *rotation*, *border radius*, *fill*, dan *border*. Berbeda dengan komponen *Image*, atribut *border* dan *fill* pada komponen *Icon* diterapkan ke setiap *vector* di dalamnya.

Konva memiliki komponen untuk menampilkan *vector*, yaitu *Path*. Komponen ini menerima parameter berupa data *vector* dan atribut kustomisasi *fill*, *stroke*, *shadow*, dan *dash* yang sama dengan elemen SVG pada HTML. Akan tetapi, method *getClientRect* pada komponen ini menghitung setiap *vector* yang ada sehingga memakan waktu lebih lama dibandingkan komponen lain yang tidak berbasis *vector*. Komponen *Image* dapat menjalankan method ini dengan memakan waktu hanya 0.06 milidetik, sedangkan komponen *Path* memerlukan waktu 1,3 milidetik atau 20 kali lebih lambat.

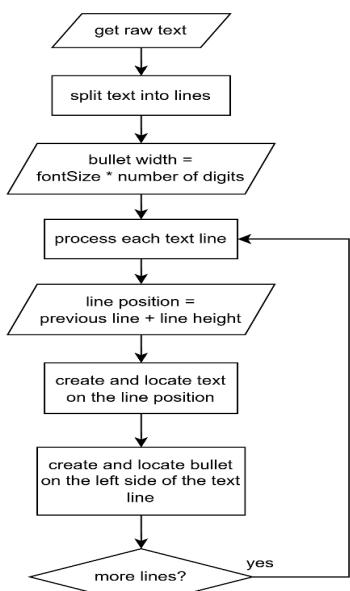
Oleh karena itu, komponen *Image* digunakan untuk menampilkan *vector*. Seperti yang telah dijelaskan sebelumnya, elemen *Image* yang digunakan sebagai sumber komponen *Image* dapat menerima base64. Data *vector* yang berupa elemen SVG perlu dikonversi menjadi base64 untuk dapat digunakan ke komponen *Image*. Selanjutnya, untuk mengatasi perbedaan perilaku atribut *border* dan *fill*, proses konversi ini dilakukan sembari menambahkan atribut tersebut ke setiap data *vector*. Proses konversi akan dilakukan saat terjadi perubahan data pada atribut yang berkaitan. Alur *render* komponen *Icon* dapat dilihat pada Gambar 4.



Gambar 4. Flowchart Render Komponen Icon Menggunakan Komponen Image pada Konva

4) Text

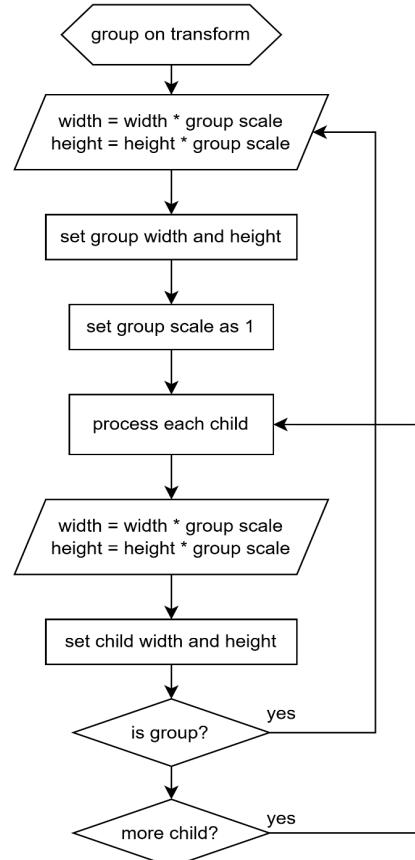
Komponen *Text* digunakan untuk menampilkan teks dengan mendukung atribut *font*, *weight*, *size*, *letter*, *line*, *align*, *italic*, *bullet/numbering*, *decoration*, dan *letter case*. Atribut yang berkaitan dengan *border* dan *fill* diterapkan ke setiap *font*. Untuk mendukung *bullet/numbering* yang stabil, proses render teks dilakukan dengan memisahkan teks berdasarkan baris yang di-render menggunakan komponen *Text* masing-masing. Komponen *bullet/numbering* ditempatkan ke sisi kiri setiap baris. Alur render komponen *Text* dapat dilihat pada Gambar 5.



Gambar 5. Flowchart Render Komponen Text

5) Group

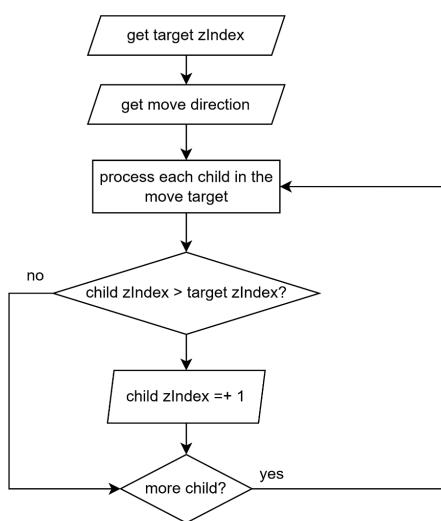
Komponen *Group* adalah kumpulan yang memuat beberapa komponen di dalamnya. Suatu *Group* dapat berperan melakukan satu aksi untuk semua komponen yang ada di dalamnya. Komponen *Group* dari Konva dapat memuat komponen lain di dalamnya. Atribut *scale* pada komponen *Group* diterapkan ke semua *children*. Pembuatan *Group* yang bertingkat dapat membuat ukuran komponen di dalamnya sulit untuk diketahui. Untuk mengabaikan atribut *scale* yang relatif terhadap *parent*, maka perlu dilakukan normalisasi dengan menerapkan atribut *scale* ke semua *children* dan membuat komponen *Group* memiliki *scale* bernilai 1. Alur kerja normalisasi atribut *scale* dapat dilihat pada Gambar 6.



Gambar 6. Flowchart Proses Normalisasi Atribut scale Komponen Group

6) Layer

Fitur *Layer* digunakan untuk memindahkan posisi komponen yang saling tumpang tindih. Pengguna dapat menggeser suatu komponen ke atas atau ke bawah. Komponen tersebut akan menerima atribut *zIndex* baru, sedangkan atribut *zIndex* pada komponen lain akan diperbarui sesuai urutan terbaru. Alur perpindahan komponen dapat dilihat pada Gambar 7.



Gambar 7. Flowchart Alur Perpindahan Layer Komponen

7) Generative AI

Fitur *Generative AI* mendukung fungsi *text-to-image*, *remove background*, dan *replace background*. Pengguna perlu memasukkan *prompt*, memilih gambar yang ingin dimodifikasi, dan menentukan fungsi yang akan digunakan. Parameter tersebut akan dikirim ke penyedia layanan *Generative AI* melalui *API* (*Application Programming Interface*). Hasilnya akan ditampilkan ke pengguna berupa beberapa pilihan hasil.

e. Build by Feature

Pada tahap ini, proses pengembangan aplikasi dilakukan sesuai hasil perencanaan *timeline* dan desain

yang telah dibuat. Proses inisiasi proyek dilakukan dengan menggunakan *boilerplate React-vite*. *Library* pendukung digunakan untuk menjalankan fungsi tertentu seperti *Zustand* sebagai penyimpanan data global yang mendukung sifat *reactive*, serta *Axios* untuk melakukan panggilan *API*. Desain *user interface* yang dibuat oleh tim desain diterapkan ke dalam kode *HTML* dan *CSS*. *Library Tailwind* digunakan untuk memudahkan penerapan *style* dengan penggunaan *class* yang sesuai. Komponen *React-Konva* yang telah direncanakan pada tahap sebelumnya digunakan beserta *hook* dari *React* untuk mendukung beberapa kasus seperti *useMemo* untuk mengambil *children* pada setiap *Group* dan *useEffect* untuk mengembalikan sifat *reactive* pada komponen *Icon* yang menggunakan komponen *Image*.

2. Pengujian Aplikasi

a. Hasil Pengujian Functional Suitability

Pengujian dilakukan oleh Ahmad Amirudin (CEO PT Enamo Inovasi Rekatama) dan Ahsan Firdaus (*Software Engineer* di PT Solusi Pembayaran Elektronik) dengan menguji daftar fitur pada Tabel 1. Masing-masing penguji memberikan skor 100% sehingga aplikasi yang dikembangkan termasuk kategori “Sangat Baik”.

b. Hasil Pengujian Performance Efficiency

Pengujian dilakukan dengan skenario terdapat 15 komponen *Image*, 15 komponen *Text* dan 15 komponen *Icon* yang dikelompokkan ke dalam 10 komponen *Group* secara bertingkat. Hasilnya adalah penggunaan rata-rata CPU sebesar 10% dan *memory* sebesar 20,2 MB hingga 37,5 MB. Pengujian pada

frame rate menghasilkan angka sebesar 133,9 *frame per seconds* yang telah melampaui batas minimal pengguna untuk tidak merasakan *delay* yaitu 60 *frame per seconds* seperti yang dijelaskan oleh (Li and Bao 2014). Rata-rata *response time* pada setiap fitur menghasilkan angka sebesar 132 milidetik yang melampaui batas maksimal untuk pengguna merasa sistem memberikan respon secara instan seperti yang dijelaskan oleh (Nielsen 1993) yaitu 100 milidetik. Akan tetapi, hasil tersebut bisa dikatakan mendekati, sehingga aplikasi dapat dinyatakan "**Baik**".

KESIMPULAN

Pengembangan aplikasi desain grafis berbasis web menggunakan framework React-Konva dengan pendekatan *Feature-Driven Development* telah berhasil dilaksanakan melalui lima tahapan pengembangan yang sistematis. Dalam implementasinya, fitur *Generative AI* berhasil diintegrasikan menggunakan API pihak ketiga yang memungkinkan pengguna memasukkan parameter sesuai kebutuhan. Hasil pengujian kualitas berdasarkan standar ISO 25010:2023 menunjukkan bahwa aplikasi memenuhi kriteria "**Baik**", dengan aspek *functional suitability* mencapai nilai sempurna 100% dalam kategori "**Sangat Baik**". Dari segi performance efficiency, aplikasi menunjukkan penggunaan sumber daya yang efisien dengan konsumsi CPU rata-rata 10% dan penggunaan memori antara 20,2 MB hingga 37,5 MB. Performa aplikasi mencapai 133,9 *frame per seconds*, melampaui standar minimal yang ditetapkan (Li and Bao 2014). Meskipun demikian, *response time* rata-rata 132 milidetik masih belum memenuhi kriteria optimal yang direkomendasikan (Nielsen

1993). Temuan ini berimplikasi pada perlunya optimisasi lebih lanjut, khususnya dalam aspek *response time* untuk meningkatkan pengalaman pengguna. Pengembangan ke depan dapat difokuskan pada implementasi teknik-teknik optimisasi performa untuk mencapai *response time* yang lebih baik sesuai standar yang direkomendasikan.

DAFTAR RUJUKAN

- AbuMusab, Syed. 2023. "Generative AI and Human Labor: Who Is Replaceable?" *AI & SOCIETY* s00146-023-01773-3. doi: 10.1007/s00146-023-01773-3.
- Ahdiat, Adi. 2022. "Banyak UMKM Belum Bisa Bisnis Online, Ini Kendala Utamanya." *Banyak UMKM Belum Bisa Bisnis Online, Ini Kendala Utamanya*. Retrieved January 22, 2025 (<https://databoks.katadata.co.id/ekonomi/makro/statistik/e212973eac08562/banyak-umkm-belum-bisa-bisnis-online-ini-kendala-utamanya>).
- Feuerriegel, Stefan, Jochen Hartmann, Christian Janiesch, and Patrick Zschech. 2024. "Generative AI." *Business & Information Systems Engineering* 66(1):111–26. doi: 10.1007/s12599-023-00834-7.
- Lavrenov, Anton. 2020. "Getting Started with React and Canvas via Konva." *Getting Started with React and Canvas via Konva*. Retrieved March 12, 2024 (<https://konvajs.org/docs/react/index.html>).
- Li, Xianfeng, and Zhiqiang Bao. 2014. "Performance Characterization of Web Applications with HTML5 Enhancements." Pp. 252–58 in 2014 *IEEE 12th International Conference on Dependable, Autonomic and*

- Secure Computing. Dalian, China: IEEE.
- Moss, Sebastian. 2024. "OpenAI Training and Inference Costs Could Reach \$7bn for 2024, AI Startup Set to Lose \$5bn - Report." *Data Center Dynamics*. Retrieved August 24, 2024 (<https://www.datacenterdynamics.com/en/news/openai-training-and-inference-costs-could-reach-7bn-for-2024-ai-startup-set-to-lose-5bn-report/>).
- Mucharomah, Nur Laily. 2023. "Survei: 70% UMKM Lokal Di Indonesia Kesulitan Memasarkan Produk." *Survei: 70% UMKM Lokal Di Indonesia Kesulitan Memasarkan Produk*. Retrieved March 8, 2024 (<https://dailysocial.id/post/survei-70-persen-umkm-lokal-di-indonesia-kesulitan-memasarkan-produk>).
- Nielsen, Jacob. 1993. "Response Times: The 3 Important Limits." *Response Times Limits: Article by Jacob Nielsen*. Retrieved November 6, 2024 (<https://www.nngroup.com/articles/response-times-3-important-limits/>).
- Palmer, Stephen R., and John M. Felsing. 2002. *A Practical Guide to Feature-Driven Development*. Upper Saddle River, NJ: Prentice Hall PTR.
- Pilgrim, Mark. 2010. *HTML5: Up and Running; [Dive into the Future of Web Development]*. 1. ed. Beijing Köln: O'Reilly.
- Rozza, Sylvia, and Elisabeth Y. Metekohy. 2023. "RENDAHNYA ADOPSI DIGITAL MARKETING PADA UMKM." 22(1):24–34.
- Sachs, Goldman. 2023. "Generative AI Could Raise Global GDP by 7%." *Goldman Sachs*. Retrieved March 12, 2024 (<https://www.goldmansachs.com/intelligence/pages/generative-ai-could-raise-global-gdp-by-7-percent.html>).
- Sugiyono. 2013. *METODE PENELITIAN KUANTITATIF, KUALITATIF, DAN R&D. ALFABETA*.
- Wan, Alvin. 2019. "When to Use HTML5's Canvas." *LogRocket Blog*. Retrieved August 16, 2024 (<https://blog.logrocket.com/when-to-use-html5s-canvas-ce992b100ee8/>).
- Yao, Fan, Chuanhao Li, Denis Nekipelov, Hongning Wang, and Haifeng Xu. 2024. "Human vs. Generative AI in Content Creation Competition: Symbiosis or Conflict?"