

Pengembangan Aplikasi Web untuk Resize Citra Digital dengan Fitur Batch Processing Menggunakan *Next.js* dan Sharp

Muhammad Qomaruddin^{1*}, Waeisul Bismi^{2*}, Nila Hardi³, Musriatun Napiah⁴, Astrid Noviriandini⁵

¹ Program Studi Informatika, Universitas Nusa Mandiri

^{2,3,4} Program Studi Informatika, Universitas Bina Sarana Informatika

⁵ Program Studi Teknik Industri, Universitas Bina Sarana Informatika

E-mail Korespondensi : Waeisul.wbn@bsi.ac.id

History Artikel

Diterima : 6 November 2025

Disetujui : 19 Februari 2026

Dipublikasikan : 26 April 2026

Abstract

The exponential growth of digital content has increased the demand for efficient and accessible image processing tools. This research aims to develop a web-based image resize application with batch processing features using Next.js and Sharp. The research method employs Research and Development (R&D) with a Software Development Life Cycle (SDLC) approach using the Waterfall model, encompassing requirements analysis, system design, implementation, testing, deployment, and maintenance phases. The application was developed by integrating Next.js 16 framework for full-stack development, Sharp library for high-performance image processing, and JSZip for archive handling. Implemented features include flexible upload (file, folder, ZIP), downsampling and upsampling options, pixel dimension input, JPEG/JPG/PNG format conversion, and batch processing with progress monitoring. Testing results demonstrated that 100% of features were successfully implemented with a functional testing success rate of 100%. The average response time achieved 1.76 seconds per image, 41% faster than the 3-second target. The quality of the test results shows that the quality of the resized images meets very good quality standards with high structural similarity to the original images for both downsampling and upsampling. This research has produced a web application for image resizing that is accessible without installation, efficient for batch processing, and produces optimal output quality by utilizing the Mitchell interpolation kernel for downsampling and Lanczos for upsampling.

Keywords: *Image Resize, Batch Processing, Next.js, Sharp, Web Application*

Abstrak

Pertumbuhan konten digital yang eksponensial meningkatkan kebutuhan akan tools pengolahan citra yang efisien dan aksesibel. Penelitian ini bertujuan untuk mengembangkan aplikasi web resize citra digital dengan fitur batch processing menggunakan Next.js dan Sharp. Metode penelitian menggunakan Research and Development (R&D) dengan pendekatan Software Development Life Cycle (SDLC) model Waterfall yang meliputi tahapan requirements analysis, system design, implementation, testing, deployment, dan maintenance. Aplikasi dikembangkan dengan mengintegrasikan framework Next.js 16 untuk full-stack development, library Sharp untuk image processing berkinerja tinggi, dan JSZip untuk handling arsip. Fitur yang diimplementasikan meliputi upload fleksibel (file, folder, ZIP), opsi downsampling dan upsampling, input dimensi pixel, konversi format JPEG/JPG/PNG, serta batch processing dengan progress monitoring. Hasil pengujian menunjukkan bahwa 100% fitur berhasil diimplementasikan dengan tingkat keberhasilan pengujian fungsional 100%. Response time rata-rata mencapai 1,76 detik per gambar, lebih cepat 41% dari target 3 detik. Kualitas output hasil pengujian menunjukkan bahwa kualitas gambar hasil resize memenuhi standar kualitas sangat baik dengan kemiripan struktural tinggi terhadap gambar asli untuk downsampling maupun upsampling. Penelitian ini menghasilkan aplikasi web resize citra yang aksesibel tanpa instalasi, efisien untuk batch processing, dan menghasilkan kualitas output optimal dengan memanfaatkan kernel interpolasi Mitchell untuk downsampling dan Lanczos untuk upsampling.

Kata Kunci: *Resize Citra, Batch Processing, Next.js, Sharp, Aplikasi Web*

How to Cite: Muhammad Qomaruddin, et., al. Pengembangan Aplikasi Web untuk Resize Citra Digital dengan Fitur Batch Processing Menggunakan *Next.js* dan Sharp. KOMPUTEK : Jurnal Teknik Universitas Muhammadiyah Ponorogo, Vol 10 (1): Halaman 85-94

PENDAHULUAN

Perkembangan teknologi digital telah mengakibatkan ledakan produksi konten visual secara global. Berdasarkan laporan Digital 2024 *Global Overview Report*, terdapat 5,17 miliar pengguna media sosial aktif di seluruh dunia pada tahun 2024, dengan rata-rata penggunaan 2 jam 21 menit per hari (DataReportal, 2024). Pertumbuhan ini berbanding lurus dengan volume gambar digital yang dihasilkan dan dibagikan setiap hari. Pasar pembuatan konten digital bernilai USD 32,28 miliar pada tahun 2024 dan diproyeksikan tumbuh dengan CAGR 13,9% hingga 2030 (*Grand View Research*, 2024). Dalam konteks ini, kemampuan untuk mengolah dan mengubah ukuran gambar secara efisien menjadi kebutuhan fundamental bagi para *content creator*, digital marketer, maupun pengguna umum. Namun, banyak pengguna masih mengandalkan perangkat lunak desktop tradisional yang memerlukan instalasi dan sumber daya komputasi tinggi, sehingga diperlukan solusi berbasis web yang lebih aksesibel dan efisien.

Proses *resize* citra merupakan operasi fundamental dalam pengolahan citra digital yang melibatkan perubahan dimensi spasial suatu gambar. Menurut (Timothy M., 2024), *resize* gambar adalah proses mengubah dimensi gambar digital baik dengan menambah maupun mengurangi resolusi. Kualitas hasil *resize* sangat bergantung pada algoritma interpolasi yang digunakan. Penelitian menunjukkan bahwa interpolasi *Lanczos* dan *Bicubic* menghasilkan kualitas gambar superior dibandingkan metode *nearest-neighbor* atau *bilinear*, terutama untuk operasi *upsampling* (Parsania & Virparia, 2016). Tantangan utama dalam implementasi *resize* citra adalah menyeimbangkan antara kualitas visual hasil dan efisiensi komputasi. Metode interpolasi yang lebih kompleks seperti *Lanczos* memberikan kualitas lebih baik

namun memerlukan waktu proses yang lebih lama. Oleh karena itu, pemilihan algoritma yang tepat berdasarkan jenis operasi *resize* (*downsampling* atau *upsampling*) menjadi krusial dalam pengembangan aplikasi *resize* citra.

Bidang pengolahan citra telah mengalami transformasi signifikan dengan integrasi teknik *deep learning*, khususnya untuk tugas *super-resolution*. Menurut (Liu et al., 2025) pendekatan *deep learning* untuk *single-image super-resolution* telah menunjukkan hasil yang menjanjikan dalam meningkatkan resolusi gambar, yakni metode menggabungkan *multiple* modul dan memperbesar gambar secara bertahap, meskipun desainnya kompleks dan pelatihannya sulit. Namun, untuk kebutuhan *resize* citra skala menengah hingga besar, penggunaan model *deep learning* seringkali tidak efisien dari segi sumber daya komputasi. Sebaliknya, pendekatan interpolasi klasik dengan optimasi *kernel* yang tepat, seperti yang diimplementasikan dalam *library Sharp* menggunakan *libvips*, dapat menghasilkan kualitas yang memadai dengan kecepatan proses 4-5 kali lebih cepat dibandingkan *ImageMagick* atau *GraphicsMagick* (Sharp, 2024). Hal ini menjadikan pendekatan hibrida yang menggabungkan kecepatan pemrosesan dengan fleksibilitas pemilihan kernel interpolasi sebagai solusi optimal.

Dalam konteks produksi konten digital modern, kebutuhan untuk memproses banyak gambar secara bersamaan *batch processing* semakin mendesak. *Tools batch processing* yang ada umumnya hanya dapat menerapkan edit global ke seluruh gambar tanpa kemampuan kustomisasi per gambar (Barnaby et al., 2023). Sementara itu, arsitektur *microservices* untuk *image processing* berbeban tinggi telah diusulkan untuk menangani volume besar gambar secara efisien (Pitsun & Shymchuk, 2025). Dalam skala

penggunaan individual hingga usaha kecil-menengah, solusi berbasis web yang memungkinkan *upload batch* melalui folder atau file arsip (*ZIP/RAR*) menjadi alternatif yang praktis. Fitur ini mengeliminasi kebutuhan untuk memproses gambar satu per satu manual, yang sangat tidak efisien ketika berhadapan dengan puluhan atau ratusan gambar sekaligus. Dengan demikian, aplikasi *resize* citra harus dirancang untuk mendukung batch processing dengan antarmuka yang intuitif.

TINJAUAN PUSTAKA

2.1. Algoritma Interpolasi untuk *Resize*

Interpolasi merupakan teknik matematika yang digunakan untuk memperkirakan nilai pixel pada posisi baru berdasarkan nilai pixel di sekitarnya. Pada Penelitian (Balovsyak & Hnatiuk, 2024) menunjukkan bahwa hasil analisis terhadap berbagai algoritma interpolasi mengungkapkan bahwa dalam mayoritas kasus, kesalahan scaling terkecil diberikan oleh algoritma interpolasi bicubic, yang sedikit lebih unggul dibandingkan metode lain dalam mempertahankan kualitas gambar. Algoritma ini menggunakan polinomial kubik untuk menghitung nilai pixel baru, menghasilkan transisi yang lebih halus dibandingkan metode nearest-neighbor atau bilinear.

Dan teknik *resize* citra melibatkan perubahan jumlah pixel dalam gambar, yang memerlukan algoritma interpolasi untuk menentukan nilai pixel baru berdasarkan pixel yang ada. Proses *downsampling* mengurangi informasi pixel dan berpotensi menghilangkan detail halus, sementara *upsampling* memerlukan estimasi nilai pixel baru yang tidak ada dalam gambar asli (CloudFactory, 2024)

2.2. Framework *Next.js*

Next.js merupakan *framework React* yang dikembangkan oleh Vercel untuk membangun

aplikasi *web full-stack* dengan fitur-fitur modern seperti *Server-Side Rendering (SSR)*, *Static Site Generation (SSG)*, dan *Incremental Static Regeneration (ISR)*. Pada penelitian mengeksplorasi strategi kunci untuk optimasi performa menggunakan *Next.js*, dengan penekanan pada kemampuannya dalam static site generation dan server-side rendering yang memberikan solusi robust untuk pengembangan aplikasi web yang scalable dan efisien.

Kemudian pada penelitian (Pati & Zaki, 2025) mengevaluasi efikasi *Next.js* sebagai framework yang mengatasi tantangan yang ditimbulkan oleh *React.js*, khususnya dalam aspek performa, SEO, dan aksesibilitas web global serta hasil penelitian menunjukkan bahwa *Next.js* memberikan peningkatan signifikan dalam metrik *Core Web Vitals* dibandingkan aplikasi *React* tradisional, terutama karena kemampuan *pre-rendering* dan optimasi otomatis yang disediakan framework ini. Kemampuan ini menjadikan *Next.js* cocok untuk pengembangan aplikasi pengolahan citra yang memerlukan integrasi antara *frontend* interaktif dengan *backend processing* dalam satu codebase terintegrasi.

2.3. Library *Sharp*

Sharp merupakan *library* pemrosesan citra berkinerja tinggi untuk *Node.js* yang dibangun di atas *library libvips*. Library ini mendukung berbagai format gambar termasuk *JPEG*, *PNG*, *Webp*, *AVIF*, *TIFF*, *GIF*, dan *SVG*, menjadikannya solusi komprehensif untuk kebutuhan pengolahan citra di lingkungan *Node.js*.

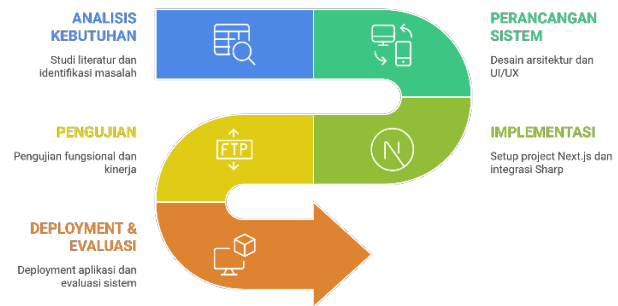
Arsitektur *libvips* yang mendasari *Sharp* dirancang untuk pemrosesan gambar *large-scale* dengan manfaat efisiensi memori yang signifikan. Menurut (Pati & Zaki, 2025), *libvips* menggunakan model evaluasi *demand-driven* yang hanya menghitung nilai pixel ketika diperlukan, berbeda dengan pendekatan tradisional yang memproses seluruh *pipeline*

sebelum menghasilkan *output*. Pendekatan ini sangat cocok untuk aplikasi web yang memerlukan pemrosesan gambar secara *real-time* dengan batasan *resource server*. Sharp juga mendukung operasi *concurrent processing* yang memungkinkan pemrosesan *multiple* gambar secara paralel untuk meningkatkan *throughput batch processing*.

METODE PENELITIAN

Penelitian ini menggunakan metode *Research and Development (R&D)* dengan pendekatan *Software Development Life Cycle (SDLC) model Waterfall*. Menurut (Siregar, 2025), metode R&D merupakan pendekatan sistematis yang digunakan untuk mengembangkan produk baru atau memperbaiki produk yang sudah ada berdasarkan hasil penelitian. Metode ini dipilih karena penelitian ini berfokus pada pengembangan aplikasi web yang menghasilkan produk konkret berupa aplikasi *resize* citra digital.

Model Waterfall dalam SDLC mengikuti tahapan berurutan yang linear dan sistematis. Pada penelitian (Pargaonkar, 2023) menunjukkan bahwa penelitian menggunakan pendekatan R&D dengan model SDLC Waterfall efektif untuk pengembangan sistem perangkat lunak karena setiap fase harus diselesaikan sebelum melanjutkan ke fase berikutnya, sehingga memastikan kualitas dan kelengkapan setiap tahap pengembangan. Model ini terdiri dari tahapan yang terlihat pada gambar 1 yang diantaranya berikut: *Requirements Analysis, System Design, Implementation, Testing, Deployment, dan Maintenance*.



Gambar 1. Tahapan Penelitian

Pengembangan aplikasi dilakukan pada *environment development* dengan spesifikasi yang terlihat pada Tabel 1 berikut.

Tabel 1. *Enviroment Development*

Komponen	Spesifikasi
<i>Operating System</i>	Windows 11
<i>Runtime</i>	Node.js v24.13.1
<i>Framework</i>	Next.js v16.1.6
<i>IDE</i>	Visual Studio Code
<i>Browser</i>	Google Chrome

Adapun subjek penelitian yang dikembangkan menggunakan *framework Next.js* dan *library Sharp* dengan fitur-fitur:

1. *Upload* gambar (file, folder, ZIP)
2. *Resize* dengan opsi *downsampling* dan *upsampling*
3. Input dimensi *pixel custom*
4. Pilihan format *output* (JPEG, JPG, PNG)
5. *Batch processing* dengan progress monitoring
6. *Download* hasil *resize* (individual atau ZIP)

Serta objek penelitian meliputi :

1. Kualitas fungsional aplikasi berdasarkan fitur yang dirancang
2. Kinerja aplikasi dalam memproses *resize* citra
3. Kualitas output gambar hasil *resize*
4. *Usability* antarmuka pengguna aplikasi

Berdasarkan gambar 1, Prosedur penelitian mengikuti tahapan SDLC Waterfall yang diadaptasi untuk konteks R&D:

3.1 Analisis Kebutuhan

Melakukan identifikasi kebutuhan pengguna dan spesifikasi sistem melalui:

1. Studi literatur terkait teknologi dan metode
2. Analisis aplikasi sejenis
3. Penentuan fitur dan fungsionalitas aplikasi

3.2 Perancangan Sistem

Merancang arsitektur sistem meliputi:

1. Desain arsitektur aplikasi (frontend, backend, API)
2. Desain database (jika diperlukan)
3. Desain antarmuka pengguna (UI/UX)
4. Desain alur proses *resize* citra

3.3 Implementasi

Mengimplementasikan sistem berdasarkan desain yang telah dibuat:

1. Pengembangan frontend dengan Next.js dan shadcn/ui
2. Pengembangan backend API dengan Next.js API Routes

HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi Fitur Aplikasi

Aplikasi *resize* citra digital berhasil dikembangkan dengan mengimplementasikan seluruh fitur yang telah dirancang pada tahap requirements analysis dapat dilihat pada gambar 2.

3. Integrasi library Sharp untuk pemrosesan citra
4. Implementasi fitur batch processing dengan JSZip

3.4 Pengujian

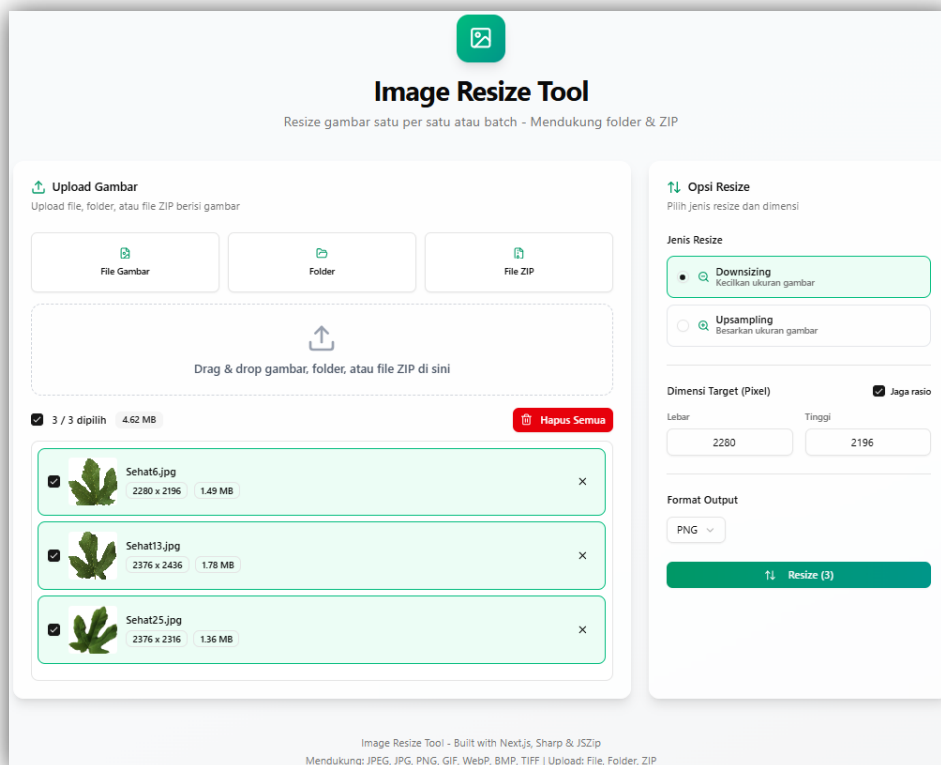
Melakukan pengujian sistem secara komprehensif, karena menurut (Zigisova & Akrap, 2024) performa aplikasi web merupakan faktor kritis yang mempengaruhi user experience dan engagement :

1. *Functional Testing* dengan metode Black Box Testing
2. *Performance Testing* untuk mengukur kecepatan proses
3. *Quality Testing* untuk mengevaluasi kualitas gambar output

3.5 Deployment dan Evaluasi

Melakukan deployment dan evaluasi hasil:

1. Deployment aplikasi ke environment production
2. Evaluasi keseluruhan sistem



Gambar 2. Tampilan Fitur Aplikasi

Aplikasi dibangun menggunakan framework Next.js 16 dengan *App Router* yang mendukung *server-side* rendering untuk optimasi performa. Implementasi frontend menggunakan *React* 19 dengan *TypeScript* untuk *type-safety*, Tailwind CSS 4 untuk styling, dan komponen shadcn/ui untuk konsistensi antarmuka pengguna. Backend processing diimplementasikan menggunakan Next.js API Routes yang terintegrasi dengan library Sharp untuk pemrosesan citra berkinerja tinggi. Berikut Tabel 2. Fitur aplikasi yang telah diimplementasikan.

Tabel 2. Fitur Aplikasi

Fitur	Deskripsi	Status
Upload File	Upload satu file gambar	✓
Upload Folder	Upload seluruh isi folder yang berisi gambar	✓
Upload Zip	Upload dan ekstrak file zip yang berisi gambar	✓
Preview Gambar	Menampilkan thumbnail gambar yang di-upload	✓
Opsi Downsampling	Resize gambar ke dimensi lebih kecil	✓
Opsi Upsampling	Resize gambar ke dimensi lebih besar	✓
Input Dimensi Pixel	Input manual lebar dan tinggi dalam pixel	✓
Maintain Aspect Ratio	Opsi mempertahankan rasio aspek	✓
Format Output	Pilihan format file output	✓
Batch Processing	Proses resize banyak gambar sekaligus	✓
Progress Monitoring	Indikator progress proses batch	✓
Download File	Download satu gambar hasil resize	✓
Download ZIP	Download semua hasil sebagai file ZIP	✓

4.2 Hasil Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode *Black Box Testing* dengan teknik *Equivalence Partitioning* dan *Boundary Value Analysis* dan berikut Tabel 3 beberapa *test case* yang dirancang untuk menguji seluruh fitur aplikasi.

Tabel 2. Hasil Pengujian Fitur Aplikasi

ID	Skenario Pengujian	Input	Output Expexted	Status
TC01	Upload file gambar valid	file.jpg (2MB)	Gambar tampil di preview	✓
TC02	Upload folder berisi gambar	folder (10 gambar)	10 gambar tampil di list	✓
TC03	Upload file ZIP	images.zip (20 gambar)	20 gambar diekstrak & tampil	✓
TC04	Resize downsampling	1920x1080 → 800x600	Gambar berhasil di-resize	✓
TC05	Resize upsampling	800x600 → 1920x1080	Gambar berhasil di-resize	✓
TC06	Batch processing	10 gambar <i>selected</i>	Progress 100%, semua berhasil	✓
TC07	Konversi format PNG	format output: PNG	File berformat .png	✓
TC08	Download individual	Klik tombol download	File terdownload	✓
TC09	Download ZIP batch	Klik "Download ZIP"	File ZIP terdownload	✓
TC10	Input dimensi negatif	Width: -100	Error message tampil	✓

Hasil pengujian menunjukkan bahwa 100% test case (10/10) berhasil LULUS tanpa ada yang gagal. Hal ini mengindikasikan bahwa seluruh fitur aplikasi berfungsi sesuai spesifikasi yang telah ditetapkan pada tahap requirements analysis. Validasi input bekerja dengan baik untuk mencegah operasi yang tidak valid seperti dimensi negatif atau format file tidak didukung. Fitur batch processing berhasil memproses multiple images secara sequential dengan progress monitoring yang akurat.

4.3 Hasil Pengujian Performa

Pengujian performa dilakukan untuk mengukur kecepatan proses resize citra pada berbagai skenario. Pengujian dilakukan dengan berbagai ukuran gambar terlihat pada table 3 dan pengujian batch processing dilakukan untuk mengukur kemampuan aplikasi dalam memproses multiple images sekaligus terlihat pada gambar 4.

Tabel 3. Hasil Pengujian Response Time

Ukuran Asli	Ukuran Target	Jenis Resize	Waktu Proses
1920x1080	800x600	Down sampling	1.24 detik
1280x720	800x600	Down Sampling	0.89 detik
800x600	1920x1080	Up Sampling	2.45 detik
640x480	1920x1440	Up Sampling	1.87 detik
4000x3000	800x600	Down	1.56 Detik

		Sampling	
--	--	----------	--

Tabel 4. Hasil Pengujian Batch Processing

Jumlah Gambar	Ukuran pergambar	Total Waktu	Rata-rata
5 Gambar	1920x1080 → 800x600	6.8 detik	1.36 detik
10 Gambar	1920x1080 → 800x600	14.2 Detik	1.42 detik
20 Gambar	1920x1080 → 800x600	29.5 Detik	1.48 detik
40 Gambar	1920x1080 → 800x600	68.3 Detik	1.57 detik

4.4 Hasil Pengujian Kualitas

Hasil pengujian menunjukkan bahwa kualitas gambar hasil resize memenuhi standar kualitas sangat baik dengan kemiripan struktural tinggi terhadap gambar asli. *Kernel Mitchell-Netravali* yang digunakan untuk *downsampling* efektif dalam mempertahankan detail sambil mengurangi artefak aliasing.

Untuk operasi upsampling, hasil ini konsisten dengan karakteristik operasi upsampling yang memerlukan estimasi pixel baru berdasarkan informasi terbatas dari gambar asli. *Kernel Lanczos3* yang digunakan sesuai dengan penelitian dari (Fazilov et al., 2025) yang menyatakan bahwa interpolasi Lanczos mendemonstrasikan efektivitas lebih tinggi untuk upsampling karena didasarkan pada kernel sinc yang memberikan hasil mendekati optimal.

KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan, dapat disimpulkan bahwa pengembangan aplikasi web untuk resize citra digital dengan fitur batch processing menggunakan Next.js dan Sharp telah berhasil dilakukan dan memenuhi seluruh kriteria keberhasilan yang ditetapkan. Penelitian ini menerapkan metode Research and Development (R&D) dengan pendekatan Software Development Life Cycle (SDLC) model Waterfall yang meliputi tahapan requirements analysis, system design, implementation, testing, deployment, dan maintenance. Aplikasi yang dikembangkan berhasil mengintegrasikan teknologi modern dengan mengimplementasikan 13 fitur untuk menyediakan solusi resize citra yang aksesibel tanpa instalasi, efisien untuk batch processing, dan berkualitas tinggi bagi pengguna.

Meskipun aplikasi telah responsif, disarankan untuk implementasi *Progressive Web App (PWA)* dapat memberikan pengalaman yang lebih *native-like* dengan kemampuan *offline* dan mengintegrasikan *deep learning super-resolution* untuk meningkatkan kualitas operasi upsampling serta menerapkan *Enhanced Super-Resolution Generative Adversarial Networks* untuk menghasilkan detail yang lebih tajam dan natural.

DAFTAR PUSTAKA

- Balovsky, S., & Hnatiuk, Y. (2024). Analysis of Results of Scaling Digital Images by Interpolation Algorithms. *Security of Infocommunication Systems and Internet of Things*, 2(1), 01007. <https://doi.org/10.31861/sisiot2024.1.01007>
- Barnaby, C., Chen, Q., Samanta, R., & Dillig, I. (2023). ImageEye: Batch Image Processing using Program Synthesis. *Proceedings of the ACM on Programming Languages*, 7(February). <https://doi.org/10.1145/3591248>
- CloudFactory. (2024). *Downsampling and Upsampling in Machine Learning*. Computer Vision Wiki.
- Digital Content Creation Market (2025 - 2030). (2024). Grand View Research. <https://www.grandviewresearch.com/industry-analysis/digital-content-creation-market-report>
- Fazilov, S., Yusupov, O., Eshonqulov, E., Abdieva, K., & Malikov, Z. (2025). YUV Color Model-Based Adaptive Pansharpening with Lanczos Interpolation and Spectral Weights. *Mathematics*, 13(17), 1–23. <https://doi.org/10.3390/math13172868>
- Kemp, S. (2024). Digital 2024 Global Overview Report. In *DataReportal*. <https://datareportal.com/reports/digital-2024-global-overview-report>
- Liu, Z., Jiang, S., Feng, S., Song, Q., & Zhang, J. (2025). Comprehensive Review of Deep Learning Approaches for Single-Image Super-Resolution. *Sensors*, 25(18), 1–43. <https://doi.org/10.3390/s25185768>
- M., T. (2024). *What is Image Resizing? A Computer Vision Guide*. Roboflow. <https://blog.roboflow.com/image-resizing>
- Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. *International Journal of Scientific and Research Publications*, 13(8), 120–124. <https://doi.org/10.29322/ijsrp.13.08.2023.p14015>
- Parsania, M. P. S., & Virparia, D. P. V. (2016). A Comparative Analysis of Image Interpolation Algorithms. *Ijarccce*, 5(1), 29–34. <https://doi.org/10.17148/ijarccce.2016.5107>
- Pati, S., & Zaki, Y. (2025). Evaluating the Efficacy of Next.js: A Comparative Analysis with React.js on Performance, SEO, and Global Network Equity. In *WWW Companion 2025 -*

- Companion Proceedings of the ACM Web Conference 2025* (Vol. 1, Issue 1). Association for Computing Machinery. <https://doi.org/10.1145/3701716.3715565>
- Pitsun, O., & Shymchuk, M. (2025). A high-load architecture for image processing based on. *CIAW-2025: Computational Intelligence Application Workshop*, 10.
- Sharp. (2024). *High performance Node.js image processing*. Sharp. <https://sharp.pixelplumbing.com/performance/>
- Siregar, T. (2025). Research and Development (R&D) Method: Definition, Types, and Stages. *Research*, 1(2), 1–33.
- Zigisova, J., & Akrap, I. (2024). *Performance*. Almanac. <https://almanac.httparchive.org/en/2024/performance>