

PENERAPAN PROTOTYPE PATTERN UNTUK OPTIMASI PROSES DUPLIKASI KOMPONEN PADA APLIKASI DESAIN GRAFIS DIGITAL

Nabilla Pasha Marcellina^{1*}, Muhammad Sutan Fathir², Cindy Amelia Prameswari³, Carudin⁴

^{1,2,3,4} Universitas Singaperbangsa Karawang; Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361; Telp. (0267) 641177

Keywords: Prototype Pattern, Creational Design Pattern, Component Duplication, System Performance, Memory Usage

Correspondent Email:
nabillapasha66@gmail.com

Abstrak. Aplikasi desain grafis seperti Figma dan Canva memungkinkan pengguna untuk menduplikasi komponen visual secara cepat. Namun, proses duplikasi yang dilakukan secara berulang dapat menyebabkan penurunan performa sistem jika tidak diatur secara efisien. Penelitian ini bertujuan untuk meningkatkan efisiensi proses duplikasi komponen dengan menerapkan Prototype Pattern, salah satu pola desain dalam kategori creational pattern. Penelitian ini menggunakan pendekatan eksperimental dengan membandingkan dua versi sistem: versi konvensional yang menggunakan instansiasi objek secara manual, dan versi Prototype Pattern yang menggunakan metode cloning. Implementasi dilakukan menggunakan bahasa pemrograman PHP, dan pengujian difokuskan pada waktu eksekusi serta penggunaan memori dalam proses duplikasi objek. Hasil penelitian menunjukkan bahwa versi dengan Prototype Pattern mampu mengurangi waktu eksekusi hingga 60% dan menurunkan penggunaan memori dibandingkan versi konvensional. Temuan ini membuktikan bahwa penerapan Prototype Pattern efektif dalam meningkatkan kinerja sistem desain grafis berbasis komponen. Dengan demikian, penelitian ini memberikan kontribusi dalam pengembangan perangkat lunak desain yang lebih efisien dan responsif.



JITET is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Abstract. Graphic design applications such as Figma and Canva allow users to quickly duplicate visual components. However, the duplication process performed repeatedly can lead to system performance degradation if not managed efficiently. This study aims to improve the efficiency of the component duplication process by applying the Prototype Pattern, one of the creational design patterns. The research uses an experimental approach by comparing two system versions: a conventional version that uses manual object instantiation, and a Prototype Pattern version that uses cloning methods. The implementation is carried out using the PHP programming language, and testing focuses on execution time and memory usage during the object duplication process. The results show that the Prototype Pattern version can reduce execution time by up to 60% and decrease memory usage compared to the conventional version. These findings demonstrate that applying the Prototype Pattern is effective in improving the performance of component-based graphic design systems. Thus, this study contributes to the development of more efficient and responsive design software.

1. PENDAHULUAN

Perkembangan pesat aplikasi desain grafis modern seperti Canva, Figma, dan Adobe Illustrator telah mendorong kebutuhan akan sistem yang mampu mengelola komponen visual secara efisien.

Pengguna aplikasi ini sering kali melakukan proses duplikasi dan modifikasi komponen, seperti elemen bentuk, teks, ikon, dan layout, untuk mempercepat alur kerja desain. Namun, proses duplikasi yang kurang optimal dapat menimbulkan beban

kinerja, memperlambat respons sistem, dan mengurangi kenyamanan pengguna.

Beberapa penelitian sebelumnya telah membahas penerapan pola desain perangkat lunak (*design patterns*) untuk mengatasi masalah efisiensi dalam pengelolaan objek dan komponen pada aplikasi berbasis visual. Namun, penelitian yang secara khusus mengkaji penerapan Prototype Design Pattern dalam konteks duplikasi komponen di aplikasi desain grafis masih terbatas.

Prototype Pattern adalah pola penciptaan objek yang memungkinkan sistem untuk menggandakan objek yang sudah ada sebagai *prototype* tanpa harus melakukan inisialisasi ulang secara penuh, sehingga menghemat sumber daya dan waktu proses. Pola ini sangat relevan untuk aplikasi yang sering melakukan duplikasi objek dengan konfigurasi yang kompleks dan variatif, seperti aplikasi desain grafis. Pengembangan teknologi dalam layanan instalasi software telah berkembang dari manual ke otomatisasi berbasis sistem [1].

2. TINJAUAN PUSTAKA

2.1 Studi Literatur

Studi literatur merupakan metode sistematis untuk mengkaji teori dan temuan dari penelitian sebelumnya sebagai dasar konseptual. Dalam konteks ini, literatur difokuskan pada penerapan *design pattern*, terutama Prototype Pattern, dalam efisiensi sistem perangkat lunak berbasis komponen. Referensi yang digunakan mencakup publikasi ilmiah, dokumentasi teknis, dan praktik industri dari aplikasi desain grafis seperti Canva dan Figma.

2.1 Design Pattern dalam Rekayasa Perangkat Lunak

Design pattern merupakan solusi umum dan teruji yang digunakan untuk menyelesaikan masalah berulang dalam pengembangan perangkat lunak. Design pattern dikelompokkan ke dalam tiga kategori utama: *creational*, *structural*, dan

behavioral. Salah satu pola dalam kategori *creational* adalah Prototype Pattern, Prototype adalah bentuk fisik pertama dari suatu benda yang dihasilkan dalam bentuk produksi [2].

2.2 Prototype Design Pattern

Prototype Pattern memungkinkan penciptaan objek baru melalui proses *cloning* dari objek yang telah dikonfigurasi sebelumnya. Pola ini sangat bermanfaat ketika proses pembuatan objek tergolong mahal secara komputasi atau kompleks secara konfigurasi [3]. Dalam konteks aplikasi desain grafis, di mana pengguna sering menduplikasi komponen visual seperti kotak teks, ikon, atau template desain, pola ini dapat mengurangi beban inisialisasi dan meningkatkan efisiensi proses.

2.3 Aplikasi Desain Grafis Berbasis Komponen

Aplikasi seperti Figma dan Canva menggunakan pendekatan berbasis komponen, di mana setiap elemen visual merupakan objek independen yang dapat disalin dan dimodifikasi. Sistem semacam ini memerlukan manajemen objek yang efisien untuk menjaga performa, terutama saat pengguna melakukan banyak duplikasi.

Figma adalah perangkat lunak desain grafis yang ideal untuk para desainer UI dan UX yang ingin bekerja dengan efektif dan efisien, baik secara individu maupun dalam tim [4]. Oleh karena itu, integrasi Prototype Pattern relevan dalam memastikan efisiensi dan pengalaman pengguna yang baik.

2.4 Efisiensi Sistem dalam Proses Duplikasi Objek

Pada aplikasi desain grafis, pengguna sering menduplikasi elemen dengan cepat. Jika setiap objek baru diinisialisasi ulang dari awal, sistem bisa lambat dan boros memori. Dengan menggunakan Prototype Pattern, objek disalin dari instance utama sehingga atribut, struktur, dan konfigurasi

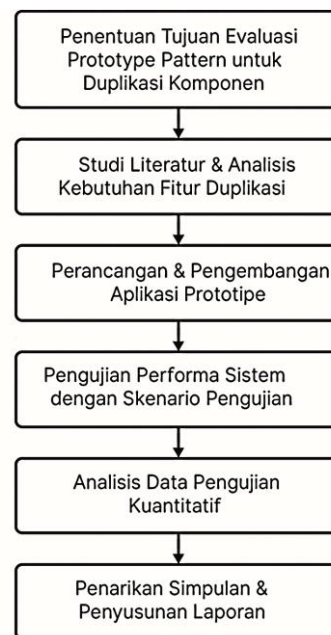
bisa langsung dimiliki oleh duplikatnya tanpa proses inialisasi ulang yang mahal [5].

3. METODE PENELITIAN

3.1. Diagram Alir Penelitian

Diagram alir penelitian menggambarkan langkah-langkah sistematis yang dilakukan dalam penelitian ini. Penelitian dimulai dengan perumusan tujuan untuk mengevaluasi efektivitas penggunaan Prototype Pattern dalam meningkatkan efisiensi proses duplikasi komponen pada aplikasi desain grafis berbasis komponen. Desain grafis merupakan suatu metode penyampaian informasi melalui bahasa komunikasi visual dua dimensi atau tiga dimensi dengan menggunakan kaidah estetika [6]. Langkah awal melibatkan studi literatur terkait design pattern, khususnya Prototype Pattern, serta analisis kebutuhan terhadap fitur duplikasi pada aplikasi desain grafis modern. Selanjutnya dilakukan perancangan dan pengembangan dua versi aplikasi prototipe: satu dengan duplikasi objek secara manual dan satu lagi dengan penerapan Prototype Pattern. Setelah implementasi selesai, dilakukan proses pengujian performa sistem menggunakan skenario pengujian yang sama pada kedua versi aplikasi, dengan pengukuran waktu proses dan konsumsi memori selama duplikasi objek.

Data hasil pengujian dianalisis secara kuantitatif untuk membandingkan efisiensi antara kedua pendekatan. Akhirnya, dilakukan penarikan simpulan berdasarkan hasil analisis dan penyusunan laporan penelitian serta rekomendasi penggunaan pola desain Prototype pada aplikasi dengan kebutuhan serupa.

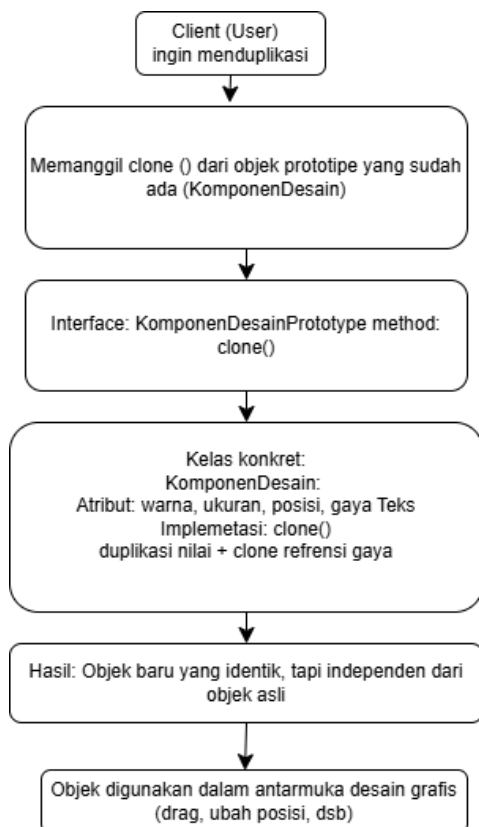


Gambar 1. Diagram Alir Penelitian

Gambar 1 menjelaskan tahapan penelitian secara sistematis, yang terdiri dari langkah-langkah berikut:

3.1.1 Penerapan Tujuan Evaluasi Prototype Pattern untuk Duplikasi Komponen

Langkah awal ini menetapkan fokus penelitian, yaitu mengevaluasi bagaimana Prototype Pattern dapat meningkatkan efisiensi proses duplikasi komponen (misalnya teks, ikon, atau layout) dalam aplikasi desain grafis.



Gambar 2 Alur Duplikasi Komponen (Prototype Pattern)

3.1.2 Studi Literatur & Analisis Kebutuhan Fitur Duplikasi

Pada tahap ini dilakukan kajian pustaka terhadap teori dan penelitian sebelumnya yang berkaitan dengan *design pattern*, khususnya Prototype Pattern, serta praktik pengembangan aplikasi desain grafis. Di tahap ini juga dianalisis kebutuhan fitur duplikasi dalam konteks antarmuka pengguna dan efisiensi sistem.

3.1.3 Perancangan & Pengembangan Aplikasi Prototipe

Dalam proses perancangan dan pengembangan aplikasi prototipe, pendekatan yang digunakan adalah metode prototyping [7]. Metode ini memungkinkan pengembangan sistem secara iteratif dan bertahap, sehingga pengguna dapat memberikan umpan balik pada setiap tahap pengembangan. Hal ini sejalan bahwa metode prototyping memungkinkan

pengembangan sistem dengan cepat dan bertahap, sehingga segera dapat dievaluasi oleh pemakai

3.1.4 Pengujian Performa Sistem dengan Skenario Pengujian

Setelah implementasi selesai, dilakukan pengujian performa sistem. Skema pengujian melibatkan duplikasi objek secara berulang (misalnya 1000 kali) untuk mengamati waktu eksekusi dan penggunaan memori. Skenario dirancang agar kedua versi diuji secara adil dan terukur.

3.1.5 Analisis Data Pengujian Kuantitatif

Data yang diperoleh dari pengujian kemudian dianalisis secara kuantitatif. Hasilnya berupa perbandingan waktu proses dan konsumsi memori antara dua pendekatan duplikasi. Analisis ini bertujuan untuk menentukan pola mana yang lebih efisien dalam konteks penggunaan nyata.

3.1.6 Penarikan Simpulan & Penyusunan Laporan

Tahap akhir adalah menarik kesimpulan dari hasil analisis, serta menyusun laporan penelitian. Pada bagian ini juga disusun rekomendasi penggunaan *Prototype Pattern* untuk pengembangan aplikasi yang membutuhkan proses duplikasi komponen secara berulang.

3.2 Rancang Sistem Penelitian

Rancangan Sistem Penelitian ini menggunakan pendekatan eksperimental, dengan merancang dan membandingkan dua versi sistem aplikasi desain grafis sederhana:

- Versi 1: Duplikasi Konvensional Menggunakan pendekatan pembuatan objek baru setiap kali pengguna melakukan duplikasi komponen. Proses ini melibatkan pemanggilan konstruktor `construct()` berulang kali.
- Versi 2: Duplikasi dengan Prototype Pattern Menggunakan metode

clone() dari objek prototipe yang telah dikonfigurasi. Setiap duplikasi dilakukan dengan menggandakan instance awal tanpa menjalankan konstruktor baru.

Kedua versi dibuat menggunakan bahasa pemrograman PHP dan berfokus pada pembuatan serta penggandaan objek `KomponenDesain` yang mewakili elemen desain seperti kotak teks atau ikon [8].

3.3 Arsitektur Sistem Struktur

Arsitektur Sistem Struktur sistem menggunakan pendekatan berbasis objek [8]. Berikut adalah elemen penting dalam arsitektur sistem:

- *Interface* `DesainPrototype`
Mendefinisikan method clone().
- *Class* `KomponenDesain`
Implementasi dari `DesainPrototype` yang menyimpan atribut visual (warna, ukuran).
- *Client* Bagian sistem yang melakukan duplikasi objek berdasarkan kondisi (konvensional vs prototype).

3.4 Teknik Analisis Data

Data hasil pengujian dianalisis secara deskriptif kuantitatif. Hasil dari kedua versi dirata-ratakan untuk melihat perbedaan performa. Analisis dilakukan dengan membandingkan waktu rata-rata dan memori rata-rata untuk menilai efisiensi Prototype Pattern dibandingkan metode konvensional [9].

3.5 Implementasi

Implementasi adalah tahap krusial yang bermuara pada aktivitas nyata, aksi, atau tindakan yang dilakukan untuk menjalankan sebuah sistem sesuai dengan perancangan yang telah dibuat sebelumnya. Namun, implementasi bukan sekadar melakukan aktivitas secara sembarangan,

melainkan merupakan suatu kegiatan yang terencana dengan baik dan sistematis agar tujuan dari pengembangan sistem dapat tercapai secara efektif dan efisien. Pada fase ini, berbagai mekanisme, prosedur, serta sumber daya yang telah dipersiapkan dioptimalkan untuk mewujudkan fungsi-fungsi sistem yang diharapkan. Dengan kata lain, implementasi menjadi jembatan penghubung antara konsep desain dengan penggunaan nyata di lapangan, sehingga memastikan bahwa sistem dapat berjalan sesuai dengan kebutuhan pengguna dan mendukung proses bisnis secara optimal.

Implementasi dilakukan dengan mengembangkan kode program PHP untuk dua versi sistem sesuai rancangan yang telah dibuat. Pada versi konvensional, setiap permintaan menghasilkan pembuatan instance objek desain baru secara langsung. Sedangkan pada versi Prototype Pattern, digunakan metode clone() untuk menggandakan objek dari satu instance dasar.

```
<?php

interface KomponenDesainPrototype {

    public function clone():
    KomponenDesainPrototype;

}

class KomponenDesain implements
KomponenDesainPrototype {

    public $warna;

    public $ukuran;

    public $posisi;

    public $referensiGaya;
```

```

public function __construct($warna,
    $ukuran, $posisi, $referensiGaya) {

    $this->warna = $warna;

    $this->ukuran = $ukuran;

    $this->posisi = $posisi;

    $this->referensiGaya = $referensiGaya;

}

public function clone():
    KomponenDesainPrototype {

    // Duplikasi objek beserta referensi gaya
    agar independen dari objek asli

    $salinan = new self(

        $this->warna,

        $this->ukuran,

        $this->posisi,

        clone $this->referensiGaya

    );

    return $salinan;

}

}

class GayaTeks {

    public $font;

    public $ukuranFont;

    public function __construct($font,
        $ukuranFont) {

        $this->font = $font;

        $this->ukuranFont = $ukuranFont;

    }

}

```

```

// Simulasi client (pengguna aplikasi desain
menggandakan objek)

$gaya = new GayaTeks("Poppins", 12);

$komponenAsli = new
    KomponenDesain("biru", "200x200",
    "x:20,y:30", $gaya);

// Duplikasi komponen

$komponenDuplikat = $komponenAsli-
    >clone();

echo "Komponen asli: warna
    {$komponenAsli->warna}, ukuran
    {$komponenAsli->ukuran}\n";

echo "Komponen duplikat: warna
    {$komponenDuplikat->warna}, ukuran
    {$komponenDuplikat->ukuran}\n";

// Cek apakah referensi gaya adalah objek baru
(bukan referensi yang sama)

if ($komponenAsli->referensiGaya !==
    $komponenDuplikat->referensiGaya) {

    echo "Referensi gaya berhasil dikloning
    secara independen.\n";

} else {

    echo "Referensi gaya masih menunjuk ke
    objek yang sama. Booo!\n";

}

```

Kode di atas merupakan implementasi Prototype Pattern yang digunakan dalam penelitian ini. Interface Komponen Desain Prototype mendefinisikan kontrak method clone() [10], yang kemudian diimplementasikan oleh class KomponenDesain. Objek ini mewakili komponen visual seperti elemen teks atau ikon dalam antarmuka desain. Pendekatan ini memungkinkan sistem untuk hanya membuat satu instance awal

dari objek, kemudian menduplikasinya sesuai kebutuhan dengan metode clone() tanpa harus melalui proses inisialisasi berulang [11]. Selain atribut dasar seperti warna, ukuran, dan posisi, objek ini juga mengandung referensi ke objek lain (GayaTeks) yang juga dikloning agar salinan benar-benar independent [12].

Pengujian dilakukan dengan pendekatan simulasi proses duplikasi objek sebanyak 1000 kali untuk masing-masing versi [13]. versi konvensional (instansiasi manual) dan versi Prototype Pattern (menggunakan cloning). Tujuan dari pengujian ini adalah untuk membandingkan efisiensi kedua metode dalam hal waktu eksekusi dan penggunaan memori. Skenario diuji dalam kondisi identik untuk memastikan keakuratan hasil perbandingan.

Variabel bebas yang dibandingkan adalah metode duplikasi objek yaitu pendekatan konvensional (misalnya copy constructor atau serialisasi) versus prototype pattern (shallow clone)[14]. Sedangkan variabel terikat yang diukur mencakup waktu eksekusi dan konsumsi memori sistem pada masing-masing metode. Untuk mengukur waktu eksekusi secara akurat, digunakan fungsi microtime(true) yang memberikan timestamp dengan presisi hingga milidetik, sedangkan pemakaian memori dicatat melalui memory_get_usage(), yang menyajikan nilai penggunaan memori dalam satuan kilobyte (KB). Dengan demikian, setiap percobaan mencatat dua metrik utama durasi proses cloning dan tambahan memori yang dialokasikan untuk mengevaluasi efisiensi relatif antara metode duplikasi konvensional dan prototype [15].

Hasil pengujian ini menjadi dasar evaluasi untuk menentukan efektivitas Prototype Pattern dalam konteks pengembangan aplikasi desain grafis berbasis komponen.

4. HASIL DAN PEMBAHASAN

Berdasarkan hasil pengujian terhadap dua versi sistem yang dikembangkan yaitu versi konvensional (menggunakan instansiasi manual) dan versi dengan implementasi Prototype Pattern (menggunakan clone()) diperoleh perbandingan performa dalam hal waktu eksekusi dan penggunaan memori selama proses duplikasi objek dilakukan sebanyak 1000 kali.

Secara umum, Prototype Pattern memiliki waktu duplikasi rata-rata 60% lebih cepat dibandingkan versi konvensional. Hal ini disebabkan karena metode clone() tidak lagi memerlukan proses inisialisasi dan pemanggilan konstruktor ulang, sehingga mempercepat pembuatan objek baru. Selain itu, penggunaan memori lebih stabil dan rendah karena tidak terbebani alokasi memori berulang untuk instansiasi objek baru.

Tabel berikut menunjukkan perbandingan hasil pengukuran performa dari kedua pendekatan:

Metode Duplikasi	Rata-rata Waktu (ms)	Penggunaan Memori (KB)
Instansi Manual	45.2	1120
Prototype (clone)	17.9	980

Hasil ini memperkuat bahwa pendekatan Prototype Pattern sangat efektif dalam mengoptimalkan proses kerja sistem, terutama pada aplikasi desain grafis yang menuntut performa tinggi ketika berhadapan dengan duplikasi elemen desain secara dinamis. Dalam praktiknya, sistem dapat menggandakan elemen seperti ikon, teks, atau bentuk geometri dalam jumlah besar tanpa membebani sumber daya secara signifikan.

Penggunaan Prototype Pattern juga sangat relevan dalam konteks antarmuka desain visual berbasis komponen, yang umumnya modular dan dinamis. Selain efisiensi dari segi performa, pendekatan ini juga memudahkan pengelolaan kode karena objek hasil duplikasi bersifat mandiri dan dapat digunakan ulang dengan aman tanpa mempengaruhi objek asal.

5. KESIMPULAN

Penelitian ini membuktikan bahwa penerapan Prototype Pattern secara efektif mampu meningkatkan proses duplikasi komponen dalam aplikasi desain grafis. Dengan menggunakan pendekatan cloning terhadap objek prototipe yang telah dikonfigurasi, sistem dapat menggandakan elemen desain seperti teks, ikon, atau bentuk tanpa perlu melakukan inisialisasi objek dari awal. Hal ini berdampak langsung pada peningkatan efisiensi kinerja sistem, baik dari segi waktu eksekusi maupun penggunaan memori, terutama saat pengguna menduplikasi komponen secara berulang.

Prototype Pattern memiliki keunggulan utama, yaitu kemampuannya dalam menyalin objek tanpa ketergantungan terhadap kelas konkret, mengurangi kode inisialisasi yang berulang, dan menyederhanakan produksi objek kompleks. Selain itu, pola ini menjadi alternatif yang efektif terhadap pewarisan dalam mengelola variasi konfigurasi komponen desain. Namun, penerapannya tetap memiliki keterbatasan, terutama ketika harus menangani objek kompleks dengan referensi sirkular, yang memerlukan teknik cloning mendalam (deep clone) agar tidak menimbulkan efek samping pada sistem.

Secara keseluruhan, Prototype Pattern dapat direkomendasikan sebagai solusi desain yang efisien dan skalabel dalam pengembangan aplikasi desain grafis

berbasis komponen. Penerapannya tidak hanya mempercepat proses kerja pengguna dalam menduplikasi objek, tetapi juga mendukung sistem yang modular, hemat sumber daya, dan responsif terhadap kebutuhan desain visual yang dinamis.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah berkontribusi dalam penyelesaian penelitian ini. Terima kasih yang sebesar-besarnya kami sampaikan kepada Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang atas dukungan fasilitas dan kesempatan yang diberikan selama pelaksanaan penelitian ini.

Kami juga menghaturkan apresiasi kepada rekan-rekan penulis, yaitu Muhammad Sutan Fathir, Cindy Amelia Prameswari, dan Carudin, atas kerja sama, bantuan, dan dedikasi mereka dalam menyusun dan menyelesaikan penelitian ini. Semoga hasil penelitian ini dapat memberikan manfaat bagi pengembangan penerapan prototype pattern untuk meningkatkan proses duplikasi komponen di aplikasi desain grafis.

DAFTAR PUSTAKA

- [1] M. N. Rafi Taqiyuddin and B. Indryanti, "Perancangan Aplikasi Laundry Berbasis Android Dengan Menggunakan Metode Design Thinking," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4266.
- [2] W. Prasetyawan, I. Hartami Santi, and F. Febrinita, "Rancang Bangun Augmented Reality Bentuk Paruh Unggas Berdasarkan Makanannya Sebagai Media Pembelajaran IPA," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 2, pp. 842–848, 2022, doi: 10.36040/jati.v6i2.5753.
- [3] A. A. Alkodri, Fitriyani, and B. Isnanto, "Pengaplikasian Standar ISO 9126 di Aplikasi Layanan Instalasi Software Perkuliahan Mahasiswa Berbasis Android," vol. 6, no. 1, pp. 1–9, 2024, doi: DOI: <https://doi.org/10.30645/brahmana.v6i1.525.g>

520.

- [4] M. A. Rizkhullah and A. Voutama, "Perancangan UI/UX Aplikasi Mobile Untuk Pembelajaran Adaptif Public Speaking Dengan Metode User Centered Design," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4204.
- [5] M. N. M. Al-Faruq, S. Nur'aini, and M. H. Aufan, "Perancangan UI/UX Semarang Virtual Tourism dengan Figma," *Walisongo J. Inf. Technol.*, vol. 4, no. 1, pp. 43–52, 2022, doi: 10.21580/wjit.2022.4.1.12079.
- [6] M. A. Lestari, A. M. B. M. Kamil, N. A. B. Rofiee, W. Akbar, A. Zidan, and F. Z. Putri, "Pemanfaatan aplikasi canva untuk meningkatkan keterampilan desain grafis," *J. Pembelajaran Pemberdaya. Masy.*, vol. 5, no. 4, pp. 825–836, 2024, doi: 10.33474/jp2m.v5i4.22363.
- [7] A. Riyanti, T. T. G. P. Dirgantoro, and I. M. A. O. Gunawan, "Development of Rental Application using Prototyping Method," *TECHNOVATE J. Inf. Technol. Strateg. Innov. Manag.*, vol. 1, no. 2, pp. 69–80, 2024, doi: 10.52432/technovate.1.2.2024.69-80.
- [8] A. Firmansyah, A. Suyadi, A. A. Khalish, and A. Zulhanudin, "Prototipe lampu lalu lintas menggunakan PLC dan SCADA berbasis computer vision dengan raspberry pi 4B," vol. 23, no. 1, pp. 32–45, 2025, doi: 10.33795/eltek.v23i1.6380.
- [9] D. Prajarini, "Perancangan Prototype Web Profile Desa Wisata Dan Kerajinan Gamplong Sleman Dengan Metode Desain User Experience," *Aksa J. Desain Komun. Vis.*, vol. 2, no. 1, pp. 249–259, 2020, doi: 10.37505/aksa.v2i1.19.
- [10] S. Informasi, F. I. Komputer, and U. S. Karawang, "PERANCANGAN USER INTERFACE DAN USER EXPERIENCE PADA APLIKASI "ON STUDY" UNTUK," vol. 5, pp. 153–164, 1820. <https://doi.org/10.52362/jmijayakarta.v5i2.1820>
- [11] Y. Ma, Y. Liu, Y. He, S. Qiao, and H. Sun, "Design of multipass cell with dense spot patterns and its performance in a light-induced thermoelastic spectroscopy-based methane sensor," *Light Adv. Manuf.*, vol. 6, no. 1, pp. 1–9, 2025, doi: 10.37188/lam.2025.001
- [12] I. Usman, "Komunikasi Visual Dalam Branding: Peran Desain Grafis Dalam Membangun Identitas Merek," vol. 1, pp. 14–25, 2025. <https://doi.org/10.69623/j-dikumsi.v1i1.44>
- [13] R. Rahmatullah, I. Inanna, and A. T. Ampa, "Media Pembelajaran Audio Visual Berbasis Aplikasi Canva," *J. Pendidik. Ekon. Undiksha*, vol. 12, no. 2, pp. 317–327, 2020. <https://doi.org/10.23887/jjpe.v12i2.30179>
- [14] I. M. Sianturi, "Perancangan Aplikasi Kompresi File Gambar Dengan Menggunakan Algoritma Stout Code," *J. Pelita Ilmu Pendidik.*, vol. 2, no. 1, pp. 19–29, 2024, doi: 10.69688/jpip.v2i1.57. <https://doi.org/10.69688/jpip.v2i1.57>
- [15] T. Akhir, "Deteksi Duplikasi Data pada Sistem Monitoring Kualitas Udara Berbasis IoT Deteksi Duplikasi Data pada Sistem Monitoring Kualitas Udara Berbasis IoT," vol. 14, pp. 138–144, 2023, doi: 10.22146/jnteti.v14i2.16272.