

Contents list available at www.jurnal.unimed.ac.id

CESS
(Journal of Computing Engineering, System and Science)

journal homepage: <https://jurnal.unimed.ac.id/2012/index.php/cess>



**Analisis Kelebihan dan Kekurangan Pemrograman Berorientasi Objek pada
Generator Form Web**

***Analysis of the Advantages and Disadvantages of Object-Oriented
Programming in Generators Form Web***

Katon Wijana¹, Gabriel Indra Widi Tamtama^{2*}

^{1,2} Sistem Informasi, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin 5-25 Yogyakarta, Indonesia

email: ¹katony@staff.ukdw.ac.id, ²gabriel@staff.ukdw.ac.id

ABSTRAK

Form merupakan salah satu bagian penting dari pengembangan aplikasi berbasis web dengan tujuan mengumpulkan data pengguna seperti nama, alamat email, nomor telepon, dan data lainnya tergantung dari aplikasi yang dikembangkan. Membuat *form* dapat memakan waktu lama dan membosankan karena harus dibuat sesuai dengan kebutuhan yang diperlukan dan semenarik mungkin sehingga memberikan pengalaman tambahan bagi pengguna akhir. Penelitian ini bertujuan untuk menganalisis kelebihan dan kekurangan dari antarmuka grafis yang dibuat secara manual dibandingkan dengan *generator form* berbasis *web* yang dirancang menggunakan pendekatan pemrograman berorientasi objek (OOP). Metode penelitian ini mencakup analisis terhadap implementasi generator form berorientasi objek yang tersedia, serta pengujian terhadap efisiensi kode program. Data yang dikumpulkan dari penelitian ini digunakan untuk mengidentifikasi kelebihan, seperti fleksibilitas, efisiensi dan reusabilitas kode, serta kekurangan, seperti kompleksitas dan overhead yang mungkin terjadi. Analisis dilakukan dengan mengidentifikasi kelebihan dan kekurangan OOP secara umum ataupun spesifik pada *generator form web* dan dibandingkan dengan pembuatan *form* secara manual. Hasil penelitian ini menunjukkan bahwa pendekatan pemrograman berorientasi objek pada generator form web secara efektif meningkatkan efisiensi dan kecepatan pengembangan aplikasi web. Namun, tantangan yang dihadapi termasuk dalam manajemen kompleksitas dan overhead yang terkadang diperlukan untuk mengelola abstraksi objek yang lebih tinggi.

Kata Kunci: *form; aplikasi berbasis web; generator form web; pemrograman berorientasi objek.*

ABSTRACT

Forms are an important part of web-based application development to collect user data such as name, email address, telephone number, and other data depending on the developed application. Creating a form can be time-consuming and tedious because it must be made

*Penulis Korespondensi:

email: gabriel@staff.ukdw.ac.id

according to the required requirements and as attractive as possible to provide an additional experience for the end user. This research analyzes the advantages and disadvantages of manually created graphical interfaces compared to web-based form generators designed using an object-oriented programming (OOP) approach. This research method includes analysis of the available object-oriented form generator implementations and testing the program code's efficiency. Data collected from this research is used to identify advantages, such as flexibility, efficiency, and reusability of the code, as well as disadvantages, such as complexity and possible overhead. The analysis was carried out by manually identifying the advantages and disadvantages of OOP in general or specifically in web form generators and comparing it with creating forms. This research shows that the object-oriented programming approach in web form generators effectively increases the efficiency and speed of web application development. However, challenges include managing the complexity and overhead that is sometimes required to manage higher object abstractions.

Keywords: *forms; web-based applications; web form generators; object-oriented programming*

1. PENDAHULUAN

Membuat dan mengembangkan aplikasi berbasis web membutuhkan perhatian khusus dan mempertimbangkan banyak hal seperti keamanan, skalabilitas, kemudahan penggunaan, dan efisiensi waktu dalam pengembangan [1]. Salah satu komponen yang krusial dalam pengembangan web ada adalah fitur *form*, yang berfungsi sebagai jembatan antara pemilik sistem dan *end user*. Form memungkinkan pengumpulan data, pemrosesan input dari pengguna ke dalam sistem, dan interaksi pengguna dengan sistem. Oleh karena itu, desain *form* yang efisien, menarik, dan *user-friendly* menjadi tantangan tersendiri bagi seorang pengembang web [2][3].

Pemrograman berorientasi obyek (OOP) telah lama menjadi paradigma dominan dalam membuat atau mengembangkan perangkat lunak, termasuk aplikasi berbasis web. OOP menawarkan pendekatan yang sistematis dalam penggunaannya dengan memanfaatkan konsep seperti kelas, objek, pewarisan, dan polimorfisme [4][5]. Sedangkan *generator form web* sendiri merupakan sebuah metode yang digunakan untuk membuat dan mengelola form secara otomatis tanpa harus menuliskan kode dari awal. Penerapan OOP pada *generator form* memiliki peranan yang cukup penting dalam membuat *form* yang dinamis, modular, dan mudah dikelola.

Penggunaan OOP dalam *generator form web* memungkinkan pengembang dalam membuat sistem yang lebih terstruktur dan fleksibel. Setiap elemen form seperti teks, *checkbox*, atau *dropdown* dapat diwakili sebagai objek dengan atribut dan metode yang spesifik. Objek-objek tersebut dapat dikombinasi dan dikonfigurasi ulang sesuai dengan kebutuhan sehingga memungkinkan membuat *form* yang kompleks menjadi lebih efisien dan mudah [5]. Penerapan OOP sudah banyak dilakukan oleh para pengembang atau pembuat aplikasi berbasis web, seperti halnya yang dilakukan oleh [6] yang membuat aplikasi berbasis web untuk membantu mengelola pengaduan nasabah pada suatu bank perkreditan rakyat. Demikian juga dengan yang dilakukan oleh [7] yang mengimplimentasikan konsep OOP untuk membuat aplikasi pembukuan keuangan penjualan jus buah. Selama ini topik penelitian sebelumnya lebih umum membahas pemrograman berorientasi objek (OOP) atau pengembangan aplikasi web secara umum, sedangkan penelitian ini fokus yang lebih

mendalam pada generator form web, baik dari segi kelebihan maupun kekurangannya. Selain itu penelitian sebelumnya lebih berfokus pada teori atau konsep, pada penelitian ini penulis menekankan penelitian pada efektivitas dan efisiensi dari implementasi OOP dalam konteks pengembangan generator form web yang meliputi studi kasus, pengujian kinerja, atau evaluasi praktis terhadap kegunaan dan kepraktisan implementasi OOP dalam proses pengembangan nyata.

Namun, seperti halnya pendekatan pemrograman lain, konsep OOP juga memiliki kelebihan dan kekurangan. Kelebihan OOP meliputi kemampuan untuk menciptakan struktur kode yang lebih terorganisir dan modular, yang dapat menunjang proses *debugging* dan pengembangan yang berkelanjutan. Di sisi lain, kekurangan OOP dapat mencakup kompleksitas tambahan dalam desain awal (*blue print*) pembuatan dan analisis kebutuhan sistem [8].

Penelitian ini mencoba untuk menganalisis dan menggali lebih dalam keuntungan dan kekurangan dalam penerapan OOP untuk mengembangkan aplikasis berbasis web untuk kebutuhan mendata identitas jemaat di salah satu Gereja Kristen di Yogyakarta. Kami mengajukan pertanyaan penelitian yaitu apa saja kelebihan dan kekurangan penerapan OOP dalam pembuatan generator form web dibandingkan pembuatan form secara manual? Adapun struktur artikel tersusun dari bagian pendahuluan, dasar/tinjauan teori, metode penelitian, hasil dan pembahasan, dan kesimpulan.

2. DASAR/TINJAUAN TEORI

2.1. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (*Object Oriented Programming/OOP*) merupakan metode pemrograman yang menggunakan "objek" sebagai elemen dasar penyusunan program. Objek ini berisi data (atribut) dan perilaku (metode) yang merepresentasikan entitas nyata di dunia [4][9]. Tujuan dari diciptakan model OOP untuk mempermudah pekerjaan dalam membuat ataupun mengembangkan aplikasi baik berbasis *desktop* maupun *web*. Selain itu juga memberikan manfaat antara lain kebebasan pengembangan, meningkatkan kualitas, mempermudah pemeliharaan, dan meningkatkan reusabilitas.

Komponen-komponen yang ada di dalam pemrograman berorientasi objek yaitu [4][5][10]:

- a. Kelas
Kelas merupakan tempat untuk membuat objek. Dalam kelas ini pula atribut dan metode yang akan dimiliki objek didefinisikan.
- b. Objek
Merupakan *instance* dari kelas yang memiliki atribut dan metode. Objek ini merepresentasikan dunia nyata ke dalam bentuk program.
- c. Abstraksi
Merupakan proses penyederhanaan kompleksitas dengan hanya menampilkan fitur yang penting dari suatu objek. Hal ini memberikan kemungkinan untuk fokus pada kinerja dari objek tersebut.
- d. Enkapsulasi
Merupakan mekanisme menyembunyikan detail implementasi internal dari suatu objek, dan hanya menyediakan antarmuka yang diperlukan untuk berinteraksi dengan objek tersebut. Terdapat tiga *modifier access* yang memberikan akses untuk berinteraksi dengan objek yaitu "private", "protected", dan "public".

e. Pewarisan

Merupakan mekanisme untuk mewariskan satu kelas ke kelas lainnya. Kelas yang mewarisi disebut kelas turunan dan kelas yang diwarisi disebut kelas induk. Tujuan dari pewarisan ini adalah kode dapat digunakan kembali dan memudahkan pemeliharaan.

f. Polimorfisme

Merupakan konsep dalam OOP yang memungkinkan objek dari kelas yang berbeda dapat diakses melalui antarmuka yang sama. Ini memungkinkan kode yang lebih fleksibel dan mudah dimodifikasi sesuai dengan kebutuhan.

2.2. Generator Form Web

Generator form web merupakan perangkat atau metode untuk membuat formulir dalam pembangunan atau pengembangan aplikasi berbasis web dengan cara yang mudah dan efisien. Dengan metode ini, pengembang maupun orang yang masih awam dalam dunia pengembangan ataupun pembuatan web dapat merancang dan membuat formulir tanpa harus menuliskan kode dari awal [11]. Kegunaan dari formulir dalam suatu aplikasi berbasis web yaitu untuk mengumpulkan data pengguna, pendaftaran, survey, dan lain sebagainya.

Generator form web memiliki beberapa fungsi dan fitur utama yaitu [2][12][13]:

a. Antarmuka pengguna yang mudah digunakan.

Fitur ini memungkinkan pengguna untuk menambahkan elemen formulir seperti teks, tombol, *checkbox*, dan menu *dropdown* dengan mudah.

b. Kustomisasi.

Fitur ini memungkinkan pengguna untuk menyesuaikan tampilan dan perilaku dari formulir yang akan dibuatnya. Penyesuaian tampilan dapat berupa *label*, *placeholder*, validasi input, dan *style* (CSS).

c. Validasi data

Fungsi ini memungkinkan pengembang untuk mengatur dan memastikan bahwa data yang akan dimasukkan ke dalam formulir sudah sesuai dengan kriteria tertentu.

d. Integrasi

Beberapa model generator sudah mendukung integrasi dengan layanan lain seperti *email*, basis data, CRM (*Customer Relationship Management*), dan fungsi analitik.

e. *Reporting and Analysis*

Fungsi ini berguna untuk memberikan pelaporan dan analisis kepada pengembang ataupun pembuat formulir terkait data yang diterima. Beberapa contoh yang dapat dianalisis adalah jumlah pengisi formulir, hasil survey, dan lain sebagainya.

f. Keamanan

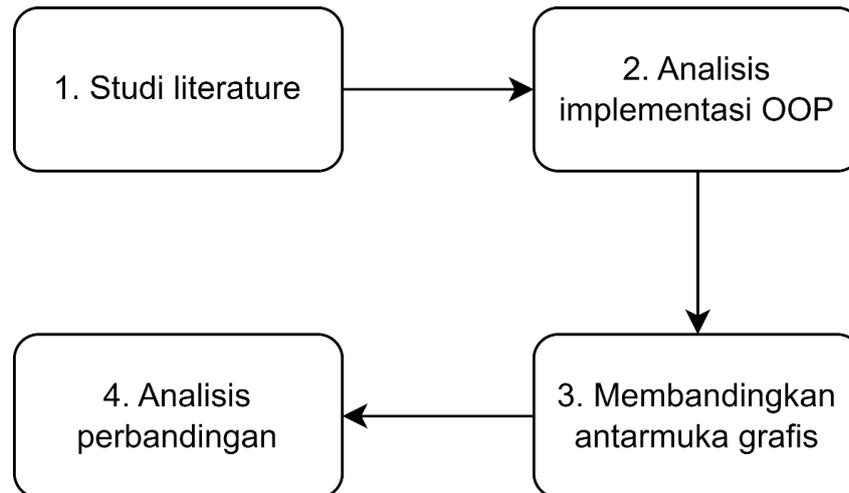
Fungsi ini sangat penting untuk menjaga integritas dan keamanan data yang dikumpulkan. Contoh fitur keamanan yang sering dipakai adalah CAPTCHA, anti spam, dan enkripsi data.

g. Respon otomatis

Beberapa generator memungkinkan pengaturan respon otomatis seperti mengirimkan email konfirmasi kepada pengguna yang sudah mengisi formulir.

3. METODE

Metode yang digunakan dalam penelitian ini terdiri dari empat tahap seperti yang ditampilkan dalam gambar berikut:



Gambar 1. Metode penelitian

Tahap 1: Studi literatur

Tahap pertama ini dilakukan untuk mengkaji teori, temuan, dan bahan penelitian lain yang relevan. Studi literatur dilakukan dengan mengumpulkan dan menganalisis berbagai sumber akademik seperti buku, artikel jurnal, konferensi, dan publikasi lainnya yang berkaitan dengan OOP, pengembangan aplikasi berbasis *web*, dan *generator form web*. Tujuan dari tahap ini adalah untuk memperoleh pemahaman yang mendalam mengenai konsep dasar dan aplikasi praktis dari OOP dalam konteks pengembangan *form web*.

Tahap 2: Analisis implementasi OOP pada *generator form web*

Pada tahap kedua, penelitian difokuskan pada analisis implementasi OOP dalam *generator form web*. Data dikumpulkan melalui studi kasus yang melibatkan *generator form web* yang ada. Aspek-aspek yang dianalisis meliputi struktur kelas dan objek, prinsip-prinsip OOP seperti enkapsulasi, pewarisan, dan polimorfisme, serta bagaimana prinsip-prinsip tersebut diterapkan dalam pengembangan *form web*. Analisis ini bertujuan untuk mengidentifikasi kekuatan dan kelemahan penggunaan OOP dalam konteks ini.

Tahap 3: Membandingkan antarmuka grafis manual vs *generator form*

Tahap ketiga melibatkan perbandingan antara antarmuka grafis yang dibuat secara manual dan yang dihasilkan oleh *generator form*. Studi kasus menggunakan aplikasi berbasis *web* “Sistem Informasi Jemaat Gereja” yang telah dibuat pada penelitian sebelumnya [14]. Data dikumpulkan dengan mengidentifikasi perbedaan dalam hal efisiensi pengembangan, kemudahan penggunaan, fleksibilitas, dan kinerja. Perbandingan ini akan memberikan wawasan tentang kelebihan dan kekurangan pendekatan manual dan otomatis dalam pembuatan *form web*.

Tahap 4: Analisis perbandingan

Tahap terakhir adalah menganalisis data perbandingan secara kualitatif untuk mengevaluasi kelebihan dan kekurangan OOP pada *generator form web*. Analisis ini akan melihat bagaimana aspek-aspek OOP seperti modularitas, pemeliharaan kode, dan skalabilitas dapat dimanfaatkan untuk meningkatkan pengembangan *generator form web*. Temuan dari tahap ini diharapkan dapat memberikan rekomendasi praktis untuk pengembang dalam mengoptimalkan penggunaan OOP dalam proyek pengembangan *web* mereka.

4. HASIL DAN PEMBAHASAN

4.1. Analisis Implementasi OOP pada *Generator Form Web*

Pada bagian ini, kami melakukan analisis terhadap implementasi OOP pada *generator form web* yang telah dikembangkan sebagai berikut.

Data Jemaat			
Id Anggota	00049008886	Rw	1
Nama Lengkap	Katon Wijana	Kelurahan	Kricak
Panggilan	Katon	Kecamatan	Tegalrejo
Nama Baptis	Paulus	Kota	Yogyakarta
Pasangan	0 - Belum terdaftar - Belum ada	Kode Pos	55242
Kks	0 - Belum terdaftar - Belum ada	Propinsi	Daerah Istimewa Yogyakarta
Tgl Gabung	12/25/2000	Negara	Indonesia
Jabatan	Announcer	Telpon	557919
No Ktp	3471012707680001	Hp	0811255488
Tempat Lahir	Klaten	Email	katony@staff.ukdw.ac.id
Tanggal Lahir	07/27/1968	Sosmed	facebook.com/katonwijana
Gender	<input checked="" type="radio"/> Pria <input type="radio"/> Wanita	Pekerjaan	Dosen
Gol Darah	A	Pendidikan	S2
Status	Menikah	Keterampilan	Mengajar
Gereja	GBIKA JOGJAKARTA	Skema Pr 1	<input checked="" type="checkbox"/> Komsei <input checked="" type="checkbox"/> Baptis <input checked="" type="checkbox"/> Retreat Encounter <input checked="" type="checkbox"/> ESBC
Alamat	Jl. Jambon Yogyakarta	Skema Pr 2	<input checked="" type="checkbox"/> Som 1 <input checked="" type="checkbox"/> Som 2 <input checked="" type="checkbox"/> Som 3 <input checked="" type="checkbox"/> Som 4 <input checked="" type="checkbox"/> Som 5 <input checked="" type="checkbox"/> Som 6
Rt	2	Status Kta	Dewasa
Submit		Reset	

Gambar 2. Hasil tampilan form yang dihasilkan generator

Hasil tampilan tersebut dibuat memakai blueprint atau kelas objek yang sudah dibuat sebelumnya, secara ringkas program PHP untuk membuat form HTML dengan OOP adalah sebagai berikut:

- Kelas Form menyediakan metode untuk menambahkan elemen-elemen form seperti text box, date, password, hidden field, list box, checkbox, radio button, dan textarea.
- Konstruktors dari kelas ini akan menentukan nilai atribut aksi form (action) dan judul.
- Kelas ini menyediakan metode-metode untuk menambah Elemen: `addTextBox`, `addDate`, `addPassword`, `addHidden`, `addListBox`, `addCheckBox`, `addRadio`, dan `addTextArea` untuk menambahkan elemen-elemen form ke dalam form.
- Metode yang hanya boleh diakses dari dalam kelas dibuat private yaitu `textBox`, `date`, `passWord`, `hidden`, `listBox`, `checkbox`, `radio`, dan `textArea`.
- Metode `getForm` digunakan untuk menghasilkan output akhir dari form HTML dengan menata elemen-elemen form dalam grid Bootstrap responsif. Memasukkan tombol submit dan reset.

Bentuk potongan program tersebut dituliskan dengan *source code* berikut:

```

<?php
spl_autoload_register(
    function($kelas) {
        require_once $_SERVER["DOCUMENT_ROOT"]."/SIMG2022/kelas/$kelas.php";
    }
);
$pasangan = array();
$pasangan[] = "0 - Belum terdaftar - Belum ada";
$kkks = array();
$kkks[] = "0 - Belum terdaftar - Belum ada";
include("koneksi.php");
$hasil = $con->query("SELECT id_anggota, nama_baptis, nama_lengkap, panggilan FROM
anggota");
while($baris = $hasil->fetch_assoc()){
    $pasangan[] = $baris['id_anggota']." - ".$baris['nama_baptis']." -
".$baris['nama_lengkap']." - ".$baris['panggilan'];
    $kkks[] = $baris['id_anggota']." - ".$baris['nama_baptis']." -
".$baris['nama_lengkap']." - ".$baris['panggilan'];
}
$gender = array("Pria","Wanita");
$status = array("Menikah","Single");
$gereja = array("GBIKA SOLO","GBIKA JOGJAKARTA","GBIKA JAKARTA","GBIKA MAGELANG","GBIKA
WONOGIRI","GBIKA NGAWI", "GBIKA MADIUN","GBIKA BANDUNG","GBIKA PURWOKERTO","GBIKA
SEMARANG","GBIKA KLATEN","GBIKA BOYOLALI","GBIKA SAMARINDA", "GBIKA BATURETNO","GBIKA
SALATIGA","GBIKA JAYAPURA","GBIKA SURABAYA","GBIKA GIRIBELAH","GBIKA TANGERANG","GBIKA
SANGATTA", "GBIKA PEKALONGAN","GBIKA NIAS","GBIKA LAMPUNG","GBIKA SRAGEN","GBIKA
WONOSARI","GEREJA INTERNET");
$darah = array("O","A","B","AB");
$pd = array("SD","SMP","SMA","S1","S2","S3");
$spr1 = array("Konsel", "Baptis","Retreat Encounter","ESBC");
$spr2 = array("Som 1", "Som 2", "Som 3", "Som 4", "Som 5", "Som 6");
$kt = array("Dewasa","Anak","Non Aktif","Almarhum");
$form = new Form("simpan_jemaat.php","Data Jemaat");
    $form->addTextBox("id_anggota","");
    $form->addTextBox("nama_lengkap","");
    $form->addTextBox("panggilan","");
    $form->addTextBox("nama_baptis","");
    $form->addListBox("pasangan",$pasangan,"");
    $form->addListBox("kkks",$kkks,"");
    $form->addDate("tgl_gabung","");
    $form->addTextBox("jabatan","");
    $form->addTextBox("no_ktp","");
    $form->addTextBox("tempat_lahir","");
    $form->addDate("tanggal_lahir","1968-07-27");
    $form->addRadio("gender",$gender,"Pria");
    $form->addListBox("gol_darah",$darah,"O");
    $form->addListBox("status",$status,"");
    $form->addListBox("gereja",$gereja,"");
    $form->addTextBox("alamat","");
    $form->addTextBox("rt","");
    $form->addTextBox("rw","");
    $form->addTextBox("kelurahan","");
    $form->addTextBox("kecamatan","");
    $form->addTextBox("kota","Yogyakarta");
    $form->addTextBox("kode_pos","");
    $form->addTextBox("propinsi","Daerah Istimewa Yogyakarta");
    $form->addTextBox("negara","Indonesia");
    $form->addTextBox("telpon","");
    $form->addTextBox("hp","");
    $form->addTextBox("email","");
    $form->addTextBox("sosmed","facebook.com");
    $form->addTextBox("pekerjaan","");
    $form->addListBox("pendidikan",$pd,"S1");
    $form->addTextBox("keterampilan","");
    $form->addCheckBox("skema_pr_1",$spr1,"");
    $form->addCheckBox("skema_pr_2",$spr2,"");
    $form->addListBox("status_kta",$kt,"");
    $form->getForm();
?>

```

Untuk menghasilkan antarmuka grafis seperti Gambar 2 di atas, diperlukan 2.617 karakter,

tampak pada program di atas untuk komponen antarmuka grafis yang sama hanya perlu memanggil nama metodenya saja.

Sedangkan untuk menghasilkan tampilan antarmuka grafis seperti Gambar 2 jika ditulis secara manual, atas dibutuhkan 11.673 karakter, data ini diperoleh dari kode HTML yang diterima oleh *web browser*.

4.2. Temuan dan Diskusi

4.2.1. Temuan

Hasil perbandingan antarmuka grafis yang dibuat secara manual dengan menggunakan *generator form* menghasilkan beberapa temuan sebagai berikut:

Tabel 1. Hasil Perbandingan

Komponen	Manual	Generator Form
Efisiensi pengembangan	Pembuatan antarmuka grafis secara manual membutuhkan waktu lebih lama dan lebih banyak karakter kode yang harus ditulis, pada contoh kasus ini efisiensinya mencapai 77,5%.	Menggunakan generator form yang dikembangkan dengan OOP, waktu pengembangan berkurang secara signifikan, dengan jumlah karakter kode yang lebih sedikit. Implementasi OOP menghasilkan kode program yang jumlah karakternya jauh lebih sedikit dibandingkan jika antarmuka grafis tersebut ditulis secara manual.
Kemudahan penggunaan	Membutuhkan pemahaman mendalam tentang HTML dan CSS, serta lebih rentan terhadap kesalahan manusia.	Mempermudah pengguna dalam membuat form dengan antarmuka yang lebih intuitif dan otomatisasi pengaturan layout.
Fleksibilitas dan kinerja	Memberikan fleksibilitas penuh dalam desain, namun kinerja dapat terpengaruh oleh kompleksitas kode.	Meskipun sedikit mengurangi fleksibilitas desain, generator form meningkatkan konsistensi dan kinerja dengan mengoptimalkan kode yang dihasilkan.

Data perbandingan dianalisis secara kualitatif untuk mengevaluasi kelebihan dan kekurangan OOP pada generator form web. Analisis ini fokus pada aspek modularitas, pemeliharaan kode, dan skalabilitas. Sedangkan untuk kelebihan dan kekurangan OOP untuk diterapkan pada *generator form* diterangkan dalam tabel 2 berikut:

Tabel 2. Kelebihan dan Kekurangan OOP pada *Generator Form Web*

Kelebihan	Kekurangan
<p>Modularitas: Kode yang modular memungkinkan pengembangan yang lebih terstruktur dan mudah dipahami.</p> <p>Pemeliharaan: Memudahkan dalam melakukan debugging dan pembaruan kode, karena perubahan dapat dilakukan pada satu kelas atau objek tanpa mempengaruhi keseluruhan sistem.</p> <p>Skalabilitas: Menyederhanakan penambahan fitur baru, karena kelas dan objek baru dapat ditambahkan tanpa perlu mengubah kode yang sudah ada secara signifikan.</p> <p>Efisiensi Kode: Implementasi OOP menghasilkan kode program yang jumlah karakternya jauh lebih sedikit dibandingkan dengan penulisan antarmuka grafis secara manual, sehingga mengurangi beban penulisan dan pemeliharaan kode.</p>	<p>Kompleksitas Awal: Implementasi awal menggunakan OOP bisa lebih kompleks dibandingkan dengan pendekatan prosedural, membutuhkan waktu dan pemahaman yang lebih dalam tentang konsep OOP.</p> <p>Overhead Kinerja: Dalam beberapa kasus, penggunaan OOP dapat menyebabkan overhead kinerja, terutama jika tidak dioptimalkan dengan baik.</p>

4.2.2. Diskusi

Penelitian ini bertujuan untuk menganalisis kelebihan dan kekurangan pemrograman berorientasi objek (OOP) dalam pengembangan generator form web. Berdasarkan hasil penelitian, ditemukan bahwa jumlah karakter pada program yang ditulis menggunakan OOP jauh lebih sedikit dibandingkan dengan apabila antarmuka grafis tersebut ditulis secara manual. Hal ini menunjukkan efisiensi dalam penulisan kode, yang merupakan salah satu keunggulan utama dari OOP.

OOP memungkinkan pengembang untuk menulis kode yang lebih modular dan terstruktur. Dalam konteks generator form web, penggunaan kelas dan objek mempermudah pengelolaan dan pengembangan fitur baru. Sebagai contoh, dengan menerapkan konsep inheritance dan polymorphism, pengembang dapat mengembangkan berbagai jenis form tanpa harus menulis ulang seluruh kode, cukup dengan menambah atau memodifikasi kelas yang ada.

Penemuan ini konsisten dengan penelitian sebelumnya yang menunjukkan bahwa OOP dapat meningkatkan efisiensi pengembangan perangkat lunak. Studi oleh [8] menunjukkan bahwa OOP mengurangi duplikasi kode dan mempermudah pemeliharaan perangkat lunak. Hasil penelitian ini memperkuat temuan tersebut dengan bukti konkret dari implementasi generator form web. Namun, penelitian ini juga menemukan beberapa kekurangan dari penggunaan OOP. Proses belajar dan implementasi awal OOP memerlukan waktu dan usaha yang lebih dibandingkan dengan pendekatan *procedural* [9]. Selain itu, untuk aplikasi kecil

atau yang tidak memerlukan modularitas yang tinggi, overhead dari OOP bisa menjadi kurang efisien.

Temuan penelitian ini memiliki beberapa implikasi penting. Pertama, bagi pengembang perangkat lunak, OOP menawarkan pendekatan yang lebih efisien dan terstruktur dalam pengembangan form web, terutama untuk proyek besar dan kompleks. Kedua, bagi akademisi dan pendidik, hasil ini menunjukkan pentingnya memasukkan OOP dalam kurikulum pengembangan perangkat lunak untuk membekali mahasiswa dengan keterampilan yang relevan dan efisien.

Meskipun penelitian ini memberikan wawasan berharga tentang kelebihan dan kekurangan OOP dalam pengembangan form web, masih banyak area yang memerlukan eksplorasi lebih lanjut. Penelitian masa depan dapat fokus pada:

- Studi Komparatif: Membandingkan OOP dengan metode pengembangan lain, seperti pemrograman fungsional atau prosedural, dalam konteks pengembangan web.
- Studi Kasus yang Lebih Luas: Menerapkan OOP pada berbagai jenis aplikasi web dan membandingkan efisiensinya.
- Pengembangan Alat Bantu: Mengembangkan alat bantu atau framework yang dapat mempermudah implementasi OOP dalam pengembangan form web.
- Pengaruh Terhadap Kinerja: Menganalisis pengaruh OOP terhadap kinerja aplikasi web, seperti kecepatan pemrosesan dan penggunaan memori.

Dengan melanjutkan penelitian di area ini, kita dapat terus meningkatkan efisiensi dan efektivitas pengembangan perangkat lunak, serta memastikan bahwa teknik-teknik yang digunakan sesuai dengan kebutuhan dan tantangan industri perangkat lunak yang terus berkembang.

5. KESIMPULAN

Penelitian ini telah menganalisis kelebihan dan kekurangan pemrograman berorientasi objek (OOP) dalam konteks pengembangan *generator form web*. Beberapa temuan utama dari penelitian ini yaitu efisiensi penulisan kode, modularitas dan pemeliharaan, pembelajaran dan implikasi, konsistensi dengan penelitian sebelumnya, implikasi untuk pengembangan perangkat lunak, dan arah penelitian selanjutnya. Secara keseluruhan, penelitian ini menegaskan bahwa OOP adalah pendekatan yang kuat dan efisien dalam pengembangan generator form web, meskipun memerlukan investasi awal yang lebih besar dalam hal pembelajaran dan implementasi. Keuntungan modularitas, efisiensi penulisan kode, dan pemeliharaan jangka panjang membuat OOP menjadi pilihan yang baik untuk pengembangan perangkat lunak, terutama untuk proyek-proyek besar dan kompleks.

REFERENSI

- [1] P. McFedries, *Web Coding & Development All-in-One For Dummies*. John Wiley & Sons, 2024.
- [2] L. Wroblewski, *Web form design: filling in the blanks*. Rosenfeld Media, 2008.
- [3] C. Jarrett and G. Gaffney, *Forms that work: Designing Web forms for usability*. Morgan Kaufmann, 2009.
- [4] P. Wegner, "Concepts and paradigms of object-oriented programming," *ACM Sigplan Oops Messenger*, vol. 1, no. 1, pp. 7–87, 1990.
- [5] A. Ojha, R. Tiwari, K. Kadam, and K. Khot, "Web Application Development with Object

- Oriented Programming,” *Pridobljeno iz <https://s3.Amaz.com/academia.edu>. Doc.*, vol. 9, 2017.
- [6] E. D. Andriano, I. Ahmad, and R. D. Gunawan, “Pengembangan Aplikasi Pengaduan Nasabah Berbasis Website Menggunakan Framework Laravel (Studi Kasus: PT BPR UTOMO MSL),” *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 4, no. 3, pp. 246–252, 2023.
- [7] Z. A. W. Sugandi, Y. A. Nugraha, S. N. Anam, and I. Darmayanti, “Implementasi Konsep Pemrograman Berorientasi Objek Dalam Aplikasi Pembukuan Keuangan Penjual Jus Buah Menggunakan Bahasa Pemrograman Java,” *J. Ilm. IT CIDA*, vol. 8, no. 1, pp. 1–8, 2022.
- [8] J. Reaves, “Comparing ladder logic and object-oriented programming: Many young automation professionals are comfortable with object-oriented programming (OOP), but ladder logic remains the standard for industrial automation applications. See advantages and disadvantages,” *Control Eng.*, vol. 65, no. 7, pp. 18–21, 2018.
- [9] S. C. Agu and F. Elugwu, “Object Oriented Programming Approach: A Panacea for Effective Software Development,” *African J. Adv. Sci. Technol. Res.*, vol. 6, no. 1, pp. 1–14, 2022.
- [10] R. Yesputra and J. Hutahaean, “Implementasi object oriented programming dalam pengembangan aplikasi berbasis web.(studi kasus: e-alumni stmik royal),” 2011.
- [11] M. Seckler, S. Heinz, J. A. Bargas-Avila, K. Opwis, and A. N. Tuch, “Designing usable web forms: empirical evaluation of web form improvement guidelines,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 1275–1284.
- [12] A. Negametzyanov, S. L. Lau, and C. F. Ng, “Web-based Interactive Form Generator for Public Kiosks,” in *2015 IEEE Conference on Open Systems (ICOS)*, 2015, pp. 108–113.
- [13] D. Strmečki, D. Radošević, and I. Magdalenić, “Web form generators design model,” in *Central European Conference on Information and Intelligent Systems*, 2015, no. November, pp. 255–260.
- [14] Y. Oslan, H. Kristanto, and R. G. Santosa, “Pendampingan Pembuatan Dan Pengelolaan Data Elektronik Jemaat GKJ Wates,” *Servirisma*, vol. 1, no. 1, pp. 73–82, 2021.