



## **PERANCANGAN APLIKASI WEB MANAJEMEN DATA PRODUK BISNIS PERHIASAN BERBASIS FLASK DAN MONGODB**

**Usman Syach<sup>1</sup>, Sri Winarso Martyas Edi<sup>2</sup>**

<sup>1,2</sup> Program Studi Teknik Informatika, Fakultas Teknologi Informasi

Universitas Kristen Satya Wacana

Jl. O.Notohamidjojo, no 1-10, Blotongan, Salatiga

Email: 672020201@student.uksw.edu<sup>1</sup>, winarso@uksw.edu<sup>2</sup>

### **Riwayat artikel:**

*Submitted: 23-02-2024*

*Revised: 25-03-2024*

*Published: 21-06-2024*

**Abstrak** – Penelitian ini bertujuan untuk merancang aplikasi web manajemen data produk berbasis *Flask* dan *MongoDB* untuk studi kasus usaha yang bergerak dalam bisnis perhiasan. Tujuan pengembangan aplikasi adalah untuk meningkatkan efisiensi pemasaran, pemesanan, dan manajemen stok produk. Penelitian ini mengidentifikasi beberapa masalah utama dalam konteks manajemen data produk perhiasan yang sering mengalami perubahan. Batasan penelitian mencakup pada tiga fungsi utama aplikasi yaitu manajemen data produk, efisiensi pemasaran, dan pemesanan, dengan teknologi utama berupa *Flask* dan *MongoDB*. Metode penelitian mencakup analisis kebutuhan, pembangunan prototipe, evaluasi prototipe, pengkodean sistem, pengujian sistem, dan evaluasi sistem. Penelitian ini menghasilkan aplikasi yang dapat memberikan kontribusi pada peningkatan efisiensi operasional perusahaan, pengembangan keterampilan pengembang, serta potensi pengembangan dan referensi untuk industri perhiasan terkait manajemen data produk dan pemasaran online.

**Kata Kunci** – Bisnis Perhiasan, Manajemen Data Produk, Flask, MongoDB.

**Abstract** – This research aims to design a product data management web application based on *Flask* and *MongoDB* for a case study of a business engaged in the jewelry business. The aim of application development is to increase marketing efficiency, ordering and product stock management. This research identifies several main problems in the context of jewelry product data management which frequently changes. Research boundaries include three main application functions, namely product data management, marketing efficiency, and ordering, with the main technologies being *Flask* and *MongoDB*. Research methods include requirements analysis, prototype development, prototype evaluation, system coding, system testing, and system evaluation. This research produces applications that can contribute to increasing company operational efficiency, developing developer skills, as well as potential development and references for the jewelry industry related to product data management and online marketing.

**Keywords** – Jewelry Business, Product Data Management, Flask, MongoDB.

## I. PENDAHULUAN

Pemanfaatan internet semakin mendominasi dalam ranah bisnis, dimana banyak perusahaan memandang internet sebagai sarana utama untuk mengembangkan usaha mereka. Bisnis yang sepenuhnya beroperasi melalui internet, mencakup seluruh proses pemesanan hingga pengantaran barang, dikenal sebagai *e-commerce* [1]. Sebagai contoh, banyak perusahaan saat ini mengadopsi pendekatan menggunakan situs web sebagai alat untuk mempromosikan produk mereka, sehingga memungkinkan mereka lebih efisien dalam menjalankan kegiatan penjualan. Strategi ini juga membantu dalam menyederhanakan proses pemasaran dan manajemen data perusahaan [2]. Dengan memanfaatkan website, para pengusaha dapat lebih efektif memperkenalkan dan mempromosikan produk unggulan mereka kepada berbagai lapisan masyarakat.

Penelitian ini menggunakan studi kasus di PT XYZ, sebuah perusahaan perhiasan yang membutuhkan solusi untuk mengelola data produknya secara efisien dan terintegrasi. Penggunaan media sosial untuk pemasaran, memang membantu produktifitas penjualan, akan tetapi membuat manajemen pengelolaan data tidak terorganisir dengan baik. Hal ini mengakibatkan inefisiensi, kesulitan dalam melacak data, dan potensi kehilangan data. Manajemen data produk melalui web dapat menjadi solusi yang tepat. Platform web memungkinkan akses data *real-time*, meningkatkan efisiensi dan pemasaran, serta memudahkan pemesanan dan pengelolaan data. Selain itu, platform web yang aman dan andal dapat membantu menjaga kerahasiaan data. Peningkatan layanan pelanggan melalui platform web juga dapat membantu perusahaan untuk meningkatkan kepuasan pelanggan, meningkatkan penjualan serta membantu organisasi manajemen data secara lebih efisien dan terstruktur [3].

Dari permasalahan tersebut, diperlukan suatu aplikasi web yang memiliki fitur pemasaran, pemesanan, dan pengendalian stok produk. Penggunaan Flask *Framework* dapat memberikan keleluasaan dan kemudahan bagi pengembang dalam membangun aplikasi. Walaupun Flask dikenal sebagai *Framework* ringan dengan inti yang sederhana, framework ini menonjol berkat sifat simpel dan fleksibilitasnya. Hal ini memudahkan pengembangan serta penyesuaian sesuai kebutuhan, terutama melalui penambahan ekstensi yang diperlukan [4][5]. Penggunaan Flask *Framework* memberikan keuntungan signifikan. Flask, dengan sintaksis yang mudah dipahami dan ringan, memungkinkan pengembangan yang cepat. Dukungan komunitas yang kuat membuat penyelesaian masalah atau dukungan dapat ditemukan dengan mudah. Fleksibilitas dalam skalabilitas serta kemampuan untuk berintegrasi dengan MongoDB sebagai basis data mendukung manajemen data produk perhiasan yang efisien. Selain itu, penggunaan Flask juga dapat meningkatkan kemampuan aplikasi untuk menangani kebutuhan yang lebih kompleks [6].

Penggunaan MongoDB dalam aplikasi ini sangat relevan karena MongoDB adalah *database* NoSQL yang sangat cocok untuk mengelola data yang cenderung berubah dalam bisnis perhiasan. Bisnis perhiasan seringkali mengalami rotasi stok yang cepat, dengan desain dan tren yang berubah seiring waktu, berbeda dengan bisnis lain seperti manufaktur, properti, dan keuangan/asuransi. Dengan menggunakan MongoDB, data dapat disimpan dalam format dokumen yang mudah disesuaikan tanpa perlu migrasi data yang krusial dalam manajemen data produk yang sering berubah. MongoDB juga dirancang untuk mendukung skalabilitas horizontal dengan mudah, memberikan dukungan bagi pertumbuhan bisnis, dan mempertahankan efisiensi

manajemen data dalam jangka panjang. Selain itu, MongoDB juga mudah diinstal dan dikonfigurasi [7][8].

## II. KAJIAN PUSTAKA

Penelitian berjudul "Model *Waterfall* Dalam Rancang Bangun Sistem Informasi Pemesanan Produk dan Outlet Berbasis Web" yang dilakukan Handrianto dan Sanjaya [9] membahas pembangunan sistem informasi pemesanan produk dan outlet dengan menggunakan model *waterfall*. Dalam penelitian tersebut, teknologi dasar yang digunakan adalah PHP dan MySQL. Penggunaan MySQL memerlukan definisi skema tabel sebelumnya saat menambah atau menghapus bidang dalam dokumen. Sebaliknya, MongoDB, dengan model data NoSQL berbasis dokumen, memungkinkan penyimpanan data dalam format dokumen *Binary JSON* (BSON) yang lebih fleksibel. Hal ini membuat penggunaan MongoDB menjadi lebih mudah dalam menambahkan atau menghapus bidang dalam dokumen tanpa mengganggu data yang sudah ada.

Selanjutnya penelitian berjudul "Rancang Bangun Sistem Informasi Penjualan Laptop Berbasis WEB" oleh Manis dan kawan-kawan [10] membahas pembangunan sistem informasi penjualan laptop menggunakan metode *waterfall*. Dalam penelitian tersebut, teknologi dasar yang digunakan adalah PHP dan HTML. Meskipun PHP umumnya digunakan dalam pengembangan web, namun memiliki potensi kerapuhan yang dapat menyebabkan celah keamanan jika tidak diatur dengan benar. Selain itu, dalam aplikasi web yang besar dan kompleks, PHP dapat menghasilkan kode yang sulit dipelihara. Sebaliknya, Flask Python memiliki mekanisme keamanan bawaan yang membantu melindungi aplikasi dari serangan umum dan didukung oleh komunitas yang aktif untuk pengembangan dan pembaruan pustaka keamanan. Oleh karena itu, dari segi keamanan dan pemeliharaan, penggunaan Flask Python lebih disarankan.

Penelitian yang berjudul "Pembangunan Aplikasi Web Sirkulasi Buku Perpustakaan Universitas Atma Jaya Yogyakarta" oleh Kwa, dan kawan-kawan [11] membahas pengembangan aplikasi web sirkulasi buku perpustakaan dengan menggunakan metode *waterfall* [11]. Meskipun metode *waterfall* digunakan dalam perancangan aplikasi, pendekatan ini dianggap kurang sesuai karena kurangnya fleksibilitas terhadap perubahan kebutuhan dan keterlibatan pengguna yang terbatas. Sebaliknya, metode *prototype* ditunjukkan sebagai alternatif yang lebih baik karena lebih fleksibel dalam mengakomodasi perubahan, melibatkan pengguna lebih awal dan lebih sering, serta meminimalkan risiko ketidakcocokan antara pengguna dan pengembang.

Penelitian berjudul "Perancangan Aplikasi Point of Sale Untuk Manajemen Pemesanan Bahan Pangan Berbasis *Framework* Laravel" oleh Yuniarti [12] membahas perancangan aplikasi Manajemen Pemesanan Bahan Pangan berbasis *Framework* Laravel. Meskipun Laravel menawarkan fitur-fitur yang kuat seperti *Eloquent ORM*, yang sangat cocok untuk proyek-proyek yang lebih kompleks, aplikasi manajemen data produk seringkali memerlukan adaptasi yang cepat terhadap perubahan dalam struktur dan logika bisnis. Flask, sebagai kerangka kerja yang ringan dan minimalis, memberikan kebebasan lebih kepada pengembang untuk mengelola *database*, mengontrol alur logika, dan menyesuaikan model-data sesuai kebutuhan [6]. Oleh karena itu, Flask lebih cocok untuk membangun aplikasi manajemen data produk yang mungkin membutuhkan perubahan konstan dalam pengembangan dan persyaratan

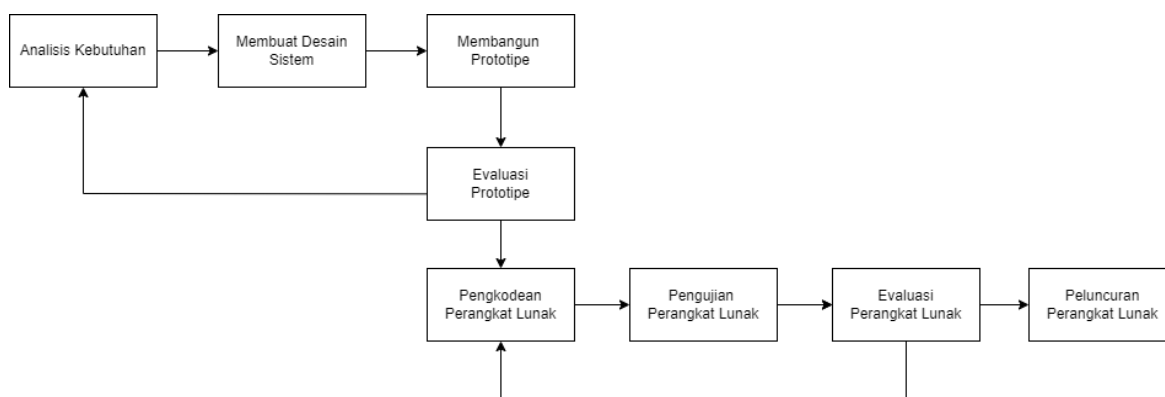
bisnis yang berkembang. membuat tampilan dengan sintaks yang bersih dan ekspresif, serta menyediakan fitur-fitur seperti warisan (*inheritance*) dan kontrol struktural yang kuat. Disamping itu, Laravel memiliki *Community Support* yaitu komunitas yang besar dan aktif, banyak sumber daya *online*, forum, dan paket-paket tambahan yang dapat membantu pengembang ketika mengalami masalah atau membutuhkan bantuan.

Penelitian "Perancangan Sistem Informasi Manajemen Invoice Menggunakan Generator *Framework* Django-Python Berbasis Website Pada Pt. Lampuind Tekno Elektrik" oleh Risyda, F., & Nuryamin, Y. membahas perancangan Sistem Informasi Manajemen *Invoice* berbasis Django-Python [13]. Django memiliki banyak fitur bawaan yang mungkin tidak diperlukan dengan ukuran yang besar. Oleh karena itu, Flask dianggap lebih cocok karena lebih ringan, minimalis, dan memberikan pengembang lebih banyak kontrol dalam merancang serta mengintegrasikan komponen yang sesuai dengan kebutuhan proyek. Ini memungkinkan fokus pada fitur esensial seperti manajemen stok produk dan pemasaran. Selain itu, Flask memiliki kurva pembelajaran yang lebih mudah, memungkinkan tim pengembang yang belum terlalu berpengalaman dengan kerangka kerja web untuk lebih cepat memahami dan mengembangkan solusi sesuai dengan tujuan proyek [14].

Memperhatikan berbagai penelitian sebelumnya, maka penelitian ini mengusulkan pengembangan aplikasi web untuk manajemen data produk dengan keunikan yang terletak pada penggunaan teknologi terkini, seperti database MongoDB, dan *framework* Flask Python. Flask sebagai *framework* yang ringan dan fleksibel, memungkinkan pengembangan aplikasi yang cepat dan mudah. MongoDB, sebagai database NoSQL, memungkinkan penyimpanan data yang fleksibel dan skalabel. Keunggulan ini menjadikan kombinasi Flask dan MongoDB menjadi unik serta ideal untuk menangani perubahan data yang sering terjadi dalam manajemen produk.

### III. METODE PENELITIAN

Metode atau tahapan yang dilakukan dalam penelitian ini digambarkan pada Gambar 1 di bawah ini. Gambar tersebut mengilustrasikan serangkaian langkah dalam implementasi perangkat lunak, yang bertujuan memastikan bahwa sistem dapat memenuhi kebutuhan dan tujuan perancangan yang ditetapkan.



Gambar 1. Metode penelitian prototipe

Langkah-langkah dalam penelitian ini mencakup analisis kebutuhan, perancangan arsitektur, penyusunan desain sistem, pembuatan prototipe, evaluasi

prototipe, pengkodean perangkat lunak, uji perangkat lunak, evaluasi kinerja perangkat lunak, dan implementasi. Dengan mengikuti serangkaian langkah ini, diharapkan sistem dapat dikembangkan dengan optimal sesuai dengan spesifikasi yang telah ditetapkan.

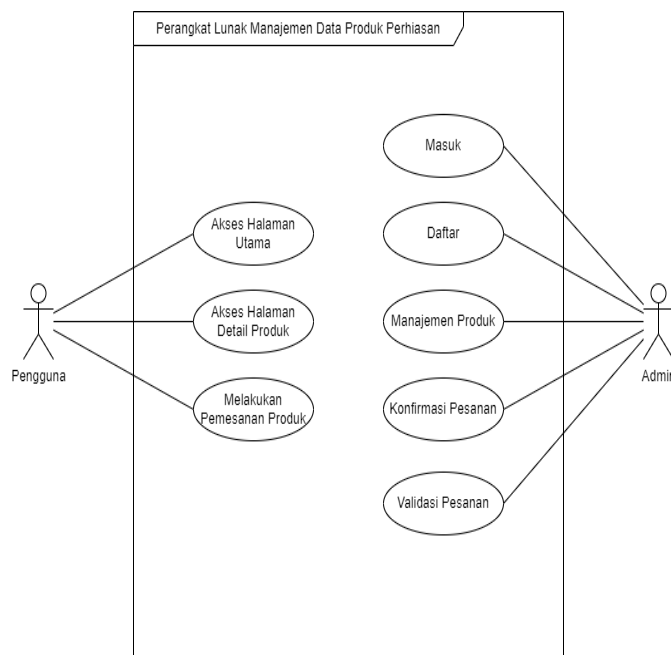
Tahap analisis kebutuhan merupakan langkah awal dalam mengidentifikasi dan mencatat kebutuhan yang esensial untuk sistem [15]. Menghadapi tantangan yang dihadapi oleh perusahaan, diperlukan suatu aplikasi manajemen data produk yang dapat meningkatkan efisiensi dalam proses pemasaran, pemesanan, dan manajemen stok produk. Aplikasi web ini dirancang menggunakan bahasa pemrograman Python, *Framework* Flask, dan basis data MongoDB. Pemilihan Python didasarkan pada fleksibilitasnya yang dapat digunakan dalam berbagai konteks, termasuk pengembangan perangkat lunak, analisis data, dan keperluan lainnya. Selain itu, Python menyediakan berbagai pustaka yang memudahkan para pengembang dalam menjalankan tugas mereka.

Setelah menetapkan kebutuhan sistem, langkah berikutnya adalah merancang sistem agar memastikan bahwa keluaran perangkat lunak sesuai dengan kebutuhan pengguna. Proses perancangan sistem dapat dilakukan dengan memanfaatkan *Unified Modeling Language* (UML), suatu metode yang memanfaatkan representasi visual untuk mendokumentasikan, menetapkan spesifikasi, dan mengembangkan perangkat lunak [15]. Beberapa jenis diagram UML yang diterapkan dalam penelitian ini melibatkan diagram *usecase*, dan diagram aktivitas.

#### IV. HASIL DAN PEMBAHASAN

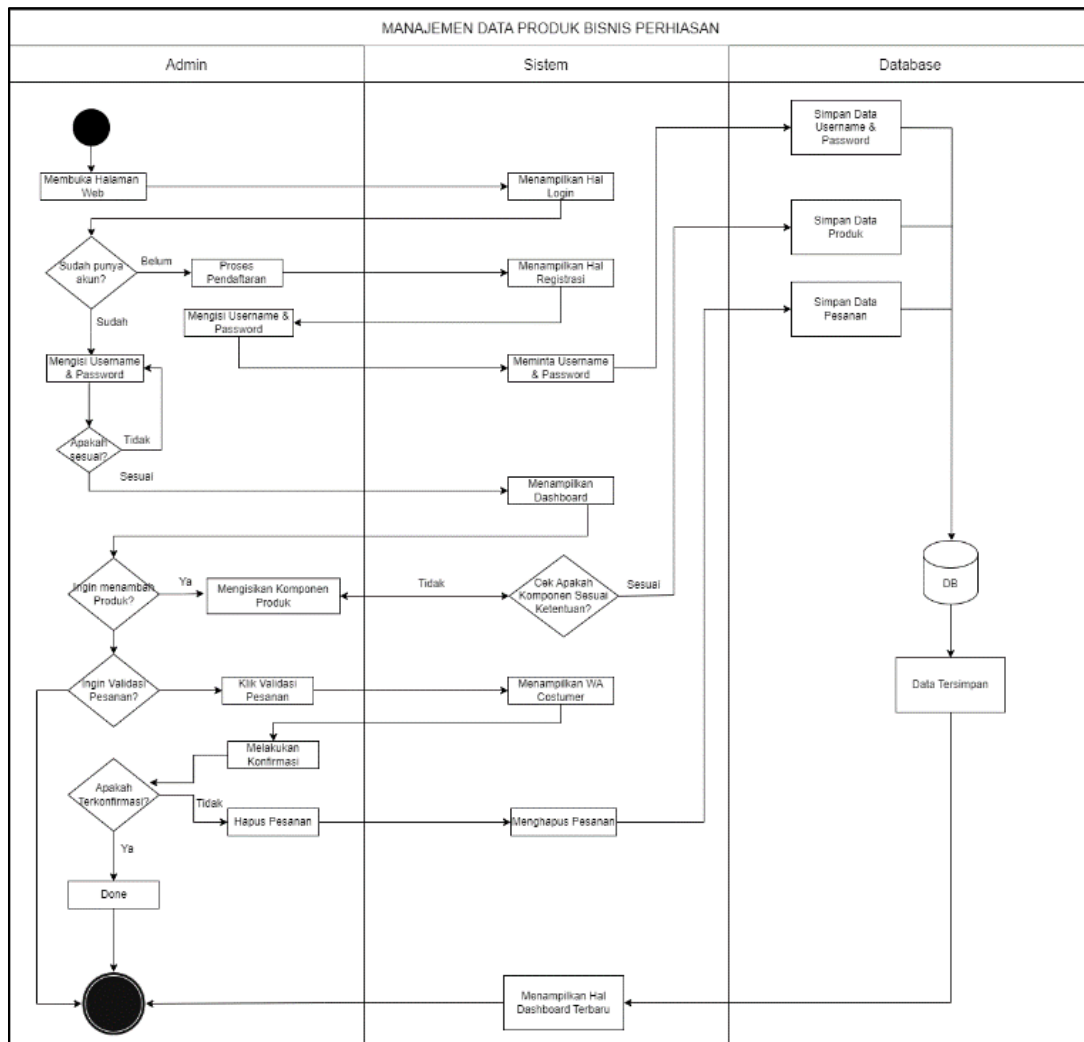
##### A. Tahap Pemodelan Sistem

Beberapa jenis diagram UML yang diterapkan dalam penelitian ini melibatkan diagram *usecase*, dan diagram aktivitas.



Gambar 2. Diagram *usecase*

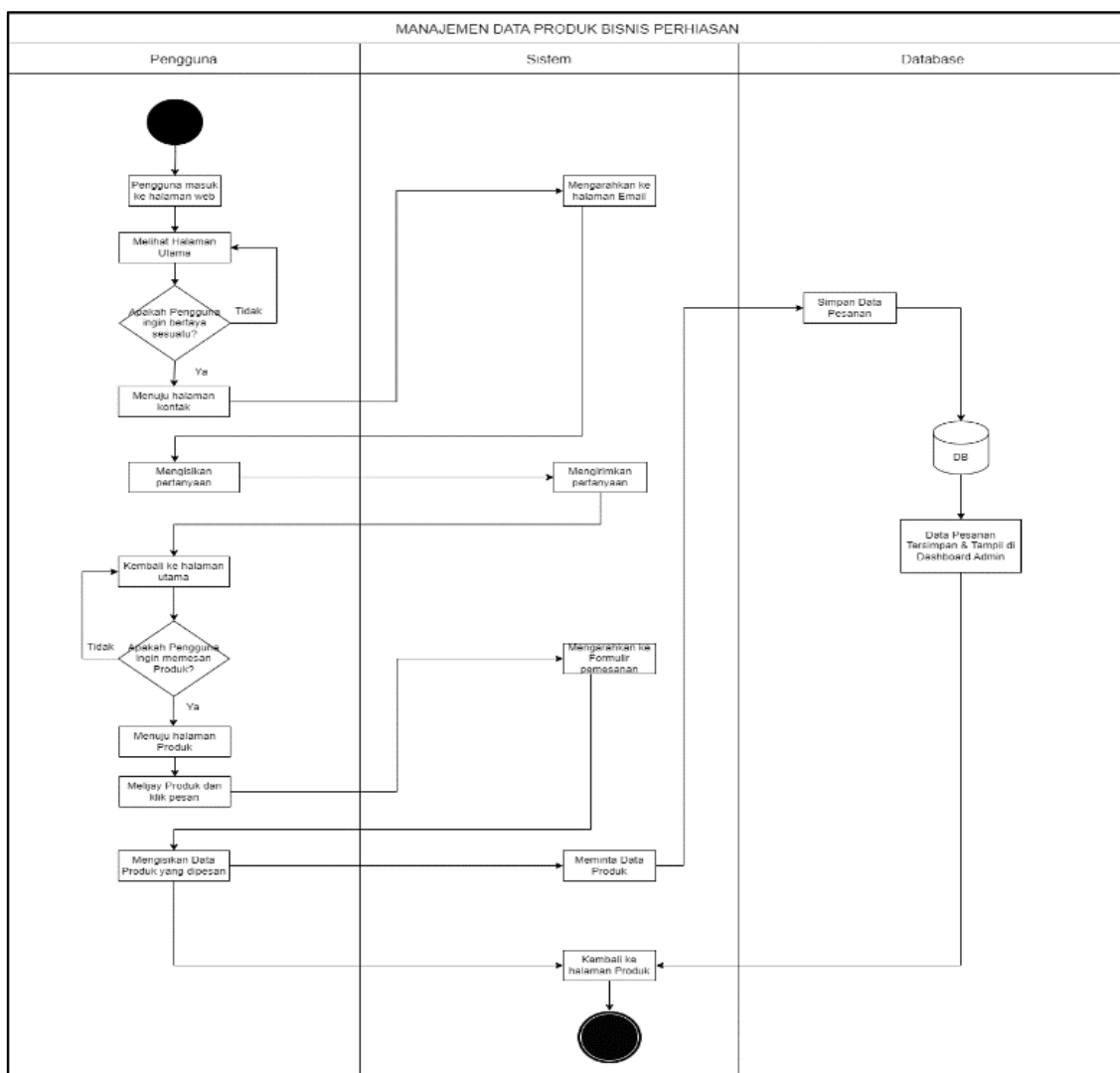
Diagram *usecase* pada Gambar 2 digunakan untuk memvisualisasikan hubungan antar kebutuhan sistem. Komponen-komponen dalam diagram *usecase* melibatkan aktor dan *usecase* itu sendiri, sebagaimana dijelaskan dalam referensi [14]. Terdapat dua aktor utama dalam diagram ini, yaitu admin dan pengguna. Peran admin memberikan akses ke sejumlah fungsi kunci, seperti halaman *dashboard*, penambahan produk, pengeditan produk, melihat detail produk, meninjau pesanan masuk, melakukan validasi pesanan, konfirmasi pesanan, dan mengelola produk. Di sisi lain, pengguna memiliki hak akses ke fungsi-fungsi seperti halaman utama melalui *URL*, halaman detail produk, serta kemampuan untuk melakukan pemesanan produk.



Gambar 3. Diagram aktivitas admin

Gambar 3 menampilkan diagram aktivitas yang mengilustrasikan serangkaian langkah yang diambil oleh admin selama berinteraksi dengan sistem. Proses dimulai dengan admin melakukan login menggunakan akun admin yang dimilikinya. Setelah berhasil login, admin akan diarahkan ke halaman utama yang mencakup dashboard data produk serta informasi tentang pesanan masuk. Jika admin ingin menambahkan produk, langkah selanjutnya adalah menekan tombol *add*, yang akan mengarahkan sistem ke formulir penambahan produk. Admin kemudian harus menginput komponen produk sesuai dengan formulir, dan data tersebut akan disimpan ke dalam *database*, hasilnya akan ditampilkan di halaman produk. Selanjutnya, jika admin hendak melakukan validasi

pesanan, admin perlu menuju halaman pesanan. Sistem akan menampilkan tabel pesanan yang dilakukan oleh pengguna. Di halaman ini, admin dapat melakukan validasi pesanan atau bahkan membatalkan pesanan tersebut.



Gambar 4. Diagram aktivitas pengguna

Gambar 4 memperlihatkan diagram aktivitas yang menggambarkan langkah-langkah yang diambil oleh pengguna selama berinteraksi dengan sistem. Proses dimulai saat pengguna melihat halaman *company profile*. Jika pengguna tertarik untuk melakukan pemesanan produk, pengguna harus menuju halaman *product* dan menekan tombol *order* untuk melakukan pemesanan dengan cara melakukan pengisian form pemesanan yang nantinya data yang diinputkan akan masuk ke halaman pemesanan *dashboard* admin.

Setelah menyelesaikan desain sistem, langkah selanjutnya adalah membangun prototipe perangkat lunak yang difokuskan pada fungsi utama, khususnya manajemen data produk. Evaluasi dilakukan untuk memastikan bahwa prototipe memenuhi kebutuhan dan sesuai dengan tujuan perancangan yang telah ditetapkan. Jika prototipe memenuhi harapan, pengembang dapat melanjutkan ke tahap selanjutnya dalam pengembangan perangkat lunak. Jika tidak, iterasi dan penyesuaian mungkin diperlukan

sebelum melanjutkan pengembangan. Evaluasi ini krusial untuk memastikan proyek bergerak sesuai kebutuhan dan tujuan yang ditetapkan.

Setelah prototipe sesuai, langkah selanjutnya adalah pengkodean. Dilanjutkan dengan pengujian *black box* untuk memastikan kinerja sesuai kebutuhan. Evaluasi tambahan dilakukan sebelum merilis aplikasi. Jika diperlukan perbaikan atau peningkatan, kembali ke pengkodean dan pengujian. Jika semua sesuai, luncurkan perangkat lunak dan masuk ke fase pemeliharaan dan pembaruan.

## B. Implementasi Sistem

Melalui serangkaian langkah-langkah dalam desain dan perancangan perangkat lunak, penelitian ini berhasil menciptakan sebuah aplikasi web yang efektif dalam mengelola data produk pada bisnis perhiasan. Aplikasi web ini dikembangkan menggunakan bahasa pemrograman Python, *Framework* Flask, dan MongoDB, memastikan keefisienan dalam manajemen data produk pada tingkat bisnis perhiasan.

### Kode Program 1. *Library* Flask yang digunakan pada Python

```
from flask import Flask, render_template, url_for, redirect, request,
session, flash, jsonify
from werkzeug.security import generate_password_hash, check_password_hash
import requests
from flask_pymongo import PyMongo
from bson import ObjectId

app = Flask(__name__)
```

Kode Program 1 menggunakan berbagai *library* di *Backend* (Python) dengan Flask sebagai kerangka kerja web utamanya. Pustaka-pustaka seperti `render_template`, `url_for`, `request`, `session`, `flash`, `jsonify`, `Werkzeug.security`, dan `requests` digunakan untuk manajemen tampilan, penanganan HTTP, keamanan kata sandi, komunikasi dengan layanan eksternal, serta manipulasi data JSON dan integrasi MongoDB. Pustaka-pustaka ini bekerja bersama untuk memudahkan pengembangan aplikasi web dengan Flask.

### Kode Program 2. Contoh kode program manajemen data produk

```
@app.route("/add_product", methods=["GET", "POST"])
def add_product():
    if request.method == "POST":
        # Dapatkan data dari formulir
        id = request.form['id']
        category = request.form['category']
        name = request.form['name']
        weight = request.form['weight']
        description = request.form['description']
        stok = request.form['stok']
        price = request.form['price']
        image_url = request.form['image_url'] # Ambil URL gambar dari
        formulir
    ...
```

Kode Program 2 digunakan untuk manajemen data produk dengan fungsi `add_product` pada rute `/add_product`. Saat metodenya adalah POST, data produk seperti ID, kategori, nama, berat, deskripsi, harga, *URL* gambar, dan jumlah stok diambil dari formulir, lalu disimpan ke dalam *database*. Ini memungkinkan pengguna menambahkan produk baru dengan informasi seperti harga, gambar, dan jumlah stok. Setelah



penambahan produk berhasil, pengguna diarahkan kembali ke *dashboard*.

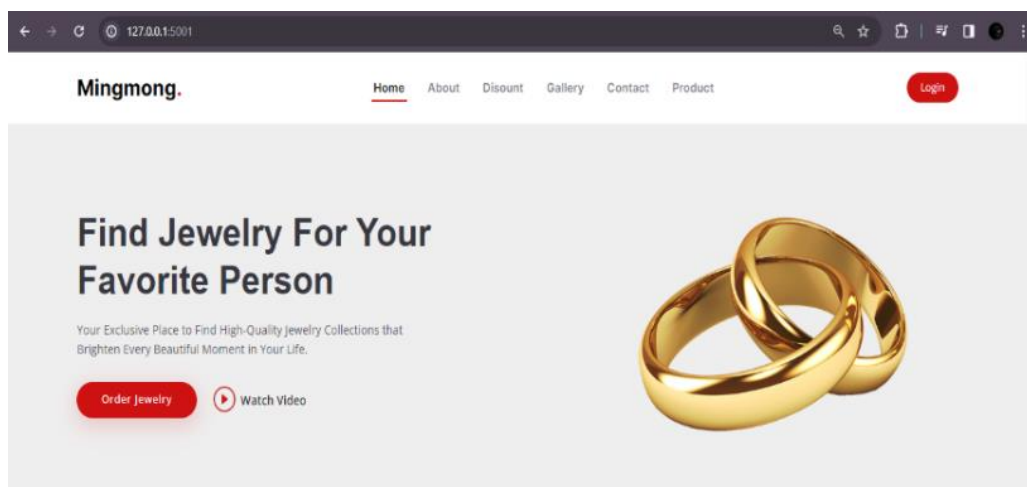
### Kode Program 3. Kode program manajemen stok produk

```
@app.route("/reduce_stock/<product_id>/<amount>", methods=["POST"])
def reduce_stock(product_id, amount):
    # Temukan produk berdasarkan ID
    product = products.find_one({'_id': ObjectId(product_id)})

    if product:
        # Ubah nilai stok menjadi integer sebelum mengurangi
        current_stock = int(product['stok'])

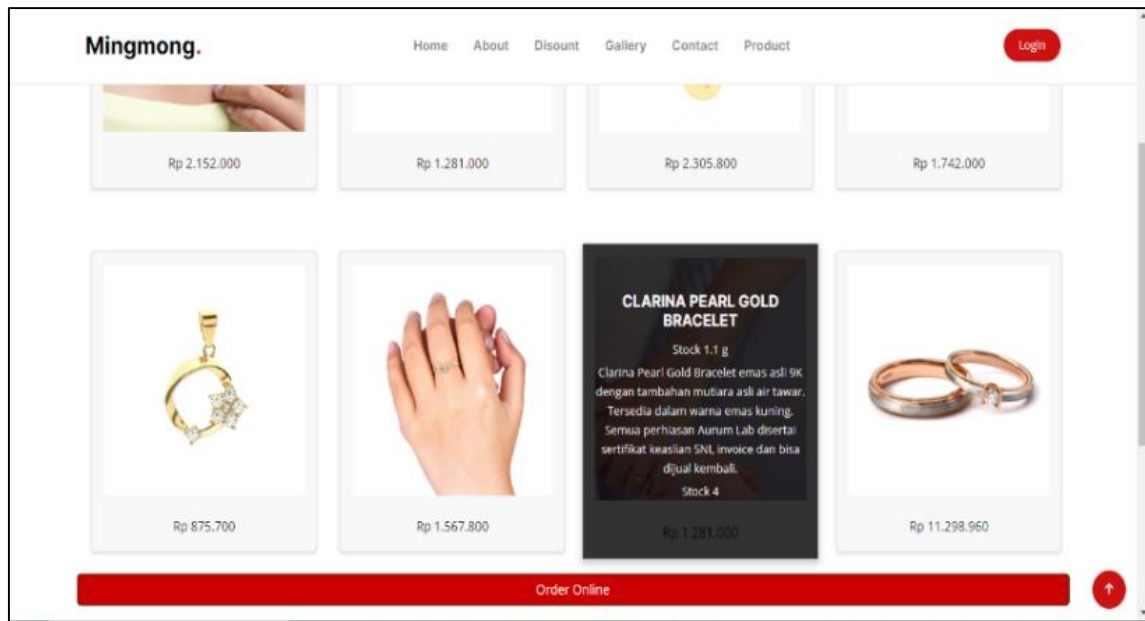
        # Kurangi jumlah stok
        new_stock = max(current_stock - int(amount), 0)
        products.update_one({'_id': ObjectId(product_id)}, {'$set':
{'stok': new_stock}})
```

Kode Program 3 berfungsi untuk pemesanan dan manajemen stok produk dengan dua rute tambahan. Pertama, pada rute */reduce\_stock/<product\_id>/<amount>*, fungsi *reduce\_stock* mengurangi jumlah stok produk berdasarkan ID dan jumlah yang diinginkan, memastikan stok tidak menjadi negatif, dan memberikan respons JSON. Kedua, rute */done\_order/<order\_id>* pada fungsi *done\_order* digunakan untuk menandai selesainya pesanan dengan menghapusnya dari database. Program ini terkait dengan manajemen stok produk dan penyelesaian pesanan, sesuai dengan aplikasi yang mungkin berfokus pada penjualan atau manajemen inventaris.



Gambar 5. Tampilan halaman *Home*

Gambar 5 menampilkan tampilan halaman utama aplikasi untuk pengguna. Sesuai dengan tujuan aplikasi untuk meningkatkan efisiensi pemasaran, desain tampilan ini dirancang dengan estetika yang menarik, bertujuan untuk memberikan kemudahan dan menarik minat pengguna dalam melakukan pembelian.



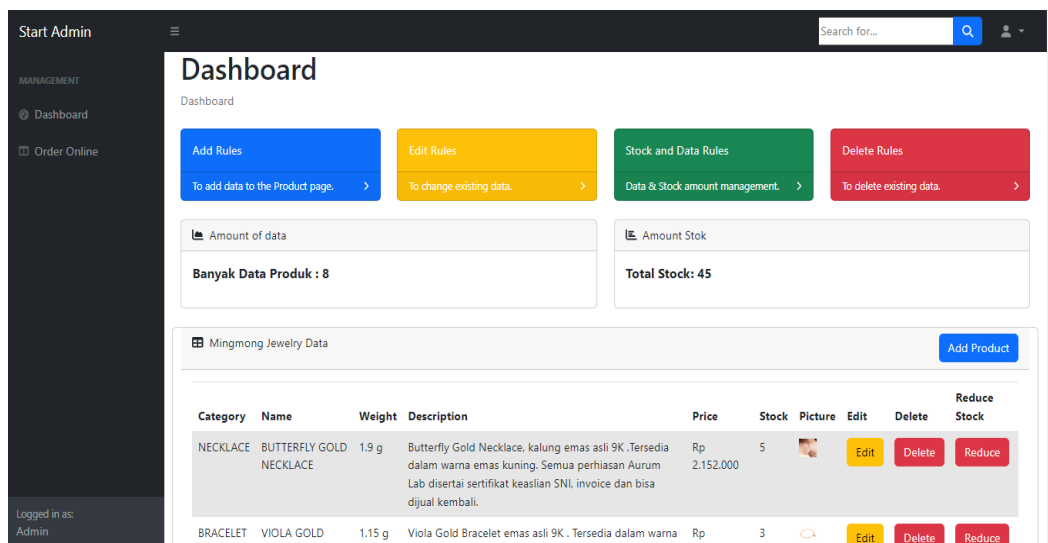
Gambar 6. Tampilan halaman penawaran produk

Gambar 6 menunjukkan tampilan halaman produk, di mana pengguna dapat menelusuri penawaran produk terbaru. Jika tertarik, pengguna dapat melakukan pemesanan dengan menekan tombol *Order Online* dan diarahkan untuk mengisi formulir pemesanan. Desain ini memudahkan pengguna dalam menavigasi produk serta melakukan pesanan secara online.

The image displays a screenshot of an online ordering form. The form is centered on a light gray background and contains several input fields: 'Product Name' (filled with 'CLARINA PEARL GOLD BRACELET'), 'Client Name' (filled with 'Tikaw'), 'Telephone' (filled with '08763211884'), and 'Amount' (filled with '2'). Below these fields is a blue 'Simpan' (Save) button. At the bottom of the form, there is a red link labeled 'Kembali ke Produk' (Return to Product).

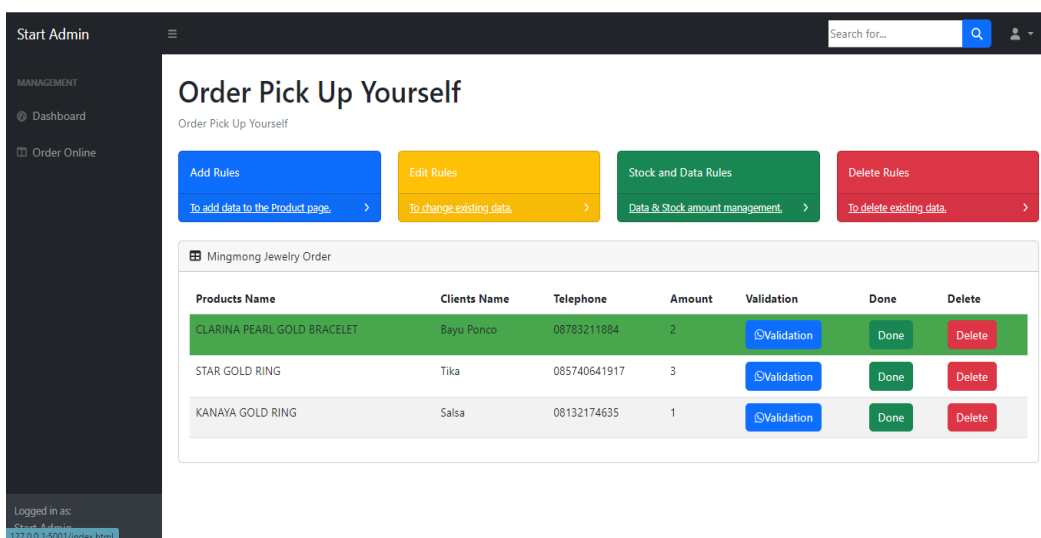
Gambar 7. Tampilan halaman form pemesanan

Gambar 7 menunjukkan tampilan halaman pemesanan, di mana pengguna diminta mengisi detail pesanan seperti produk yang ingin dipesan, nomor telepon, dan jumlah pesanan. Desain halaman ini dirancang untuk memudahkan pengguna dalam mengisi informasi pesanan, meningkatkan kejelasan, dan efisiensi proses pemesanan.



Gambar 8. Tampilan *dashboard* admin

Gambar 8 memperlihatkan tampilan *dashboard* admin, di mana admin dapat melakukan manajemen data produk dan stok. Fungsionalitas ini mencakup penambahan produk baru, pemeriksaan stok, penghapusan produk, serta pengeditan data produk. Desain halaman ini dirancang untuk memberikan admin kemudahan dan kontrol penuh dalam mengelola informasi produk dan stok.



Gambar 9. Tampilan *dashboard* admin untuk pesanan

Gambar 9 menampilkan tampilan *dashboard* admin untuk pesanan, di mana admin dapat melihat, validasi, dan mengelola pesanan. Admin dapat menghubungi pengguna melalui WhatsApp untuk validasi pembayaran. Setelah pembayaran diverifikasi, admin menekan tombol *done* dan mengurangi stok produk yang telah dibeli. Desain halaman ini memberikan admin kendali penuh dalam pengelolaan pesanan dan stok.

### C. Pengujian Sistem

Tabel 1. Tabel hasil pengujian *black box testing*

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Admin menekan <i>add product</i>	Formulir tampil lalu admin dapat mengisi formulir	Formulir terbuka dan dapat di isi	Sukses
Admin mengisi komponen-komponen yang ada pada formulir	Ketika formulir di klik simpan, maka data akan masuk di <i>dashboard</i> admin	Data tampil di halaman <i>dashboard</i> admin	Sukses
Admin menekan tombol simpan pada formulir	Data menampilkan gambar di <i>dashboard</i> dan juga data tampil di hal produk	Data dapat tampil dengan lengkap di <i>dashboard</i> dan halaman produk beserta gambarnya	Sukses
Admin menekan edit dan di klik simpan pada formulir	Formulir terbuka dan admin dapat mengedit formulir	Formulir terbuka dan dapat di isi serta hasil data dapat terbaharui	Sukses
Admin menekan <i>delete</i> data	Data dari <i>index</i> yang di klik <i>delete</i> akan terhapus	Data dari <i>index</i> yang di klik tombol <i>delete</i> dapat terhapus	Sukses
Admin selesai melakukan penambahan atau pengurangan data	<i>Amount of data &amp; Amount of product</i> menampilkan jumlah data dan jumlah <i>stock</i> sesuai data yang ada di <i>dashboard</i> .	<i>Amount of data &amp; Amount of product</i> berhasil menampilkan jumlah data dan <i>stock</i> yang ada pada <i>dashboard</i> baik itu ditambah atau dikurangi	Sukses
User menekan tombol <i>Order Online</i> di halaman produk	Formulir terbuka dan <i>user</i> dapat mengisi data yang ada.	Formulir terbuka dan dapat di isi oleh user	Sukses
User menekan simpan pada formulir	Data pesanan yang diinput <i>user</i> akan masuk ke <i>dashbaord order online</i> .	Data pesanan masuk ke <i>dashboard order online</i>	Sukses
Admin menekan tombol <i>validation</i>	Akan masuk ke halaman whatsapp yang tertuju pada nomor <i>user</i> .	Halaman whatsapp dari nomor <i>user</i> terbuka dan dapat langsung melakukan validasi pemesanan	Sukses
Admin menekan tombol <i>done</i> ketika pesanan sudah tervalidasi di whatsapp	Data pada <i>dashboard order online</i> yang ditekan tombol <i>done</i> akan berwarna hijau	Data pada <i>dashboard order online</i> berwarna hijau yang menandakan sudah di validasi	Sukses
Admin menekan <i>delete</i> data pada <i>dashboard order online</i>	Data yang ditekan tombol <i>delete</i> akan otomatis terhapus.	Data berhasil dihapus sesuai <i>index</i> yang ditekan tombol <i>delete</i>	Sukses

Admin menekan tombol <i>reduce</i> pada kolom <i>Reduce Stock</i>	Muncul <i>pop up</i> untuk memasukan jumlah <i>stock</i> yang sudah divalidasi	Muncul <i>pop up</i> dan dapat di isi	Sukses
Admin memasukan jumlah <i>stock</i> yang sudah di validasi	Jumlah <i>stock</i> pada <i>dashboard</i> dan <i>Amount Stock</i> akan berkurang.	<i>Stock</i> produk berkurang sesuai dengan jumlah <i>stock</i> yang diinputkan pada <i>pop up</i>	Sukses
Admin menekan tombol <i>Log Out</i>	Admin akan keluar dari halaman <i>dashboard</i> menuju ke halaman <i>home</i>	Admin berhasil <i>logout</i> dan menuju ke halaman <i>home</i>	Sukses

Hasil pengujian blackbox menunjukkan bahwa aplikasi yang telah dibangun berhasil memenuhi tujuan PT XYZ dalam mengatasi tantangan pemasaran online. Dengan mengintegrasikan pemasaran, pemesanan, dan pengelolaan stok produk, aplikasi memberikan solusi efisien untuk promosi produk, memudahkan pelanggan dalam mencari dan memesan produk online, serta meningkatkan organisasi manajemen data secara efisien.

## V. SIMPULAN DAN SARAN

Penelitian ini merancang aplikasi web manajemen data produk berbasis Flask dan MongoDB untuk PT XYZ dalam industri perhiasan. Fokusnya adalah meningkatkan efisiensi pemasaran, pemesanan, dan manajemen stok produk. Meskipun terbatas pada PT XYZ dan tiga fungsi utama, penelitian ini memberikan kontribusi signifikan dengan potensi pengembangan untuk industri perhiasan secara umum. Metode penelitian, melibatkan analisis kebutuhan, pembangunan prototipe, dan pengujian sistem, berhasil mencapai tujuannya. Hasil penelitian dapat menjadi panduan bagi perusahaan sejenis untuk meningkatkan operasional dan beradaptasi dengan tren pemasaran modern menggunakan Flask dan MongoDB.

Dalam merancang aplikasi web manajemen data produk bisnis perhiasan berbasis Flask dan MongoDB, ditemukan potensi pengembangan lebih lanjut. Sebagai saran, perangkat lunak tersebut masih perlu memperkuat fungsi pengecekan kesalahan dengan implementasi sistem log yang *komprehensif*. Penambahan fitur ini akan memberikan kemudahan bagi pengguna dan pengembang dalam melacak dan menangani potensi kesalahan, serta meningkatkan efisiensi perawatan dan perbaikan perangkat lunak. Tidak hanya itu saja perangkat lunak juga masih belum ada manajemen transaksi, sehingga hal itu dapat di kembangkan lagi. Karena dengan adanya manajemen transaksi maka akan ada fitur yang dapat menghasilkan *invoice* yang dapat digunakan sebagai bukti pemesanan.

## DAFTAR PUSTAKA

- [1] Gede, P., Cipta Nugraha, S., Putu, I., Indrawan, Y., Kadek, I., & Asmarajaya, A. Rancang Bangun Sistem Informasi e-Commerce Berbasis Website. *INSERT: Information System and Emerging Technology Journal*, 3(1), 2022.

- [2] Khairil Ahsyar, T., P. E-Marketplace Media Pengembangan Promosi Usaha Mikro Kecil dan Menengah Dinas Koperasi UMKM Kota Pekanbaru. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 6(1) 2020, 43–54.
- [3] Eko Pratama, A., Junaidi, A., Rudianto, R., & Yunita, Y. (2020). Cross Platform Aplikasi Manajemen Data Karyawan pada PT Maxindo Mitra Solusi. *Journal of Computer Science and Engineering (JCSE)*, 1(1), 2020, 52–61, <https://doi.org/10.36596/jcse.v1i1.16>
- [4] Mufid, M., Basofi, A., Rasyid, M., Rochimansyah, I., & rokhim, A. Design an MVC Model using Python for Flask Framework Development. *International Electronics Symposium (IES)*, 27-28 September 2019 <https://doi.org/10.1109/ELECSYM.2019.8901656>
- [5] Rama Vyshnavi, V., & Malik, A. Efficient Way of Web Development Using Python and Flask. In *International Journal of Recent Research Aspects* (Vol. 6), 2019.
- [6] Budianto, A. J., Ocsa, P., & Saian, N. Pengembangan Modul Inventory Management pada Aplikasi Master Distribution Centre System Menggunakan Framework Flask di PT XYZ. *Jurnal Teknologi Informasi Dan Komunikasi*, 7(2), 2023. <https://doi.org/10.35870/jti>
- [7] Wijaya, A. Penerapan Konsep NoSQL Pada Perangkat Lunak Manajemen Properti. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 3(6), 2023, 823–835. <https://doi.org/10.30865/klik.v3i6.913>
- [8] Cahyani, A. D., Basuki, A., & Nafisah, D. *Pengolahan Basis Data Mongo DB*. Media Nusa Creative, 2022.
- [9] Handrianto, Y., & Sanjaya, B. Model Waterfall Dalam Rancang Bangun Sistem Informasi Pemesanan Produk Dan Outlet Berbasis Web, *Jurnal Inovasi Informatika*, Vol. 5 No. 2, 2020.
- [10] Manis, R., Setyaningsih, W., & Kuswinardi, W. Rancang Bangun Sistem Informasi Penjualan Laptop Berbasis Web dengan Metode Waterfall. *Jurnal Terapan Sains & Teknologi*, vol 3, nomor 3, 2021.
- [11] Alison Kwa, S., Ardhyanti Mita Nugraha, J., & Adi Purnomo Sidhi, T.. Pembangunan Aplikasi Web Sirkulasi Buku Perpustakaan Universitas Atma Jaya Yogyakarta, *Jurnal Informatika Atma Jogja*, Volume 4, Nomor 1, Mei 2023: 37-44.
- [12] Yuniarti, R., Santi, I. H., & Puspitasari, W. D. Peresanan Bahan Berbasis Framework Lavarel. *Jurnal Mahasiswa Teknik Informatika*, Vol. 6, Issue 1, 2022.
- [13] Risyda, F., Nuryamin, Y, et al. Perancangan Sistem Informasi Manajemen Invoice Menggunakan Generator Freamwork Django-Python Berbasis Website Pada Pt. Lampuind Tekno Elektrik, *JSI: Jurnal Sistem Informasi*, Vol 10, No 1 (2023).
- [14] Ningtyas, D. F., & Setiyawati, N. (2021). Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request. *Jurnal Janitra Informatika dan Sistem Informasi*, 1(1), 2021, 19–34.

- [15] Frederica, R. R., et al. Perancangan Web Design Aplikasi E-Learning dengan Metode Prototype pada Tingkat SMA. *Majalah Ilmiah UPI YPTK*, 2021, 13–18. <https://doi.org/10.35134/jmi.v28i1.62>