

# PERANCANGAN WEBSITE COMMERCE TOKO PESONA BANGUNAN

Justin Hensel<sup>1)</sup>, Darius Andana Haris<sup>2)</sup>

<sup>1)</sup>Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jl. Letjen S.Parman St No. 1, RT. 6 / RW. 16, Tomang, Grogol Pertamburan, Jakarta Barat, DKI Jakarta 11440  
email: [justinhensel.825210127@stu.untar.ac.id](mailto:justinhensel.825210127@stu.untar.ac.id)

<sup>2)</sup>Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jl. Letjen S.Parman St No. 1, RT. 6 / RW. 16, Tomang, Grogol Pertamburan, Jakarta Barat, DKI Jakarta 11440  
email: [dariush@fti.untar.ac.id](mailto:dariush@fti.untar.ac.id)

## ABSTRACT

*Proyek ini merancang dan mengembangkan website e-commerce untuk Toko Pesona Bangunan, yang fokus pada bahan dan barang keperluan property. Pesona Bangunan menghadapi tantangan dalam menjangkau konsumen di pasar yang didominasi oleh Sandimas dan Roman. Website ini bertujuan untuk memperluas jangkauan pasar, meningkatkan penjualan, dan memperkuat identitas toko. Desain website yang menarik dan ramah pengguna diharapkan menciptakan pengalaman berbelanja yang personal dan nyaman bagi konsumen. Pengembangan website menggunakan model SDLC Waterfall, yang meliputi tahap analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Fitur utama yang disediakan antara lain tampilan koleksi produk, sistem pemesanan, wishlist, serta halaman admin untuk pengelolaan produk dan pesanan. Pengujian dilakukan dengan metode user acceptance testing (UAT) untuk memastikan fungsionalitas sesuai kebutuhan. Hasil pengembangan menunjukkan bahwa website e-commerce ini berhasil meningkatkan efisiensi dalam transaksi dan pengelolaan produk. Website ini diharapkan dapat membantu Pesona Bangunan bersaing lebih baik di pasar dan memperkuat hubungan dengan pelanggan melalui pengalaman belanja yang eksklusif dan menarik.*

## Key words

*Kata kunci: E-commerce, Website, Waterfall*

## 1. Pendahuluan

Di Indonesia, e-commerce mengalami pertumbuhan pesat seiring dengan meningkatnya akses internet di kalangan masyarakat. Menurut data dari beberapa studi, pertumbuhan ini tidak hanya memberikan peluang besar bagi bisnis, tetapi juga menghadirkan tantangan baru dalam persaingan pasar yang semakin ketat [1]. Strategi ini memang menonjolkan keunikan dan eksklusivitas, namun di sisi lain juga menjadi tantangan dalam menjangkau konsumen yang lebih luas. Salah satu solusi yang potensial untuk tantangan ini adalah dengan

merancang *web e-commerce* yang mampu merepresentasikan keunikan dan nilai eksklusivitas. Selain itu, penelitian menunjukkan bahwa tampilan visual yang menarik dan fungsionalitas yang baik pada *web e-commerce* dapat meningkatkan kepuasan pengguna dan, pada akhirnya, meningkatkan penjualan [2]. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengembangkan sebuah website e-commerce bagi Toko Pesona Bangunan yang dapat membantu memperluas jangkauan pasar, meningkatkan penjualan, serta memperkuat identitas brand di tengah persaingan yang ketat.

Perkembangan teknologi digital telah mengubah banyak aspek kehidupan, salah satunya adalah pola belanja masyarakat yang kini beralih ke platform daring atau e-commerce. Di Indonesia, e-commerce mengalami pertumbuhan pesat seiring dengan semakin luasnya akses internet dan adopsi teknologi oleh masyarakat. Penetrasi internet yang semakin meningkat di Indonesia secara langsung berkontribusi terhadap pertumbuhan e-commerce di negara ini. Kondisi ini menciptakan peluang besar bagi bisnis untuk memperluas pasar dan mengoptimalkan penjualan melalui platform digital [3]. Pertumbuhan pesat e-commerce juga memicu persaingan yang ketat antar pelaku bisnis. Banyak perusahaan yang berlomba-lomba menawarkan produk serupa, sehingga menonjolkan keunikan dan diferensiasi menjadi faktor penting dalam mempertahankan eksistensi di pasar menunjukkan bahwa keunikan produk dan eksklusivitas brand menjadi strategi yang efektif untuk menarik perhatian konsumen di tengah pasar yang kompetitif. Namun, tantangan muncul ketika perusahaan berusaha menjangkau konsumen yang lebih luas tanpa kehilangan identitas brand yang eksklusif[4].

Selain keunikan, tampilan visual dan fungsionalitas pada website e-commerce juga berperan penting dalam menarik minat pengguna. Penelitian menemukan bahwa desain antarmuka yang menarik serta navigasi yang intuitif dapat meningkatkan pengalaman pengguna saat mengakses situs e-commerce. Faktor-faktor ini, seperti estetika visual dan kemudahan dalam melakukan transaksi, terbukti memiliki korelasi positif dengan

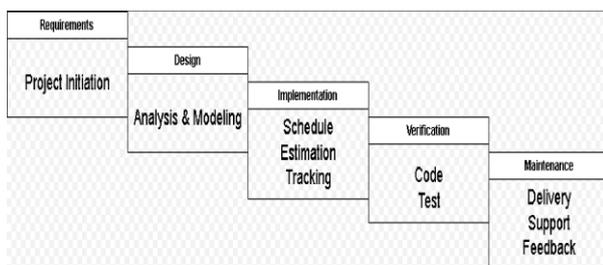
kepuasan pengguna, yang pada akhirnya dapat berdampak pada peningkatan loyalitas dan penjualan [5].

Mengacu pada hal tersebut, penelitian ini bertujuan untuk merancang dan mengembangkan website *e-commerce* bagi Toko Pesona Bangunan. Toko ini ingin memperluas jangkauan pasar, meningkatkan penjualan, serta memperkuat identitas brand di tengah persaingan yang semakin ketat. Melalui pendekatan ini, diharapkan website yang dirancang dapat mewakili nilai-nilai eksklusivitas Toko Pesona Bangunan, memberikan pengalaman belanja yang menyenangkan, serta memudahkan konsumen dalam mengakses produk yang ditawarkan.

## 2. Metode Penelitian

### 2.1 Metode Waterfall

Metode yang digunakan dalam proses perancangan *web e-commerce* ini menggunakan *waterfall*, model ini melibatkan proses bertahap yang dimulai dari perencanaan sistem hingga pemeliharaannya. Model *waterfall* menawarkan pendekatan pengembangan perangkat lunak yang sangat terstruktur dengan tahapan yang jelas, seperti *requirements, design, implementation, verification, dan maintenance*. Struktur ini membantu mengatur proyek secara efisien, dimana setiap tahap memiliki batasan waktu dan tanggung jawab yang jelas [6]. Berikut merupakan tahapan-tahapan dari metode *waterfall* untuk perancangan *web e-commerce* ini seperti yang tertera pada **Gambar 1**:



**Gambar 1.** Metode Waterfall

#### 1. Requirements

Langkah awal dalam metode *waterfall* adalah melakukan identifikasi masalah yang ada dan menetapkan kebutuhan yang harus dipenuhi oleh sistem yang dibangun. Tahapan ini dimulai dengan menentukan permasalahan lalu mengumpulkan informasi. Informasi ini yang akan membantu dalam menentukan kebutuhan spesifik yang harus dipenuhi oleh sistem yang akan dikembangkan [6].

#### 2. Design

Langkah kedua dalam metode *waterfall* adalah melakukan perancangan. Perancangan yang dibuat, yaitu alur sistem, desain *database*, dan desain antarmuka [6].

#### 3. Implementation

Langkah ketiga dalam metode *waterfall* adalah mengimplementasikan yang merupakan bagian pengkodean. Pada tahap ini, desain basis data dan desain antarmuka mulai dibangun dengan menggunakan bahasa pemrograman [6].

#### 4. Verification

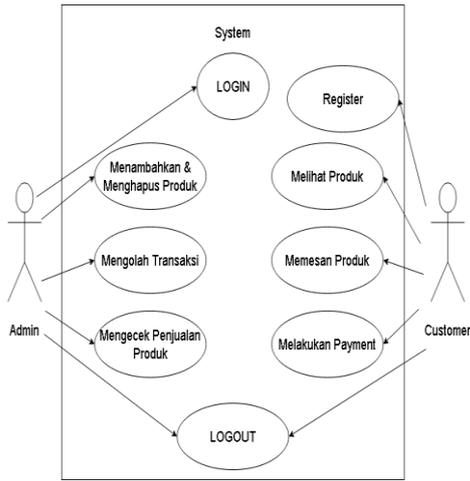
Langkah keempat dalam metode *waterfall* adalah pengujian. Ditahap ini sistem yang telah dirancang akan diuji coba untuk menguji kelayakan sistem untuk menghindari terjadinya kesalahan sistem [7].

#### 5. Maintenance

Langkah kelima atau langkah terakhir dalam metode *waterfall* adalah pemeliharaan sistem. Tahap dari pemeliharaan ini adalah untuk memastikan bahwa sistem telah dirancang secara optimal untuk memenuhi kebutuhan pengguna. Tujuan dari pemeliharaan ini adalah untuk memperbaiki kesalahan, memperbaiki implementasi sistem, dan meningkatkan kinerja sistem [7].

### 2.2 Use Case Diagram

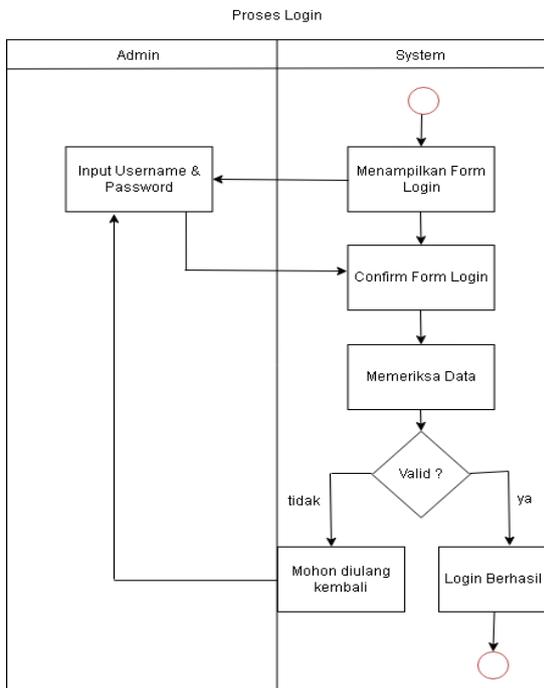
*Use Case Diagram* adalah representasi visual dari interaksi antara aktor dan sistem dalam skenario tertentu. Diagram ini menampilkan berbagai *use case* yang menunjukkan bagaimana aktor berinteraksi dengan sistem untuk mencapai tujuan tertentu. *Use case* pada sistem yang dibuat adalah admin dan user, **Gambar 2** merupakan perancangan *use case diagram web e-commerce Toko Pesona Bangunan*.



Gambar 2. Use Case Diagram

2.3 Activity Diagram

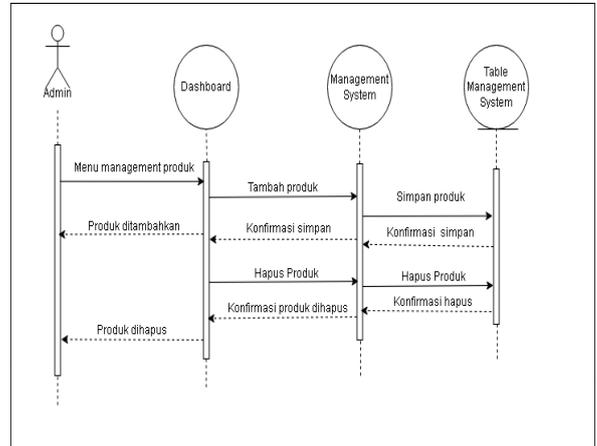
Activity diagram adalah rancangan aliran aktivitas yang digunakan pada sebuah sistem yang dijalankan. Activity diagram juga digunakan untuk mendefinisikan alur dari tampilan suatu sistem. Gambar 3 merupakan perancangan activity diagram web e-commerce Toko Pesona Bangunan.



Gambar 3. Activity Diagram

2.4 Sequence Diagram

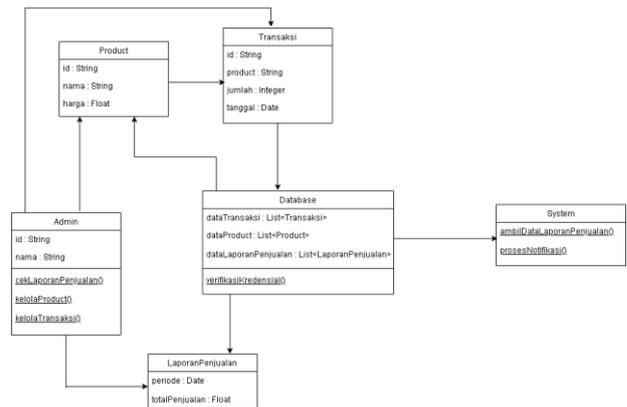
Sequence diagram adalah diagram interaksi yang menggambarkan urutan waktu dari pesan atau interaksi yang terjadi antara objek-objek di dalam sistem. Gambar 4 merupakan perancangan sequence diagram web e-commerce Toko Pesona Bangunan.



Gambar 4. Sequence Diagram

2.5 Class Diagram

Class diagram adalah diagram yang menggambarkan struktur statis dari sistem. Class diagram digunakan untuk mendefinisikan atribut dan hubungan antar kelas dalam sistem berbasis objek. Gambar 5 merupakan perancangan class diagram web e-commerce Toko Pesona Bangunan.



Gambar 5. Class Diagram

3. Hasil dan Pembahasan

3.1 Implementasi Sistem dan Jadwal

Pada bulan Agustus, dimulai proses perencanaan program aplikasi. Tahapan ini melibatkan identifikasi kebutuhan dan tujuan aplikasi, serta pembuatan rencana umum. Setelah menyelesaikan tahap perencanaan, fokus

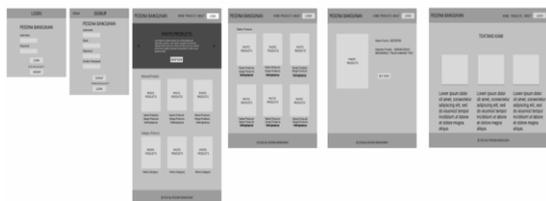
berpindah ke perancangan aplikasi yang melibatkan pembuatan use case diagram, activity diagram, class diagram, dan elemen desain lainnya.

No	Bulan	Pekerjaan
1	Agustus	Perencanaan dan perancangan
2	September	Perancangan dan membuat tampilan design
3	Oktober	Melakukan pengumpulan data dan mulai proses membuat kode
4	November	Melakukan uji coba pada program aplikasi

Gambar 6. Implementasi Sistem dan Jadwal

### 3.2 Perancangan Tampilan Design

Pada bulan September, langkah selanjutnya adalah melanjutkan perancangan yang belum selesai dan menghasilkan desain tampilan yang memenuhi kebutuhan pengguna. Pada tahap ini, penting untuk memastikan keselarasan antara aspek fungsional dan antarmuka pengguna aplikasi. Berikut ini adalah hasil dari design rancangan awal menggunakan figma.



Gambar 7. Rancangan Design

Beralih ke bulan Oktober, fokus berubah ke pengumpulan data yang diperlukan dan implementasi

kode. Proses ini mencakup penulisan kode program serta pengintegrasian berbagai komponen untuk memastikan aplikasi dapat berjalan sesuai rencana pada Gambar 8 – 15.

App.js merupakan komponen utama dalam mengontrol struktur utama aplikasi dan mengelola berbagai fungsi dan data penting yang dibutuhkan di seluruh aplikasi. Secara singkat App.js bertindak sebagai pusat manajemen alur dan data untuk aplikasi e-commerce berbasis React, mengatur navigasi, manajemen keranjang belanja, dan tampilan halaman utama.

Dalam Gambar 8 dibawah ini App.js berfungsi sebagai komponen utama aplikasi e-commerce, mengelola state keranjang belanja, mengatur fungsi penambahan produk ke keranjang, dan merender elemen-elemen UI utama aplikasi, termasuk navigasi, detail produk, beranda, dan footer.

```

App.js x productdetail.js reportWebVitals.js home.js homeproduct.js index.js
eCommerce_website > src > App.js | default
7  const App = () => {
8
9  }
10 const addtocart = (product) => {
11   setCart([...cart, {...product, qty:1}])
12   alert("Produk berhasil ditambahkan")
13 }
14
15
16
17 console.log(cart)
18
19 return (
20   <div>
21     <BrowserRouter>
22       <div searchbtn={searchbtn}/>
23       <div product={product} setProduct={setProduct} detail={detail} view={view} close={close} setClose={setClose} cart={ca
24     </div>
25     </BrowserRouter>
26   </div>
27 )
28
29 export default App
  
```

Gambar 8. Code Program App.js

Cart.js merupakan komponen dalam aplikasi e-commerce berbasis React yang bertanggung jawab untuk mengelola dan menampilkan keranjang belanja. Secara singkat Cart.js adalah komponen yang mengelola fungsi dan tampilan keranjang belanja dalam aplikasi e-commerce. Ini merupakan tempat di mana pengguna dapat meninjau dan mengelola produk yang ingin mereka beli, serta mempersiapkan diri sebelum melanjutkan ke tahap pembayaran.

Dalam Gambar 9 dibawah ini Cart.js berfungsi sebagai menambah atau mengurangi jumlah item yang ingin dibeli., menghapus item dari keranjang., melihat ringkasan produk yang dipilih sebelum melanjutkan ke tahap pembayaran.

```

1 import React from 'react'
2 import { AiOutlineClose } from 'react-icons/ai';
3 import { Link } from 'react-router-dom';
4 import { Cart } from './cart';
5 const Cart = ({ cart, setCart }) => {
6   //decrease qty
7   const incQty = (product) => {
8     const exist = cart.find(x =>
9       x.id === product.id
10      );
11     return x.id === product.id
12     ? setCart(cart.map((curEle) => {
13       return curEle.id === product.id ? {...exist, qty: exist.qty + 1} : curEle
14     })) : null;
15   }
16   //increase qty
17   const decQty = (product) => {
18     const exist = cart.find(x =>
19       x.id === product.id
20      );
21     return x.id === product.id
22     ? setCart(cart.map((curEle) => {
23       return curEle.id === product.id ? {...exist, qty: exist.qty - 1} : curEle
24     })) : null;
25   }
26 }
27 export default Footer;

```

Gambar 9. Code Program cart.js

Contact.js memainkan peran penting dalam menyediakan fungsionalitas inti untuk halaman kontak pada sebuah website *e-commerce*.

Dalam Gambar 10 Contact.js merupakan sebuah file JavaScript yang umumnya digunakan untuk mengelola logika dan interaksi pada halaman kontak di sebuah website.

```

1 import React, { useState } from 'react'
2 import { Formik, Field } from 'formik';
3 const Contact = () => {
4   const [user, setUser] = useState({
5     name: '', email: '', keterangan: '', message: ''
6   });
7   const [data, setData] = useState({});
8   const handleSubmit = (e) => {
9     e.preventDefault();
10    const { name, value } = e.target;
11    const { email, value: emailValue } = e.target;
12    const { message, value: messageValue } = e.target;
13    const { keterangan, value: keteranganValue } = e.target;
14    setUser({ name, email: emailValue, message: messageValue, keterangan: keteranganValue });
15    return { name, email, message, keterangan };
16  };
17  return (
18    <div className="contact">
19      <div className="form">
20        <input type="text" name="name" value={user.name} placeholder="Masukan Nama disini" required autoComplete="name" />
21        <input type="text" name="email" value={user.email} placeholder="Masukan email disini" autoComplete="email" />
22        <input type="text" name="keterangan" value={user.keterangan} placeholder="Masukan keterangan disini" autoComplete="text" />
23        <input type="text" name="message" value={user.message} placeholder="Masukan pesan" autoComplete="off" onChange={data} />
24        <button type="submit" value="Kirim" />
25      </div>
26    </div>
27  );
28 }
29 export default Contact;

```

Gambar 10. Code Program contact.js

Footer.js sebuah file JavaScript yang umumnya digunakan untuk mengelola tampilan dan fungsionalitas bagian footer pada sebuah website, termasuk website *e-commerce*.

Dalam Gambar 11 Footer.js komponen penting dalam pengembangan website *e-commerce* dengan merancang footer yang informatif dan menarik.

```

1 import React from 'react'
2 import { RiFacebookFill } from 'react-icons/ri';
3 import { AiOutlineInstagram } from 'react-icons/ai';
4 import { AiOutlineTwitter } from 'react-icons/ai';
5 import { BsYoutube } from 'react-icons/bs';
6 const Footer = () => {
7   return (
8     <div className="Footer">
9       <div className="container">
10        <div className="social">
11          <div className="logo">
12            
13          </div>
14          <div className="detail">
15            <p>PT188 Pesma Bangunan, Pekayon, Bekasi, Jawa Barat</p>
16          </div>
17          <div className="icon">
18            <span><RiFacebookFill /></span>
19            <span><AiOutlineInstagram /></span>
20            <span><AiOutlineTwitter /></span>
21            <span><BsYoutube /></span>
22          </div>
23        </div>
24      </div>
25    </div>
26  );
27 }
28 export default Footer;

```

Gambar 11. Code Program footer.js

Home.js komponen utama yang bertanggung jawab untuk menampilkan halaman beranda (homepage) pada sebuah aplikasi *e-commerce* yang dibangun menggunakan React.

Dalam Gambar 12 Home.js komponen inti dalam aplikasi *e-commerce* yang bertanggung jawab untuk menyajikan tampilan awal yang menarik dan informatif bagi pengguna.

```

1 import React from 'react'
2 import { Link } from 'react-router-dom'
3 import { BsArrowRight } from 'react-icons/bs';
4 import { AiTrack } from 'react-icons/ai';
5 import { BsCurrencyDollar } from 'react-icons/bs';
6 import { CiPercent } from 'react-icons/ci';
7 import { BiShoppingBag } from 'react-icons/bi';
8 import { AiOutlineShoppingCart, AiOutlineCloseCircle } from 'react-icons/ai';
9 import { AiEdge } from 'react-icons/ai';
10 import { AiOutlineShare } from 'react-icons/ai';
11 import HomeProduct from './homeProduct';
12 import './home.css';
13 const Home = ({ detail, view, close, setClose, addtoCart }) => {
14   return (
15     <div className="home">
16       <div className="product_detail">
17         <div className="container">
18           <div className="close">
19             <button onClick={() => setClose(true)} className="close"><AiOutlineCloseCircle /></button>
20           </div>
21           <div className="detail">
22             <div className="product">
23               <img src={detail.img} alt={detail.title} />
24             </div>
25           </div>
26         </div>
27       </div>
28     </div>
29   );
30 }
31 export default Home;

```

Gambar 12. Code Program home.js

Index.js file JavaScript yang sangat penting dalam sebuah aplikasi React, termasuk aplikasi *e-commerce*. File ini seringkali bertindak sebagai titik masuk utama aplikasi dan memiliki peran krusial dalam menginisialisasi aplikasi, menghubungkan komponen, dan mengelola state global.

Dalam Gambar 13 Index.js file yang sering digunakan sebagai titik awal untuk memulai aplikasi atau modul, mengimpor, dan mengeksplor bagian penting dari kode program.

```

1 import React from 'react'
2 import ReactDOM from 'react-dom/client'
3 import App from './App';
4 import reportWebVitals from './reportWebVitals';
5 import { AuthProvider } from './auth/auth';
6
7 const root = ReactDOM.createRoot(document.getElementById('root'));
8 root.render(
9   <AuthProvider>
10     <App />
11   </AuthProvider>
12 );
13
14 // If you want to start measuring performance in your app, pass a function
15 // to log results (for example: reportWebVitals(console.log))
16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17 reportWebVitals();

```

Gambar 13. Code Program index.js

Nav.js merupakan konteks pengembangan website, khususnya *e-commerce*, biasanya merupakan file JavaScript yang bertanggung jawab untuk mengelola tampilan dan fungsionalitas dari navigasi atau navbar pada sebuah halaman website.

Dalam **Gambar 14** Nav.js menangani elemen-elemen navigasi dalam aplikasi, baik itu berupa menu dropdown, sidebar, atau menu utama yang dapat diubah sesuai dengan status pengguna dan perangkat yang digunakan.

```

ecommerce_website > src > # nav.js > 1
1 import React, { useState } from 'react'
2 import { FontAwesomeIcon } from 'react-icons/fa'
3 import { AiOutlineHeart } from 'react-icons/ai'
4 import { RiBagCheck } from 'react-icons/ri'
5 import { AiOutlineHome } from 'react-icons/ai'
6 import { CiLogin } from 'react-icons/ci'
7 import { CiLogout } from 'react-icons/ci'
8 import { Link } from 'react-router-dom'
9 import { useAuth } from 'auth/auth-react'
10
11 const Nav = ({ searchBtn }) => {
12   const [search, setSearch] = useState('')
13   const { isLoggedIn, logout, user, isAuthenticated } = useAuth()
14   return (
15     <div className="flex">
16       <div className="icon">
17         <FontAwesomeIcon />
18       </div>
19       <div>
20         <p>Free Shipping Pembelian diatas Rp500.000</p>
21       </div>
22       <div className="main_header">
23         <div className="container">
24           <div className="logo">
25             
26           </div>
27           <div className="search_bar">
28             <input type="text" value={search} placeholder="Masukkan Nama Produk Yang Dicari" autoComplete="off" onChange={setSearch} />
29           </div>
30         </div>
31       </div>
32     </div>
33   )
34 }
35
36 export default Nav
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

**Gambar 14.** Code Program nav.js

Product.js merupakan file JavaScript yang umumnya digunakan untuk mengelola tampilan dan logika terkait produk dalam sebuah aplikasi *e-commerce*.

Dalam **Gambar 15** Product.js komponen penting dalam aplikasi *e-commerce* yang bertanggung jawab untuk menampilkan dan mengelola informasi produk. Komponen ini berinteraksi dengan pengguna dan juga dengan komponen lain seperti keranjang belanja, halaman kategori, dan halaman pencarian.

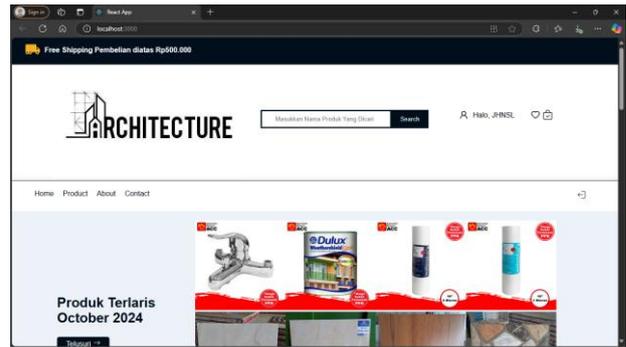
```

ecommerce_website > src > # product.js > 1
1 import React from 'react'
2 import ProductDetail from './productDetail'
3 import { AiOutlineShoppingCart } from 'react-icons/ai'
4 import { BiBag } from 'react-icons/bi'
5 import { AiOutlineHeart, AiOutlineCloseCircle } from 'react-icons/ai'
6 import { useState } from 'react'
7 import './product.css'
8
9 const Product = ({ product, setProduct, detail, view, close, addtoCart }) => {
10
11   const { isLoggedIn, isAuthenticated } = useAuth()
12   const filterProduct = (product) => {
13     const update = productDetail.filter(x => {
14       return x.Cat === product;
15     })
16     setProduct(update);
17   }
18   const AllProducts = () => {
19     setProduct(ProductDetail)
20   }
21   return (
22     <div>
23       <ProductDetail />
24     </div>
25   )
26 }
27
28 export default Product
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

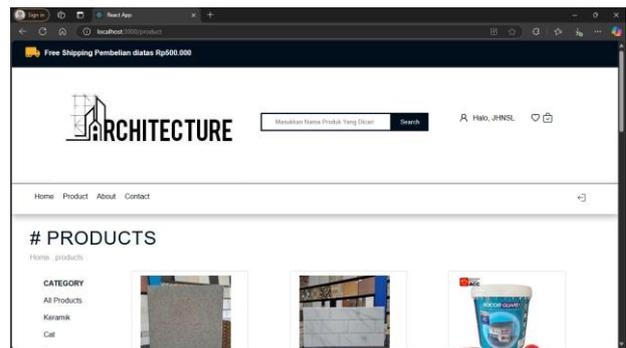
```

**Gambar 15.** Code Program product.js

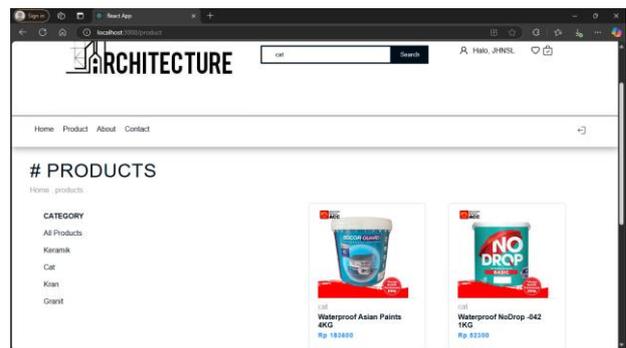
Pada Bulan November, dilakukan uji coba secara berkelanjutan untuk mendeteksi dan memperbaiki potensi masalah. Uji coba ini mencakup pengujian fungsionalitas, keamanan, dan kinerja aplikasi. Langkah ini penting untuk memastikan bahwa program aplikasi dapat beroperasi dengan lancar dan dapat diandalkan tanpa adanya masalah yang signifikan. Dapat dilihat pada **Gambar 16 - 22** proses pembuatan program *web e-commerce Toko Pesona Bangunan*. Berikut ini tampilan yang dihasilkan dengan menggunakan *code program* React dan Node.js



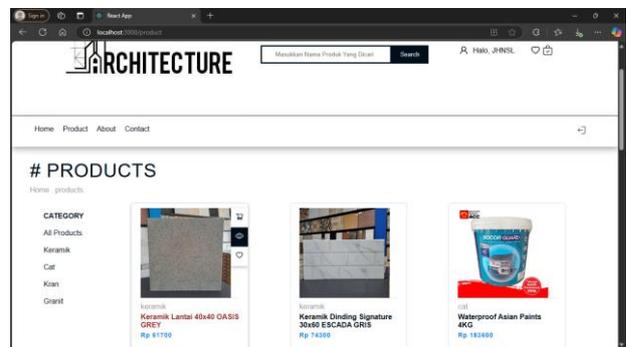
**Gambar 16.** Tampilan Home Page



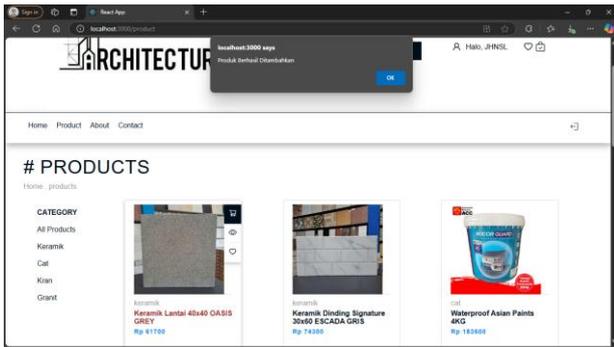
**Gambar 17.** Tampilan Product Page



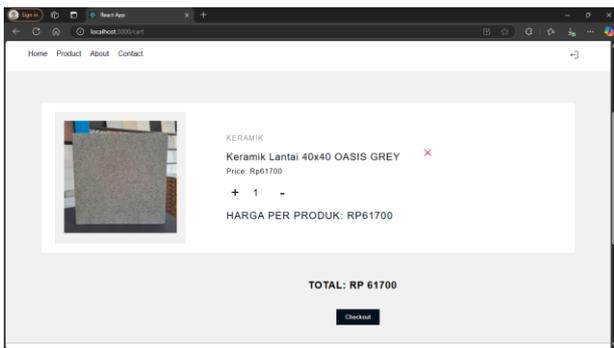
**Gambar 18.** Tampilan Fitur Search Bar Product



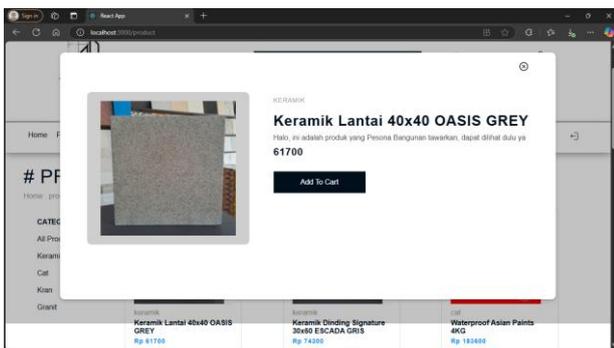
**Gambar 19.** Tampilan Sidebar Category Product



Gambar 20. Tampilan Fitur Add To Cart



Gambar 21. Tampilan Cart Order



Gambar 22. Tampilan Deskripsi Product

#### 4. Kesimpulan

Penelitian ini bertujuan untuk merancang dan mengembangkan website *e-commerce* bagi Toko Pesona Bangunan, yang berfokus pada produk bahan bangunan dan kebutuhan properti. Dengan metode pengembangan *Software Development Life Cycle* (SDLC) model Waterfall, proyek ini melalui beberapa tahapan, yaitu analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Website ini dilengkapi dengan fitur-fitur utama seperti tampilan produk, sistem pemesanan, wishlist, dan halaman admin untuk pengelolaan produk dan pesanan.

Penulis mengembangkan website ini menggunakan React dan Node.js dimana React digunakan untuk membangun antarmuka pengguna yang interaktif dan

responsif. Keunggulannya dalam *component-based architecture* memudahkan pengelolaan dan pengembangan antarmuka yang modular serta dapat diubah sesuai kebutuhan. React juga mendukung pengalaman pengguna yang cepat dan dinamis, yang sangat penting dalam menarik dan mempertahankan minat konsumen di platform *e-commerce*.

Node.js juga penting dalam sisi server aplikasi ini, memungkinkan pemrosesan data yang cepat dan real-time untuk manajemen produk, pemesanan, dan pengolahan data lainnya. Dengan lingkungan JavaScript yang menyatukan *frontend* (React) dan *backend* (Node.js), pengembangan menjadi lebih efisien, karena keduanya dapat berbagi bahasa pemrograman yang sama.

#### REFERENSI

- [1] A. A. A. P. Y. Cahyadi, "The Impact of E-Commerce Adoption on Indonesian Industry Performance. Journal of Digital Commerce," *Journal of Digital Commerce*, pp. 102-117, 2020.
- [2] S. S. M. & D. A. Hermawan, "Web Technology and User Experience: How Visual and Functional Elements Affect E-Commerce Success," *Journal of Information Systems*, pp. 211-228, 2021.
- [3] S. Sahputra, L. Mursyida and S. , "Pengembangan Platform Daring untuk Penjadwalan Mata Pelajaran dan Pengelolaan Nilai dai MAN 1 Padang dengan Metode Waterfall," *Journal of Hypermedia & Technology-Enhanced Learning*, vol. 2, pp. 137-152, 2014.
- [4] A. Utami, D. Aldo, Y. S. R. Nur and T. Yuniati, "Web based Yogyakarta food recipe application using sdlc waterfall method," *Jurnal Mantik*, pp. 1-9, 2023.
- [5] Shodiq, M., & Wibowo, P. (2020). *Pengaruh Penetrasi Internet terhadap Pertumbuhan E-commerce di Indonesia*. Jurnal Ekonomi Digital, 12(1), 45-55.
- [6] Rahayu, D., & Kurniawan, A. (2019). *Eksklusivitas Brand sebagai Strategi Diferensiasi dalam E-commerce*. Jurnal Manajemen Pemasaran, 11(3), 233-242.
- [7] Prasetyo, Y., & Hartono, S. (2021). *Pengaruh Desain Antarmuka dan Fungsionalitas Website terhadap Kepuasan Pengguna E-commerce*. Jurnal Sistem Informasi dan Teknologi, 14(2), 87-95.

**Justin Hensel**, Mahasiswa Program Studi Sistem Informasi Fakultas Teknologi Informasi Universitas Tarumanagara sejak 2021.

**Darius Andana Haris**, memperoleh gelar S.Kom pada tahun 2009 dari Universitas Tarumanagara, M.T.I pada tahun 2012 dari Universitas Bina Nusantara, Saat ini sebagai staff pengajar di Fakultas Teknologi Informasi Universitas Tarumanagara.