



1. BUKTI SUBMIT JURNAL

#2019 Summary

[SUMMARY](#)[REVIEW](#)[EDITING](#)

Submission

Authors	Zarkasih Akhmad, Muhamad Azrino Gustalika
Title	Rancang Bangun E-Commerce Berbasis Single Page Application (SPA) Menggunakan ReactJS
Original file	2019-6166-1-SM.DOCX 2022-08-30
Supp. files	None
Submitter	Bapak Muhamad Azrino Gustalika 
Date submitted	August 30, 2022 - 12:54 AM
Section	Articles
Editor	Kuntum Imania 
Abstract Views	340

Status

Status	Published Vol 8, No 2 (2022): Petik vol 8 no 2 tahun 2022
Initiated	2022-09-29
Last modified	2022-09-30

Submission Metadata

Authors

Name	Zarkasih Akhmad 
Affiliation	Institut Teknologi Telkom Purwokerto
Country	Indonesia
Bio Statement	—
Name	Muhamad Azrino Gustalika 
Affiliation	Institut Teknologi Telkom Purwokerto
Country	Indonesia
Bio Statement	—

Principal contact for editorial correspondence.

2. Proses Review

review lanjutan

1 pesan

Pendidikan Teknologi Informasi IPI <pti@institutpendidikan.ac.id>
Kepada: 18102217@ittelkom-pwt.ac.id, azrino@ittelkom-pwt.ac.id

31 Agustus 2022 pukul 17.05

hasil keputusan tim editor jurnal PETIK untuk artikel saudara adalah "**Revisions Required**" dengan judul artikel :

" Pengembangan E-Commerce Berbasis Single Page Application (SPA) Menggunakan ReactJS "

Hasil review : **Revisions Required**

Login Link : <https://journal.institutpendidikan.ac.id/index.php/petik>

(silahkan di periksa kembali hasil review artikelnya)

Mohon diperhatikan :

1. urutan dalam daftar pustaka tidak berdasarkan alpabet, tapi berdasarkan kemunculan di dialam artikel dengan gaya (IEEE), mohon disesuaikan.
2. Jumlah halaman artikel minimal 10 halaman, mohon disesuaikan.
3. kutipan dalam artikel menggunakan IEEE Style.
4. Wajib menyesuaikan dengan format artikel Petik
5. lengkapi dengan ucapan terima kasih

3. LoA / BUKTI DI TERIMA



Garut, 23 September 2022

No : 062/PETIK/IX/2022
Lampiran : -
Hal : Surat Penerimaan Naskah Publikasi Jurnal

Kepada Yth,
Zarkasih Akhmad
Program Studi Teknik Informatika
Institut Teknologi Telkom
Purwokerto

Terimakasih telah mengirimkan artikel ilmiah untuk diterbitkan pada Jurnal PETIK Sinta 5 dengan (e-ISSN 2614-6606) dengan Judul :

Rancang Bangun E-Commerce Berbasis Single Page Application (SPA) Menggunakan ReactJS

Penulis: Zarkasih Akhmad, Muhamad Azrino Gustalika

Berdasarkan hasil review, artikel tersebut dinyatakan **DITERIMA** untuk dipublikasikan di Jurnal kami untuk Volume 8, Nomor 2, September 2022

Artikel tersebut akan lebih dahulu tersedia secara online di <https://journal.institutpendidikan.ac.id/index.php/petik/issue/archive>

Demikian informasi ini disampaikan, dan atas perhatiannya, kami ucapkan terimakasih.

Garut, 23 September 2022
Jurnal Manager



Dr. Dian Rahadian, M.Pd

Rancang Bangun E-Commerce Berbasis Single Page Application (SPA) Menggunakan ReactJS

Zarkasih Akhmad^{#1}, Muhamad Azrino Gustalika^{*2}

[#]Program Studi Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom

JL. DI Panjaitan No.128, Karangreja Purwokerto Indonesia

¹18102217@ittelkom-pwt.ac.id

²azrino@ittelkom-pwt.ac.id

Abstract — Advances in information technology currently have a significant influence on all sectors, including changes in the business sector. Electronic commerce or more often referred to as e-commerce is one approach to the use of information technology in business. The Ozchic store is located in the city of Solo, focusing on selling fashion needs but does not yet have e-commerce so that it does not meet the needs of users in shopping. In the development of website applications, traditional architecture has a weakness in the use of wasteful bandwidth when making requests from the web server so that the Single Page Application (SPA) architecture using ReactJs is used in developing this website with efficient data management from a fast and interactive web server. SPA uses AJAX (Asynchronous JavaScript and XML) to dynamically update pages. On the website that has been created, the application only makes requests to the server only if the user needs it, so the application loads faster the first time than without using SPA. If the user has previously visited the website page, then the website does not need to make a request again if it has previously made a request. In developing using the ReactJs library, the source code can be separated into a component so that development becomes efficient. In testing the functionality of the Ozchic e-commerce website application using Blackbox testing, the results of testing the functionality of the application run well. Then testing using usability testing with 30 respondents who filled out 16 questionnaire questions resulted in a score of 76.91% which according to usability standards had a good grade. From the results of testing using the method on the application, it can be concluded that the Ozchic web application can run well.

Keywords— Single Page Application, ReactJs, E-Commerce

Abstrak— Kemajuan teknologi informasi telah memberikan pengaruh yang cukup signifikan pada semua sektor, termasuk perubahan pada sektor bisnis. *Electronic commerce* atau yang lebih sering disebut dengan *e-commerce* merupakan salah satu contoh dalam pemanfaatan teknologi informasi di sektor bisnis. Toko Ozchic terletak di kota Solo berfokus menjual kebutuhan fashion namun belum memiliki *e-commerce* sehingga belum memenuhi kebutuhan pengguna dalam berbelanja. Pada pengembangan aplikasi website, arsitektur tradisional memiliki kelemahan pada penggunaan *bandwidth* yang boros pada saat melakukan request dari web server sehingga arsitektur *Single Page Application* (SPA) dengan menggunakan ReactJs digunakan pada pengembangan website ini dengan pengelolaan data yang efisien dari web server yang bekerja cepat dan interaktif. SPA menggunakan *Asynchronous JavaScript and XML* (AJAX) untuk memperbaharui halaman secara dinamis. Pada website yang telah dibuat, aplikasi hanya melakukan request ke server hanya jika pengguna membutuhkannya, sehingga aplikasi lebih cepat dimuat pada pertama kali dibanding tanpa menggunakan SPA. Jika pengguna sebelumnya sudah mengunjungi halaman *website*, maka website tidak perlu melakukan request kembali jika sebelumnya sudah melakukan request. Dalam melakukan pengembangan menggunakan *library* ReactJs, *source code* dapat dipisahkan menjadi sebuah komponen sehingga pengembangan menjadi efisien. Pada pengujian fungsionalitas aplikasi website e-commerce Ozchic menggunakan *blackbox testing*, hasil pengujian fungsionalitas pada aplikasi berjalan dengan baik. Lalu pengujian menggunakan pengujian *Usability* dengan 30 responden yang mengisi 16 pertanyaan kuesioner menghasilkan skor 76,91% dimana menurut standar *usability* memiliki grade yang baik. Dari hasil pengujian menggunakan metode pada aplikasi, maka dapat disimpulkan aplikasi web Ozchic dapat dijalankan dengan baik

Kata kunci — Single Page Application, ReactJs, E-Commerce

I. PENDAHULUAN

Kemajuan teknologi informasi saat ini memberi dampak yang besar di segala bidang sektor salah satunya perkembangan di sektor bisnis. Salah satu strategi untuk menggunakan teknologi informasi dalam bisnis adalah perdagangan elektronik atau lebih sering disebut dengan *e-commerce*. Sebelum teknologi internet ada, pelaku bisnis menggunakan model tradisional untuk menjalankan dan mendukung kegiatan usahanya, namun seiring

berkembangnya zaman manusia membutuhkan kepraktisan dalam menjalankan aktivitasnya sehingga pelaku usaha melakukan perdagangan dengan sarana elektronik atau *electronic commerce* (*e-commerce*) [1]. Asosiasi *e-Commerce* Indonesia (IdEA) mencatat kenaikan penjualan sebesar 25% selama pandemi Covid-19 pada *platform e-commerce* [2]. Menteri Keuangan Indonesia, Sri Mulyani dalam penelitiannya menyatakan bahwa Indonesia memiliki potensi ekonomi digital hingga US\$ 133 miliar pada 2025. [3]. Keuntungan didapatkan oleh penjual dan pembeli karena

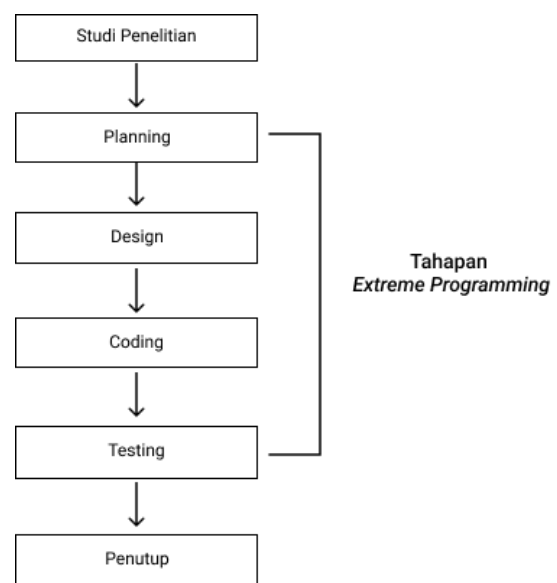
transaksi internet menghemat waktu dan tenaga kedua belah pihak. Toko Ozchic didirikan pada tahun 2020 yang berfokus menjual kebutuhan fashion yang terletak di kota Solo dimana memiliki cakupan pelanggan tidak hanya di solo tetapi juga pesanan datang dari kota-kota besar lainnya. Namun toko Ozchic belum mempunyai website untuk memenuhi kebutuhan pengguna dalam berbelanja sehingga website dibutuhkan agar para konsumen dapat memilih produk dengan lebih baik. Menurut Bank Indonesia (BI), selama wabah covid-19 peningkatan penjualan sebanyak 26% dan jumlah nasabah baru meningkat 51% [4]. Didukung dengan konsumen yang akan berpindah ke *website* saat sebelum melakukan pembelian pada toko Ozchic, *website* akan membutuhkan sistem yang efektif untuk mengurangi borosnya penggunaan *bandwidth* dalam upaya untuk mengantisipasi banyaknya pengunjung.

Dalam mengatasi permasalahan tersebut, arsitektur *Single Page Application* (SPA) digunakan karena memiliki waktu respons yang lebih cepat, lebih menyenangkan untuk digunakan, memproses data dalam jumlah besar lebih cepat dengan satu halaman kali penyegaran halaman, dan menghasilkan hasil yang akurat jika dibandingkan dengan metode lainnya. [5]. Penggunaan arsitektur ini digunakan pada *website e-commerce Ozchic* karena banyaknya sumber daya seperti gambar yang memakan waktu pemuatan lebih lama jika melakukan pengembangan *website* secara tradisional yang harus memuat seluruh halaman web [6]. Setelah halaman pertama di-refresh, semua interaksi server dalam implementasi SPA berlangsung melalui *Asynchronous JavaScript and XML (AJAX)* yang mengembalikan data dalam format *JavaScript Object Notation (JSON)* kemudian menggunakan data JSON untuk memperbarui halaman secara dinamis [7]. *ReactJs* menawarkan kecepatan, kesederhanaan, dan skalabilitas selain itu memungkinkan pengembang untuk membuat komponen UI yang lebih interaktif, *stateful*, dan dapat digunakan kembali [8].

Pada penelitian sebelumnya dapat dijadikan gambaran dan acuan dalam penelitian selanjutnya. Pada penelitian selanjutnya yang dilakukan oleh [9] yang menerapkan *Single Page Application (SPA)* menggunakan teknologi *AJAX* dan *REST API* pada sistem informasi untuk reservasi wisma tamu, kemudian penelitian milik [10] melakukan pengembangan *e-commerce* menggunakan aplikasi *content management system (CMS)*. Kemudian penelitian yang dilakukan oleh [11] yang melakukan analisis penerapan *code splitting* dalam pemanfaatan *library react* pada pengembangan aplikasi penjualan mebel berbasis *website*. Selanjutnya pada penelitian yang dilakukan oleh [12] menerapkan teknik layout web menggunakan *library React Js* dan *Bootstrap* dalam pengembangan aplikasi web *Single Page Application (SPA)*.

II. METODE PENELITIAN

Tahap ini menjelaskan bagan diagram alir pengembangan aplikasi *website e-commerce Ozchic*, pada penelitian ini memanfaatkan pendekatan metode *Extreme Programming (XP)*. Metode ini merupakan bagian dari teknik *Agile* untuk pengembangan aplikasi, yang dirancang untuk memenuhi tuntutan pengembangan tertentu dan didasarkan pada *Software Development Life Cycle (SDLC)*. [13]. *Extreme Programming (XP)*, sering dikenal sebagai pendekatan *technical how to*, adalah seperangkat konsep dan metodologi pengembangan perangkat lunak yang berguna yang membantu tim teknis membuat perangkat lunak dengan cepat dan efektif [14]. Penggunaan metode ini memiliki kelebihan dalam pengembangan *website e-commerce Ozchic* dimana tahapannya membutuhkan waktu yang cepat. Tahapan dari *Extreme Programming* yakni *Planning, Design, Coding dan Testing*. Gambar 1 merupakan alur dari penelitian yang dimulai dari *Studi Penelitian* untuk melakukan pengumpulan data-data yang dibutuhkan untuk penelitian, selanjutnya melakukan *planning* sebagai langkah awal pengembangan aplikasi lalu melakukan desain untuk perancangan sistem. Selanjutnya melakukan coding untuk implementasi pembuatan *website* dari desain yang sudah dibuat, setelah itu melakukan testing untuk menguji sistem yang telah dibuat pada tahapan sebelumnya lalu ditutup dengan kesimpulan dan saran.



Gambar 1. Diagram alir Penelitian

1. Studi Penelitian

Tahapan penelitian ini melakukan pengumpulan data yang berkaitan dengan *Single Page Application*. Peneliti melakukan literatur mengenai *Single Page Application* sebagai bahan ajar mata pelajaran. Data yang diperoleh berasal dari karya ilmiah maupun buku. Tujuan dari penelitian ini adalah untuk mengidentifikasi kelemahan-kelemahan penelitian sebelumnya sehingga dapat diperbaiki

dan dimasukkan ke ini. Dengan begitu, aplikasi yang dibuat dapat lebih efektif dan efisien.

2. Planning

Tahap ini merupakan langkah awal dalam proses pengembangan *Extreme Programming* (XP) dimana melibatkan pemeriksaan masalah, identifikasi alur proses dari aplikasi, menentukan kebutuhan pengguna, fungsi dari aplikasi, fitur aplikasi, alur proses pengembangan aplikasi dan menentukan semua kebutuhan pengguna yang harus dipenuhi oleh sistem.

3. Design

Tahapan desain merupakan tahapan dalam pembuatan pemodelan sistem aplikasi berdasarkan kebutuhan pengguna. Perancangan desain sistem yang dibuat akan bersifat fleksibel dan dinamis yang akan disesuaikan dengan masukan-masukan dari pengguna. Penulis membuat perancangan sistem aplikasi berbasis website dengan menyiapkan *use case diagram* sebagai perancangan website yang akan dibuat untuk pengembangan sistem ini. *Use case diagram* salah satu dari banyak tipe *diagram Unified Modeling Language* (UML) yang digunakan dalam memberikan gambaran interaksi antara aktor dan sistem. Cara di mana pengguna sistem dan sistem berinteraksi dapat dijelaskan *use case*.

4. Coding

Coding merupakan proses implementasi rancangan desain aplikasi yang telah dibuat ke dalam bentuk kode program hingga aplikasi dapat bentuk dan digunakan. Implementasi rancangan dari desain yang sudah dibuat pada website e-commerce Ozchic ini dilakukan dengan bahasa pemrograman JavaScript, sedangkan untuk rancangan data (basis data) dilakukan dengan menggunakan database MongoDB.

5. Testing

Pengujian dilakukan dengan menggunakan pendekatan metode *Black box testing* dimana pengujian ini tidak melakukan pengujian pada *source code* program melainkan melakukan pengujian secara langsung dengan pengujian pada tampilan sistem [15]. Kemudian dilanjut dengan pengujian aspek *usability* dengan menghitung rata-rata jawaban berdasarkan skoring menggunakan skala Likert. Hasil pengujian kemudian digunakan untuk membuat kesimpulan dan memberikan referensi pengembangan yang dapat digunakan sebagai panduan saat membuat aplikasi di masa mendatang.

III. HASIL DAN PEMBAHASAN

E-commerce Ozchic adalah sebuah website yang menjual berbagai kebutuhan *fashion* untuk wanita yang terletak di kota solo. Aplikasi dibangun dengan arsitektur *Single Page Application* (SPA) menggunakan *library* ReactJs, penggunaan arsitektur ini digunakan dalam meningkatkan pengalaman pengguna karena cepat memproses data dalam jumlah yang banyak serta tidak perlu memuat ulang saat berpindah halaman. Setiap perpindahan halaman atau melakukan aktivitas transaksi, website tidak perlu me-refresh halaman website sehingga aplikasi lebih nyaman digunakan. Penerapan *Single Page Application* juga membuat *website*

menjadi lebih cepat karena konten yang tampil akan disesuaikan dengan kebutuhan pengguna sehingga dapat mengurangi pemborosan *bandwidth* dan *website* berjalan lebih cepat dibanding menggunakan arsitektur tradisional.

A. Planning

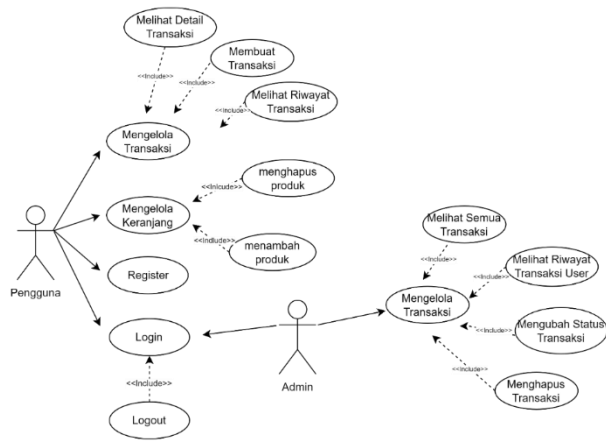
Tahapan planning dimulai dengan mencari permasalahan yang ada pada toko Ozchic, dimana toko Ozchic belum memiliki *website* untuk menjual produk kemudian kebutuhan sistem akan dibangun. Dalam mengembangkan *website e-commerce* memiliki permasalahan dimana membutuhkan sistem yang efektif untuk mengurangi borosnya penggunaan *bandwidth* dalam upaya untuk mengantisipasi banyaknya pengunjung, sehingga arsitektur *Single Page Application* (SPA) digunakan untuk pengembangan sistem ini. Kebutuhan fungsional diperlukan untuk menetapkan tugas atau sumber daya mana yang dapat dikelola sistem, dan pengguna yang dapat menjalankan sistem. Dalam pengguna sistem terdapat dua aktor yaitu *user* dan *admin*.

TABEL I
KEBUTUHAN PENGGUNA

No	Aktor	Kebutuhan Sistem
1	User	Melakukan <i>register</i> Melakukan <i>login</i> Melakukan <i>logout</i> Menampilkan produk Menambah produk ke dalam keranjang Menghapus produk dari keranjang Melakukan pemesanan Menampilkan riwayat transaksi Menampilkan detail riwayat transaksi
2	Admin	Melakukan <i>login</i> Melakukan <i>logout</i> Melihat detail transaksi Melakukan <i>update</i> status Melihat detail produk Menghapus transaksi pengguna
3	Guest	Melakukan <i>register</i> Melakukan <i>login</i> Menampilkan produk

1) Use Case Diagram

Activity Diagram digunakan untuk penggambaran sebuah aktivitas atau sebuah kegiatan yang dilakukan oleh pengguna dalam memakai setiap langkah yang ada di menu. Pada *e-commerce* Ozchic terdapat beberapa aktifitas yang ada pada aplikasi dari pengguna memesan produk hingga barang diterima oleh pengguna



Gambar 1. Use Case Diagram

Pada Gambar Gambar 1 menunjukkan *activity diagram* pada *e-commerce* Ozchic dengan penjelasan sebagai berikut:

1. Admin melakukan *update* pada setiap informasi seperti produk, pengguna hingga transaksi.
2. Selanjutnya admin mengelola produk dengan menambahkan produk atau menghapus produk.
3. Setelah produk tersedia, web *e-commerce* Ozchic akan produk sesuai dengan persediaan yang dikelola oleh admin.
4. Produk yang ditampilkan selanjutnya dapat dipilih oleh pengguna untuk ditambahkan kedalam keranjang pada website .
5. Setelah produk ditambahkan ke keranjang, pengguna dapat melakukan transaksi.
6. Setelah transaksi dibuat, admin dapat mengelola transaksi hingga produk dikirimkan ke pengguna.
7. Lalu setelah dikirimkan, pengguna menerima produk yang sesuai pilihannya.

2) Activity Diagram

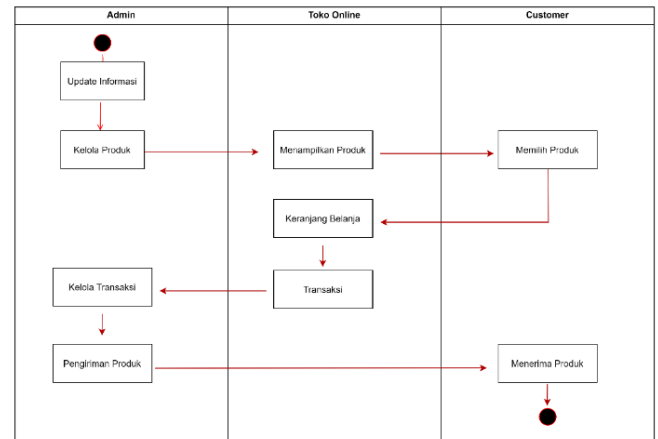
Activity Diagram digunakan untuk memberikan gambaran sebuah aktivitas atau sebuah kegiatan yang dilakukan oleh pengguna dalam memakai setiap langkah yang ada di menu. Pada *e-commerce* Ozchic terdapat beberapa aktifitas yang ada pada aplikasi dari pengguna memesan produk hingga barang diterima oleh pengguna.

Pada Gambar Gambar 4.1.2 menunjukkan *activity diagram* pada *e-commerce* Ozchic dengan penjelasan sebagai berikut:

1. Admin melakukan *update* pada setiap informasi seperti produk, pengguna hingga transaksi.
2. Selanjutnya admin mengelola produk dengan menambahkan produk atau menghapus produk.
3. Setelah produk tersedia, web *e-commerce* Ozchic dapat menampilkan produk sesuai dengan persediaan yang

dikelola oleh admin.

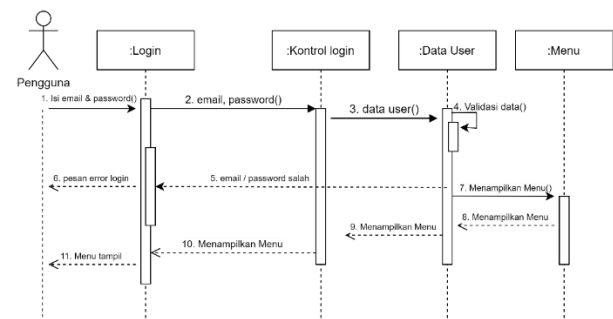
4. Produk yang ditampilkan selanjutnya dapat dipilih oleh pengguna untuk ditambahkan kedalam keranjang pada website .
5. Setelah produk ditambahkan ke keranjang, pengguna dapat melakukan transaksi.
6. Setelah transaksi dibuat, admin dapat mengelola transaksi hingga produk dikirimkan ke pengguna.
7. Lalu setelah dikirimkan, pengguna menerima produk yang sesuai pilihannya.



Gambar 2. Activity Diagram

3) Sequence Diagram

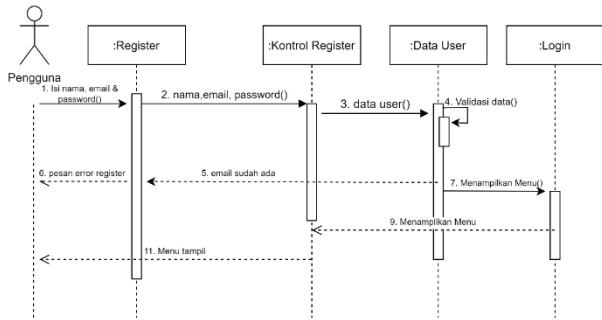
Sequence Diagram berfungsi untuk mengetahui semua perilaku pengguna yang dalam setiap langkah menggunakan beberapa menu yang terdapat didalam *e-commerce* Ozchic yang telah dibuat. Pada urutan *sequence diagram* pengguna harus melewati beberapa tahapan sebelum melakukan transaksi. Gambar 4.1.3 dibawah ini adalah contoh dari *sequence diagram* dari *e-commerce* Ozchic.



Gambar 3. Sequence Diagram Login

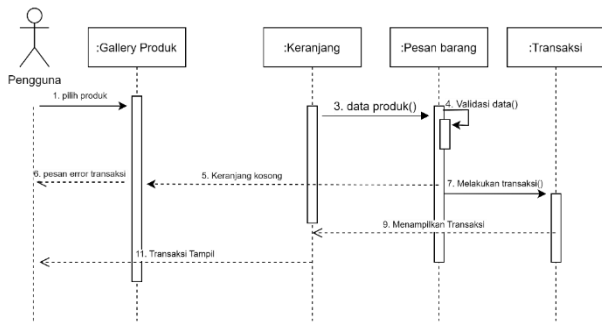
Pada Gambar 3 dimana *sequence diagram* pada login dimana pengguna melakukan login dengan memasukkan *email* dan *password* akan di lakukan validasi, jika *error* maka akan menampilkan pesan *error* sedangkan jika berhasil maka pengguna akan diarahkan ke menu yang tersedia pada

website.



Gambar 4. Sequence Diagram Register

Pada Gambar 4 dimana sequence diagram pada register untuk mendaftar akun pada pengguna jika ingin melakukan transaksi. Data pengguna akan di validasi jika error maka akan muncul pesan error saat register, sedangkan jika berhasil maka pengguna akan diarahkan ke halaman login.



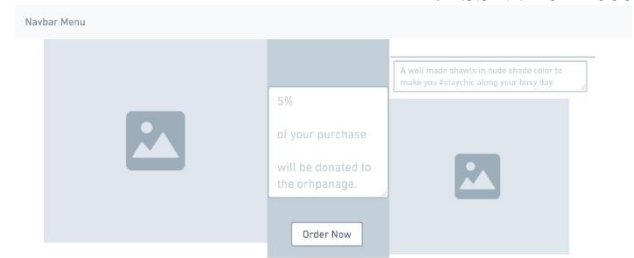
Gambar 5. Sequence Diagram Transaksi

Pada Gambar 5 dimana sequence diagram pada transaksi memiliki alur dimana pengguna memilih produk dengan menentukan jumlah produk yang diinginkan untuk dimasukkan ke dalam keranjang. Jika keranjang terdapat produk di dalamnya, maka pengguna dapat melakukan transaksi, jika tidak maka akan terdapat pesan error kepada pengguna. Pengguna akan diarahkan ke halaman transaction jika pesanan berhasil dipesan.

B. Design

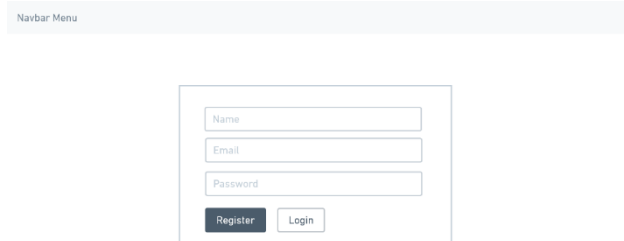
Desain Low Fidelity merupakan sebuah rangkaian desain awal pada sebuah tampilan user yang nantinya akan dijadikan sebagai acuan untuk tampilan UI pada aplikasi. Tampilan UI dibuat agar memudahkan untuk dipahami dan digunakan. Pada e-commerce Ozchic dibuat dengan dua desain yaitu desain low fidelity Pengguna dan desain low fidelity Admin yang dibuat sesuai dengan kebutuhan masing-masing.

1) *Halaman User - Home*: Pada Gambar 4 menunjukkan halaman home dimana konsumen akan mengakses antarmuka utama website yang dapat melihat seluruh isi fitur pada website seperti melihat produk dan menu yang tersedia, produk yang ditawarkan hingga pilihan kategori produk yang dapat mempermudah pengguna dalam menemukan produk yang di inginkan.



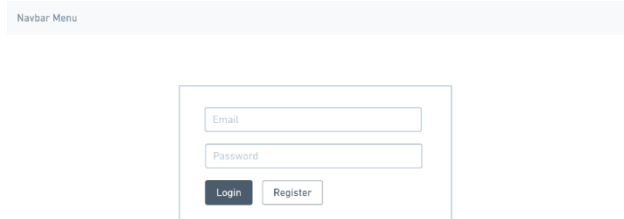
Gambar 4. Halaman User - Home

2) *Halaman User - Register*: Pada Gambar 5 halaman register pengguna dapat mendaftar akun dengan mengisi nama, email dan password. Setelah register selesai, pengguna dapat melakukan login pada halaman login.



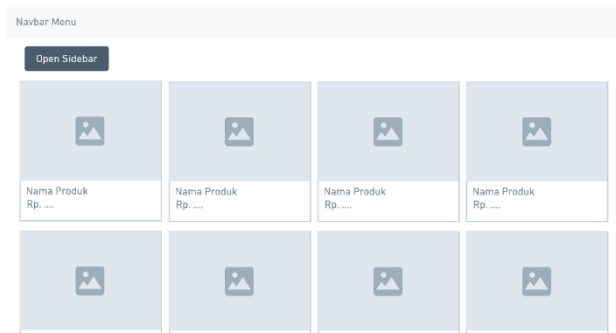
Gambar 5. Halaman User - Register

3) *Halaman User - Login*: Setelah selesai melakukan register, pengguna dapat melakukan login seperti pada gambar 5. Pengguna dapat melakukan login dengan memasukan akun yang telah didaftarkan sebelumnya pada halaman register.



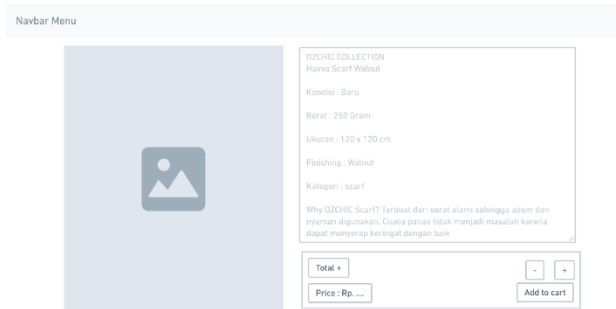
Gambar 6. Halaman User - Login

4) *Halaman User - Gallery*: Pada halaman gallery, pengguna dapat memilih produk yang terjual di toko Ozchic seperti yang terlihat pada Gambar 7. Jika pengguna memilih salah satu produk pada gallery, pengguna akan diarahkan ke halaman detail gallery untuk melihat lengkap deskripsi produk.



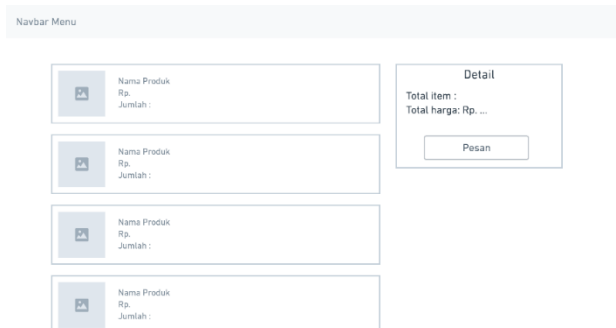
Gambar 7. Halaman User - Gallery

5) *Halaman User – Detail Product* : Pada Gambar 8 menunjukkan halaman detail *product* dimana halaman ini konsumen dapat melihat lengkap detail produk mulai dari nama, kondisi, ukuran, kategori, deskripsi hingga harga produk. Pengguna dapat memasukan produk ke dalam keranjang dengan menentukan jumlah produk yang ingin ditambahkan dan harga akan menyesuaikan jumlah produk yang ditambahkan. Setelah pengguna menambahkan produk, pengguna akan diarahkan ke halaman *cart* secara otomatis agar dapat melihat produk yang sudah ditambahkan.



Gambar 8. Halaman User – Detail Product

6) *Halaman User - Cart*: Pengguna dapat melihat seluruh produk yang telah ditambahkan ke keranjang seperti yang terlihat pada Gambar 9. Pengguna dapat menghapus produk dari keranjang dan pengguna dapat melihat detail produk pada keranjang yang akan diarahkan ke halaman detail produk seperti pada gambar 8.



Gambar 9. Halaman User - Cart

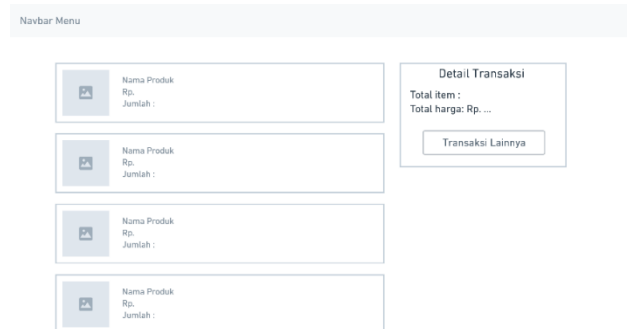
7) *Halaman User - Transactions*: Pada halaman transactions, pengguna dapat melihat seluruh transaksi yang

sudah dilakukan oleh pengguna seperti yang terlihat pada Gambar 10. Jika pengguna memilih salah satu list transaksi, pengguna akan diarahkan ke halaman detail transaksi seperti yang terlihat pada gambar 11.



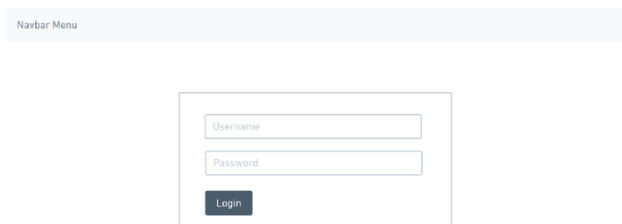
Gambar 10. Halaman User - Transactions

8) *Halaman User - Detail Transaction*: Pada halaman detail transactions, pengguna dapat melihat detail transaksi yang telah dilakukan dimana berisi produk dan harga yang dibeli seperti yang terlihat pada Gambar 11.



Gambar 11. Halaman User - Detail Transaction

9) *Halaman Admin - Login*: Admin diharuskan melakukan *login* sebelum masuk ke halaman beranda dengan memasukan *username* dan *password* seperti pada Gambar 12. Setelah login admin dapat mengunjungi ke halaman *dashboard*.



Gambar 12. Halaman Admin - Login

10) *Halaman Admin - Dashboard*: Admin akan diarahkan ke halaman dashboard seperti yang terlihat pada gambar 14 setelah *login*. Di halaman ini admin dapat melihat pesanan yang sudah masuk sehingga admin dapat melakukan pengelolaan pesanan transaksi.



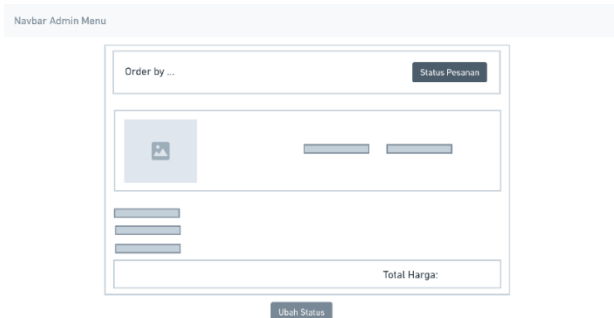
Gambar 13. Halaman Admin - Dashboard

11) *Halaman Admin - Riwayat Transaksi User:* Admin dapat melihat riwayat transaksi dari akun user yang sudah didaftarkan seperti yang terlihat pada gambar 14. Admin dapat memilih list transaksi user dimana admin akan diarahkan ke detail transaksi seperti pada gambar 15.



Gambar 14. Halaman Admin - Riwayat Transaksi User

12) *Halaman Admin - Detail Transaksi User:* Pada detail transaksi *user*, admin dapat mengubah status dari “Belum lunas” menjadi sudah “Sudah Lunas” serta admin dapat menghapus transaksi seperti pada gambar 15.



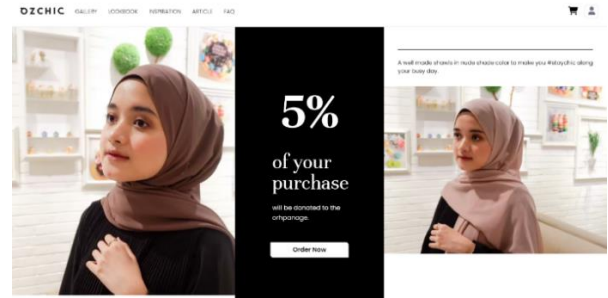
Gambar 15. Halaman Admin - Detail Transaksi User

C. Coding

Coding merupakan proses implementasi rancangan desain yang sudah dibuat ke dalam bentuk kode program hingga aplikasi dapat bentuk dan digunakan. Implementasi rancangan dari desain yang sudah dibuat pada website *e-commerce* Ozchic ini dilakukan dengan bahasa pemrograman

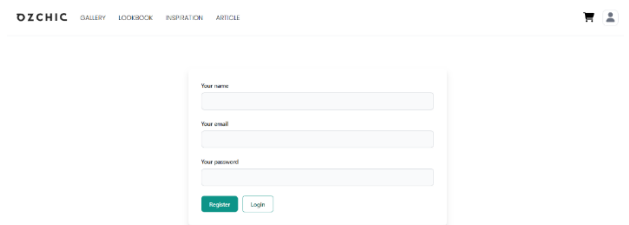
JavaScript, sedangkan untuk rancangan data (basis data) dilakukan dengan menggunakan database MongoDB. Proses implementasi menggunakan library ReactJS dengan menggunakan aplikasi WebStorm dalam proses pengkodean. Dalam proses pengkodean, debugging dilakukan jika ada error dalam program menggunakan *developer tools* yang tersedia di dalam *browser* chrome. Berikut merupakan hasil pengkodean yang telah dilakukan sesuai dengan desain yang sudah dibuat sebelumnya.

1) Coding Halaman User - Home



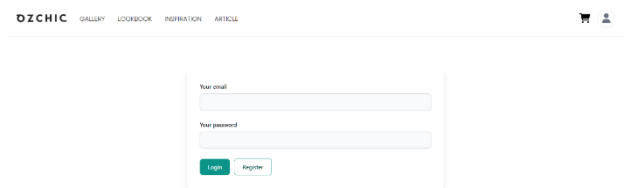
Gambar 16. Coding Halaman User - Home

2) Coding Halaman User - Register



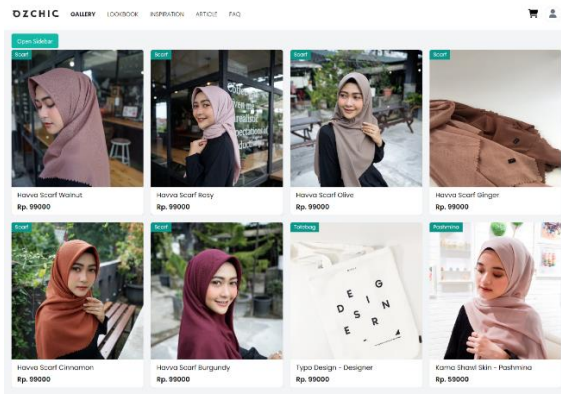
Gambar 17. Coding Halaman User - Register

3) Coding Halaman User - Login



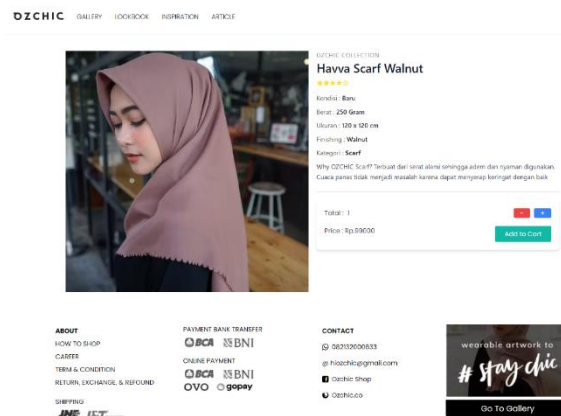
Gambar 18. Coding Halaman User - Login

4) Coding Halaman User - Gallery



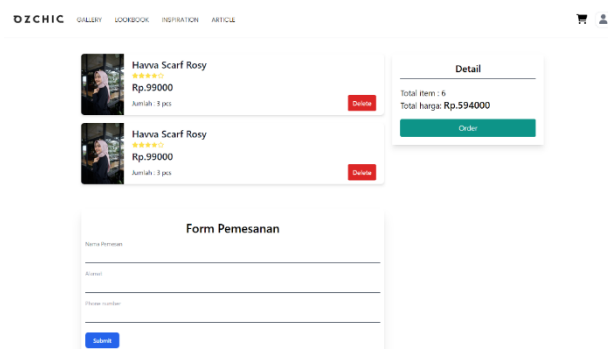
Gambar 19. Coding Halaman User - Gallery

5) Coding Halaman User – Detail Product



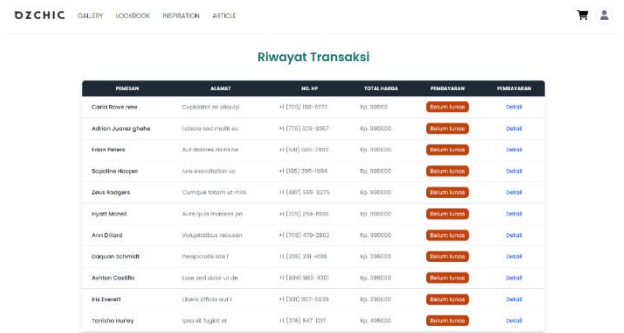
Gambar 20. Coding Halaman User – Detail Product

6) Coding Halaman User - Cart



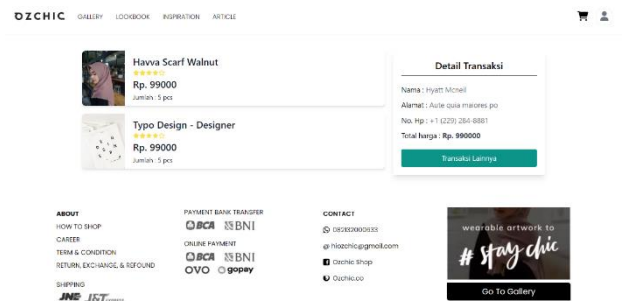
Gambar 21. Coding Halaman User – Cart

7) Coding Halaman User - Transactions



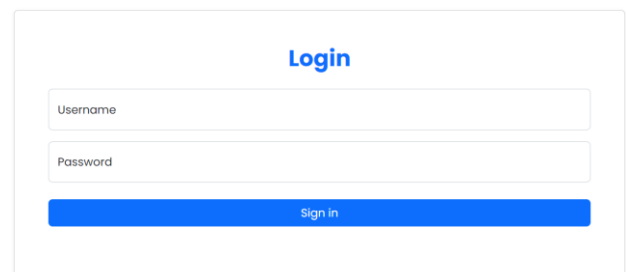
Gambar 22. Coding Halaman User – Transaction

8) Coding Halaman User - Detail Transaksi



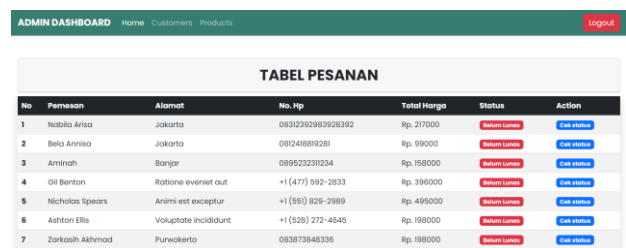
Gambar 23. Coding Halaman User - Detail Transaksi

9) Coding Halaman Admin - Login



Gambar 24. Coding Halaman Admin - Login

10) Coding Halaman Admin - Dashboard



Gambar 25. Coding Halaman Admin – Dashboard

11) Coding Halaman Admin - Riwayat Transaksi User

No	Nama	Email	Pemesanan
1	Desi	khassanahdes20@gmail.com	Cek pesanan
2	Desilli	1910219@ittekam-pwt.ac.id	Cek pesanan
3	Reza Ahmad	rezaahmad@gmail.com	Cek pesanan
4	Whitemina Powers	zureno@mailinator.com	Cek pesanan
5	admin232	akhmadzarkasih@gmail.com	Cek pesanan
6	thomas	thomas@gmail.com	Cek pesanan
7	Zarkasih Akhmad	zarkasih@gmail.com	Cek pesanan
8	Mursid	mursid@gmail.com	Cek pesanan
9	Aminah	mamangmamang@gmail.com	Cek pesanan

Gambar 26. Coding Halaman Admin - Riwayat Transaksi User

12) Coding Halaman Admin - Detail Transaksi User

Gambar 27. Coding Halaman Admin - Detail Transaksi User

D. Testing

1) *Pengujian Black Box testing*: Pada pengujian *black box*, pengujian akan menguji fungsionalitas dari aplikasi web *e-commerce* Ozchic yang mana pengujian berfokus pada fungsi - fungsi dan fitur yang terdapat pada aplikasi tersebut. Hasil pengujian ditunjukkan pada tabel

TABEL II
BLACKBOX TESTING HALAMAN PENGGUNA

No	Parameter Pengujian	Harapan	Hasil
1	Melakukan <i>register</i>	Pengguna dapat mendaftar akun baru pada website	valid
2	Melakukan <i>login</i>	Pengguna dapat melakukan <i>login</i> ke dalam website	valid
3	Melakukan <i>logout</i>	Pengguna dapat <i>logout</i> dari website	valid
4	Menampilkan produk berdasarkan kategori	Pengguna dapat melihat seluruh produk berdasarkan kategori yang dipilih	valid
5	Menambah produk ke keranjang	Pengguna dapat melihat seluruh produk berdasarkan kategori	valid

		yang dipilih	
6	Menghapus produk dari keranjang	Pengguna dapat menghapus produk dari halaman keranjang	valid
7	Melakukan transaksi	Pengguna dapat melakukan pemesanan berdasarkan produk yang terdapat di keranjang dan data diri yang telah di isi pada form pemesanan	valid
8	Menampilkan riwayat transaksi	Pengguna dapat melihat seluruh riwayat transaksi pada halaman transactions	valid
9	Menampilkan detail transaksi	Pengguna dapat melihat detail riwayat transaksi sesuai dengan riwayat transaksi	valid

TABEL III
BLACKBOX TESTING HALAMAN ADMIN

No	Parameter Pengujian	Harapan	Hasil
No	Target Pengujian	Hasil Yang Diharapkan	valid
1	Melakukan <i>Login</i>	Admin dapat <i>Login</i> ke halaman admin	valid
2	Melakukan <i>logout</i>	Admin dapat <i>logout</i> untuk keluar	valid
3	Melihat detail transaksi	Admin dapat melihat detail transaksi milik pengguna	valid
4	Melakukan <i>update</i> status	Admin dapat mengupdate status "belum lunas" ke sudah "lunas"	valid
5	Melihat detail produk	Admin dapat melihat detail produk	valid
6	Menghapus transaksi pengguna	Admin dapat menghapus transaksi pengguna	valid

2) *Pengujian Usability*: Setelah melakukan pengujian dengan menggunakan *blackbox testing*, selanjutnya dilakukan pengujian kedua dengan metode *usability testing* yang dilakukan dengan memberikan kuesioner yang diisi beberapa responden yang hasil akhirnya akan muncul sebagai

kelayakan aplikasi dengan kriteria untuk instrumen *usability* dengan lima pilihan jawaban berdasarkan skala Likert.

TABEL IV
KISI – KISI INSTRUMEN USABILITY

No	Pertanyaan	Penilaian				
		SS	S	R	TS	STS
<i>Usefulness</i>						
1	Aplikasi ini membantu saya untuk membeli produk pada toko ozchic					
2	Aplikasi ini mempermudah saya dalam melakukan transaksi					
3	Aplikasi ini sesuai dengan kebutuhan					
4	Aplikasi ini berjalan sesuai dengan harapan					
<i>Ease of use</i>						
5	Saya tidak kesulitan saat menggunakan aplikasi ini					
6	Saya mudah memahami cara penggunaan aplikasi ini					
7	Aplikasi ini praktis saat digunakan					
8	Saya tidak kesulitan saat menggunakan aplikasi ini					
<i>Ease of learning</i>						
9	Saya dapat dengan cepat memahami cara menggunakan aplikasi ini					
10	Saya mudah mengingat bagaimana menggunakan aplikasi ini					
11	Aplikasi ini mudah untuk dipelajari bagaimana cara penggunaannya					
12	Saya terampil menggunakan aplikasi ini dengan cepat					

Satisfaction

13	Saya merasa puas dengan aplikasi ini					
14	Aplikasi ini memiliki tampilan yang bagus					
15	Aplikasi ini menyenangkan saat digunakan					
16	Aplikasi ini bekerja sesuai dengan apa yang saya inginkan					

Setelah pengumpulan data penelitian dari kuesioner, data tersebut akan digunakan untuk menentukan hasil akhir dengan mencari nilai rata-rata respons berdasarkan skor dari semua respon yang diberikan oleh responden terhadap kuesioner. Skala Likert dan lima pilihan respon digunakan dalam kriteria untuk instrumen kegunaan pada Tabel V dibawah ini.

TABEL V
INTERVAL SKALA LIKERT PERNYATAAN POSITIF

Alternatif Jawaban	Nilai
Sangat Tidak Setuju	1
Tidak Setuju	2
Ragu - ragu	3
Setuju	4
Sangat Setuju	5

Dalam pengujian Usability ada beberapa faktor yang diujikan seperti *Usefulness*, *Ease of Use*, *Ease of Learning* dan *Satisfaction*. Pengujian dilakukan dengan menyebarkan kuesioner kepada masyarakat umum dan mahasiswa. Hasilnya ada 30 jawaban kuesioner yang telah dikumpulkan. Nilai total keseluruhan yang didapat adalah 2164, sedangkan untuk nilai maksimal tiap pertanyaan adalah 5 (Sangat Setuju) sehingga nilai maksimal yang diperoleh adalah 2.400. Nilai tersebut didapat dari hasil nilai maksimal tiap pertanyaan yaitu $16 \times 30 \times 5 = 2.400$. Setelah mendapat nilai maksimal, maka hasil secara keseluruhan adalah sebagai berikut:

$$\begin{aligned}
 \text{Persentase kelayakan (\%)} &= \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\% \\
 &= \frac{1888}{2400} \times 100\% \\
 &= 78,67\%
 \end{aligned}$$

Setelah mendapatkan hasil persentase kelayakan sebesar 76,91%, maka berdasarkan hasil observasi dapat aplikasi web *e-commerce* Ozchic memiliki respon yang baik dari

responden. Pada Tabel VI dibawah ini merupakan tabel kriteria interpretasi skor berdasarkan hasil pengujian *Usability* dengan skala Likert

TABEL VI
KRITERIA INTERPRETASI SKOR

Skala Jawaban	Nilai
0 - 20	Sangat Tidak Baik
21 - 40	Tidak Baik
41 - 60	Cukup
61 - 80	Baik
81 - 100	Sangat Baik

Berdasarkan hasil observasi maka aplikasi web e-commerce Ozchic dapat memenuhi standar *usability*, yakni sebesar 78,67% atau bisa dikatakan "Baik".

IV. KESIMPULAN DAN SARAN

1. Kesimpulan

Berdasarkan hasil dan pengujian dari Pengembangan *E-Commerce Berbasis Single Page Application* (SPA) Menggunakan React Js pada toko Ozchic telah dilakukan dapat diambil kesimpulan diantaranya, Ketika pengguna berpindah halaman website tidak perlu melakukan *refresh* sehingga aplikasi berjalan lebih baik, konten yang ditampilkan juga disesuaikan kebutuhan pengguna sehingga website berjalan lebih cepat. Pada pengujian *Black Box*, semua hasil pengujian fungsionalitas pada website masing - masing berjalan dengan baik. Kemudian untuk hasil pengujian *Usability* mendapatkan hasil skor yaitu 78,67%, maka hasil skor mendapatkan rating yang "Baik" sesuai dengan aspek *usability* yang telah dibuat.

2. Saran

Adapun saran yang dapat dilakukan pada pengembangan penelitian yaitu aplikasi dapat menerapkan fitur *payment gateway* sehingga pengguna dapat lebih mudah saat melakukan pembayaran langsung melalui aplikasi. Aplikasi juga dapat menerapkan *library* *redux* agar penggunaan *state management* lebih efektif pada pengembangan aplikasi.

UCAPAN TERIMA KASIH

Terima kasih untuk Allah SWT yang telah melimpahkan berkat dan karunia-Nya, kepada seluruh pihak yang mendukung penelitian ini kami ucapkan terima kasih. Terima Kasih untuk Kampus kami Institut Teknologi Telkom Purwokerto dan juga Terima Kasih untuk semua pihak atas dukungan selama ini.

DAFTAR PUSTAKA

[1] Melisa Setiawan Hotana, "Industri E-Commerce Dalam

Menciptakan Pasar Yang Kompetitif Berdasarkan Hukum Persaingan Usaha," *J. Huk. Bisnis Bonum Commune*, vol. 1, no. 1, pp. 28–38, 2018, [Online]. Available: <http://jurnal.untag-sby.ac.id/index.php/bonumcommune/article/view/1754>

[2] A. Waworuntu, "Rancang Bangun Aplikasi e-Commerce Dropship Berbasis Web," *Ultim. J. Tek. Inform.*, vol. 12, no. 2, pp. 118–124, 2020, doi: 10.31937/ti.v12i2.1823.

[3] A. Sulaiman, "Pengembangan Sistem Informasi Public E-Marketplace pada PT XYZ," *J. Ultim. InfoSys*, vol. 6, no. 2, pp. 92–101, 2016, doi: 10.31937/si.v6i2.225.

[4] S. Ayu and A. Lahmi, "Peran e-commerce terhadap perekonomian Indonesia selama pandemi Covid-19," *J. Kaji. Manaj. Bisnis*, vol. 9, no. 2, p. 114, 2020, doi: 10.24036/jkmb.10994100.

[5] E. Sorongan, D. R. Sari, and P. Apriliza, "Sistem Pendukung Keputusan Pemilihan Gudang Menggunakan Metode Single Page Application Dan Simple Additive Weighting," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 3, p. 485, 2021, doi: 10.25126/jtiik.0813257.

[6] Nasution and L. Iswari, "Penerapan React JS Pada Pengembangan FrontEnd Aplikasi Startup Ubaform," *Automata*, vol. 2, no. 2, 2021.

[7] F. Nurpandi and D. M. R. Liki, "Perancangan dan Pembangunan Aplikasi Electronic Menu Restoran," *Media J. Inform.*, vol. 13, no. 1, p. 1, 2021, doi: 10.35194/mji.v13i1.1270.

[8] I. Panjaitan and A. F. Pakpahan, "Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 20–34, 2021, doi: 10.28932/jutisi.v7i1.3098.

[9] A. Luqman and Azhari, "Analisis Penerapan Single Page Application Menggunakan Teknologi AJAX dan REST API (Studi Kasus : Sistem Informasi Reservasi Wisma Tamu UKSW)," *J. Tek. Inform. dan Sist. Inf.*, vol. 3, no. 1, pp. 116–126, 2017.

[10] B. Huda and B. Priyatna, "Penggunaan Aplikasi Content Management System (CMS) Untuk Pengembangan Bisnis Berbasis E-commerce," *SYSTEMATICS*, vol. 1, no. 2, p. 81, Dec. 2019, doi: 10.35706/sys.v1i2.2076.

[11] D. Tanudjaja and R. Tanone, "Analisis Penerapan Code Splitting Library React pada Aplikasi Penjualan Mebel Berbasis Website," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 2, pp. 344–356, 2021, doi: 10.28932/jutisi.v7i2.3493.

[12] M. F. Santoso, "Teknik Single Page Application (Spa) Layout Web Dengan Menggunakan React Js Dan Bootstrap," vol. 9, no. 2, p. 6, 2021.

[13] Basri, "Pendekatan Kriptografi Hybrid pada Keamanan Dokumen Elektronik dan HypertextTransfer Protocol Secure (HTTPS) (Analisis Potensi Implementasi Pada Sistem Keamanan Informasi)," *J. Ilmu Komput.*, vol. 1, no. 2, pp. 32–37, 2015.

[14] L. Rusdiana, "Extreme programming untuk rancang bangun aplikasi pengelolaan surat keterangan kependudukan," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 4, no. 1, pp. 49–55, 2018, doi: 10.26594/register.v4i1.1191.

[15] M. A. Gustalika, D. P. Rakhmadani, and A. J. T. Segara, "Penerapan Metode Simple Additive Weighting (SAW) pada Sistem Informasi Pemilihan Asisten Praktikum," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 813, 2021, doi: 10.30865/mib.v5i3.3065.