

Pengembangan *Back-End* Aplikasi *Mobile Web* Untuk Pedagang Keliling Di Lingkungan Rt 04 Palem Kota Bandung

1st Nisa Aisyatunnabilah Hasyim
Fakultas Rekayasa Industri
Universitas Telkom

Bandung, Indonesia
nisanabb@student.telkomuniversity.ac.id

2nd Ahmad Musnansyah
Fakultas Rekayasa Industri
Universitas Telkom

Bandung, Indonesia
ahmadanc@telkomuniversity.ac.id

3rd Zalina Fatima Azzahra
Fakultas Rekayasa Industri
Universitas Telkom

Bandung, Indonesia
zalinaza@telkomuniversity.ac.id

Abstrak — Penelitian ini mengkaji permasalahan yang dihadapi oleh warga dan pedagang keliling di RT04 Palem, Kota Bandung, Jawa Barat terkait ketidaktersediaan pedagang jasa keliling saat dibutuhkan, yang menyebabkan warga harus mencari alternatif lain. Permasalahan ini berdampak pada kurang maksimalnya profit pedagang keliling dan ketidakseimbangan antara supply dan demand akibat tidak adanya sistem koordinasi yang efektif antara calon konsumen dan pedagang keliling. Untuk mengatasi masalah tersebut, penelitian ini mengembangkan sebuah aplikasi berbasis website mobile menggunakan metode iterative incremental. Aplikasi ini dirancang untuk meningkatkan koordinasi antara pedagang keliling dan konsumen dengan fitur-fitur utama seperti pemesanan melalui chat, fitur share location, informasi mengenai pedagang keliling, dan fitur review. Pengembangan aplikasi dilakukan dalam dua fase dan diuji menggunakan metode Blackbox Testing dan Stress Testing. Hasil pengujian menunjukkan bahwa aplikasi ini berjalan lancar dan dapat memenuhi kebutuhan pengguna, serta metode iterative incremental terbukti efektif dalam meningkatkan efisiensi pengembangan dan penyesuaian aplikasi. Stress Testing juga menunjukkan sistem mampu menangani permintaan hingga 200 pengguna virtual secara bersamaan dengan baik.

Kata kunci — Pedagang keliling, *Back-end*, *Testing*.

I. PENDAHULUAN

Teknologi Dalam Dalam lingkungan RT, pedagang keliling memiliki peran penting dalam memenuhi kebutuhan sehari-hari warga sekitar. Menurut Mc Gee, Orang-orang yang berjualan dengan bergerak dari satu tempat ke tempat lain disebut mobile hawkers. Mereka akan berkeliling sampai bertemu dengan calon pelanggan untuk menjual produk dan menawarkan jasa mereka. Seringkali terjadi masalah yang dihadapi oleh pedagang dan konsumennya. Contohnya adalah ketidaktersediaan barang/jasa yang dibutuhkan oleh konsumen pada saat mereka membutuhkannya, dan sebaliknya, barang/jasa seringkali tersedia ketika konsumen tidak memerlukannya. Terlewatnya pedagang keliling oleh konsumen adalah masalah umum dalam proses bisnis

berdagang keliling. Pembeli tidak tahu saat pedagang berjalan di sekitar mereka.

Hasil kuisioner menurut Saputra, menunjukkan bahwa 56 dari 70 konsumen membatalkan transaksi hanya karena mereka tidak mengetahui keberadaan dan lokasi pedagang keliling. Dalam wawancara yang dilakukan penulis dengan Muslih, seorang pedagang jasa sol sepatu berusia 47 tahun yang beroperasi di RT 04 Palem, kota Bandung, terungkap bahwa seringkali beliau harus mengulangi rute yang sudah dilewati. Hal ini dilakukan karena beliau khawatir ada warga yang tidak mengetahui bahwa beliau sudah melewati daerah tersebut dan mungkin membutuhkan jasanya. Yanto (35 tahun) yang merupakan pedagang sayur juga menyatakan seringnya membeli stok yang terlalu banyak di hari-hari tertentu dimana terkadang konsumen yang dijumpai lebih sedikit dari perkiraannya.

Sebanyak 7 dari 10 warga di daerah tersebut mengeluhkan ketidakadaanya pedagang jasa keliling saat dibutuhkan. Sehingga beberapa warga termasuk Rafi (25 Tahun) mencari alternatif lain dengan mengunjungi pasar kosambi jika jasa tertentu. Hayati (42 Tahun) juga mencari alternatif lain dengan mengunjungi pasar kordon dikarenakan stock dari pedagang sayur yang lewat seringkali tidak lengkap. Masalah tersebut mengakibatkan:

1. Tidak maksimalnya profit yang diperoleh pedagang keliling dikarenakan konsumen memilih alternatif lain.
2. Pedagang harus mengulangi rute karena tidak tahu lokasi pasti konsumen.
3. Tidak terpenuhinya supply dan demand dikarenakan tidak ada sistem koordinasi dan komunikasi yang efektif antara pedagang keliling dan konsumen.
4. Kerugian pedagang karena memiliki stock barang terlalu banyak di hari tertentu.

Dalam usaha mengatasi permasalahan yang belum terselesaikan tersebut, Penulis ingin memanfaatkan peran teknologi dengan mengembangkan sebuah aplikasi berbasis website mobile dengan menggunakan metode iterative incremental. Aplikasi ini dirancang untuk meningkatkan koordinasi antara pedagang keliling dan calon konsumen dimana aplikasi ini nantinya memberikan fitur-fitur seperti

informasi produk dan jasa pedagang untuk memesan barang/jasanya, request barang/produk melalui fitur chat, fitur share location, serta fitur review agar konsumen dapat memberikan ulasan dan saran yang dapat dilihat pedagang dan diharapkan dapat memberi peluang bagi pedagang keliling untuk terus meningkatkan layanan mereka. Sementara pedagang keliling dapat mengetahui permintaan konsumen guna menghindari stock yang terlalu sedikit atau terlalu banyak juga dapat mengetahui dimana saja barang dan jasa mereka dibutuhkan agar tidak perlu berkeliling ke rute yang tidak pasti juga melalui fitur chat dan share location dari calon konsumen. Dengan ini, diharapkan pedagang keliling dapat memperoleh profit yang maksimal.

Penelitian ini diharapkan dapat memberikan wawasan yang berharga tentang bagaimana aplikasi berbasis website mobile dapat memecahkan masalah yang ada antara pedagang dan konsumen yang lumrah terjadi. Selain itu, penelitian ini juga diharapkan dapat memberikan kontribusi terhadap pemahaman lebih mendalam untuk penelitian selanjutnya

II. KAJIAN TEORI

Pedagang keliling merupakan individu atau kelompok yang menjual barang dan jasa dengan berpindah-pindah dari satu lokasi ke lokasi lain, tanpa memiliki toko fisik tetap. Fleksibilitas ini memungkinkan mereka menjangkau daerah-daerah yang kurang terlayani, seperti pedesaan atau daerah terpencil. Mereka menawarkan berbagai barang seperti makanan, pakaian, dan layanan lain seperti perbaikan. Meski menghadapi tantangan dari regulasi dan persaingan dengan toko fisik serta e-commerce, beberapa pedagang keliling telah memanfaatkan teknologi digital untuk promosi dan penjualan online, menunjukkan kemampuan adaptasi dalam lanskap yang berubah.

E-commerce melibatkan pembelian dan penjualan barang melalui platform elektronik, yang memungkinkan komunikasi langsung antara produsen dan pembeli, mempermudah proses transaksi, dan memperluas jangkauan pasar. E-commerce menawarkan keuntungan bagi konsumen, produsen, dan penjual dengan mengurangi biaya dan waktu dalam transaksi.

Mobile computing adalah kemajuan teknis yang memungkinkan komunikasi nirkabel dan memberikan mobilitas tinggi. Ini mencakup perangkat seperti smartphone yang memainkan peran penting dalam mengatasi berbagai kendala mobilitas manusia. Smartphone dan perangkat mobile lainnya memungkinkan penggunaannya untuk berkomunikasi dan menjalankan berbagai fungsi yang sebelumnya terbatas pada komputer.

API adalah serangkaian aturan yang memungkinkan aplikasi berbeda berkomunikasi satu sama lain. API memainkan peran penting dalam pengembangan perangkat lunak modern dengan memungkinkan integrasi dan komunikasi antara berbagai sistem dan layanan. Contoh khususnya adalah Google Maps API, yang memungkinkan pengembang untuk mengintegrasikan peta dan data lokasi ke dalam aplikasi mereka.

Java adalah bahasa pemrograman serbaguna yang populer karena kemampuannya untuk dijalankan di berbagai platform tanpa perubahan kode. Dengan pendekatan

berorientasi objek, Java memudahkan pengembangan aplikasi yang lebih efisien, modular, dan aman.

Perangkat Pengembangan:

1. **UML (Unified Modelling Language):** Bahasa pemodelan visual yang digunakan untuk merancang dan mendokumentasikan sistem perangkat lunak yang kompleks.
2. **Visual Paradigm:** Alat untuk menggambarkan desain aplikasi menggunakan diagram yang mendukung pemodelan bisnis dan pengembangan aplikasi.
3. **Visual Studio Code:** Text editor yang mendukung berbagai bahasa pemrograman dan sistem operasi.
4. **React JS:** Library JavaScript untuk membangun antarmuka pengguna yang deklaratif dan berbasis komponen.
5. **Node JS:** Platform yang memungkinkan pengembangan server dengan JavaScript, mendukung pemrograman asinkron dan lintas platform.
6. **Express.js:** Kerangka kerja web berbasis Node.js yang mempermudah pengembangan aplikasi web dinamis.
7. **Supabase:** Platform backend-as-a-service yang menyediakan alat pengembangan API dan manajemen database.
8. **APIDog:** Platform untuk desain, pengujian, dan dokumentasi API dengan efisiensi yang tinggi.
9. **Ngrok:** Alat yang memungkinkan pengembangan mengakses aplikasi lokal melalui internet dengan aman.
10. **Blackbox Testing:** Metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi tanpa melihat struktur internalnya

III. METODE

Pengembangan Iterative Incremental atau berulang dan bertahap adalah proses pengembangan perangkat lunak yang berfokus pada penyampaian pekerjaan dalam porsi kecil dan dapat dikelola yang dikenal sebagai peningkatan. Pendekatan ini sering digunakan dalam manajemen proyek Agile, khususnya dalam metodologi Scrum, karena memungkinkan tim merespons perubahan dengan cepat dan terus meningkatkan produk mereka (Idratherbewriting, 2023). Dalam proses pengembangan yang berulang dan bertahap, tim pengembangan memecah proyek menjadi tugas atau peningkatan yang lebih kecil, dan setiap peningkatan dibuat berdasarkan versi sebelumnya. Hal ini memungkinkan adanya umpan balik dan perbaikan yang berkelanjutan, memastikan bahwa produk akhir lebih mungkin memenuhi persyaratan dan tujuan yang diinginkan (Scrum, 2022). Sifat berulang dari pendekatan ini membantu tim untuk menangani proyek yang kompleks dengan lebih baik, karena memungkinkan mereka menyempurnakan pekerjaan mereka dan melakukan penyesuaian berdasarkan umpan balik dan perubahan persyaratan (Scrum, 2022).

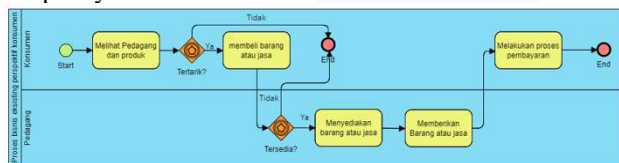
Proses pengembangan Iterative Incremental menjadi penting karena memberikan beberapa manfaat (Scrum, 2022): Dengan membagi proyek menjadi bagian-bagian yang lebih kecil, tim dapat menyelesaikan pekerjaan lebih sering, sehingga memungkinkan umpan balik dan penyesuaian yang

lebih cepat, Setiap kenaikan dapat dikelola, sehingga memudahkan tim untuk menangani dan menyelesaikan tugas. Dengan mengembangkan versi sebelumnya, tim dapat terus meningkatkan produk mereka, memastikan bahwa produk tersebut memenuhi persyaratan dan sasaran yang diinginkan. Proses pengembangan yang berulang dan bertahap memungkinkan tim merespons perubahan dengan cepat, sehingga produk akhir lebih mungkin memenuhi kebutuhan bisnis yang terus berkembang.

Maka dapat dipahami bahwa Iterative Incremental atau pengembangan berulang dan bertahap adalah proses pengembangan perangkat lunak yang berfokus pada penyampaian pekerjaan dalam porsi kecil yang dapat dikelola, sehingga memungkinkan adanya umpan balik dan perbaikan berkelanjutan yang memberikan beberapa manfaat, termasuk pengiriman yang sering, porsi yang dapat dikelola, perbaikan berkelanjutan, dan adaptasi yang lebih baik terhadap perubahan

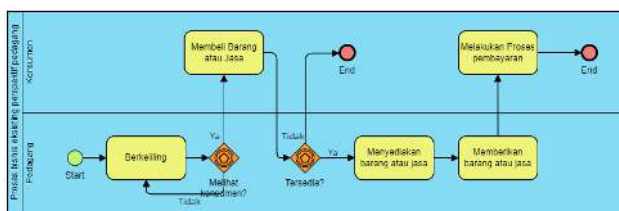
IV. HASIL DAN PEMBAHASAN

Pada bagian ini merupakan analisis bagaimana langkah-langkah yang dilakukan oleh Pedagang keliling untuk mendapatkan konsumen. Selanjutnya akan dirancang sebuah proses bisnis eksisting dimana proses bisnis ini menggambarkan serangkaian kegiatan yang dilakukan oleh aktor dalam menjalankan bisnisnya. Perancangan ini diharapkan dapat memahami semua aktivitas yang terjadi sehingga dapat memberikan saran dan perbaikan kedepannya.



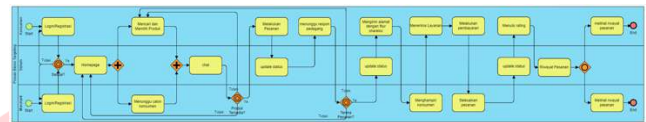
GAMBAR IV.1
Proses Bisnis Eksisting Prespektif Konsumen

Pada Gambar IV.1 menggambarkan mengenai proses bisnis eksisting dari prespektif konsumen. Pertama konsumen harus melihat atau mendengarkan tanda atau teriakan pedagang keliling, lalu konsumen memanggil pedagang untuk melakukan aktivitas pembelian dengan membeli barang atau jasa pedagang keliling yang tersedia, selanjutnya konsumen akan membeli barang atau jasa dimana jika tidak tersedia akan mengakibatkan aktivitas membeli akan selesai jika barang atau jasa tersedia selanjutnya pedagang akan menyediakan barang atau jasa sehingga konsumen dapat membelinya dan terakhir konsumen mendapatkan barang atau jasa dari pedagang keliling tersebut.



GAMBAR IV.2
Proses Bisnis Eksisting Prespektif Pedagang

Pada gambar menggambarkan mengenai proses bisnis dari prespektif pedagang keliling dimana pedagang keliling akan berkeliling ke daerah-daerah untuk mendapatkan konsumen, pada saat berkeliling konsumen akan memanggil pedagang keliling untuk membeli barang/jasanya, selanjutnya konsumen akan membeli barang atau jasa dimana jika tidak tersedia akan mengakibatkan aktivitas membeli akan selesai jika barang atau jasa tersedia lalu pedagang akan mempersiapkan barang atau jasanya kepada konsumen dan memberikannya, lalu konsumen akan menyiapkan uang untuk barang atau jasa tersebut dan pedagang keliling menerima uang tersebut dan aktivitas bisnis selesai.



GAMBAR IV.3
Proses Bisnis Targeting.

Pada modul aplikasi pedagang keliling terdapat konsumen, sistem, dan pedagang. Tahap pertama dalam proses ini dimulai dengan registrasi akun pedagang dan konsumen, user harus mendaftar dengan mengisi data yang dibutuhkan dalam proses registrasi dan menunggu sistem untuk memverifikasi akun mereka. Setelah pengguna melakukan registrasi maka pengguna akan melakukan login. Pada tahap selanjutnya untuk pedagang akan menunggu apakah ada konsumen yang akan memesan dan untuk konsumen akan ditampilkan sebuah menu atau halaman pedagang yang sedang tersedia dimana jika konsumen tertarik akan barang atau jasa pedagang, konsumen akan memilihnya dan langsung di arahkan pada halaman chat sehingga konsumen dapat mengirimkan alamat dan berkomunikasi kepada pedagang. Setelah selesai berdiskusi terkait apa yang dibutuhkan konsumen dan pedagang, jika konsumen memutuskan untuk membeli, mereka akan menekan tombol pesan dan sistem akan menerima permintaan pemesanan selanjutnya pedagang akan memeriksa ketersediaan produk atau jasanya. Jika produk tersedia, pedagang akan menyiapkan produk atau jasa yang dipesan dan pedagang mengkonfirmasi Penerimaan Pesanan, lalu sistem akan memperbarui status pesanan menjadi "dalam proses", konsumen dapat melihat pembaruan status pesanan ini melalui akun mereka. Setelah produk atau jasa siap, pedagang akan mengantarnya pada konsumen, melakukan proses transaksi, dan menyelesaikan pesanan. Setelah itu, konsumen dapat memberikan rating dan riwayat pesanan. Pedagang juga dapat melihat hasil rating yang diberikan konsumen. Kebutuhan pengguna dapat dilihat pada tabel berikut:

TABEL II.1
Kebutuhan Pengguna

No	Kebutuhan Pengguna	Solusi
1	Media untuk konsumen berkomunikasi langsung dengan pedagang agar tidak harus menunggu ketidakpastian lewatnya pedagang.	Aplikasi menyediakan fitur chat untuk konsumen dan pedagang melakukan komunikasi.
2	Media agar pedagang dapat mengetahui Dimana dan kapan saja jasa mereka dibutuhkan agar tidak perlu	Aplikasi menyediakan fitur chat dan share location untuk mengetahui posisi atau titik koordinat

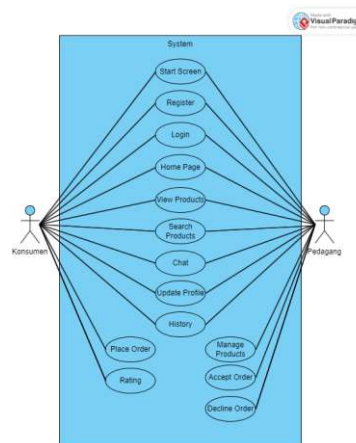
	berkeliling mengulangi rute yang sama dan hanya menunggu di satu Lokasi saja.	calon konsumen agar pedagang dapat berkomunikasi dengan cepat dan mudah.
3	Media untuk pedagang memaksimalkan keuntungan dan meminimalisir kerugian	Aplikasi menyediakan fitur produk agar konsumen dapat melakukan pre-order atau memesan sesuai deskripsi. Dan pedagang dapat membeli stock sesuai dengan request konsumen.
4	Pedagang menginginkan aplikasi yang simple dan mudah dipakai	Tampilan dan fitur pada aplikasi cukup sederhana sehingga memudahkan pedagang dalam memakai aplikasi.

Share Location	Fitur ini berada pada halaman chat dimana konsumen dapat memberikan titik lokasi mereka kepada merchant
Call	Fitur ini berada pada halaman chat dimana user dapat melakukan panggilan menggunakan nomor telepon seluler
Rating	Fitur Rating merupakan tempat untuk konsumen melakukan evaluasi dan umpan balik terkait layanan atau produk merchant. Fitur ini berada didalam halaman chat ketika pesanan telah selesai oleh merchant.
Profile	Halaman Profile merupakan tempat untuk menyediakan informasi, pusat manajemen akun, personalia user.
History	Halaman History Pesanan merupakan tempat untuk user melihat semua pesanan yang telah selesai dan rating yang diberikan.

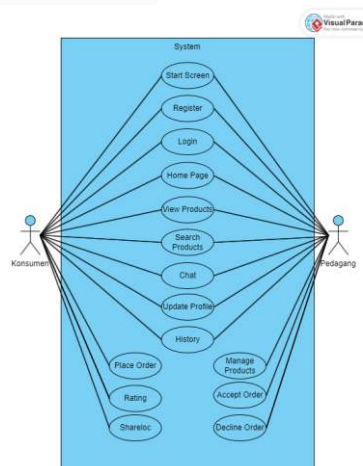
TABEL II.2
Analisis Fitur

Fitur	Deskripsi
Start Screen	Halaman Start Screen merupakan halaman pertama yang disajikan ketika pengguna mengakses sistem ini. Halaman ini berfungsi sebagai pemisah antara User dan merchant yang menampilkan dua buah button pedagang dan pengguna
Login	Halaman login merupakan halaman yang disajikan ketika konsumen menekan tombol masuk. Halaman ini berfungsi untuk menghubungkan antara isian username dan password dengan sistem yang diolah pada bagian controller
Registration	Halaman daftar merupakan halaman untuk bisa mengakses menu login. Disini Merchant akan mengisi data diantaranya nomor telepon, nama pengguna, dan kata sandi
Regis Success	Halaman Regist Success merupakan halaman ketika konsumen telah berhasil melakukan proses pendaftaran
Homepage	Halaman Homepage merupakan halaman ketika Merchant telah melakukan proses login dan daftar atau ketika user membuka aplikasi ini. Halaman Homepage berisikan fitur menambah produk, melihat semua produk, dan mencari produk, profile, chat, dan history.
Tambah produk	Halaman Produk Merchant merupakan halaman dimana merchant dapat menambahkan produk dan jasa yang mereka sediakan.
Edit Produk	Halaman Edit produk Merchant merupakan halaman dimana merchant dapat mengubah dan menghapus produk dan jasa yang telah mereka sediakan.
All Chat	Halaman All Chat merupakan halaman yang memberikan informasi berupa history chat user Dimana mereka dapat melihat keseluruhan chat berupa list
Chat Konsumen	Halaman Chat konsumen merupakan halaman dimana tempat konsumen dan merchant saling berkomunikasi, saling bertukar informasi, dan tempat terjadinya proses jual dan beli pada aplikasi ini. Disini konsumen dapat melakukan pemesanan, memberi rating, menelepon, dan mengirim lokasi mereka. Sedangkan merchant dapat menelepon, menerima pesanan, menolak pesanan, dan menyelesaikan pesanan.

Selanjutnya use case diagram akan mendeskripsikan sebuah interaksi antara satu atau lebih aktor sistem yang akan dibuat serta merupakan permodelan untuk tingkah laku (behaviour). Pada sistem ini use case diagram yang akan dibuat terdapat dua aktor, yaitu pedagang dan konsumen. Adapun use case diagram yang digunakan dalam sistem ini ditunjukkan pada Gambar IV.4.

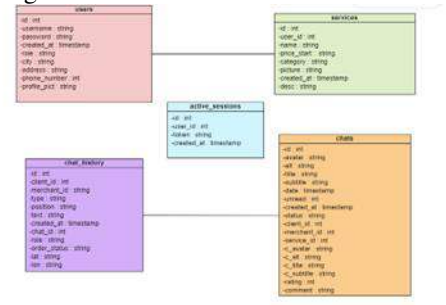


GAMBAR IV.4
Use Case Diagram iterasi pertama



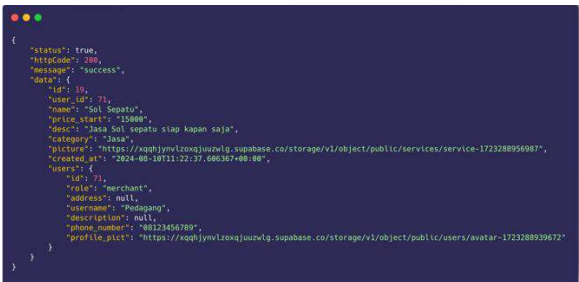
GAMBAR II.5
Use Case Diagram Iterasi kedua

Basis data digunakan sebagai media penyimpanan data yang kemudian akan ditampilkan guna untuk menghasilkan informasi. Dalam sistem ini perancangan basis data dibuat berdasarkan Class Diagram yang nantinya dapat digunakan dalam pembuatan tabel-tabel basis data. Class Diagram dapat dilihat pada gambar IV.5



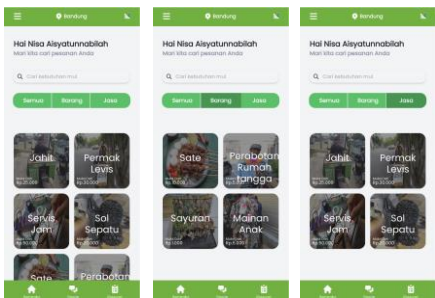
GAMBAR IV.5
Class Diagram

Pada tahapan ini sudah memasuki tahap konstruksi dan transisi interactive incremental fase pertama, pada fase pertama akan berfokus kepada fitur utama aplikasi. Dima menggunakan framework Node Js, Express Js, dan java sebagai bahasa dari pemograman dan juga menggunakan metode black-box-testing dan stress testing. Berikut adalah contoh salah satu fitur yang dikembangkan sesuai dengan kebutuhan sistem yang diperlukan untuk aplikasi ini.



GAMBAR IV.6
Hasil output API Memilih merchant

Gambar IV.6 merupakan output yang diberikan dari berhasilnya respon pada permintaan memilih merchant yang ada pada halaman homepage. Dimana pada data pada gambar memiliki atribut utama yaitu id, nama produk, kategori, harga, deskripsi, foto produk, dan timestamp. Halaman Homepage Konsumen merupakan halaman ketika konsumen telah melakukan proses login dan registrasi atau ketika user membuka aplikasi ini. Halaman Homepage berisikan tampilan pertama dari layanan, produk, dan fitur-fitur pada aplikasi. Tampilan Homepage Konsumen ditunjukkan pada gambar IV.7



GAMBAR IV.7
Halaman Homepage Konsumen

Setelah fase development, akan dilakukan proses testing fase pertama yang menggunakan metode dari black-box-testing dan stress testing. Tahap ini dilaksanakan setelah pengembangan selesai. Pada tahap ini bertujuan untuk mengetahui apakah semua fitur dapat berfungsi dengan baik dan memenuhi keinginan user dan dapat memenuhi tujuan dari pembuatan aplikasi ini sehingga dapat segera diimplementasikan dan akan dapat menjawab permasalahan yang ada.

TABEL IV.3
Hasil Blackbox Testing Berhasil Melakukan Pesanan

Skenario	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesesuaian
Pesanan Berhasil	1.Konsumen memilih produk atau jasa 2.Konsumen menekan tombol <i>contact the seller</i> 3.Berdiskusi melalui <i>chat</i> dengan <i>merchant</i> 4.Konsumen menekan tombol pesan 5. <i>Merchant</i> menekan tombol terima pesanan 6.Mengirim Alamat 7. <i>Merchant</i> mengunjungi Lokasi 8.Pesanan diproses 9.Pesanan selesai 10. <i>Merchant</i> menekan tombol selesaikan pesanan	Konsumen berhasil berkomunikasi dan memesan produk atau jasa dari <i>merchant</i> tanpa adanya kendala.	<i>Konsumen</i> berhasil memesan produk dari <i>merchant</i> namun terkendala komunikasi terkait Lokasi dikarenakan tidak adanya fitur gps pada aplikasi.	Sesuai
Merchant Menolak Pesanan	1. <i>User</i> mengakses aplikasi 2.Menekan Tombol Pengguna 3.Melakukan <i>login</i> 4.Memilih produk atau jasa 5.Berdiskusi melalui <i>chat</i> 6.Produk atau layanan tidak tersedia 7. <i>Merchant</i> menekan tombol tolak pesanan 8.Pesanan dibatalkan	<i>User</i> gagal melakukan pesanan	<i>User</i> gagal melakukan pesanan	Sesuai

Setelah melakukan blackbox testing, akan dilakukan stress testing. Tahap ini bertujuan untuk menguji kinerja sistem ketika berada dalam kondisi yang ekstrim seperti simulasi kondisi di mana banyak pengguna mengakses aplikasi secara bersamaan. Tujuannya adalah untuk memastikan sistem dapat beroperasi secara stabil dalam kondisi tersebut. Berikut adalah hasil dari stress testing menggunakan Jmeter

TABEL IV.4
Stress Testing 100 Samples

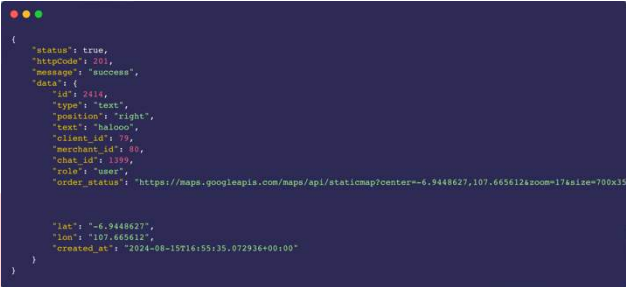
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
/login	100	1772	1802	2233	2342	2503	1019	2682	0%	33.875	32.19	6.91
/logout	100	7	6	12	17	26	0	28	0%	91.743	87.35	18.81
/client-all-order/id	100	380	332	594	865	1005	122	1183	0%	67.659	214.80	9.18
/client-add-rating/id	100	281	271	422	482	504	99	532	0%	63.451	59.30	15.74
/api/new-order-message/id	100	279	264	399	489	547	113	664	0%	64.020	60.46	14.32
/new-message	100	281	255	421	477	602	103	685	0%	70.028	80.35	28.65
/client-new-chat	100	241	196	389	484	585	94	605	0%	61.012	71.91	34.20
/users	100	273	238	408	513	601	111	637	0%	48.851	75.81	5.96
/user/id	100	242	224	375	399	504	106	575	0%	49.358	46.13	6.12
/user/update/id	100	248	222	413	464	598	93	628	0%	54.171	50.63	15.24
/user/role/role	100	256	221	413	487	643	87	681	0%	55.772	71.46	7.52
/services	100	252	225	395	419	523	82	620	0%	61.690	119.04	7.71
/services/id	100	218	187	355	396	450	77	535	0%	66.666	62.30	8.53
/services/update/id	100	173	155	262	296	377	62	387	0%	76.687	71.67	25.09
/services/category/category	100	154	158	228	270	282	61	282	0%	95.785	90.45	13.47
TOTAL	1500	337	223	500	1429	2185	0	2682	0%	247.524	309.03	55.60

TABEL IV.5
Stress Testing 200 Samples

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
/login	200	914	1001	1137	1163	1363	472	1688	0%	107.18	101.84	21.88
/logout	200	85	87	104	111	114	49	115	0%	245.09	233.37	50.26
/client-all-order/id	200	367	371	420	448	471	290	513	0%	169.92	539.47	23.07
/client-add-rating/id	200	385	386	463	478	511	230	515	0%	173.16	161.83	42.95
/api/new-order-message/id	200	426	412	509	521	1176	271	1244	0%	97.087	91.68	21.71
/new-message	200	387	273	524	1559	1855	145	1961	0%	79.176	90.84	32.40
/client-new-chat	200	443	290	750	1620	1812	87	2090	0%	72.648	85.62	40.72
/users	200	468	280	1507	1774	1864	102	1992	0%	74.654	115.85	9.11
/user/id	200	426	228	1557	1761	1828	102	1854	0%	66.334	61.99	8.23
/user/update/id	200	392	206	1521	1674	1774	82	1789	0%	68.775	64.28	19.34
/user/role/role	200	295	214	420	655	1532	102	1654	0%	67.934	87.04	9.16
/services	200	268	213	438	558	1322	105	1355	0%	71.047	137.10	8.88
/services/id	200	246	210	429	498	537	70	565	0%	75.786	70.83	9.70
/services/update/id	200	208	190	313	370	525	50	577	0%	80.482	75.22	26.33
/services/category/category	200	197	199	285	304	372	51	404	0%	82.440	77.85	11.59
TOTAL	3000	367	268	680	1131	1774	49	2090	0%	432.900	540.47	97.23

Tahap selanjutnya adalah tahap evaluasi, pada tahapan ini mendapatkan umpan balik dari hasil pengujian yang telah dilakukan dengan pengguna aplikasi. Dimana dari hasil blackbox testing didapatkan sebuah masukan pada aplikasi dimana diperlukannya fitur konsumen mengirimkan alamat mereka secara langsung sehingga pedagang dapat mengetahui secara pasti alamat atau lokasi real time konsumen secara akurat. Lalu dari hasil Stress testing didapatkan bahwa aplikasi dapat berjalan dengan stabil ketika diuji dengan simulasi pertama sebanyak 100 pengguna virtual yang mengakses aplikasi, dan simulasi kedua sebanyak 200 pengguna virtual yang mengakses aplikasi, dilihat dari nilai Error % yang 0% menunjukkan bahwa semua permintaan berhasil diproses tanpa adanya kesalahan. Nilai throughput yang tinggi juga menunjukkan bahwa aplikasi mampu menangani banyak permintaan secara bersamaan. Waktu respon rata-rata bervariasi untuk setiap permintaan. Permintaan logout dan staging merchant memiliki waktu respon tercepat.

Setelah melakukan testing dan mendapatkan umpan balik, maka sampailah pada tahap iterative incremental tahap kedua. Pada tahapan ini berfokus kepada bagaimana mengembangkan aplikasi sesuai dengan keresahan pengguna sehingga dapat menjawab kebutuhan pengguna yang didapatkan dari hasil uji. Pengembangan yang terjadi pada tahap ini ialah penambahan fitur share Share Location di fitur chat sehingga dapat memudahkan pedagang mengetahui alamat konsumen dengan google maps API.



Gambar IV.8 merupakan output yang diberikan dari berhasilnya respon pada fitur share location yang ada pada halaman chat ketika konsumen dan merchant berdiskusi pada halaman chat. Dimana data pada gambar memiliki atribut utama yaitu id, type, position, text, id client, id merchant, id chat, role, order status, lat dan lon sebagai titik koordinat ketika menggunakan fitur share location, dan timestamp.



GAMBAR IV.9
Halaman Share Location

Pada Gambar IV.9 adalah tampilan dari fitur Share Location di chat dimana konsumen dapat mengirimkan alamatnya secara langsung kepada pedagang sehingga dapat mengetahui lokasi konsumen secara akurat. Selanjutnya peneliti akan kembali melakukan pengujian blackbox testing dan stress testing dari hasil pengembangan yang telah diperbaharui pada aplikasi, yaitu penambahan fitur Share Location di dalam fitur chat.

TABEL IV.6
Hasil Blackbox Testing Konsumen Share Location

Skenario	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesesuaian
Konsumen mengirim Lokasi	1.User mengakses aplikasi 2.Menekan Tombol Pengguna 3.Melakukan login 4.Memilih produk atau jasa 5.Berdiskusi dengan merchant 6.Menekan tombol Share Location 7.Menekan tombol pesan	User berhasil mengirim lokasinya dengan akurat	User berhasil mengirim lokasinya dengan akurat	Sesuai

	8.Menunggu merchant			
Merchant mendapatkan Lokasi konsumen	1.User mengakses aplikasi 2.Menekan Tombol pedagang 3.Melakukan login 4.Berdiskusi dengan calon konsumen 5.User meminta titik lokasi calon konsumen 6.Mendapatkan lokasi calon konsumen 7.Menekan tombol terima pesanan 8.User mengunjungi titik lokasi calon konsumen	User berhasil menerima lokasi dengan akurat	User berhasil menerima lokasi dengan akurat	Sesuai

TABEL IV.6

Hasil Stress Testing Share Location 100 Samples Iterasi Kedua

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
/new-message	100	221	218	270	274	284	141	285	0%	115,74	132,91	47,36
TOTAL	1500	301	201	401	1374	1992	48	2121	0%	283,879	378,79	63,76

TABEL IV.6

Hasil Stress Testing Share Location 100 Samples Iterasi Kedua

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
/new-message	200	425	408	569	956	1044	126	2626	0%	53,504	61,44	21,89
TOTAL	3000	446	289	1024	1248	2530	53	2743	0%	311,428	367,49	82,52

Pada tahapan ini merupakan hasil dari pengujian tahap kedua yang merupakan Final product. Secara keseluruhan fungsional aplikasi sudah berjalan sebagaimana seharusnya. Tidak ada tombol ataupun fitur yang error selama proses dilakukannya blackbox testing. Lalu dari hasil Stress testing didapatkan bahwa waktu respon rata-rata adalah 221ms, dengan median 218ms. 90% permintaan selesai dalam 270ms, dan 99% dalam 284ms. Nilai Error % yang 0% menunjukkan bahwa semua permintaan berhasil diproses tanpa adanya kesalahan. Ini menunjukkan kinerja yang umumnya baik dan stabil setelah ditambahkan fitur share location. Pengguna merasa puas dan terbantu akan aplikasi dimana fitur yang diberikan sesuai dengan output apa yang mereka keluhkesahkan sebagai pedangan keliling dan konsumennya. Dengan ini dapat disimpulkan bahwa aplikasi pedagang keliling ini sudah dapat diluncurkan kepada masyarakat.

V. KESIMPULAN

Penelitan ini bertujuan untuk memberikan kemudahan kepada pedagang keliling dan konsumen pedagang keliling untuk berkomunikasi dengan mudah. Pengembangan aplikasi menggunakan metode interactive incremental selama 2 fase pengulangan. Aplikasi pedagang keliling ini menggunakan Node Js, Express Js, dan diuji dengan metode Blackbox Testing dan Stress Testing. Berdasarkan hasil pengujian dan analisa yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

1. perancangan aplikasi memiliki tujuan untuk mengembangkan fitur yang dapat memudahkan berkomunikasi antara pedagang keliling dan konsumen dengan memberikan fitur yang telah dibuat. Fitur yang diberikan dan dikembangkan sudah berjalan dengan baik

2. Penelitian ini menunjukkan bahwa metode interactive incremental memiliki nilai penting dalam mendukung pengembangan aplikasi. Pendekatan ini memungkinkan peningkatan efisiensi dalam pengembangan aplikasi, penyesuaian terhadap pengguna aplikasi, dan dapat menghasilkan produk sesuai dengan masukan pengguna.
3. Pengujian dan evaluasi menggunakan BlackBox Testing dapat memberikan hasil yang memuaskan dalam hal pengujian aplikasi sehingga dapat memberikan pengalaman yang memuaskan kepada pengguna. Dengan menggunakan metode testing ini aplikasi dapat dikembangkan secara berkelanjutan dan memberikan hasil yang memuaskan pengguna. Lalu pengujian Stress Testing menggunakan Jmeter menunjukkan sistem secara keseluruhan sudah cukup baik dan stabil dalam menangani permintaan 100 sampai 200 pengguna virtual yang mengakses aplikasi secara bersamaan.

REFERENSI

[1] Ahdan, S. (2015). Stress Testing To The Network Topology Using Ns2 Modeling And Simulation Of Network.

[2] Ali Ibrahim, A. (n.d.). Rancang Bangun Aplikasi E-Commerce Furnitur Berbasis Android. JUIT, 2(2).

[3] Afser, A. (2023). 10 Best Practices to Secure Your Node.js Application in Production

[4] Abdillah, M. F., Triayudi, A., & Hayati, N. (2023). Sistem Aplikasi Pelaporan Pelanggaran dalam Pemilihan Umum di Kota Depok Berbasis Web menggunakan Node. js. KLIK: Kajian Ilmiah Informatika dan Komputer, 3(6), 1226-1233.

[5] Andrianto, L. D., & Suyatno, D. F. (2024). Analisis Performa Load Testing Antara Mysql Dan Nosql MongoDB Pada RestAPI Nodejs Menggunakan Postman. Journal of Emerging Information System and Business Intelligence (JEISBI), 5(1), 18-26.

[6] Bachtiar, D. H., Paniran, P., & Suksmadana, I. M. B. (2024). Perancangan Back-end Api pada Aplikasi Mobile Fruityfit Menggunakan Framework Express JS. Mars: Jurnal Teknik Mesin, Industri, Elektro Dan Ilmu Komputer, 2(3), 107-117.

[7] Bastari, D. I., Pradana, F., & Priyambadha, B. (2017). Pengembangan Sistem Pembelajaran Pemrograman Java yang Atraktif Berbasis Website. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 1(12), 1493-1499

[8] Barus, A. C., Harungguan, J., & Manulu, E. (2021). Pengujian api website untuk perbaikan performansi aplikasi ditunen. Journal of Applied Technology and Informatics Indonesia, 1(2), 14-21.

[9] Buhl, A., Schmidt-Keilich, M., Muster, V., Blazejewski, S., Schrader, U., Harrach, C., & Süßbauer, E. (2019). Design thinking for sustainability: Why and how design thinking can foster sustainability-oriented innovation development. Journal of Cleaner Production, 231, 1248–1257.

[10] Budiman, E., & Hairah, U. (2017). End-to-End QoS tool development and performance analysis for network mobile. International Journal of Scientific Research in

- Science, Engineering and Technology (IJSRSET), 3(2), 128-135.
- [11] Costa, C., Ha, J., & Lee, S. (2021). Spatial disparity of income-weighted accessibility in Brazilian Cities: Application of a Google Maps API. *Journal of Transport Geography*, 90, 102905.
- [12] Dam, R. F. (2023). The 5 Stages in the Design Thinking Process. In *Interaction Design Foundation - IxDF*.
- [13] Dewi, C., & Sasongko, A. W. (2018). Sistem Pelaporan Infrastruktur Dinas Bina Marga Dan PSDA Kota Salatiga Menggunakan NodeJs Berbasis Web. *Indonesian Journal of Computing and Modeling*, 1(1), 10-17.
- [14] Eric, T.C. (2021). Your first steps with Express.js
- [15] Gulo, E., & Ferdiansyah, I. (2024). Pengujian Performa E-commerce Meningkatkan Skalabilitas dan Responsivitas Menggunakan Jmeter. *Kohesi: Jurnal Sains dan Teknologi*, 3(12), 1-10.
- [16] Hartolo, RA, Febriansyah, FE, Pribadi, IA, & Favorisen RL, (2023). Sistem Web Real Time untuk Pelacakan Lokasi Pedagang Keliling. *Jurnal Pepadun*, Vol. 4No. 2, 2023, pp. 166-173
- [17] Henim, S. R., & Sari, R. P. (2020). User Experience Evaluation of Student Academic Information System of Higher Education Using User Experience Questionnaire. *Jurnal Komputer Terapan*, 6(1), 471553.
- [18] Irviani, R., Setyorini, E., Muslihudin, M., Manajemen Informatika, P., Pringsewu, S., Sistem Informasi, P., & Pringsewu Jl Wisma Rini No, S. (2018). Perancangan Aplikasi E-Commerce Berbasis Android Pada Kelompok Swadaya Masyarakat Desa Margakaya Pringsewu. *Jurnal Ilmiah Ilmu Komputer*, 4(1).
- [19] Joseph, W. (2024). Exploring Supabase: Self-Hosting & GraphQL Integration
- [20] Krisnada, F. E., & Tanone, R. (2019). Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis Mobile menggunakan Flutter. *Jurnal Teknik Informatika Dan Sistem Informasi*, 5(3).
- [21] Kuang, Z., Li, L., Gao, J., Zhao, L., & Liu, A. (2019). Partial offloading scheduling and power allocation for mobile edge computing systems. *IEEE Internet of Things Journal*, 6(4), 6774–6785.
- [22] Mao, J., Chen, X., Nixon, K. W., Krieger, C., & Chen, Y. (2017). Modnn: Local distributed mobile computing system for deep neural network. *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1396–1401.
- [23] Maulana, H., Kasmawi, K., & Enda, D. (2020). Buku Penghubung Berbasis Android Menggunakan Metode Prototyping. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(3).
- [24] Maulana, M. (2023). Black Box Testing Adalah: Pengertian dan Contohnya. *ITBox Blog Cyber Security*.
- [25] Mc Gee, Y. (1977). Hawkers in Southeast Asian Cities: Planning for the Bazaar Economy.
- [26] Natsir, M. (2017). Pengembangan Prototype Sistem Kriptografi Untuk Enkripsi Dan Dekripsi Data Office Menggunakan Metode Blowfish Dengan Bahasa Pemrograman Java. vol, 6, 87-105.
- [27] Meol, EY, Nababan, D, & Kelen, YPK (2024). Sistem Informasi Penjualan Ikan pada Kefamenanu Berbasis Android Menggunakan Metode Waterfall. *Jurnal Krisnadana*, vol 3, 2
- [28] Nugraha, H., Nurhidayah, F., & Angelina, D. (2021). Desain Gerobak Pedagang Keliling pada Masa New Normal Covid-19. *Jurnal Desain Idea: Jurnal Desain Produk Industri Institut Teknologi Sepuluh Nopember Surabaya*, 20(2), 72–79.
- [29] Oktavia, W. (2018). Penamaan bunyi segmental dan suprasegmental pada pedagang keliling. *Jurnal Bahasa Lingua Scientia*, 10(1), 1–16.
- [30] Ozkaya, M. (2019). Are the UML modelling tools powerful enough for practitioners? A literature review. *IEt software*, Wiley Online Library.
- [31] Panjaitan, J., & Pakpahan, A. F. (2021). Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1).
- [32] Permana, Y. A., & Rohadlon, P. (2019). Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode Sdlc Pada Pt. Mandiri Land Prosperous Berbasis Mobile, 10(2).
- [33] Putra, C (2021). Rancang Bangun Sistem Informasi Penjualan Sayur Online Berbasis Android. *Jurnal Sistem Informasi dan Sains Teknologi Vol.3 No.1*
- [34] Randula, K. (2020). Connecting Azure DevOps Repo with Visual Studio Code.
- [35] Rodrigo, A.A. (2022). JMeter good practices. *onesalt platform blog*.
- [36] ReqTest, (2019). Stress Testing Tutorial: Understanding the Basics.
- [37] Saputra, A. (2019). Sistem Penjualan Pedagang Keliling Berbasis Geolocation (Vol. 1).
- [38] Sari, Y., & Riyansah, H. (2021). Aplikasi Tracking Pedagang Keliling Dengan GPS Google Maps API Berbasis Android. *IKRA-ITH Informatika: Jurnal Komputer Dan Informatika*, 5(3), 178–191.
- [39] Safitri, R., Setiawan, H., Ariyanti, N., & Rohman, D., (2023). Rancang Bangun Aplikasi Notifikasi Dan Geolocation Pada Pedagang Keliling Terdekat Berbasis Android. *Jurnal Umkendari Vol.4, No. 1*, 52-64.
- [40] Sibarani, N. S., Munawar, G., & Wisnuadhi, B. (2018, October). Analisis performa aplikasi android pada bahasa pemrograman java dan kotlin. In *Prosiding Industrial Research Workshop and National Seminar (Vol. 9, pp. 319-324)*.
- [41] Suhasini, G. (2023). What is iterative and incremental development? Process, examples.
- [42] Syamsudin, A. (2020). Analisis Kesalahan Coding Pemrograman Java pada Matakuliah Algoritma Pemrograman Mahasiswa Tadris Matematika IAIN Kediri. *Journal Focus Action of Research Mathematic (Factor M)*, 2(2), 102-114.
- [43] Widyanto, A., Aprizal, Y., Akmal Wardani, K., & No, J. J. B. R. Pengenalan dan Pengaplikasian Tunelling (ngrok. com) Bagi Siswa SMA Guna Mengakses Aplikasi Berbasis Web.