Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
https://ojs.stmik-banjarbaru.ac.id/index.php/jutisi/index
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0893

Model Aplikasi Sistem *E-Commerce* untuk Rekomendasi dan Pemantauan Harga Produk Penjual

DOI: http://dx.doi.org/10.35889/jutisi.v14i3.3293

Grace Levina Dewi[1*], Brigitta Angeline[2], Andrew Jonathan Hadikusuma[3], Kevin Setiono[4], Mikhael Setiawan[5], Suhatati Tjandra[6], Hendrawan Armanto[7]
[1,2,4,5,6,7]Informatika, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Indonesia
[3]Information and Technology Management Program, Ming Chuan University, Taoyuan City, Taiwan
*e-mail *Corresponding Author.* gracelevina@stts.edu

Abstract

Sellers on e-commerce platforms face difficulties in determining strategic pricing due to dynamic competition. To address this problem, an integrated system application model was developed as a solution to assist sellers. This research utilized the Research and Development (R&D) methodology with a waterfall model, which included the stages of needs analysis, system design, implementation, performance verification and data analysis, and final evaluation and revision. The result is a functional application capable of providing features such as search, strategic price recommendations at three levels (lowest, average/middle, highest), viewing more detailed information about products sold in graphical form, and other features. Usability testing conducted on target users yielded average scores of 95.54% for free-type customers, 91.26% for premium-type customers, and 96.8% for buyer-type cutomers. This indicates excellent ease of use and a high level of acceptance. This website model proved to be an effective and practical tool for sellers to establish competitive pricing strategies.

Keywords: *E-commerce; Price Monitoring; Strategic Pricing; Research and Development; Recommender System*

Abstrak

Penjual pada platform e-commerce menghadapi kesulitan dalam menentukan harga strategis akibat persaingan dinamis. Untuk mengatasi masalah ini, sebuah model aplikasi sistem terintegrasi dikembangkan sebagai solusi untuk membantu penjual. Penelitian ini menggunakan metodologi *Research and Development* (R&D) dengan *waterfall* yang mencakup tahap analisis kebutuhan; perancangan sistem; implementasi; verifikasi kinerja dan analisis data; dan evaluasi akhir dan revisi. Hasilnya adalah sebuah aplikasi fungsional yang mampu menyediakan fitur pencarian, rekomendasi harga strategis pada tiga level (terendah, rata-rata/menengah, tertinggi), melihat informasi lebih detail mengenai produk yang dijual dalam bentuk grafik, dan fitur lainnya. Pengujian usabilitas menggunakan usability testing terhadap target pengguna menghasilkan skor rata-rata 95,54% pada customer yang bertipe gratis, 91,26% untuk pengguna bertipe premium, dan 96,8% pengguna yang bertipe pembeli. Hal ini yang mengindikasikan kemudahan penggunaan dan tingkat penerimaan sangat baik. Model website ini terbukti menjadi alat bantu yang efektif dan praktis bagi penjual untuk menetapkan strategi harga kompetitif.

Kata kunci*: E-commerce; Pemantauan Harga; Penentuan Harga Strategis; Research and Development; Sistem Rekomendasi*

1. Pendahuluan

E-commerce telah menjadi pilar utama dalam ekonomi digital saat ini, dengan pertumbuhan yang dipercepat oleh meluasnya akses internet dan perubahan perilaku konsumen sejak pandemi. Meskipun hal ini membuka peluang pasar global bagi para penjual, lingkungan yang hiper-kompetitif ini juga menghadirkan satu tantangan besar, yaitu bagaimana

cara untuk menentukan harga produk yang tepat sehingga pengguna juga puas dengan harga yang diberikan. Di berbagai *marketplace*, harga bersifat sangat dinamis karena para pesaing terus melakukan penyesuaian melalui diskon dan promosi. Jika penjual memantau harga secara manual (dengan cara membuka banyak situs dan mencatatnya satu per satu) maka proses ini tidak hanya memakan banyak waktu dan tenaga saja, tetapi juga sangat rentan terhadap kesalahan manusia [1]. Kesalahan kecil dalam penetapan harga ini dapat berakibat fatal, seperti kehilangan pelanggan karena harga yang terlalu mahal atau kehilangan keuntungan karena harga yang terlalu murah dalam menjual produk.

Dalam mengatasi masalah tersebut, banyak penelitian sebelumnya menawarkan solusi yang canggih dengan menggunakan teknologi seperti *machine learning* atau analisis prediktif untuk menciptakan sistem harga yang dinamis (*dynamic pricing*) [2]. Meskipun solusi-solusi ini secara teori sangat efektif, pada praktiknya sistem tersebut sangat kompleks dan mahal. Implementasinya seringkali sulit dijangkau oleh UKM (Usaha Kecil dan Menengah). Hal ini disebabkan karena dibutuhkan biaya infrastruktur yang tinggi, keahlian khusus di bidang data, dan data historis dalam jumlah besar. Adanya kesenjangan antara solusi ideal yang ditawarkan dan kemampuan nyata mayoritas penjual. Inilah yang mendasari perlunya sebuah pendekatan yang lebih sederhana, praktis, dan terjangkau.

Solusi untuk menjawab kebutuhan tersebut, pada penelitian ini mengembangkan sebuah aplikasi berbasis website yang fungsional dan kemudahan dalam penggunaannya. Pemilihan platform website ini di dasarkan pada kemampuannya yang dapat diakses secara luas tanpa instalasi tambahan, sehingga efektif untuk pengguna dengan berbagai perangkat dan sistem operasi [3]. Aplikasi ini menggunakan dua pendekatan utama untuk mengumpulkan data harga melalui integrasi API (*Application Programming Interface*) untuk menarik data secara resmi dari *marketplace* yang menyediakannya dan menggunakan teknik web *scrapping* sebagai alternatif untuk mengambil data dari *platform* yang tidak memiliki API publik [4]. Pendekatan tersebut rasional karena integrasi API memastikan akurasi dan kecepatan data secara *real-time*, sedangkan web *scraping* berfungsi sebagai mekanisme *fallback* untuk menjaga kontinuitas data ketika API tidak tersedia. Penelitian sebelumnya juga menunjukkan bahwa kombinasi ini dapat meningkatkan kelengkapan data dan efisiensi proses ekstraksi informasi pada domain *e-commerce* [5]. Dari sisi teknologi, aplikasi ini dirancang dengan arsitektur modern yang berbasis Node.Js dan Express.Js pada sisi *back-end* untuk mendukung *asynchronous processing* dan *high concurrency* yang terbukti efektif menangani banyak permintaan data dalam skala besar [6]. Pada sisi antarmuka, digunakan Vue.Js dan Boostrap untuk menciptakan tampilan yang interaktif dan responsif di berbagai perangkat [7]. Seluruh data, mulai dari informasi produk hingga riwayat harga, disimpan dan dikelola menggunakan database MySQL yang memiliki kestabilan tinggi dan kompatibilitas luas dalam aplikasi berbasis web [8]. Dengan kombinasi pendekatan teknologi dan desain ini, website diharapkan mampu menghadirkan solusi yang efisien, andal, dan mudah digunakan. Hal ini juga telah direkomendasikan dalam penelitian sebelumnya terkait pengembangan sistem web fungsional dan berpusat pada user [9], [10].

Tujuan utama dari penelitian ini adalah membantu para penjual dengan menampilkan tiga rekomendasi harga produk yang sesuai dengan pencarian produk. Tiga rekomendasi yang dimaskud adalah harga tertinggi, rata-rata/menengah, dan terendah yang telah melakukan transaksi. Tujuan lain yang ingin dicapai adalah membantu para penjual dengan menampilkan grafik pemantauan harga yang sesuai dengan produk yang diinginkan dan menerima email dari hasil pemantauan harga produk secara otomatis. Fitur-fitur utama yang telah disiapkan untuk membantu para penjual melakukan pemantauan terpusat dengan menampilkan hasil pencarian harga produk dari berbagai *marketplace*, menyajikan detail data dari produk dalam bentuk grafik sehingga penjual dapat dengan mudah melihat pola dan fluktuasi harga dari waktu ke waktu, memberikan rekomendasi harga dalam tiga kategori berdasarkan data transaksi yang berhasil terjual produknya, serta mengirimkan pemberitahuan melalui email jika terjadi perubahan yang signifikan pada produk yang dipantau. Dengan fitur-fitur tersebut, aplikasi ini diharapkan dapat menjadi alat bantu yang efektif dan efisien. Harapannya website ini dapat memungkinkan penjual menetapkan harga yang kompetitif tanpa harus melalui proses manual yang rumit.

2. Tinjauan Pustaka

Banyak penelitian tentang sistem rekomendasi di *e-commerce* bertujuan untuk menciptakan keuntungan bagi penyedia layanan. Penelitian yang dilakukan oleh Alvise De

Biasio dkk memiliki kontribusi utama dengan mengusulkan metode untuk mengintegrasikan pertimbangan keuntungan secara langsung ke dalam algoritma sistem rekomendasi (tidak hanya sebagai langkah akhir saja) [11]. Tujuannya adalah untuk menciptakan rekomendasi yang seimbang antara relevansi bagi konsumen dan keuntungan bagi penyedia layanan *e-commerce*. Pendekatan yang dilakukan terbukti lebih efisien dibandingkan dengan metode tradisional. Penelitian yang dilakukan oleh Nada Ghanem dkk memiliki kontribusi utama dengan menggunakan simulasi untuk menganalisis dampak jangka panjang dari berbagai strategi rekomendasi [12]. Temuan utamanya adalah strategi rekomendasi menyeimbangkan antara kepuasan konsumen dan profitabilitas bisnis untuk menghasilkan keuntungan tertinggi dalam jangka panjang. Hal ini membuktikan pentingnya kepercayaan konsumen untuk keberhasilan bisnis yang berkelanjutan.

Di luar orientasi keuntungan, pada penelitian yang ada juga membahas aspek teknis integrasi. Salah satu teknologi yang digunakan adalah API yang dimanfaatkan untuk menjadi jembatan standar yang memungkinkan berbagai aplikasi bertukar data secara aman dan efisien. Dengan adanya API, sebuah sistem dapat memanfaatkan layanan eksternal tanpa perlu mengetahui detail teknis internalnya sehingga dapat mempercepat pengembangan dan memudahkan dalam pemeliharaan [5]. Namun Ketika API resmi tidak tersedia, teknik *web scrapping* dapat digunakan sebagai alternatif untuk mengekstrak informasi terstruktur (seperti nama produk, harga, dan ulasan) langsung dari halaman web [13]. Web *scrapping* memiliki tantangan tersendiri, seperti perubahan tata letak web, kebijakan akses dari pemilik web, dan kebutuhan untuk mengatur kecepatan permintaan agar tidak membebani server sumber data.

Pemilihan teknologi yang tepat juga sangat krusial dalam membangun aplikasi yang dapat diandalkan. Pada sisi server memanfaatkan teknologi seperti Node.Js menjadi pilihan yang popular karena kemampunanya dapat menangani banyak permintaan data secara bersamaan tanpa harus menunggu (*non-blocking I/O*), yang sangat cocok untuk aplikasi agregasi harga dari berbagai *marketplace* [6]. Framework Express.Js juga digunakan untuk melengkapi Node.Js dengan menyediakan struktur yang rapi untuk mengelola rute API, validasi data, dan penanganan *error* [14]. Pada sisi antarmuka dapat memanfaatkan Vue.Js karena Vue.Js ini memiliki keunggulan pada pembuatan aplikasi halaman tunggal (*Single Page Application*) yang cepat dan interaktif. Kemampuannya untuk memperbarui tampilan secara otomatis saat data berubah sangat membantu menjaga konsistensi informasi harga secara *real-time* [15]. Proses desain dipercepat dengan menggunakan Boostrap yang menyediakan komponen antarmuka siap pakai dan memastikan tampilan tetap optimal di berbagai layar [3].

Pada bagian manajemen data dan keamanan menggunakan MySQL karena memiliki keandalan dalam mengolah data terstruktur serta dukungannya terhadap konsistensi data transaksi yang penting untuk modul pemantauan harga [8]. Aspek keamanan dalam website, menggunakan Bcrypt. Bcrypt adalah sebuah metode *hashing* modern yang tahan terhadap serangan *brute force* [16]. Bcrypt ini dimanfaatkan dalam proteksi kata sandi yang digunakan pada website. Komponen pendukung fungsionalitas yang lain untuk fitur notifikasi email secara otomatis diimplementasikan menggunakan *Node Mailer*. *Node mailer* merupakan sebuah pustaka yang menyederhanakan proses pengiriman email transaksional melalui SMTP [17]. Untuk memperluas fungsionalitas ke arah transaksi komersial, integrasi dengan *payment gateway* seperti Midtrans dapat mempermudah proses penerimaan berbagai metode pembayaran, sangat relevan dengan konteks aplikasi di Indonesia [18].

Secara keseluruhan, literatur menunjukkan adanya pendekatan utama dalam pengembangan sistem *e-commerce*. Pendekatan utama ini berfokus pada optimasi algoritma yang kompleks untuk memaksimalkan keuntungan (dilakukan oleh Alvise De Biasio dkk) [11] dan Nada Ghanem dkk menekankan pada *profit-aware recommendation* melalui model prediktif dan simulasi berbasis *machine learning* [12]. Pendekatan ini memiliki tingkat kompleksitas implementasi yang tinggi dan memerlukan sumber daya komputasi yang besar, sehingga sulit diterapkan pada skala usaha kecil dan menengah. Sebaliknya, penelitian ini mengadopsi pendekatan pragmatis berbasis integrasi teknologi modern yang sudah teruji stabilitas dan skalabilitasnya (seperti API, web *scraping*, Node.JS, dan Vue.js). Pendekatan ini berfokus pada efektivitas penerapan dan kemudahan pemeliharaan sistem, bukan semata-mata pada aspek optimasi matematis [4], [13].

Dari sisi *state of the art*, penelitian ini berbeda dari studi terdahulu karena menggabungkan dua mekanisme pengumpulan data (integrasi API dan web *scapping fallback*) dalam satu arsitektur yang konsisten dan otomatis. Dimana hal tersebut jarang dibahas secara

integrasi dalam penelitian sebelumnya [5]. Selain itu dalam sistem juga dikembangkan dengan pendekatan *usability testing* dan pengujian *black box* untuk memastikan aspek fungsionalitas dan pengalaman pengguna dapat terukur secara kuantitatif [9],[10]. Dengan demikian, kebaruan dari penelitian ini terletak pada integrasi ganda antara API dan web *scraping* dalam satu kerangka otomatis yang adaptif terhadap ketersediaan sumber data, implementasi arsitektur *full stack modern* (Node.Js, Express.Js, Vue.Js) untuk efisiensi dan kecepatan pengembangan sistem *e-commerce* berbasis data harga, dan penerapan simultan antara *black box testing* dan *usability testing* untuk memastikan sistem berfungsi secara teknis serta efektif sesuai pengalaman pengguna. Pendekatan ini menunjukkan adanya kontribusi baru dalam menjembatani kesenjangan antara riset berbasis algoritmik yang kompleks dan kebutuhan praktis untuk solusi sistem yang mudah diimplementasikan, diukur, dan dipelihara (khususnya untuk kebutuhan data harga).

3. Metodologi

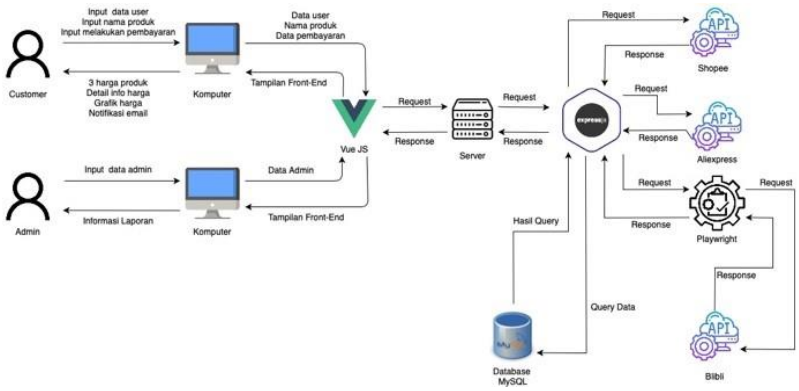
Penelitian ini menggunakan metode *Research and Development* (R&D) untuk pengembangan perangkat lunak dengan menggunakan model *waterfall* [19]. Model ini dipilih karena kebutuhan sistem telah ditetapkan secara stabil sejak awal, sehingga mendukung terciptanya fungsionalitas yang konsisten serta dokumentasi yang lebih terstruktur [20]. Selain meminimalkan resiko perubahan besar selama pengambangan, pendekatan ini diterapkan secara pragmatis dengan memberikan ruang revisi terbatas setelah tahap verifikasi. Berikut adalah prosedur dan tahapan yang dilalui:

1) Analisis Kebutuhan Sistem

Tahap ini bertujuan untuk menetapkan batasan dan spesifikasi sistem. Pengumpulan data dilakukan melalui beberapa metode, yaitu studi literatur, wawancara dengan penjual *e-commerce* untuk memahami kebutuhan nyata di lapangan, serta melakukan observasi cara para penjual tersebut melakukan pencarian produk di *marketplace*. Analisa yang dilakukan mencangkup analisa untuk kebutuhan fungsional dan kebutuhan non-fungsional. Pada kebutuhan fungsional yang dilakukan adalah menetapkan fungsi-fungsi inti seperti manajemen akun pengguna (gratis vs premium), mekanisme pencarian produk lintas *marketplace*, sistem rekomendasi harga, dan fitur notifikasi otomatis. Kebutuhan non-fungsional yang dilakukan dengan menetapkan kualitas sistem, termasuk antarmuka yang ramah pengguna, performa waktu respon yang cepat, dan jaminan keamanan data pengguna. Luaran yang dihasilkan dari tahap ini adalah dokumen spesifikasi kebutuhan perangkat lunak.

2) Perancangan Sistem

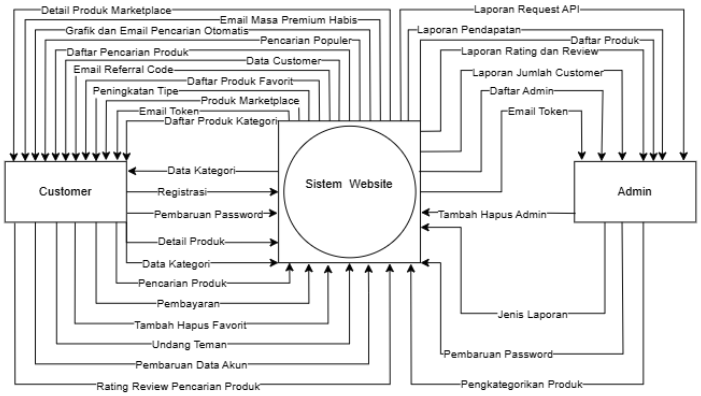
Berdasarkan dokumen spesifikasi kebutuhan perangkat lunak yang telah dibuat, selanjutnya adalah menerjemahkan spesifikasi kebutuhan tersebut dalam cetak biru. Proses perancangan dibagi menjadi tiga fase, yaitu: desain arsitektur sistem, desain proses sistem, dan desain database. Pada gambar 1 menunjukkan arsitektur sistem website dengan model *client-server*. Arsitektur ini dirancang dengan membagi pengguna ke dalam dua peran utama, yaitu: pengguna (*customer*) dan admin. Pada sisi antarmuka dibuat dengan menggunakan teknologi Vue.js dan server utama didukung dengan Node.js yang diperkuat dengan framework Express.js. Seluruh data disimpan dan dikelola menggunakan database MySQL.



Gambar 1. Arsitektur Sistem

Pada sisi pengguna terdapat dua tipe pengguna, yaitu gratis dan premium. Prosesnya dimulai saat pengguna mendaftar dengan nama dan email. Pada saat mendaftar, mereka akan menerima kata sandi unik berupa token yang dikirimkan ke email dengan menggunakan *Node Mailer*. Demi keamanan, token yang disimpan di database dienkripsi terlebih dahulu dengan library Bcrypt. Pada pengguna gratis dapat mencoba akses premium dengan dua cara. Cara pertama dengan mengaktifkan masa percobaan gratis yang hanya tersedia satu kali. Cara yang kedua dengan melakukan upgrade berbayar untuk menjadi akun premium yang berlaku selama satu bulan. Proses pembayarannya ini didukung dengan Midtrans yang melayani berbagai metode seperti transfer bank dan GoPay. Setelah masa premium berakhir, status akun akan otomatis Kembali menjadi gratis dan pengguna akan diinfokan melalui email. Berikut rincian fitur-fitur utama yang terdapat pada website berdasarkan tipe penggunanya:

a) Pengguna gratis: memberikan perbandingan harga komprehensif yang mengolah parameter pencarian (nama, berat, dan rentang harga) menjadi statistik harga terendah, rata-rata/menengah, dan tertinggi sekaligus dilengkapi kalkulasi harga *netto*. Sistem menyajikan visualisasi data yang fleksibel, baik secara terpadu maupun dikelompokkan per *marketplace*, dengan algoritma khusus pada platform Shopee yang menampilkan rekomendasi produk berbasis tingkat harga dan reputasi toko. Selain menyediakan fitur analisis tren pencarian teratas dan detail produk mendalam, website ini juga mengintegrasikan model keanggotaan gratis dengan mekanisme *referral* untuk insentif akses premium. Selain itu juga didukung fitur personalisasi yang mencakup manajemen produk favorit, riwayat aktivitas, dan sistem umpan balik pengguna.
b) Pengguna premium
Pengguna premium memiliki semua fitur pengguna gratis dengan tambahan beberapa fitur, yaitu: fitur yang mendukung pemantauan harga produk secara otomatis. Di mana pengguna dapat memilih produk tertentu untuk dipantau setiap hari dan hasilnya disajikan dalam bentuk grafik perubahan harga. Selain itu, grafik tren harga juga dapat dilihat langsung pada halaman detail produk untuk memudahkan analisis pergerakan harga. Bagi pengguna yang mengaktifkan fitur pemantauan otomatis, sistem akan mengirimkan hasil pemantauan ke email secara berkala. Ketika masa premium berakhir, status akun otomatis berubah menjadi gratis dan pengguna menerima email pemberitahuan sebagai informasi bahwa akses premium telah dihentikan.



Gambar 2. *Context Diagram*

Cara untuk menjalankan fungsi utama dalam pencarian produk ini, sistem mengandalkan pengambilan data dari berbagai *marketplace* melalui API. Data dari Shopee dan AliExpress akan ditarik menggunakan layanan RapidAPI. Pada *marketplace* Blibli sistem menggunakan API *back-end* dengan cara kerja yang berbeda. API tesebut memerlukan sebuah *browser* untuk membuka halaman produk terlebih dahulu agar data yang relevan dapat diekstrak, tidak seperti API Shopee dan AliExpress yang dapat mengambil daya secara langsung tanpa bantuan dari *browser*. Selanjutnya dibahas mengenai desain proses sistem dalam bentuk DFD (*Data Flow Diagram*). Pada gambar 2 ini merupakan *context diagram* dari website yang dibuat. Pada desain database, dilakukan perancangan struktur penyimpanan data menggunakan ERD (*Entity Relationship Diagram*). ERD digunakan untuk mendefinisikan entitas, atribut, dan relasi antar

entitas yang ada. Setelah ERD terbentuk, diturunkan menjadi skema tabel fisik. Luaran yang dihasilkan dari tahap ini adalah dokumen desain perangkat lunak yang berisi diagram arsitektur, diagram alur proses, skema database, *mockup* antarmuka pengguna (UI).

3) Implementasi Sistem

Tahap ini semua luaran dari desain dikembangkan dalam bentuk kode program fungsional. Proses implementasi ini meliputi pengembangan *server* dengan teknologi Node.Js dan Express.Js; pengembangan antarmuka menggunakan Vue.Js; pembuatan database dengan MySQL; implementasi dalam pengumpulan data primer dilakukan dengan mengintegrasikan website dengan API eksternal dari *marketplace* Shopee, Blibli, dan AliExpress; membangun *web scraping* dengan Playwright untuk memperoleh data harga dan produk; dan implementasi analisis produk dengan memanfaatkan algoritma di sisi server untuk mengolah data mentah yang telah dikumpulkan menjadi rekomendasi harga untuk tiga level (terendah, rata-rata/menengah, dan tertinggi). Luarannya adalah aplikasi fungsional tahap *alpha* yang siap untuk diverifikasi dan dilengkapi dengan dokumentasi teknis dasar.

4) Verifikasi Kinerja dan Analisis Data

Tahap ini bertujuan untuk memverifikasi apakah produk yang sudah dibangun sudah sesuai dengan spesifikasi dan memiliki kinerja yang baik. Mekanisme Verifikasi yang dilakukan dengan pengujian *black box testing* (memeriksa semua fitur berjalan dengan fungsinya dan tidak memeriksa struktur kode), *usability testing* (dilakukan *testing* pada 20 orang yang dibagi menjadi 10 orang pengusaha yang memiliki toko *online* dengan pembagian pengguna premium dan pengguna gratis, 10 orang yang tidak memiliki toko *online*/pembeli), dan mengukur respon dari *endpoint* API untuk masing-masing *marketplace*. Luaran dari tahap ini adalah laporan hasil pengujian yang mencangkup catatan *bug*, tabel skor usabilitas, dan analisa temuan lainnya.

5) Evaluasi Akhir dan Revisi

Pada penlitian ini, tahap evaluasi akhir dan revisi tidak dilaksanakan secara penuh. Evaluasi dilakukan sampai tahap verifikasi fungsional dengan *black box testing* dan verifikasi non fungsional menggunakan *usability testing*. Dari kedua hasil evaluasi tersebut telah menunjukkan bahwa sistem telah memenuhi kebutuhan dasar pengguna dan tujuan pengembangan sistem. Evaluasi performa terkait waktu respon API dan proses *scrapping* dilakukan secara terbatas untuk mengidentifikasi latensi dari API pihak ketiga.

4. Hasil dan Pembahasan

4.1. Hasil Penelitian

Hasil dari penelitian ini adalah sebuah website fungsional yang dapat digunakan untuk pemantauan dan rekomendasi harga. Pembahasan yang dilakukan pada hasil penelitian ini dibagi menjadi 2 sub subbab, yaitu: penjelasan tampilan antarmuka utama dari website untuk menggambarkan fungsionalitas utama yang telah dikembangkan.

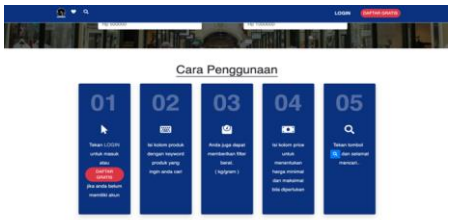
4.1.1. Tampilan Antarmuka Utama

Tampilan halaman home pada website terlihat sama untuk semua pengguna. Fitur pencarian dan ketegori produk hanya bisa diakses oleh pengguna yang telah login. Hal ini bertujuan agar pengunjung baru tetap dapat meilhat gambaran umum website tanpa akses penuh ke semua fitur. Pada gambar 3 terdapat tampilan halaman Home.

Gambar 3. Tampilan Halaman Home Bagian Atas

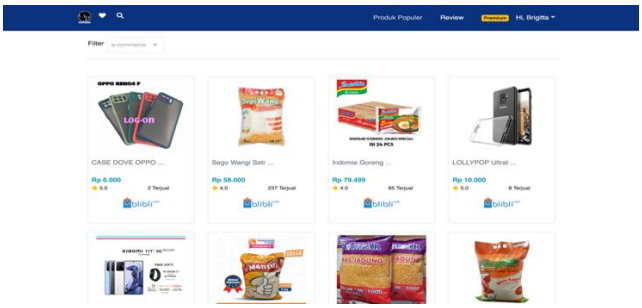
Pada gambar 4 merupakan tampilan halaman yang membedakan halaman ini dengan halaman home pengguna yang telah login. Perbedaannya tampak pada bagian petunjuk cara penggunaan. Pada halaman home (login dan belum login) dapat melihat informasi mengenai

API yang digunakan dan *marketplace* yang digunakan oleh website. Garis besar pada halaman home yang belum melakukan login adalah fitur-fitur pada halaman home tersebut mengarah ke halaman login dan terdapat petunjuk cara penggunaan website. Pada halaman home yang pengguna telah login, tidak terdapat petunjuk cara penggunaan dan terdapat kategori produk.



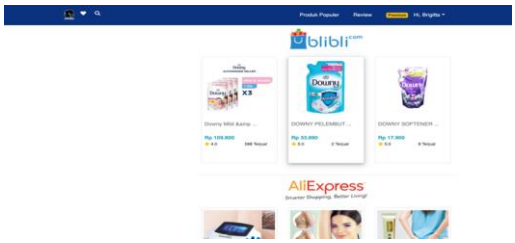
Gambar 4. Tampilan Halaman Home Bagian Tengah

Pada gambar 5 merupakan halaman tampilan produk yang ditandai dan dipantau harganya oleh pengguna lain. Pada halaman ini juga terdapat pilihan *marketplace*. Pengguna dapat memilih *marketplace* yang telah disediakan, yaitu Shopee, Blibli, dan Ali Express. Apabila pengguna melakukan pemilihan pada *marketplace*, maka daftar produk-produk tersebut akan berubah menjadi daftar produk populer dari *marketplace* yang telah dipilih pengguna.



Gambar 5. Tampilan Halaman Produk Populer

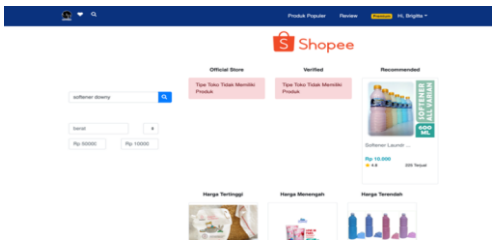
Setiap produk menampilkan gambar, nama, harga, rating, jumlah penjualan, dan asal *marketplace*. Detail seperti deskripsi, ulasan pembeli, dan varian produk, cukup dengan klik produk tersebut, pengguna dapat menandai atau mengikuti produk yang diminati saat melihat produk. Selain melihat produk populer, pengguna juga bisa mengakses daftar produk yang ditandai. Terdapat perbedaan penanganan pada produk-produk yang telah ditandai untuk pengguna gratis dan premium, dimana pengguna premium mendapat notifikasi email harian untuk pemantauan harga, sementara pengguna gratis harus mengecek langsung di website.



Gambar 6. Tampilan Halaman Hasil Pencarian di *Marketplace* Blibli dan AliExpress

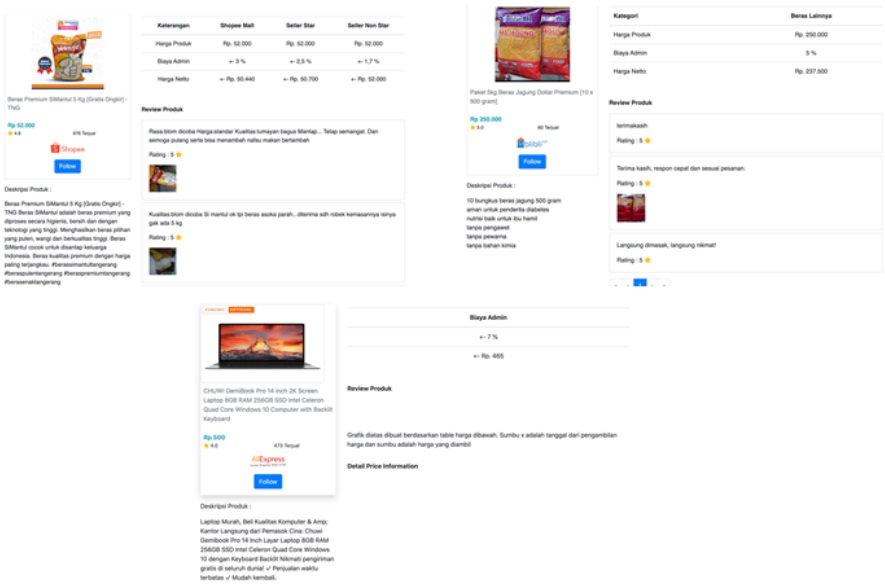
Pada gambar 6 merupakan tampilan dari halaman hasil pencarian di *marketplace* Blibli dan AliExpress beserta dengan ketiga harganya. Secara standar sistem, masing-masing *marketplace* akan menampilkan tiga produk hasil pencarian yang sesuai dengan hasil pengolahan pencarian harga produk. Pada masing-masing *marketplace* juga menampilkan tiga produk yang merupakan produk yang telah berhasil melakukan penjualan dengan harga terendah, rata-rata/menengah, dan harga tertinggi dari hasil pengolahan data. Pada gambar 7 merupakan tampilan khusus halaman *marketplace* Shopee. Hasil pencarian produk akan terdapat tiga produk yang ditampilkan, yaitu: 1 produk dari masing-masing tipe toko *official*

store, *verified*, dan *recommender* pada *marketplace* Shopee. Produk yang ditampilkan merupakan produk yang memiliki jumlah penjualan tertinggi dari data yang telah diolah oleh sistem website dari masing-masing tipe toko.



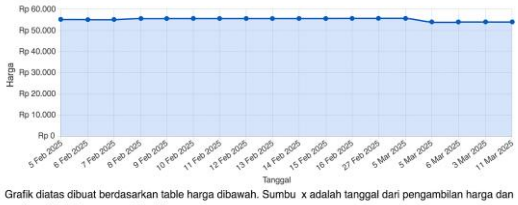
Gambar 7. Tampilan Halaman Hasil Pencarian di *Marketplace* Shopee

Pada gambar 8 merupakan tampilan halaman produk detail Shopee, Blibli, dan AliExpress. Pada masing-masing *marketplace* terdapat perbedaan perhitungan harga *netto* yang akan didapatkan oleh pengguna. Secara umum, harga *netto* akan berhubungan dengan kategori dari produk tersebut.



Gambar 8. Tampilan Halaman Produk Detail Shopee, Blibli, dan AliExpress

Pada gambar 9 menampilkan fitur pemantauan harga pada halaman detail produk yang eksklusif bagi pengguna premium. Fitur ini menyajikan data fluktuasi harga harian dalam bentuk grafik dan tabel untuk membantu pengguna menganalisis tren pasar sebagai dasar pengambilan keputusan penjualan. Visualisasi data historis ini tidak tersedia bagi pengguna dengan status akun gratis.



Grafik diatas dibuat berdasarkan table harga dibawah. Sumbu x adalah tanggal dari pengambilan harga dan sumbu adalah harga yang diambil

Detail Price Information

Created At	Price
5 Feb 2025	Rp. 56.000
6 Feb 2025	Rp. 56.000

Gambar 9. Contoh Tampilan Halaman Produk Detail Bagian Pemantauan Harga

4.1.2. Hasil Pengujian Sistem

Pengujian dilakukan untuk memverifikasi fungsionalitas dengan menggunakan *black box* testing, mengukur kualitas sistem dari perspektif pengguna dengan *usability testing*, dan mengukur respon dari *endpoint* API untuk masing-masing *marketplace*. *Black box testing* dilakukan oleh *developer*. *Black box testing* adalah uji coba yang fokus pada hasil *input* dan *output* sistem tanpa melihat struktur program [10]. Testing ini menitikberatkan pada fitur dan dilakukan setelah pengujian fungsi untuk memastikan semua berjalan sesuai dengan harapan. Pengujian *black box* ini difokuskan pada sisi pengguna.

Tabel 1. Pengujian *Black Box*

Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
Register	User mengisi form registrasi dengan nama dan email valid, lalu submit.	Sistem menyimpan data, mengirim email berisi token password, dan akun aktif.	Berhasil.
Login	User memasukkan email dan password yang valid.	Sistem memberikan ijin akses dan menampilkan dashboard utama.	Berhasil.
Forgot Password	User memasukkan email terdaftar dan meminta reset password.	Sistem mengirim email berisi token password baru.	Berhasil.
Change Password	User login lalu mengubah password lama ke password baru.	Password tersimpan dan dapat digunakan untuk login berikutnya.	Berhasil.
Edit Profile	User mengubah nama dan email akun di halaman profil.	Perubahan disimoan dan data baru muncul di profil.	Berhasil.
Mengundang Teman	User mengisi form undangan (nama dan email teman) lalu submit.	Email undangan dikirim ke penerima dengan kode refrerral.	Berhasil.
Percobaan Premium	User gratis mengaktifkan akses percobaan 7 hari.	Status akan berubah ke premium trial, durasi tercatat 7 hari.	Berhasil.
Menukar poin	User menukar 50 poin di halaman reward.	Sistem menukar poin dan mengaktifkan premium gratis 7 hari.	Berhasil.
Peningkatan ke premium	User memilih upgrade dan menyelesaikan pembayaran Midtrans.	Status berubah ke premium berbayar selama 30 hari.	Berhasil.
Rating Review Website	User mengisi form rating dan ulasan website.	Data rating dan review tersimpan dan tampil di halaman ulasan.	Berhasil.
Riwayat Pencarian	User membuka menu riwayat pencarian.	Sistem menampilkan daftar hasil pencarian sebelumnya.	Berhasil.
Pencarian Produk	User mencari produk berdasarkan kata kunci dan filter harga/berat	Sistem menampilkan hasil pencarian sesuai dengan kata kunci dan filter.	Berhasil.
Produk Detail	User memilih produk dari hasil pencarian.	Sistem menampilkan deskripsi, harga bersih, rating, dan review.	Berhasil.
Grafik Harga Produk (premium)	User premium membuka halaman detail produk.	Grafik harga muncul berdasarkan data pemantauan.	Berhasil.

Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
Produk populer	User membuka halaman produk popular.	Menampilkan 12 produk terpopuler (bisa difilter per *marketplace*)	Berhasil.
Katgori Produk	User membuka kategori produk melalui carousel atau list.	Menampilkan daftar kategori dan dapat difilter berdasarkan input.	Berhasil.
Daftar produk yang ditandai	User membuka halaman favorit produk.	Menampilkan produk yang ditandai dan bisa dihapus	Berhasil.
Pencarian Otomatis Harga Produk	Sistem melakukan pengecekan otomatis harga produk favorit.	Sistem mengirimkan notifikasi email harga terkini.	Berhasil.
Pengecekan masa berakhir premium	Sistem menjalankan proses pengecekan masa aktif premium.	Saat masa premium habis, status berubah ke gratis dan email notifikasi dikirim.	Berhasil.

Usability testing adalah metode evaluasi yang fokus pada pengalaman pengguna (UX) untuk memastikan aplikasi mudah dipahami dan digunakan [9]. Pengujian ini melibatkan pengguna potensial untuk menilai seberapa baik kegunaan website. Pengujian *usability* testing dilakukan dengan cara user melakukan uji coba pada aplikasi dan mengisi kuesioner. Kuesioner akan dibagi menjadi 5 bagian, yaitu: *learnability*, *efficiency*, *memorability*, *error*, dan *satisfaction*. Pengukuran kegunaan aplikasi dilakukan melalui tiga aspek, yaitu: efektivitas, efisiensi, dan kepuasan. *Usability testing* bertujuan menilai apakah pengguna dapat menyelesaikan tugas secara mandiri, mengukur kinerja dan kondisi mental saat menjalankan tugas (untuk menilai kualitas desain), mengetahui tingkat kenyamanan pengguna, mengidentifikasi masalah serta tingkat keparahannya, serta menemukan solusi atas masalah yang ditemukan.

Nilai *usability* dari website yang dibuat dihitung dari pertanyaan yang sudah diberikan kepada pengguna. Sebelum melakukan perhitungan, dilakukan menghitung nilai rata-rata untuk setiap pertanyaan yang telah diisi oleh pengguna. Dalam mentukan nilai rata-rata yang telah didapatkan dari hasil kuesioner, ditentukan dengan perhitungan rumus Skala *Likert pada persamaan 1*. Nilai Total adalah total hasil dari setiap pertanyaan pada semua responden, simbol Y adalah nilai pertanyaan yang paling tertinggi yaitu 5, dan simbol ∑n dalah jumlah total responden yang mengisi pertanyaan. Setelah menghitung nilai rata-rata pada setiap pertanyaan, lanjut untuk menghitung nilai *usability*.

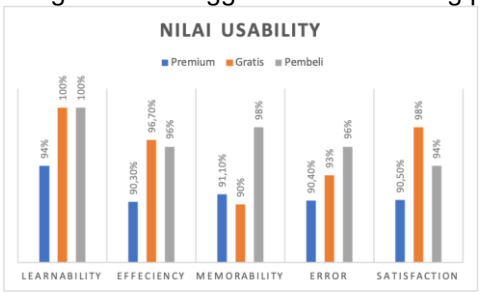
$$\text{Index}(\%) = \text{Nilai Total} / (Y \times \sum n) \times 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Persamaan 2 adalah rumus untuk mendapatkan *usability* dari website. Pertanyaan-Pertanyaan tersebut dikelompokkan menjadi 5 kelompok. Simbol A adalah presentasi nilai *Learnability*. Simbol B adalah presentasi nilai *Efficiency*. Simbol C adalah presentasi nilai *Memorability*. Simbol D adalah presentasi nilai *Error*. Simbol E adalah presentasi nilai *Satisfaction*. Perhitungan *usability* testing dilakukan pada hasil kuesioner yang telah diberikan. Untuk mendapatkan nilai dari masing-masing simbol, akan dilakukan perhitungan dengan persamaan 1.

$$\textit{Usability}(\%) = (A + B + C + D + E) / 5 \times 100\% \quad \ldots\ldots\ldots\ldots\ldots \quad (2)$$

Hasil kuesioner menunjukkan bahwa nilai *usability* website tergolong sangat baik meskipun masih ada kelemahan yang ditemukan. Kelemahan yang dimiliki website ini datang dari pengguna bertipe premium. Pada gambar 10 merupakan grafik nilai *usability* yang didapatkan. Jadi setelah dihitung dengan persamaan 2, nilai *usability* pengguna bertipe gratis memiliki nilai 95,54% dan pembeli sebesar 96,8%, yang artinya pengguna gratis dan pembeli merasa terbantu dengan adanya website ini. Nilai *usability* dari pengguna bertipe premium

mendapatkan nilai sebesar 91,26% artinya user sangat terbantu dengan adanya fitur-fitur premium yang ditawarkan. Kendala yang dihadapi oleh pengguna premium adalah loading hasil pencariannya terasa lambat. Kemudian produk yang dihasilkan dari *marketplace* AliExpress kurang banyak dan AliExpress sering sibuk sehingga membuat loading pencariannya lama.



Gambar 10. Grafik Nilai *Usability*

Selain itu juga dilakukan uji coba respon dari *endpoint* API. Pada uji coba ini berfokus kepada jumlah produk yang diterima dan waktu yang dibutuhkan untuk menerima seluruh produk dari hasil respon *endpoint search* API pada masing-masing *marketplace*. Pada bagian menampilkan hasil pencarian produk, memiliki proses menampilkan produk yang membuat pengguna harus menunggu terlebih dahulu. Hal tersebut dikarenakan menggunakan pihak ketiga yaitu RapidAPI. Jadi menggunakan pihak ketiga dengan pihak pertama akan lebih cepat yang pihak pertama.

Tabel 2. Respon API *Endpoint* Search

No	Keyword	*Marketplace*	Total Produk	Waktu (Menit)
1	Oppo Minimal : 1.000.000 Maksimal : 2.000.000	Shopee	150	0.17.56
		Blibli	80	0.03.20
		Ali Express	3	0.31.72
		TOTAL	**233**	**0.52.48**
2	Oppo Reno F4 Minimal : 2.000.000 Maksimal : 11.000.000	Shopee	150	0.12.52
		Blibli	80	0.03.99
		Ali Express	0	0.12.34
		TOTAL	**230**	**0.28.85**
3	Royal Canin Medium Adult 10 kg Minimal : 300.000 Maksimal : 1.000.000	Shopee	150	0.12.98
		Blibli	52	0.03.60
		Ali Express	1	0.20.58
		TOTAL	**203**	**0.37.16**
4	Beras 5 kg Minimal : 50.000	Shopee	150	0.10.12
		Blibli	80	0.04.45
		Ali Express	15	1.34.50
		TOTAL	**233**	**1.49.44**
5	Softener Downy	Shopee	150	0.08.32
		Blibli	80	0.03.68
		Ali Express	20	1.41.05
		TOTAL	**233**	**1.53.45**

Hal lain yang menyebabkan proses menjadi lambat dikarenakan website menggunakan paketan RapidAPI tingkat dasar. Paketan RapidAPI tingkat dasar hanya mengizinkan 1 *request* per detik. Akibatnya, program harus menunggu 1 detik setiap kali mengirim *request*, terutama untuk *marketplace* Shopee dan AliExpress. Selain itu, pada AliExpress respon dari *endpoint search* tidak menyertakan data rating dan jumlah penjualan. Oleh karena itu, program perlu melakukan *request* tambahan ke *endpoint* detail produk untuk setiap produk yang makin memperlambat proses. Proses untuk mengambil rating dan jumlah penjualan pada masing-masing produk di *endpoint* detail membuat tampilan hasil produk AliExpress menjadi lambat Setiap *request* ke *endpoint* detail memerlukan jeda 1 detik, dan ini akan dilakukan untuk setiap produk dari hasil *endpoint* search. Oleh karena itu, semakin banyak jumlah produk yang didapatkan pada respon *endpoint* search AliExpress, semakin lama proses untuk menunggu produk AliExpress. Tabel 2 merupakan total produk yang didapatkan pada respon dari *endpoint* search API dan waktu yang diperlukan untuk mendapatkan total produk tersebut.

Berdasarkan tabel 2, API Blibli terbukti paling cepat karena menggunakan API resmi Blibli. Blibli mampu mengambil hingga 80 produk karena melakukan maksimal 2 kali *request* untuk *pagination*. Rata-rata waktu yang diperlukan API Blibli untuk memberikan respon adalah 3.784 detik. Di urutan kedua terdapat *marketplace* Shopee yang menggunakan API pihak ketiga, RapidAPI. Shopee dapat mengambil hingga 150 produk karena melakukan maksimal 3 kali *request* untuk *pagination*. Rata-rata waktu respon RapidAPI Shopee adalah 12.3 detik. Di urutan ketiga terdapat *marketplace* AliExpress yang menggunakan API pihak ketiga, yaitu RapidAPI. Maksimal produk yang dapat diperoleh adalah 20 produk, karena RapidAPI Ali Express melakukan maksimal 2 kali *request* untuk *pagination*. Rata-rata waktu respon dari RapidAPI AliExpress cukup lama, yaitu 45.928 detik. Hal ini disebabkan setiap produk dari hasil pencarian harus melakukan *request* tambahan ke *endpoint* detail untuk memperoleh data rating dan jumlah transaksi produk sehingga memperpanjang waktu pemrosesan.

4.2. Pembahasan

Penelitian ini berangkat dari masalah utama yang dihadapi penjual, yaitu proses pemantauan harga manual, lambat, dan rentan adanya kesalahan. Sebagai solusinya, sebuah website yang dikembangkan ini berhasil menyelesaikan masalah tersebut dengan menyediakan platform terpusat yang mengotomatisasi pengumpulan dan penyajian data harga dari berbagai sumber eksternal. Pendekatan ini sejalan dengan penelitian M. I. Arief dan Robert Kurniawan serta Hamid Reza Saeidnia dkk yang menekankan pentingnya integrasi sistem berbasis web dan API untuk efisiensi pengolahan data [4][5]. Namun, penelitian ini menambahkan mekanisme ganda (integrasi API dan web *scraping fallback*) sebagai solusi adaptif ketika API publik tidak tersedia.

Hasil pengujian dangan *black box testing* telah berhasil mengkonfirmasi bahwa semua kebutuhan fungsional seperti pencarian produk dari berbagai *marketplace*, rekomendasi tiga level harga (harga terendah, rata-rata/menengah, dan tertinggi), dan notifikasi *email* telah terpenuhi. Temuan ini menguatkan penelitian M.T. Abdillah dkk yang juga menunjukkan bahwa pengujian berbasis *black box* efektif dalam memastikan keandalan fungsional sistem website [10]. Pada penelitian ini memberikan penguatan empiris baru karena penerapan pengujian dilakukan pada sistem real-time berbasis API dan *scraping* yang dinamis (tidak hanya pada sistem berbasis konten statis seperti pada penelitian sebelumnya).

Selain fungsional yang telah disebutkan, hasil penelitian ini menyajikan tiga wawasan utama yang signifikan. Penjelasan yang pertama adalah keberhasilan non fungsional sistem ini divalidasi melalui *usability testing*. Secara konsisten menghasilkan skor *usability* diatas 90% dari semua kelompok pengguna (untuk pengguna gratis, pengguna premium, dan pengguna yang tidak memiliki toko *online*/pembeli) yang ada. Temuan ini memperkuat penelitian Fitria dkk yang menekankan pentingnya kemudahan penggunaan dan efisiensi interaksi pengguna dalam aplikasi digital, terutama dengan mengadaptasi metrix Nielsen untuk menilai efektivitas desain antarmuka. Hasil angka yang sangat tinggi ini menjadi bukti empiris yang medukung pendekatan pragmatis. Hal ini memperkuat kesimpulan dari Nada Ghanem dkk bahwa solusi praktis dan mudah dioperasikan lebih bernilai bagi pengguna UKM dibandingkan dengan sistem yang berbasis algoritma yang kompleks [12]. Dengan demikian, penelitian yang bersifat komplementer ini telah mengisi kesenjangan (*gap*) yang ada antara solusi teoritis dengan kebutuhan nyata di lapangan sehingga merupakan penelitian yang saling melengkapi.

Penjelasan yang kedua tentang pengujian kinerja mengungkapkan adanya konsekuensi timbal balik yang fundamental antara kecepatan respon sistem dan kelengkapan data yang disajikan. Telah ditemukan bahwa ketergantunagan pada API pihak ketiga, khususnya AliExpress menjadi penyebab utama dalam keterlambatan. Hal ini dikarenakan memerlukan permintaan waktu tambahan untuk mendapatkan data rating dan jumlah transaksi produk. Dengan ini menunjukkan bahwa secara signifikan memperlambat waktu respon. Temuan ini memberikan tambahan pengetahuan baru terhadap penelitian Alvise de Biasio dkk yang memiliki fokus pada optimasi *profit-aware recommendation* tetapi tidak mengulas dampak performa jaringan pada integrasi sistem [11]. Dengan demikian, penelitian ini memperluas cakupan diskusi tentang performa praktis sistem *e-commerce* (khususnya dalam konteks integrasi multi sumber).

Penjelasan yang ketiga tentang analisis skor *usability* yang menunjukkan adanya perbedaan antara pengguna premium (91,26%), pengguna gratis (95,54%), dan pembeli (96,8%). Skor yang dimiliki pengguna premium ini sedikit lebih rendah mengindikasikan bahwa fitur-fitur yang telah disediakan dan bisa diakses oleh pengguna premium ini sifatnya lebih kompleks (butuh banyak langkah, waktu, dan usaha mental untuk mencapai tujuannya). Hal ini mendukung pernyataan Rahma Fitria dkk bahwa kompleksitas fitur berdampak langsung terhadap persepsi kemudahan penggunaan [9]. Oleh karena itu, pada penelitian ini diperlukan perancangan ulang alur interaksi untuk kompleksitas fitur premium pada sistem *e-commerce*. Dengan tujuan agar kemudahan fitur dasar dan premium dapat setara.

5. Simpulan

Berdasarkan hasil pengembangan dan pengujian, penelitian ini berhasil menghasilkan model website *e-commerce* yang fungsional, efektif, dan memiliki tingkat penerimaan pengguna yang sangat tinggi. Website yang dikembangkan ini secara langsung menjawab permasalahan utama yang dihadapi penjual. Solusi yang diberikan dengan menyediakan alat bantu yang praktis untuk memantau harga secara terpusat, memperoleh rekomendasi strategis (tiga level rentang harga), dan mengurangi proses manual yang rentan terhadap kesalahan. Temuan penelitian ini menegaskan bahwa pendekatan pengembangan yang berfokus pada integrasi pragmatis teknologi modern dan peningkatan usabilitas terbukti relevan serta diterima dengan baik oleh pelaku UKM. Namun, penelitian ini memiliki keterbatasan pada jumlah sampel pengujian dan performa sistem terhadap beberapa *marketplace* yang masih bergantung pada API pihak ketiga. Prospek penelitian ke depan diarahkan pada perluasan kolaborasi dengan penyedia *marketplace* untuk memperoleh akses API resmi, serta pengembangan modul analisis prediktif sederhana yang berguna untuk memperkuat fitur rekomendasi harga yang telah ada.

Daftar Referensi

[1] R. Mustika Sari and Prihartono, “Pengaruh Harga Dan Kualitas Produk Terhadap Keputusan Pembelian (Survey Pelanggan Produk Sprei Rise),” *J. Ilm. MEA (Manajemen, Ekon. dan Akuntansi)*, vol. 5, no. 3, pp. 1171–1184, 2021.

[2] M. Nowak and M. Pawłowska-Nowak, “Dynamic Pricing Method in the E-Commerce Industry Using Machine Learning,” *Appl. Sci.*, vol. 14, no. 24, 2024, doi: 10.3390/app142411668.

[3] A. Naik, “The Front-End Dilemma: How to Choose the Perfect Technology for your Application.,” *J. Comput. Sci. Technol. Stud.*, vol. 6, no. 1, pp. 211–216, 2024, doi: 10.32996/jcsts.2024.6.1.24.

[4] M. I. Arief and R. Kurniawan, “Pengembangan Sistem Aplikasi Web Scraper Harga Komoditas Menggunakan Metode Design Oriented Research,” *Jambura J. Informatics*, vol. 2, no. 1, 2020, doi: 10.37905/jji.v2i1.4474.

[5] H. R. Saeidnia, A. Ghorbi, M. Kozak, and S. Abdoli, “Web-based Application Programming Interface (Web APIs): Vacancies in Iranian Public Library Websites,” *Webology*, vol. 19, no. 1, pp. 133–141, 2021, doi: 10.14704/web/v19i1/web19010.

[6] G. Jadhav and F. Gonsalves, “Role of Node.js in Modern Web Application Development,” *Int. Res. J. Eng. Technol.*, vol. 07, no. 06, pp. 6145–6150, 2020, [Online]. Available: www.irjet.net

[7] Y. Balkhande, “Implementation of comprehensive information security model by using JWT & CryptoJS,” *Res. Sq.*, pp. 1–11, 2024.

[8] S. Sotnik, V. Manakov, and V. Lyashenko, “Overview: PHP and MySQL Features for

Creating Modern Web Projects," *Int. J. Acad. Inf. Syst. Res.*, vol. 7, no. 1, pp. 11–17, 2023, [Online]. Available: www.ijeais.org/ijaisr

[9] R. Fitria, Cut Meurah Nurul 'Akla, R. Meiyanti, N. Kartika, and Rifqa, "Integrating Heuristic Evaluation and Think-Aloud Protocols By Applying Nielsen's Metrics on Indonesian e-Pangan Application," *J. Artif. Intell. Eng. Appl.*, vol. 3, no. 3, pp. 712–718, 2024, doi: 10.59934/jaiea.v3i3.508.

[10] M. T. Abdillah, I. Kurniastuti, F. A. Susanto, and F. Yudianto, "Implementasi Black Box Testing dan *Usability* Testing pada Website Sekolah MI Miftahul Ulum Warugunung Surabaya," *J. Comput. Sci. Vis. Commun. Des.*, vol. 8, no. 1, pp. 234–242, 2023, doi: 10.55732/jikdiskomvis.v8i1.897.

[11] A. De Biasio, D. Jannach, and N. Navarin, "Model-based approaches to profit-aware recommendation," *Expert Syst. Appl.*, vol. 249, no. PB, p. 123642, 2024, doi: 10.1016/j.eswa.2024.123642.

[12] N. Ghanem, S. Leitner, and D. Jannach, "Balancing consumer and business value of recommender systems: A simulation-based analysis," *Electron. Commer. Res. Appl.*, vol. 55, no. August, p. 101195, 2022, doi: 10.1016/j.elerap.2022.101195.

[13] M. A. Khder, "Web scraping or web crawling: State of art, techniques, approaches and application," *Int. J. Adv. Soft Comput. its Appl.*, vol. 13, no. 3, pp. 144–168, 2021, doi: 10.15849/ijasca.211128.11.

[14] N. Nasrul and A. Izhar, "Pengembangan REST API dengan menggunakan Express JS untuk mencari Mentor Pribadi," *J. Inform. Terpadu*, vol. 9, no. 2, pp. 92–102, 2023, doi: 10.54914/jit.v9i2.974.

[15] R. Rahardian and M. William Pratama Wenas, "Rancang Bangun Sistem Informasi Koperasi Xyz Menggunakan Framework Laravel Dan Vue.Js," *J. Tek. Inform. dan Teknol. Inf.*, vol. 2, no. 3, pp. 115–122, 2022, doi: 10.55606/jutiti.v2i3.494.

[16] T. P. Batubara, S. Efendi, and E. B. Nababan, "Analysis Performance BCRYPT Algorithm to Improve Password Security from Brute Force," *J. Phys. Conf. Ser.*, 2021, doi: 10.1088/1742-6596/1811/1/012129.

[17] B. Maulana Imran and N. Santoso, "Pengembangan Sistem Manajemen Logistik Dan Order Di UBCOFFEE," *J. Pengemb. Teknol. Inf. dan Ilmu …*, vol. 5, no. 6, pp. 2381–2389, 2021, [Online]. Available: http://j-ptiik.ub.ac.id

[18] S. H. Hasibuan, M. I. P. Nasution, and S. S. A. Sundari, "Development of Payment Gateway Digitalization Using Midtrans in the Use of Halodoc," *Int. J. Adv. Technol. Eng. Inf. Syst.*, vol. 2, no. 1, pp. 9–17, 2023, doi: 10.55047/ijateis.v2i1.545.

[19] Deden Moh Alfiansyah, Wiilys, Lila Setiyani, Devi Fajar Wati, and Dedih, "Pengembangan Chatbot Berbasis Web untuk Layanan Informasi di Horizon University," *bit-Tech*, vol. 7, no. 3, pp. 1068–1077, 2025, doi: 10.32877/bt.v7i3.2318.

[20] A. ALazzawi, Q. M. Yas, and B. Rahmatullah, "A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions," *Iraqi J. Comput. Sci. Math.*, vol. 4, no. 4, pp. 173–190, 2023, doi: 10.52866/ijcsm.2023.04.04.014.