



Pengembangan Website Harga Bapokting *Real-time* dengan *Extreme Programming* dan Integrasi API SILINDA

Ridwan Setiawan¹, Rina Parlina², Erwin Gunadhi³

^{1,2,3}Jurusan Ilmu Komputer, Institut Teknologi Garut

¹ridwan.setiawan@itg.ac.id*, ²2106152@itg.ac.id, ³erwingunadhi@itg.ac.id

Abstract

This study aims to implement the SILINDA API from the West Java Provincial Government on a prototype website to present real-time price data for Basic Necessities and Important Goods (Bapokting) in Garut Regency. This addresses the problem of reporting delays and potential data inaccuracies that arise from the manual process of reporting via WhatsApp and weekly recapitulation using Microsoft Excel. The system was developed using the Extreme Programming (XP) methodology, which includes the stages of planning, design, coding, and testing. System design utilizes Unified Modeling Language (UML), specifically use case and class diagrams. The implementation uses JavaScript with the React.js library for the frontend and Node.js with the Express.js framework for the backend. The result of this research is a website prototype that is synchronized with the SILINDA API to perform automatic price updates. System testing included unit testing with a black-box approach and acceptance testing using the System Usability Scale (SUS) method, which yielded an average score of 83, categorized as Grade A (Excellent) with an "Acceptable" level of acceptance. This research contributes a system that replaces the manual reporting process with a website synchronized with SILINDA, providing real-time data for the Disperindag ESDM, Garut Satu Data, and the general public. It also demonstrates the effectiveness of the XP method in building an adaptive system that is relevant to user needs.

Keywords: Bapokting, Extreme Programming, SILINDA API, Website.

Abstrak

Penelitian ini bertujuan untuk mengimplementasikan API SILINDA dari Pemerintah Provinsi Jawa Barat pada *prototype website* untuk menyajikan data harga Bapokting secara *real-time* di Kabupaten Garut. Hal ini mengatasi masalah keterlambatan pelaporan dan potensi ketidakakuratan data yang timbul dari proses pelaporan manual melalui WhatsApp dan rekapitulasi mingguan menggunakan Microsoft Excel. Sistem dikembangkan menggunakan metodologi *Extreme Programming* (XP) dengan tahapan *planning*, *design*, *coding*, dan *test*. Perancangan sistem menggunakan *Unified Modelling Language* (UML), yang mencakup *use case diagram* dan *class diagram*. Implementasi menggunakan bahasa pemrograman JavaScript dengan library React.js untuk *front-end* dan Node.js dengan *framework* Express.js untuk *back-end*. Hasil penelitian ini adalah sebuah *prototype website* yang tersinkronisasi dengan API SILINDA untuk melakukan pembaruan harga secara otomatis. Pengujian sistem mencakup unit testing dengan pendekatan *black-box* dan *acceptance testing* menggunakan metode *System Usability Scale* (SUS), yang menghasilkan skor rata-rata 83, termasuk dalam kategori Grade A (*Excellent*) dengan tingkat penerimaan "Acceptable". Penelitian ini berkontribusi menggantikan proses pelaporan manual dengan website yang tersinkronisasi dengan SILINDA, yang dapat digunakan langsung oleh Disperindag ESDM, Garut Satu Data, dan masyarakat umum. Penelitian ini juga membuktikan efektivitas metode XP dalam membangun sistem yang adaptif dan relevan dengan kebutuhan pengguna.

Kata kunci: API SILINDA; Bapokting, Extreme Programming, Website.

1. Pendahuluan

Perkembangan teknologi informasi telah mengubah cara masyarakat mengakses informasi, salah satunya melalui website yang dapat diakses secara *real-time* [1], [2]. Dalam bidang perdagangan, pengelolaan data harga Barang Kebutuhan Pokok dan Barang Penting (Bapokting) menjadi krusial karena berperan penting dalam menjaga stabilitas harga dan pengendalian inflasi pangan [3], [4]. Di Kabupaten Garut, pengelolaan data ini dilakukan oleh Dinas Perindustrian dan Perdagangan

(Disperindag) ESDM berdasarkan laporan dari UPT (Unit Pelaksana Teknis) Pasar. Namun, proses yang berjalan saat ini memiliki kendala pelaporan dari UPT ke Disperindag masih dilakukan melalui aplikasi WhatsApp dan direkap secara manual menggunakan Microsoft Excel. Selain itu, data yang dipublikasikan ke portal Garut Satu Data bersifat mingguan, sehingga menyebabkan keterlambatan informasi dan data yang diterima masyarakat menjadi tidak akurat dengan kondisi pasar terkini. Kondisi ini menimbulkan keterlambatan, risiko kesalahan *input*, serta



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

berkurangnya akurasi informasi harga yang diterima publik.

Untuk mengatasi masalah ini, Pemerintah Provinsi Jawa Barat telah menyediakan layanan API SILINDA yang memungkinkan sinkronisasi data harga Bapokting secara otomatis dan *real-time* [5]. Namun, sejauh ini belum ada penelitian yang secara spesifik mengimplementasikan API SILINDA di tingkat daerah untuk mendukung transparansi data harga pasar.

Beberapa penelitian yang telah dilakukan di antaranya oleh Hafidz et al. merancang sistem *penginputan* data bahan pokok, namun masih bergantung pada *input* manual [6]. Penelitian oleh Mita et al. mengembangkan sistem perbandingan harga rata-rata, namun tidak mendukung integrasi data *real-time* [7]. Sementara itu, Sufri et al. serta Nurani dan Syuryadi mengusulkan sistem monitoring harga pangan, tetapi belum memanfaatkan API eksternal sebagai sumber data otomatis [1], [8]. Di sisi lain, penelitian oleh Lukas et al. dan Rokhim et al. menunjukkan keberhasilan penerapan metode *Extreme Programming* (XP) dan integrasi API pada sistem berbasis web untuk kebutuhan monitoring stok barang dan harga *cryptocurrency* [9], [10].

Metodologi pengembangan yang digunakan adalah *Extreme Programming* (XP), yang dipilih karena sifatnya yang fleksibel dan memiliki siklus iteratif pendek, sehingga mampu menghasilkan produk yang sesuai dengan kebutuhan pengguna [11]. Pendekatan ini terbukti berhasil dalam penelitian sebelumnya yang mengembangkan sistem *real-time* dan mengimplementasikan API. Kontribusi penelitian ini mencakup penerapan XP dalam pengembangan sistem informasi pemerintahan berbasis web, sekaligus mendemonstrasikan pemanfaatan API SILINDA sebagai solusi otomatisasi pelaporan harga yang sebelumnya dilakukan secara manual. Dengan demikian, hasil penelitian ini diharapkan dapat meningkatkan transparansi, akurasi, dan efisiensi pelaporan harga Bapokting di tingkat daerah.

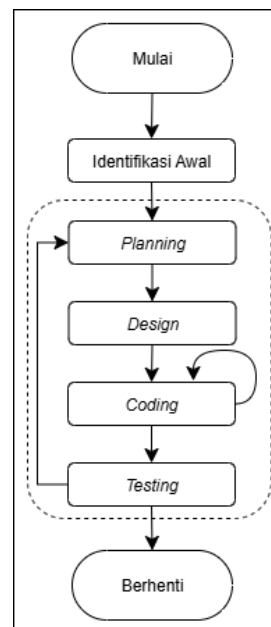
Berdasarkan penelitian sebelumnya yang masih menitikberatkan pada *input* manual atau pengolahan harga tanpa dukungan data *real-time* dan belum memanfaatkan API pemerintahan sebagai sumber data otomatis. Penelitian ini memajukan celah tersebut dengan menghubungkan API SILINDA (Jawa Barat) ke aplikasi kabupaten sehingga harga Bapokting tersaji *real-time*. Selain itu kami menggunakan metode *Extreme Programming* (XP) pada domain layanan publik daerah, bukan sekadar aplikasi komersial/umum. Evaluasi dilakukan melalui SUS lintas-pemangku kepentingan (Pemda, Admin, dan masyarakat) dengan hasil SUS=83 (*Grade A*). Dengan demikian, kontribusi utama penelitian ini ialah: arsitektur integrasi berbasis API resmi, otomatisasi alur pelaporan, dan pembuktian kegunaan pada konteks pemerintahan daerah.

2. Metode Penelitian

Penelitian ini menggunakan metodologi *Extreme Programming* (XP)[12], sebuah pendekatan *agile* yang berfokus pada fleksibilitas dan kualitas perangkat lunak melalui iterasi cepat [11]-[15]. Metode XP dipilih karena kemampuannya dalam menghasilkan sistem yang tanggap terhadap kebutuhan pengguna dan dapat divalidasi secara berkelanjutan melalui siklus pengembangan yang berulang[16]. Alur penelitian ini mengikuti empat tahapan utama metode *Extreme Programming* yaitu *planning*, *design*, *coding*, dan *testing*. Pemodelan sistem menggunakan *Unified Modelling Language* (UML), untuk memvisualisasikan struktur sistem berorientasi objek agar mudah dipahami [17].

Gambar 1 menggambarkan model tahapan *Extreme Programming* yang digunakan dalam penelitian ini. Gambar tersebut menunjukkan alur iteratif dimulai dari identifikasi kebutuhan, tahapan *Extreme Programming* sampai *testing* pengguna.

Pada tahap *planning* melibatkan identifikasi kebutuhan pengguna dengan aktivitas mendengarkan dan pengumpulan kebutuhan sistem melalui wawancara dengan pihak Disperindag ESDM Kabupaten Garut. Hasil wawancara digunakan untuk menyusun *user stories* dan *values*, yang merepresentasikan kebutuhan dan harapan pengguna akhir. Selanjutnya, ditentukan *acceptance test criteria* untuk menentukan parameter keberhasilan sistem, dan disusun *iteration plan* sebagai dasar pelaksanaan implementasi secara bertahap selama proses pengembangan [18].



Gambar 1. Proses Model *Extreme Programming*

Tahap Design dirancang menggunakan prinsip *simple design*, yang menekankan pada desain minimal namun fungsional. Sistem dimodelkan dengan menggunakan

pendekatan *Unified Modeling Language* (UML), di antaranya *use case diagram* dan *class diagram* untuk memodelkan interaksi dan struktur data sistem. Selain itu, digunakan pula *Class-Responsibility-Collaborator* (CRC) *Cards* guna merinci tanggung jawab dan kolaborasi antar kelas secara rinci [19], [20].

Tahap *coding* dalam metode *Extreme Programming* dilakukan pendekatan *pair programming* yaitu dua pengembang bekerja sama dalam satu komputer[21], kemudian *refactoring* merupakan kegiatan merestrukturisasi kode agar mudah dalam pemeliharaan kode, dan *continuous integration* di mana kode digabungkan ke dalam sistem utama[22]. Implementasi sistem dilakukan menggunakan bahasa pemrograman JavaScript, dengan *library* React.js untuk pengembangan antarmuka pengguna (*frontend*), dan Node.js dengan *framework* Express.js untuk pengembangan *back-end*. Basis data yang digunakan adalah PostgreSQL [9].

Tahap testing meliputi unit testing dan *acceptance testing*. Validasi sistem dilakukan melalui dua pendekatan: (1) *black-box* testing untuk memverifikasi fungsionalitas terhadap spesifikasi tanpa meninjau kode sumber; dan (2) *acceptance testing* menggunakan *System Usability Scale* (SUS) untuk mengukur persepsi kegunaan secara kuantitatif [23], [24]. Pemilihan SUS relevan dengan konteks *Open Government Data* (OGD) karena keberterimaan dan kemudahan akses antarmuka merupakan prasyarat pemanfaatan data oleh pemangku kepentingan; studi lintas 41 portal OGD menunjukkan pentingnya evaluasi berpusat pengguna untuk memetakan hambatan *discoverability*, navigasi, dan terminologi[25]. Selain itu, telaah komprehensif terkini merekomendasikan peningkatan *usability* melalui desain visualisasi/analitik serta uji pengguna terstruktur agar benar-benar bermanfaat bagi berbagai peran—selaras dengan *acceptance testing* berbasis SUS yang kami lakukan[26]. Dalam pembahasan, temuan SUS ditautkan pada lima aspek yang lazim digunakan dalam literatur—*learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*—sebagai lensa untuk menjelaskan pengalaman pengguna akhir.

3. Hasil dan Pembahasan

Berdasarkan tahapan metode XP yang telah dilaksanakan, penelitian ini berhasil mengembangkan sebuah *prototype website* Bapokting di Kabupaten Garut yang tersinkronisasi dengan API SILINDA untuk menampilkan dan memperbarui data harga Bapokting secara otomatis setiap harinya. *Website* ini terdiri atas fitur utama, yaitu informasi harga harian per-pasar, grafik harga interaktif berdasarkan komoditas dan pasar, laporan unduh harga dalam format *Excel*, peta lokasi UPT pasar, berita terkait. Berikut merupakan tahapan-tahapan dalam proses pengembangan sistem.

3.1. Planning

Tahap ini merumuskan kebutuhan pengguna secara sistematis melalui penyusunan *user stories* beserta nilai kegunaannya, penetapan *acceptance test criteria*, serta perancangan rencana iterasi. Perumusan tersebut diturunkan dari wawancara pihak terkait dan menjadi landasan pengembangan pada tahap berikutnya. Untuk menjaga keterlacakan, rekam *user stories* dan *values* disajikan pada Tabel 1, sedangkan kriteria penerimaan dirangkum pada Tabel 2.

Kebutuhan yang dihimpun meliputi dua peran, yakni pengguna umum dan admin. Pada sisi pengguna, fitur inti mencakup akses data harga Bapokting secara real time dari API SILINDA, pencarian serta penyaringan berdasarkan tanggal, pasar, komoditas, dan keterangan, visualisasi historis yang dapat dikustomisasi, laporan tabel dan ekspor, peta lokasi UPT pasar dengan detail yang dapat ditindaklanjuti, daftar berita, dan informasi tentang aplikasi. Pada sisi admin, kebutuhan berfokus pada keamanan akses serta pengelolaan data UPT pasar dan berita. Seluruhnya tersusun ringkas pada Tabel 1.

Untuk memastikan setiap kebutuhan dapat diverifikasi, *acceptance test criteria* didefinisikan pada tingkat *user story*. Contohnya, keberhasilan pengambilan dan pembaruan data dari API SILINDA untuk US1, ketepatan fungsi penyaring serta pencarian untuk US1.1, konsistensi dan pemutakhiran grafik historis untuk US2 dan US2.1, kelengkapan laporan serta fasilitas ekspor untuk US3 dan US3.1, serta skenario login dan pengelolaan UPT maupun berita untuk A-US1 sampai A-US7. Rincian lengkapnya dapat dilihat pada Tabel 2.

Tabel 1. Rekam *user stories* dan *values*

Kode	User Story	Value
US1	Sebagai pengguna, ingin melihat data harga Bapokting secara real time dari API eksternal	Pengguna memperoleh akses langsung ke harga terkini
US1.1	Sebagai pengguna, ingin memfilter dan mencari harga berdasarkan tanggal, pasar, komoditas, dan keterangan	Menemukan informasi yang spesifik dengan cepat dan akurat
US2	Sebagai pengguna, ingin melihat visualisasi grafik historis harga komoditas dari data API	Mengidentifikasi pola dan tren harga
US2.1	Sebagai pengguna, ingin grafik historis yang dapat dikustomisasi berdasarkan tanggal akhir, komoditas, dan pasar	Kustomisasi visualisasi sesuai kebutuhan analisis
US3	Sebagai pengguna, ingin melihat laporan harga dalam format tabel di halaman website	Meninjau ringkasan data dengan cepat sebelum mengunduh
US3.1	Sebagai pengguna, ingin mengunduh laporan harga dalam format Excel dengan memilih rentang	Data terformat rapi untuk pelaporan

Tabel 1. (Lanjutan)

Kode	User Story	Value
US4	Sebagai pengguna, ingin melihat peta lokasi UPT pasar	Mempermudah penemuan lokasi UPT dan aksesibilitas
US4.1	Sebagai pengguna, ingin melihat detail UPT di peta serta terhubung ke WhatsApp dan Google Maps	Akses cepat ke informasi lengkap dan navigasi
US5	Sebagai pengguna, ingin melihat daftar dan membaca detail berita	Mendapatkan informasi terkini yang relevan
US6	Sebagai pengguna, ingin melihat informasi tentang website	Meningkatkan kepercayaan melalui informasi latar belakang
A-US1	Sebagai admin, ingin masuk menggunakan username dan password	Melindungi data dan menjaga integritas sistem
A-US2	Sebagai admin, ingin menambahkan informasi UPT pasar baru	Memastikan lokasi pasar tetap relevan
A-US3	Sebagai admin, ingin mengubah informasi UPT pasar yang ada	Menjaga relevansi informasi UPT yang ditampilkan
A-US4	Sebagai admin, ingin menghapus informasi UPT yang tidak relevan atau tidak aktif	Menjaga kualitas dan relevansi data UPT
A-US5	Sebagai admin, ingin menambahkan berita	Mempublikasikan informasi penting dan relevan
A-US6	Sebagai admin, ingin mengubah informasi berita	Mengoreksi atau memperbarui konten agar akurat dan relevan
A-US7	Sebagai admin, ingin menghapus berita yang tidak relevan	Menjaga daftar berita tetap aktual

Tabel 2. Acceptance test criteria

Kode	User Story
US1	Sistem mengambil serta menampilkan data harga dari API SILINDA; data sesuai dengan data pada API; pembaruan berjalan otomatis pada hari berjalan.
US1.1	Pemilihan tanggal melalui date picker menampilkan harga sesuai tanggal; pemilihan pasar melalui dropdown memfilter harga; pencarian komoditas menampilkan hasil relevan; filter keterangan naik, turun, tetap menampilkan data sesuai kondisi.
US2	Grafik historis menampilkan sumbu waktu dan harga sesuai komoditas yang dipilih.
US2.1	Pengguna memilih komoditas dan pasar melalui dropdown serta tanggal akhir melalui date picker; grafik menampilkan pergerakan sampai tanggal akhir; grafik memperbarui secara dinamis; data grafik konsisten dengan data historis API.
US3	Halaman laporan menampilkan tabel harga yang jelas; tabel mencakup semua pasar pada API SILINDA; isi tabel sesuai data tersedia.
US3.1	Sistem menyediakan laporan dalam bentuk tabel dan tombol ekspor; file Excel yang diunduh merepresentasikan data yang tampil termasuk periode yang dipilih.
US4	Peta termuat serta menampilkan marker seluruh UPT; pengguna dapat memperbesar, memperkecil, dan menggeser peta.

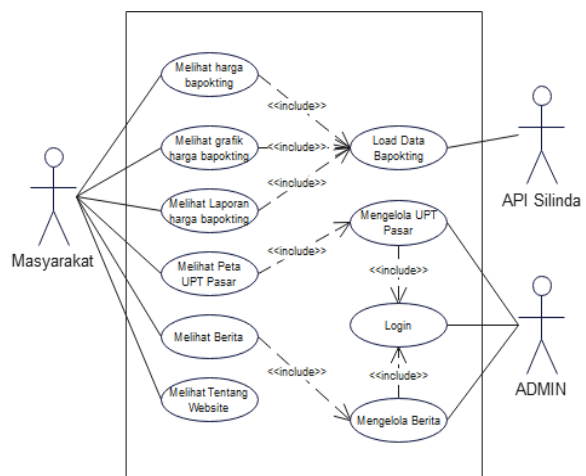
Tabel 2. (Lanjutan)

Kode	User Story
US4.1	Klik marker memunculkan informasi UPT; tombol WhatsApp membuka aplikasi dengan nomor kontak UPT; tombol Google Maps membuka rute lokasi.
US5	Halaman berita menampilkan daftar yang tersusun terbaru; pengguna dapat membuka detail yang memuat judul, isi, tanggal, dan gambar.
US6	Halaman tentang menyajikan informasi yang relevan mengenai tujuan aplikasi.
A-US1	Login berhasil dengan kredensial valid mengarahkan ke dashboard; bila salah menampilkan pesan gagal.
A-US2	Admin membuka Manajemen UPT dan menambah UPT; pengisian data lengkap menampilkan konfirmasi berhasil; titik UPT baru muncul di peta dan tabel.
A-US3	Admin mengedit UPT melalui tabel atau peta; setelah simpan tampil konfirmasi berhasil.
A-US4	Admin menghapus UPT melalui tabel atau peta; sistem menampilkan dialog konfirmasi; setelah hapus data hilang dari daftar dan peta serta ada konfirmasi berhasil.
A-US5	Admin menambah berita, mengisi judul, isi, dan gambar; setelah simpan muncul konfirmasi berhasil dan berita tampil di daftar.
A-US6	Admin mengedit judul, isi, dan gambar berita; setelah simpan muncul konfirmasi berhasil; perubahan tercermin di halaman berita.
A-US7	Admin menghapus berita; sistem menampilkan dialog konfirmasi; setelah hapus berita hilang dari daftar dan halaman pengguna serta ada konfirmasi berhasil.

3.2. Design

Pada tahap ini sistem dimodelkan menggunakan *Unified Modelling Language* (UML) untuk mendeskripsikan struktur dan interaksi sistem secara visual. Salah satu model yang digunakan adalah *use case diagram* yang menggambarkan interaksi antara aktor sistem, yaitu masyarakat, admin, dan API SILINDA, dengan berbagai fitur yang tersedia pada *prototype website*.

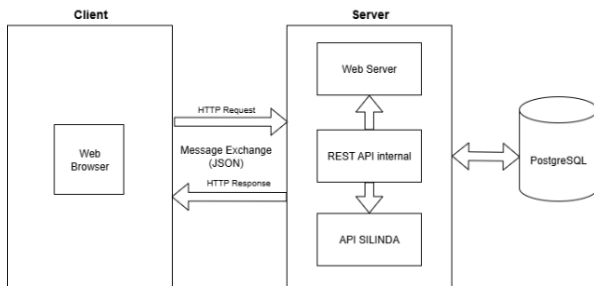
Gambar 2 menggambarkan *use case diagram* sistem yang mencakup fitur seperti tampilan harga harian, grafik historis harga, unduh laporan, peta lokasi UPT pasar, dan manajemen UPT pasar dan berita oleh admin yang terlibat dengan sistem.



Gambar 2. Use Case Diagram

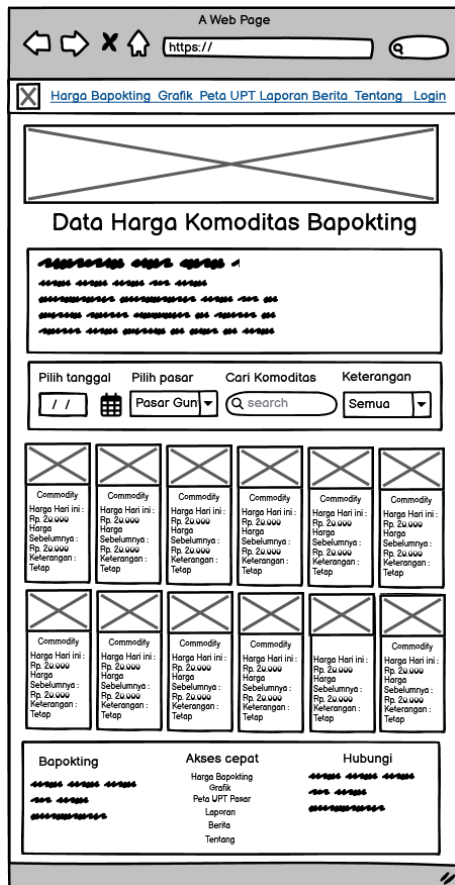
Class diagram menggambarkan relasi antar kelas-kelas yang saling berkaitan. Selain itu, untuk merinci tanggung jawab masing-masing kelas dan kolaborasinya, digunakan pemodelan *Class Responsibility Collaborator (CRC) Cards* sistem.

Gambar 3 menjelaskan alir data yang disusun dalam bentuk diagram arsitektur sistem. *Client* mengirimkan permintaan ke server, kemudian server menerima permintaan, lalu API SILINDA untuk data harga Bapokting *real-time*, ataupun mengambil data dari Postgresql untuk data UPT pasar dan berita, dan server mengembalikan respons dalam bentuk JSON ke *frontend*, dan *frontend* menampilkan data.



Gambar 3. Arsitektur Sistem

Gambar 4 Menunjukkan rancangan *interface* sistem pada halaman awal *prototype* website dengan mengimplementasikan API SILINDA.



Gambar 4. Rancangan *Interface* Sistem

3.3. Coding

Implementasi sistem dilakukan dengan menggunakan React.js dan Node.js dengan Express.js. sistem ini mengakses data dari API SILINDA untuk memperoleh harga Bapokting secara otomatis dan *real-time*.

Potongan program merupakan kode untuk mendapatkan data harga harian dari API SILINDA. Pada kode tersebut bahasa pemrograman yang digunakan adalah JavaScript dengan menggunakan Node.JS dan *framework* Express.js dalam pengimplemntasiannya.

Perintah `makeSillindaApiRequest` yang digunakan untuk melakukan permintaan HTTP POST ke endpoint API Sillinda. Fungsi ini menerima empat parameter yaitu `market_id`, `time`, `currentToken`, dan `isRetry` (dengan nilai default `false`). Di dalam fungsi, terlebih dahulu disusun objek konfigurasi `config` yang berisi header otorisasi dengan token `bearer` serta pengaturan `timeout` selama 10 detik. Kemudian disiapkan data `body` yang akan dikirim ke server, yaitu informasi `length`, `market_id`, `time`, dan `page`. URL endpoint juga disimpan dalam variabel terpisah untuk menjaga keterbacaan kode. Permintaan dikirim menggunakan `axios.post` dengan menyertakan URL, `body`, dan konfigurasi. Jika berhasil, fungsi akan mengembalikan respons dari server. Namun jika terjadi error, akan ditangkap dan ditampilkan menggunakan `console.error`.

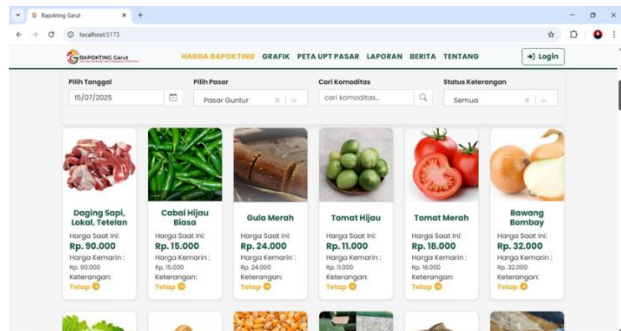
```

Potongan Program Pemanggilan API SILINDA
const makeSillindaApiRequest = async (market_id, time,
currentToken, isRetry = false) => {
  try {
    const config = {
      headers: {
        Authorization: `Bearer ${currentToken}`,
        'Content-Type': 'application/json',
      },
      timeout: 10000,
    };

    const body = { length: 1000, market_id, time, page:
1 };
    const url = 'https://api-silinda.go.id/t/jabarprov.go.id/silinda/api_v2/transaction/integration/find';

    return await axios.post(url, body, config);
  } catch (error) {
    console.error(error);
  }
};
    
```

Gambar 5 menunjukkan hasil dari pemanggilan data harga harian. Fitur yang dikembangkan meliputi fitur harga yang menampilkan data dari API dengan filter tanggal, komoditas, pasar, dan keterangan.



Gambar 5. Hasil Implementasi API SILINDA

Fitur grafik harga menggunakan Recharts untuk menampilkan tren harian harga, fitur laporan yang dapat diunduh dalam format Excel dengan menggunakan *library exceljs*, fitur lokasi UPT menggunakan *leaflet.js* dengan marker interaktif dan tautan ke Google Maps dan WhatsApp, dan fitur manajemen berita yang dapat dikelola oleh admin. Kode dikembangkan dengan menerapkan *refactoring*, dan modularisasi komponen.

3.4. Testing

Tahap pengujian (*test*) merupakan elemen esensial dalam *extreme programming* yang berfungsi untuk memastikan kualitas fungsionalitas sistem secara berkelanjutan. Dalam penelitian ini, pengujian dilakukan untuk *unit test* dan *acceptance testing*.

Unit testing dilakukan untuk memastikan bahwa setiap komponen atau bagian terkecil dari sistem bekerja sesuai dengan fungsinya. Dalam penelitian ini, pengujian unit diterapkan menggunakan pendekatan *blackbox testing*, yang menitikberatkan pada pengujian fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*), tanpa memeriksa struktur kode secara internal. Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi yang dikembangkan telah berjalan sesuai dengan spesifikasi yang ditentukan pada tahap perencanaan. Hasil dari pengujian tersebut ditampilkan pada Tabel 3.

Tabel 3. *Unit Test*

Kode Test	Fitur yang Diuji	Langkah Pengujian	Status
T-US1.1	Filter harga berdasarkan tanggal	Pilih tanggal dari <i>date picker</i>	Valid
T-US1.1	Cari berdasarkan komoditas	Ketik pada kolom pencarian	Valid
T-US1.1	Filter berdasarkan pasar	Pilih pasar dari <i>dropdown</i>	Valid
T-US1.1	Filter berdasarkan keterangan	Pilih keterangan dari <i>dropdown</i>	Valid
T-US2	Tampilkan grafik historis harga	Pilih menu grafik	Valid
T-US2.1	Pilih tanggal terakhir grafik	Pilih tanggal terakhir dari <i>date picker</i>	Valid
T-US2.1	Pilih komoditas	Pilih komoditas dari <i>dropdown</i>	Valid
T-US2.1	Pilih pasar	Pilih pasar dari <i>dropdown</i>	Valid
T-US3	Tampilkan laporan harga dalam tabel	Buka halaman laporan	Valid
T-US3.1	Pilih tanggal laporan	Pilih tanggal pada <i>datepicker</i>	Valid
T-US3.1	Cari Komoditas	Ketik pada kolom pencarian	Valid
T-US3.1	Unduh laporan harga	Unduh laporan	Valid
T-US4	Tampilkan peta UPT Pasar	Peta dimuat	Valid
T-US4.1	Info Popup UPT & Navigasi	Klik marker	Valid
T-US4.1	Popup Whatsapp pada UPT	Klik icon whatsapp	Valid

Kode Test	Fitur yang Diuji	Langkah Pengujian	Status
T-US4.2	Popup Maps pada UPT	Klik tombol Gmaps	Valid
T-US5	Menampilkan daftar berita	Klik berita dari menu	Valid
T-US5.1	Menampilkan detail berita	Klik berita dari daftar	Valid
TUS5.1.1	Kembali ke daftar berita	Menekan tombol kembali ke daftar berita	Valid
T-US6	Halaman tentang	Klik menu tentang	Valid
T-A-US1	Login	Memasukan username dan password, kemudian login	Valid
T-A-US1	Login	Memasukan username salah	Valid
T-A-US1	Login	Memasukan password salah	Valid
T-A-US1	Login	Memasukan username & password salah	Valid
T-A-US2	Menampilkan manajemen UPT	Klik menu manajemen UPT	Valid
T-A-US2	Tambah UPT Pasar	Klik tombol tambah upt, isi form, klik simpan	Valid
T-A-US3	Edit UPT Pasar	Klik Edit, ubah data, lalu simpan	Valid
T-A-US4	Hapus UPT Pasar	Klik hapus, konfirmasi penghapusan	Valid
T-A-US5	Menampilkan halaman manajemen berita	Klik menu manajemen berita	Valid
T-A-US6	Tambah berita	Klik tambah berita, isi form, klik simpan	Valid
T-A-US7	Edit berita	Klik edit, ubah data, lalu simpan	Valid
T-A-US8	Hapus berita	Klik hapus	Valid
T-A-US9	Detail berita	Menekan tombol detail	Valid
T-A-US10	Kembali ke daftar berita	Menekan tombol kembali ke daftar berita	Valid

Pengujian *acceptance* dilakukan untuk memastikan kesesuaian sistem dengan kebutuhan pengguna dan kemudahan penggunaan. Metode yang digunakan adalah *System Usability Scale* (SUS), instrumen kuesioner 10 item dengan skala *Likert*. Pengujian melibatkan 20 responden yang merepresentasikan Pemerintah Daerah, Admin, Garut Satu Data, dan masyarakat umum. Berdasarkan hasil uji, skor rata-rata SUS adalah 83 (Grade A—Excellent; Acceptable), yang menunjukkan sistem mudah dipelajari dan digunakan. Tingginya skor tersebut terutama didorong oleh: (i) kesesuaian tugas–fitur (akses harga harian per pasar, grafik tren, dan unduh Excel yang langsung mendukung pelaporan), (ii) kredibilitas data *real-time* dari integrasi API SILINDA, serta (iii) antarmuka yang sederhana dengan respons aplikasi yang cepat, sehingga meningkatkan *learnability* dan *satisfaction*.

Selain itu, metode XP efektif pada konteks ini karena integrasi API pihak ketiga menuntut iterasi pendek dan umpan balik dini. Rantai *user story* → *acceptance criteria* → testing memastikan setiap rilis ter-verifikasi terhadap tugas nyata dan menekan *scope creep*, sehingga kualitas pengalaman pengguna meningkat sebelum pengukuran SUS. Mengacu pada [27], skor SUS di atas 68 dikategorikan *acceptable*, sehingga sistem ini layak digunakan dan memberikan pengalaman pengguna yang positif.

Pengujian *acceptance testing* dilakukan untuk mengetahui apakah sistem telah sesuai dengan kebutuhan pengguna dan mudah digunakan. Pada *acceptance testing* ini menggunakan metode *System Usability Scale* (SUS), yaitu instrument evaluasi *usability* berbasis kuesioner yang terdiri dari sepuluh pertanyaan dengan skala Likert. *Acceptance test* dengan menggunakan metode SUS ini melibatkan dua puluh responden, yang merupakan representasi dari segemen pengguna baik dari Pemerintah Daerah, Admin, Garut Satu Data, dan Masyarakat umum. Data responden dikumpulkan melalui kuisisioner standar SUS yang terdiri dari sepuluh pertanyaan, dengan skala Likert untuk mengukur tingkat *usability* sistem.

Berdasarkan hasil uji coba, menunjukkan bahwa skor rata-rata SUS keseluruhan dari dua puluh responden adalah 83 yang berarti "*Excellent*" dengan *grade A* dan tingkat penerimaan "*Acceptable*". Hasil ini menunjukkan bahwa secara umum pengguna merasa puas dan memenuhi prinsip *usability* yang baik. Skor ini tidak hanya menunjukkan tingkat kepuasan pengguna yang tinggi, tetapi juga melampaui ambang batas akseptabilitas yang umum digunakan dalam evaluasi kegunaan sistem. Mengacu pada mengatakan skor SUS pada sistus web diatas dari 68 berarti pengguna merasa puas dan dianggap memenuhi standar minimum untuk diterima. Sehingga disimpulkan bahwa sistem ini layak untuk digunakan dan memberikan pengalaman pengguna yang positif.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan API SILINDA pada sebuah *prototype website* untuk pembaruan data harga Bapokting secara *real-time* di Kabupaten Garut. Pengembangan menggunakan metodologi *Extreme Programming* (XP) terbukti efektif dalam menghasilkan sistem yang fungsional dan sesuai kebutuhan pengguna, yang tervalidasi melalui pengujian sistem. *Prototype* mampu menyajikan informasi harga secara otomatis dan *real-time*, dilengkapi grafik historis, unduh laporan, peta lokasi UPT Pasar, dan berita terkait, sehingga mengatasi keterlambatan serta potensi ketidakakuratan pelaporan sebelumnya. Hasil *acceptance testing* menunjukkan SUS rata-rata 83 (kategori *Excellent*). Secara praktis, sistem meningkatkan transparansi dan akuntabilitas data bagi pemerintah daerah, mengefisienkan alur kerja dengan

menggantikan rekap manual, serta mempercepat pengambilan keputusan (misalnya koordinasi pengendalian inflasi) melalui visualisasi tren dan laporan siap-pakai. Bagi masyarakat, layanan ini menyediakan akses informasi harga yang setara lintas pasar dan mengurangi asimetri informasi yang berpotensi memicu hoaks.

Penelitian ini masih memiliki keterbatasan berupa ketergantungan pada ketersediaan API SILINDA. Oleh karena itu, arah pengembangan selanjutnya mencakup ketahanan integrasi (mekanisme *retry*, *circuit breaker*, *caching*, dan *fallback offline* saat API terganggu), analitik prediktif dan peringatan dini (*forecast* dan deteksi anomali) agar sistem tidak hanya melaporkan tetapi juga mendukung analisis strategis serta penguatan keamanan dan evaluasi lanjutan (RBAC, *audit trail*, *load testing*, dan uji aksesibilitas berbasis WCAG). Dengan demikian, kontribusi ilmiah dan manfaat praktisnya dapat diperluas sekaligus menjaga keandalan layanan bagi pemda dan masyarakat.

Ucapan Terimakasih

Kami menyampaikan terima kasih kepada Institut Teknologi Garut yang telah mendukung penelitian ini serta Disperindag ESDM Kabupaten Garut atas kontribusi mereka dalam menyediakan sumber data yang diperlukan untuk penelitian ini.

Daftar Rujukan

- [1] R. Sufri, Mukhroji, Nazuarsyah, R. B. Ginting, and S. Prayudi, "Sistem Informasi Ketersediaan Dan Perkembangan Harga Bahan Pokok Secara Real Time Berbasis Web," *Jurnal Ilmiah Informatika*, vol. 11, no. 01, pp. 102–105, 2023, doi: 10.33884/jif.v11i01.7168.
- [2] S. Azzahra, S. W. Maulidina, B. N. Natasha, F. A. Assyidiq, and M. H. N. Fadhillah, "Advocacy For The Use Of Websites To Improve RT/RW Services For The Community," *Setia Mengabdi: Jurnal Pengabdian kepada Masyarakat*, vol. 3, no. 2, pp. 47–54, Dec. 2022, doi: 10.31113/setiamengabdi.v3i2.41.
- [3] Perpres, "Perpres Nomor 59 Tahun 2020," 2020.
- [4] N. Puspita Widasari, E. Tanur, Y. R. Uli Sitanggang, and M. Situmorang, "Pengendalian Harga Pangan Melalui Penghitungan Indikator Proxy Inflasi," *Jurnal Good Governance*, pp. 117–136, Oct. 2023, doi: 10.32834/gg.v19i2.625.
- [5] Silinda, "Sistem Informasi Pengendalian Inflasi Daerah di Jawa Barat," 2020. [Online]. Available: <https://silinda.jabarprov.go.id/>
- [6] K. Hafidz, M. D. Irawan, and H. D. Nawar, "Sistem Penginputan Data Bahan Pokok pada Pasar Tradisional Sumatera Utara Berbasis Website di Disperindag Sumut," *sudo Jurnal Teknik Informatika*, vol. 1, no. 3, pp. 98–107, 2022, doi: 10.56211/sudo.v1i3.27.
- [7] M. Mita, Muharir, and Hoiriyah, "Sistem informasi perbandingan harga rata-rata bahan pokok masyarakat dan grafik kenaikan harga pada dinas perdagangan kota banjarmasin berbasis web," 2023.
- [8] Nurani and Syuryadi, "Sistem Informasi Monitoring Komoditas Harga Pangan Berbasis Web Dab Android," *Jurnal It*, vol. 13, no. 3, pp. 77–81, 2023, doi: 10.37639/jti.v13i3.339.
- [9] Lukas, R. A. Praptiwi, A. T. Priandika, Suaidah, and D. Alita, "Implementasi REST API pada Manajemen Stok Barang Berbasis Aplikasi WEB (Studi Kasus: PT Jon Kuliner

- Indonesia),” *Jurnal Teknik Komputer*, vol. 3(1), no. 1, pp. 19–24, 2024, doi: 10.14710/jtk.v3i1.46234.
- [10] A. Rokhim, A. Alimin, and M. Mustofa, “Implementasi Extreme Programming Untuk Monitoring Harga Cryptocurrency Antar Exchanger Dengan Menggunakan Websocket,” *SPIRIT*, vol. 16, no. 2, 2024.
- [11] R. S. Pressman, *Software Engineering A Practitioner’s Approach*. 2010. doi: 10.1145/336512.336521.
- [12] K. Pal, “eXtreme Programming, Agile Methodologies, Software Project Management, Customer Role, Rigorous Testing,” in *Contemporary Challenges for Agile Project Management*, IGI Global Scientific Publishing, 2022, pp. 163–182.
- [13] Yoseph Halim, Sandy Kosasi, Tony Wijaya, and Susanti M. Kuway, “Self-Service Technology Berbasis Android Menggunakan RestFul Web Service Pada Bisnis Restoran,” *Journal of Applied Computer Science and Technology*, vol. 2, no. 2, pp. 73–82, Dec. 2021, doi: 10.52158/jacost.v2i2.174.
- [14] S. Hudawiguna, N. Buchori, R. Setiawan, and R. L. Ulfa, “Pengembangan Aplikasi Monitoring Proses Penjahitan APD Berbasis Web Menggunakan Metode Extreme Programming,” *Jurnal Algoritma*, vol. 22, no. 1, pp. 666–677, 2025.
- [15] D. Kurniadi, A. Mulyani, and D. A. N. Annisa, “Rancang Bangun Sistem Informasi Rekam Medis Pada Klinik Menggunakan Metode Extreme Programing,” *Jurnal Algoritma*, vol. 17, no. 2, pp. 440–451, 2020.
- [16] R. Putri, “Perancangan Sistem Informasi Desa Menggunakan Exteme Programming,” *Journal of Information Technology, Software Engineering and Computer Science (ITSECS)*, vol. 2, no. 3, pp. 140–149, Jul. 2024, doi: 10.58602/itsecs.v2i3.156.
- [17] R. A. Sukanto and M. Shalahudin, “Rekayasa Perangkat Lunak : Terstruktur dan berorientasi objek,” 2018.
- [18] I. Rahmayuni, Y. Sonatha, A. Alanda, and A. Erianda, “Sistem Informasi Harga Komoditas Pangan untuk Pasar-Pasar di Kota Padang,” *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 1, no. 1, pp. 25–31, 2020, doi: 10.30630/jitsi.1.1.5.
- [19] N. Marthiawati, K. Kurniawansyah, H. Nugraha, and F. Khairunnisa, “Pelatihan Pembuatan UML (Unified Modelling Language) Menggunakan Aplikasi Draw.io Pada Prodi Sistem Informasi Universitas Muhammadiyah Jambi,” *Jurnal Inovasi dan Sosial Pengabdian*, vol. 1, no. 2, pp. 25–33, 2024.
- [20] I. Kurniawan and F. Rozi, “REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android,” *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 1, no. 4, pp. 127–132, 2020.
- [21] Bagas Ardianto, Jaroji Jaroji, and M.Asep Subandri, “Pengembangan Aplikasi Simasjid Menggunakan Metode Extreme Programming,” *Jurnal Penelitian Rumpun Ilmu Teknik*, vol. 3, no. 3, pp. 187–202, Aug. 2024, doi: 10.55606/juprit.v3i3.4249.
- [22] D. Irwanto, *Refactoring Pada Object Oriented Software dan Object Database*. Penerbit Andi, 2010.
- [23] I. R. Dhaifullah, M. Muttanifudin H, A. A. Salsabila, and M. A. Yaqin, “Survei Teknik Pengujian Software,” *Journal Automation Computer Information System*, vol. 2, no. 1, pp. 31–38, 2022, doi: 10.47134/jacis.v2i1.42.
- [24] E. Kurniawan, N. Nofriadi, and A. Nata, “Penerapan System Usability Scale (Sus) Dalam Pengukuran Kebergunaan Website Program Studi Di Stmik Royal,” *Journal of Science and Social Research*, vol. 5, no. 1, pp. 43–49, 2022.
- [25] A. Nikiforova and K. McBride, “Open government data portal usability: A user-centred usability analysis of 41 open government data portals,” *Telematics and Informatics*, vol. 58, p. 101539, 2021, doi: https://doi.org/10.1016/j.tele.2020.101539.
- [26] B. Ansari, M. Barati, and E. G. Martin, “Enhancing the usability and usefulness of open government data: A comprehensive review of the state of open government data visualization research,” *Gov Inf Q*, vol. 39, no. 1, p. 101657, Jan. 2022, doi: 10.1016/j.giq.2021.101657.
- [27] T. Wahyuningrum, “Mengukur Usability Perangkat Lunak,” no. 1596, p. 96, 2021.