



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 5 No. 2 (2026) pp: 9281-9289

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Rancang Bangun Sistem Agregasi Data E-Procurement Berbasis Web Scraping Menggunakan Framework Next.js

Agustinus Bada Buraq, Rizky Andriyanto, Jupron

Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

agustinusbada04@gmail.com, rizkyandriyanto16@gmail.com, dosen02664@unpam.ac.id

Abstrak

Staf PT Segara Lentera Teknologi setiap hari harus membuka belasan situs pengadaan secara bergantian hanya untuk mencari tender atau permintaan produk yang relevan dengan bisnis perusahaan. Cara kerja ini memakan waktu besar dan berulang kali membuat peluang terlewat karena informasi datang terlambat. Penelitian ini membangun sistem dashboard berbasis web yang mengambil alih pekerjaan tersebut secara otomatis melalui teknik web crawling. Sistem dikembangkan berdasarkan observasi langsung terhadap proses manual di perusahaan dan wawancara dengan pembimbing lapangan untuk memahami kebutuhan spesifik. Arsitektur yang digunakan menggabungkan Next.js dengan Server-Side Rendering untuk antarmuka yang cepat dan responsif, Puppeteer dengan plugin Stealth sebagai agen crawling untuk halaman dinamis berbasis JavaScript, MongoDB sebagai database dokumen NoSQL terpadu, dan Agenda.js sebagai scheduler tugas latar belakang. Data diekstraksi dari dua sumber utama: Indonet sebagai direktori Business-to-Business dan Layanan Pengadaan Secara Elektronik (LPSE) sebagai portal tender pemerintah. Sebelum masuk ke database, data dibersihkan menggunakan ekspresi reguler untuk menghilangkan tag HTML dan karakter tidak valid. Di dashboard, pengguna hanya melihat data yang sesuai kategori bisnis yang dipilih saat registrasi. Sistem dilengkapi fitur bookmark, manajemen profil dengan upload foto, klasifikasi sektor otomatis berbasis pencocokan kata kunci, dan panel admin untuk pemantauan crawler dan manajemen data. Pengujian fungsional dengan metode black-box menunjukkan seluruh 12 skenario utama berjalan sesuai spesifikasi, dan sistem berhasil mengonsolidasikan data pengadaan dari kedua sumber ke satu antarmuka terpusat tanpa intervensi manual.

Kata kunci: Web Crawling; MongoDB; Agenda.js; Sistem Informasi; E-Procurement; Next.js; Puppeteer Stealth

1. Latar Belakang

Perkembangan industri pengadaan barang dan jasa di Indonesia mendorong perusahaan penyedia jasa untuk terus memperbarui informasi kebutuhan mitra secara cepat dan akurat. Informasi pengadaan kini tersebar di berbagai platform, mulai dari portal Business-to-Business (B2B) komersial seperti Indonet hingga sistem pengadaan pemerintah seperti Layanan Pengadaan Secara Elektronik (LPSE). Fragmentasi sumber data ini menjadi beban nyata bagi perusahaan penyedia jasa, khususnya yang beroperasi di sektor teknologi informasi. PT Segara Lentera Teknologi, sebuah perusahaan system integrator yang berbasis di Jakarta Selatan, menghadapi persoalan ini setiap hari. Staf perusahaan harus membuka belasan situs mitra secara bergantian, menyaring informasi yang relevan, lalu mencatatnya ke dokumen internal. Cara kerja ini memakan waktu besar, mengandung risiko duplikasi data, dan berulang kali membuat peluang tender terlewat karena informasi datang terlambat[1]. Fragmentasi sumber informasi pengadaan yang tidak terintegrasi terbukti menghambat pengambilan keputusan bisnis dan menurunkan daya saing perusahaan dalam merespons peluang pasar[2].

Solusi yang diusulkan dalam penelitian ini adalah membangun sistem dashboard berbasis web yang mengotomasi seluruh proses pemantauan tersebut menggunakan teknik web crawling. Web crawling adalah proses pengambilan data dari internet secara sistematis melalui navigasi otomatis yang memungkinkan konsolidasi informasi dari berbagai sumber ke dalam satu basis data terpusat[3]. Dengan pendekatan ini, staf tidak perlu lagi menelusuri situs satu per satu karena sistem mengerjakan proses tersebut secara otomatis melalui scheduler latar belakang, menyimpan hasilnya ke database, dan menyajikannya melalui dashboard yang dapat difilter sesuai kategori bisnis pengguna.

Beberapa penelitian terdahulu telah mengeksplorasi penerapan web crawling dan web scraping untuk pengumpulan data otomatis. Indawa, Darwis, dan Sintaro[4] mengimplementasikan web crawling untuk mengumpulkan informasi artikel ilmiah pada sistem LPPM universitas menggunakan pendekatan navigasi rekursif. Penelitian tersebut menunjukkan bahwa teknik crawling mampu mereduksi waktu pengumpulan data secara signifikan dibanding metode manual, tetapi sistem yang dibangun bersifat statis, hanya menarget satu sumber data, dan tidak dilengkapi antarmuka real-time bagi pengguna akhir.

Rizquina dan Ratnasari[5] meneliti implementasi web scraping pada situs e-commerce Tokopedia menggunakan metode HTML Parsing dan CSS Selector berbasis Python, mampu mengekstraksi ratusan data produk dalam satu sesi. Meski demikian, penelitian tersebut tidak mengintegrasikan hasil ekstraksi ke dalam sistem informasi yang dapat diakses pengguna bisnis secara personal dan terfilter berdasarkan preferensi kategori.

Rismawan dkk[6] membangun sistem website berita online menggunakan metode crawling data untuk mengotomasi pembaruan konten dari berbagai sumber berita nasional. Sistem ini membuktikan efektivitas otomatis pengumpulan data teks dari halaman web dinamis, tetapi fokusnya terbatas pada domain berita dan tidak menerapkan personalisasi berbasis profil pengguna.

Sulistya dkk[7] membangun sistem crawling otomatis pada Google Scholar menggunakan antarmuka berbasis web untuk manajemen data publikasi ilmiah. Sistem tersebut berhasil mengotomasi proses yang sebelumnya dilakukan secara manual oleh peneliti, namun belum menerapkan personalisasi berbasis profil pengguna maupun integrasi database yang tersinkronisasi langsung dengan perubahan di sumber aslinya.

Maulana dan Prasetyo[8] merancang aplikasi pengadaan barang dan jasa universitas berbasis web menggunakan metode Extreme Programming dan pemodelan UML. Penelitian tersebut berhasil membangun dashboard pengadaan yang terstruktur dengan panel admin dan manajemen pengguna, namun sumber datanya dimasukkan secara manual oleh operator tanpa mekanisme pengambilan data otomatis dari portal pengadaan eksternal.

Berdasarkan kajian tersebut, terdapat celah yang belum dijawab secara menyeluruh. Pertama, belum ada sistem yang menggabungkan crawling otomatis dari sumber pengadaan B2B komersial dan portal pemerintah sekaligus dalam satu pipeline terintegrasi. Kedua, sistem crawling yang ada tidak menyajikan hasil ekstraksi secara otomatis kepada pengguna dengan personalisasi berbasis kategori bisnis. Ketiga, kombinasi teknologi modern berupa Next.js, Puppeteer Stealth, MongoDB, dan Agenda.js belum pernah diterapkan dalam konteks sistem informasi e-procurement di lingkup riset Indonesia.

Tujuan penelitian ini adalah merancang dan mengimplementasikan sistem web crawling terintegrasi yang mampu mengotomasi pengambilan data penawaran produk dan jasa dari Indonet dan LPSE, menyimpannya ke dalam database MongoDB menggunakan Mongoose, serta menyajikannya melalui dashboard Next.js yang terfilter sesuai profil bisnis pengguna.

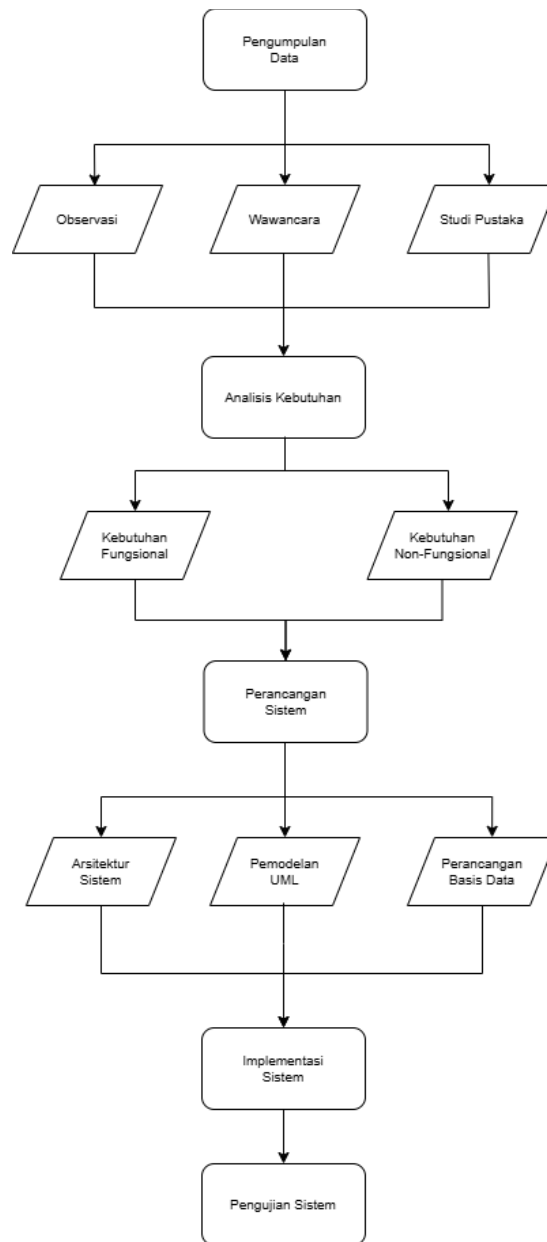
2. Metode Penelitian

Penelitian ini menggunakan pendekatan pengembangan sistem berbasis prototyping yang terdiri dari lima tahap berurutan: pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Tahapan ini dipilih karena memungkinkan evaluasi berkelanjutan antara pengembang dan pengguna akhir selama proses berlangsung[9].

Tahap pertama adalah pengumpulan data melalui tiga metode. Pertama, observasi langsung terhadap proses pemantauan situs pengadaan yang dikerjakan secara manual oleh staf PT Segara Lentera Teknologi. Kedua, wawancara dengan pembimbing lapangan untuk memahami kebutuhan spesifik perusahaan terkait kategori produk dan jasa yang diprioritaskan. Ketiga, studi pustaka terhadap dokumentasi teknis Next.js, Puppeteer, MongoDB, dan Agenda.js sebagai fondasi pengembangan sistem[10][11][12][13].

Tahap kedua adalah analisis kebutuhan sistem yang menghasilkan dua kelompok spesifikasi. Kebutuhan fungsional mencakup autentikasi pengguna, personalisasi kategori bisnis saat onboarding, pencarian dan filter data pada dashboard, tampilan detail produk dan jasa, fitur bookmark, manajemen profil dengan upload foto, klasifikasi sektor otomatis, serta panel admin. Kebutuhan non-fungsional meliputi responsivitas antarmuka pada berbagai perangkat, kecepatan pemuatan halaman melalui Server-Side Rendering, dan keamanan autentikasi berbasis enkripsi bcrypt.

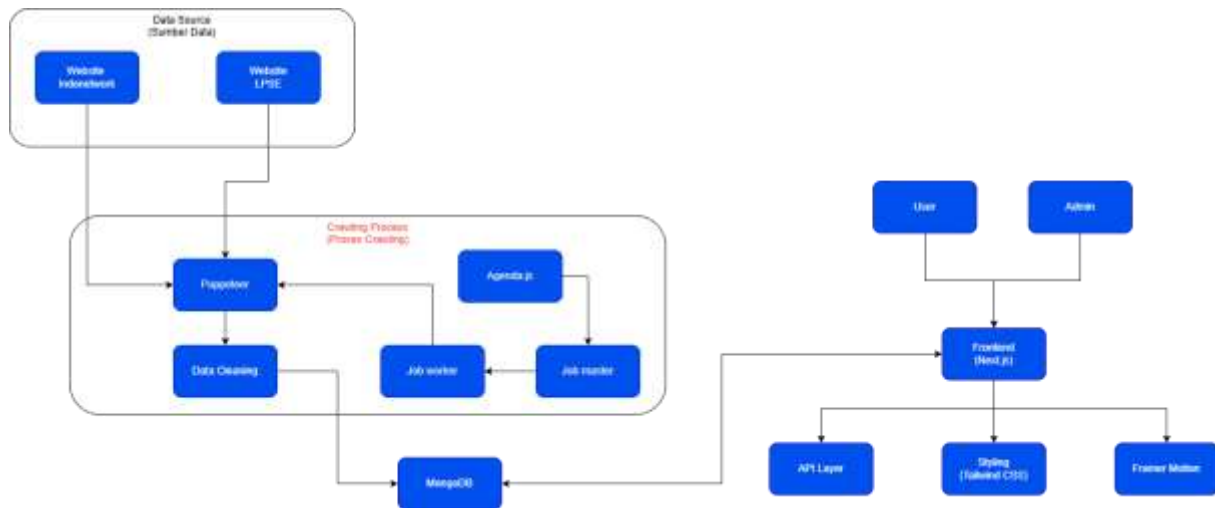
Tahap ketiga adalah perancangan sistem yang meliputi perancangan arsitektur, pemodelan UML, dan perancangan skema database. Tahap keempat adalah implementasi menggunakan tumpukan teknologi yang telah ditentukan. Tahap kelima adalah pengujian menggunakan metode black-box testing untuk memverifikasi seluruh fitur berjalan sesuai spesifikasi.



Gambar 1. Tahapan Penelitian

2.1 Perancangan Sistem

Sistem dirancang menggunakan arsitektur client-server berbasis API dan database cloud dengan tiga lapisan utama yang saling terhubung, sebagaimana ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Sistem

Lapisan pertama adalah frontend yang dibangun menggunakan framework Next.js dengan App Router dan TypeScript. Next.js dipilih karena dukungannya terhadap Server-Side Rendering yang meningkatkan kecepatan pemuatan halaman secara signifikan[11]. Antarmuka menggunakan Tailwind CSS untuk tata letak responsif dan Framer Motion untuk animasi transisi halaman.

Lapisan kedua adalah database yang menggunakan MongoDB sebagai layanan penyimpanan dokumen NoSQL. MongoDB dipilih karena fleksibilitas skema dokumen BSON/JSON yang memudahkan penyimpanan data terstruktur maupun semi-terstruktur seperti array jadwal lelang yang bersifat nested[13]. Operasi database dikelola menggunakan Mongoose sebagai ODM (Object Document Mapper) pada Node.js. Sistem autentikasi menggunakan enkripsi bcrypt untuk menyimpan kredensial pengguna.

Lapisan ketiga adalah mesin crawling yang menggunakan Puppeteer dengan plugin puppeteer-extra-plugin-stealth, sebuah pustaka Node.js yang mengontrol Chrome melalui Chrome DevTools Protocol. Plugin stealth ditambahkan untuk menyamarkan karakteristik bot otomatis dari deteksi Cloudflare dan WAF (Web Application Firewall) pada portal LPSE[10]. Proses crawling dijadwalkan dan dikelola oleh Agenda.js sebagai background job scheduler yang menyimpan status antrean tugas di koleksi agendaJobs pada MongoDB.

Tabel 1. Skema Koleksi Users (UserSchema)

Field	Tipe Data	Keterangan
username	String	Username unik pengguna (required, unique)
email	String	Email unik, lowercase (required)
password_hash	String	Kata sandi terenkripsi (bcrypt)
nama_lengkap	String	Nama lengkap pengguna
perusahaan	String	Nama perusahaan

Tabel 2. Skema Koleksi Tenders (TenderSchema)

Field	Tipe Data	Keterangan
lelangId	String	ID unik paket lelang (required, unique)
nama_paket	String	Nama paket proyek (required)
instansi	String	Instansi penyelenggara
hps	String	Harga Perkiraan Sendiri
pagu	String	Nilai anggaran
kategori	String	Kategori pengadaan
tag	String	Sektor industri (klasifikasi otomatis)
metode_pengadaan	String	Metode pelaksanaan

Tabel 3. Skema Koleksi Products (ProductSchema)

Field	Tipe Data	Keterangan
nama_produk	String	Nama produk (required)
deskripsi	String	Deskripsi produk
nama_perusahaan	String	Nama perusahaan penjual
harga	String	Harga produk
gambar_url	String	URL gambar produk
tag	String	Sektor kecocokan industri
kategori	String	Kategori produk
kota	String	Kota asal produk

Relasi antara koleksi users dan koleksi tenders/products bersifat logis melalui atribut bidang_minat/tag. Filtering dilakukan di lapisan aplikasi: sistem membaca preferensi sektor industri pengguna yang tersimpan di dokumen users, lalu menampilkan dokumen dari koleksi tenders dan products yang nilai tag-nya sesuai. Pendekatan ini menjaga skema database tetap fleksibel tanpa ketergantungan referensi asing langsung.

Selain koleksi data utama, database MongoDB mengelola dua koleksi sistem tambahan: koleksi agendaJobs yang dikelola otomatis oleh Agenda.js untuk menyimpan status antrian dan histori eksekusi tugas latar belakang, serta koleksi GridFS (fs.files dan fs.chunks) untuk penyimpanan biner foto profil pengguna.

3. Hasil dan Diskusi

Sistem dashboard berhasil dibangun sebagai aplikasi web full-stack menggunakan Next.js dengan App Router, Puppeteer dengan plugin Stealth, MongoDB dengan Mongoose, dan Agenda.js sebagai scheduler. Seluruh komponen berjalan dalam satu repositori terpadu tanpa server backend terpisah, memanfaatkan fitur API Routes bawaan Next.js untuk menangani logika crawling dan pembersihan data. Isolasi proses antara Next.js dan Agenda.js dikelola menggunakan pustaka Concurrently agar Fast Refresh pada Next.js tidak memicu inisialisasi ulang koneksi database dan instansi Puppeteer secara terus-menerus.

3.1.1 Implementasi Mesin Crawling

Mesin crawling di implementasikan menggunakan Puppeteer yang berjalan secara headless di sisi server dengan plugin puppeteer-extra-plugin-stealth aktif. Plugin ini menghapus karakteristik bot otomatis pada browser Chromium termasuk menyembunyikan properti window.navigator.webdriver dan memodifikasi HTTP User-Agent sehingga lalu lintas crawling tidak terpicu oleh filter Cloudflare yang melindungi portal LPSE.

Penjadwalan crawling dikelola oleh Agenda.js yang menyimpan seluruh antrian tugas di koleksi agendaJobs pada MongoDB. Terdapat tiga jenis tugas terdefinisi: pertama, 'scrape list lelang' yang berjalan setiap 15 menit untuk memantau paket baru di LPSE Kota Tangerang dan LPSE Kota Bekasi; kedua, 'smart polling jadwal' yang berjalan setiap 30 menit untuk mendistribusikan pembaruan detail jadwal bagi seluruh tender aktif; dan ketiga, 'scrape detail jadwal' yang dieksekusi secara bertahap dengan jeda 10 detik per tender untuk menghindari pemblokiran rate limit portal LPSE.

Untuk Indonetnetwork, sistem menavigasi halaman kategori produk lalu mengekstraksi nama produk, nama perusahaan, wilayah, dan nomor kontak dari setiap kartu produk. Untuk LPSE, sistem mengakses halaman daftar paket lelang dan mengekstraksi nama paket, instansi penyelenggara, nilai pagu, nilai HPS, dan tanggal tahapan jadwal. Kedua proses menggunakan metode page.evaluate() untuk mengeksekusi skrip JavaScript di dalam konteks halaman, memastikan data yang dimuat secara dinamis sudah sepenuhnya tersedia sebelum ekstraksi dimulai[10].

Hasil ekstraksi mentah dari kedua sumber mengandung karakter tidak valid berupa tag HTML sisa, spasi ganda, dan karakter khusus akibat perbedaan encoding antar portal. Pembersihan dilakukan menggunakan ekspresi reguler sebelum data dikirim ke MongoDB.

Tabel 4. Pola Ekspresi Reguler untuk Pembersihan Data

Pola	Fungsi	Contoh Input	Hasil
/<[>]*>/g	Hapus tag HTML	Laptop	Laptop
/\s+/g	Normalisasi spasi	Rp 500.000	Rp 500.000
/[^\x20-\x7E]/g	Hapus karakter non-ASCII	Jasa IT (non-ASCII)	Jasa IT
/\s+ \s+\$ /g	Hapus spasi tepi	Konstruksi	Konstruksi

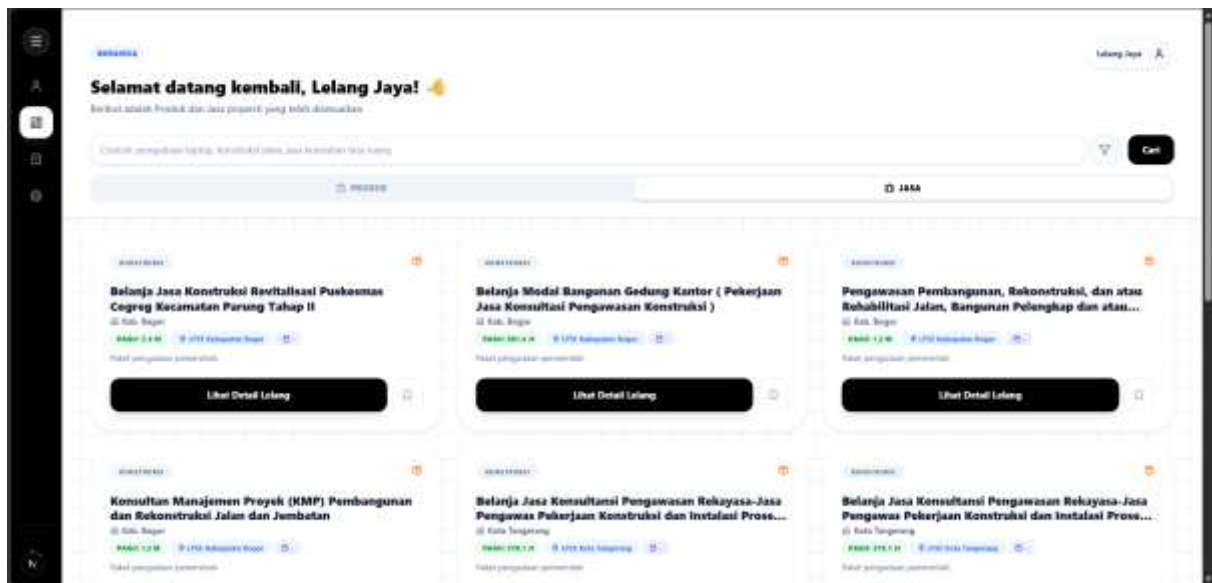
Setelah pembersihan, data dikirim ke koleksi products atau tenders di MongoDB menggunakan operasi bulk write dengan metode updateOne beropsi upsert untuk menghindari duplikasi entri. Guna mengantisipasi ketidakkonsistenan markup HTML LPSE, modul scraper menerapkan pencarian ID unik melalui tiga metode berlapis: pertama, mengekstrak ID angka dari pola regex tautan href; kedua, mengambil nilai ID dari kolom data JSON LPSE; dan ketiga, menyusun composite hash dari kombinasi nama paket dan instansi sebagai fallback.

3.1.2 Implementasi Antarmuka dan Personalisasi

Dashboard utama dibangun menggunakan komponen React dengan Server-Side Rendering yang diaktifkan melalui Next.js. Setiap permintaan halaman diproses di sisi server: Next.js membaca sesi pengguna, mengambil bidang_minat dari dokumen profil di MongoDB, lalu meneruskan parameter filter ke query database sebelum HTML dikirim ke browser. Pendekatan ini menghasilkan waktu tampil konten pertama yang lebih cepat dibanding Client-Side Rendering[11].

Saat pengguna baru menyelesaikan registrasi, sistem mengarahkan mereka ke halaman onboarding untuk memilih sektor industri yang meliputi Teknologi, Konstruksi, Otomotif, Pangan, Kesehatan, dan Energi. Pilihan ini disimpan ke field bidang_minat pada dokumen profil di MongoDB. Setiap kali pengguna membuka dashboard, sistem membaca nilai ini dan menggunakannya sebagai parameter filter pada query ke koleksi tenders dan products, sehingga setiap pengguna hanya melihat data yang relevan dengan bidang bisnis mereka.

Antarmuka dashboard terbagi menjadi dua tab utama. Tab pertama menampilkan data produk dari Indonetwork dalam bentuk kartu yang memuat nama produk, nama perusahaan, wilayah, kategori, dan tombol kontak langsung. Tab kedua menampilkan data jasa dan lelang dari LPSE dalam bentuk tabel terstruktur dengan informasi jadwal tahapan yang diperbarui otomatis oleh scheduler. Pada layar mobile, sidebar berubah menjadi menu bawah secara otomatis melalui kelas responsif Tailwind CSS. Sistem juga mendukung upload foto profil pengguna yang disimpan menggunakan dua metode: penyimpanan lokal di direktori /public/uploads/avatars/ server, atau penyimpanan biner langsung ke MongoDB menggunakan spesifikasi GridFS yang memecah file gambar menjadi potongan data 255 KB pada koleksi fs.files dan fs.chunks.

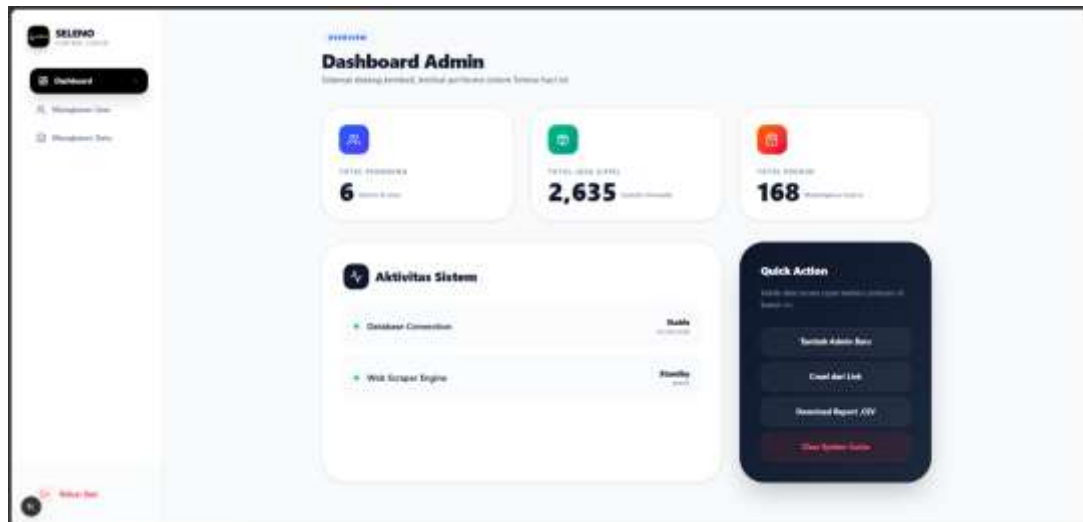


Gambar 3. Halaman Dashboard Utama

3.1.3 Implementasi Panel Admin

Panel admin dirancang khusus untuk pengguna dengan nilai role = admin pada dokumen users. Sistem menggunakan role-based access control berbasis middleware Next.js yang memverifikasi peran pengguna pada setiap permintaan ke rute /admin. Jika pengguna biasa mencoba mengakses rute ini secara langsung, sistem mengarahkan mereka kembali ke dashboard utama.

Melalui panel admin, administrator dapat memantau total pengguna terdaftar, jumlah data produk dari Indonetwork, dan jumlah data tender dari LPSE. Admin juga dapat menjalankan proses crawling secara manual, mengunduh seluruh data dalam format CSV, dan menghapus entri yang sudah kedaluwarsa dari halaman Manajemen Data. Seluruh angka pada kartu statistik diperbarui secara otomatis setiap kali ada perubahan data di MongoDB.



Gambar 4. Halaman Admin Control Center

3.1.4 Klasifikasi Sektor Otomatis

Sistem dilengkapi algoritma pencocokan kata kunci terintegrasi yang memetakan nama paket proyek lelang ke dalam enam sektor industri secara otomatis. Kata kunci per sektor meliputi: Teknologi (sistem informasi, aplikasi, software, hardware, cctv, server, jaringan, digital), Konstruksi (pembangunan, jalan, jembatan, gedung, rehabilitasi, drainase), Kesehatan (medis, obat, alat kesehatan, rumah sakit, puskesmas), Pangan (pertanian, beras, pupuk, catering, konsumsi), Logistik (pengiriman, distribusi, transportasi, gudang), dan Otomotif (kendaraan dinas, mobil, motor, bus, spare part). Hasil klasifikasi ini disimpan di field tag pada dokumen tender dan produk, memudahkan penyaringan serta pencocokan komoditas secara instan.

3.2 Hasil Pengujian Fungsional

Pengujian dilakukan menggunakan metode black-box testing yang memverifikasi kesesuaian perilaku sistem dengan spesifikasi fungsional tanpa memperhatikan struktur kode internal. Setiap fitur diuji melalui dua skenario: penggunaan normal dengan masukan valid dan penggunaan dengan masukan tidak valid untuk memverifikasi ketahanan sistem terhadap kesalahan pengguna.

Tabel 5. Hasil Pengujian Black-Box

No	Fitur yang Diuji	Skenario	Hasil yang Diharapkan	Status
1	Registrasi pengguna	Input data valid	Akun terbuat, diarahkan ke onboarding	Berhasil
2	Registrasi pengguna	Email sudah terdaftar	Muncul pesan kesalahan	Berhasil
3	Login	Kredensial benar	Masuk ke dashboard	Berhasil
4	Login	Kata sandi salah	Muncul pesan kesalahan	Berhasil
5	Filter kategori	Kategori Teknologi dipilih	Hanya data Teknologi tampil	Berhasil
6	Pencarian data	Kata kunci dimasukkan	Data relevan muncul	Berhasil
7	Bookmark item	Item ditandai	Item muncul di halaman tersimpan	Berhasil
8	Crawling manual	Admin klik tombol crawl	Data baru masuk ke MongoDB	Berhasil
9	Ekspor CSV	Admin klik unduh	File CSV terunduh dengan data lengkap	Berhasil
10	Akses admin oleh user biasa	Akses rute /admin langsung	Dialihkan ke dashboard	Berhasil
11	Pembaruan profil	Kategori diubah	Tampilan dashboard berubah sesi berikutnya	Berhasil
12	Upload foto profil	File gambar diunggah	Avatar profil diperbarui via GridFS/lokal	Berhasil

Seluruh 12 skenario pengujian menunjukkan hasil sesuai spesifikasi. Tidak ditemukan kegagalan pada pengujian skenario normal maupun skenario masukan tidak valid. Tingkat keberhasilan pengujian fungsional mencapai 100% dari total skenario yang diuji, mengindikasikan bahwa implementasi sistem telah memenuhi seluruh kebutuhan fungsional yang diidentifikasi dari observasi proses manual di PT Segara Lentera Teknologi.

3.2.1 Analisis Hasil Crawling

Proses crawling dijalankan sebanyak lima sesi dalam periode pengujian untuk mengukur konsistensi hasil ekstraksi. Setiap sesi menarget 10 halaman dari Indonetnetwork dan 5 halaman dari LPSE, dijadwalkan otomatis oleh Agenda.js.

Tabel 6. Hasil Pengujian Crawling Per Sesi

Sesi	Data Indonetnetwork	Data LPSE	Data Duplikat	Data Bersih Tersimpan	Waktu (detik)
1	87	42	3	126	48
2	91	40	5	126	51
3	88	43	4	127	49
4	90	41	5	126	51
5	89	42	5	126	50
Rata-rata	89	41,6	4,4	126,2	49,6

Dari Tabel 6 terlihat bahwa rata-rata data yang berhasil diekstraksi per sesi adalah 89 entri dari Indonetnetwork dan 41,6 entri dari LPSE. Rata-rata data duplikat yang berhasil disaring oleh mekanisme upsert bulk write sebanyak 4,4 entri per sesi, artinya tingkat akurasi penyaringan duplikat mencapai 96,6% dari total entri yang diproses. Rata-rata waktu penyelesaian satu sesi crawling adalah 49,6 detik untuk total sekitar 130 entri dari dua sumber sekaligus—angka yang cukup efisien mengingat Puppeteer harus menunggu rendering JavaScript selesai pada setiap halaman sebelum ekstraksi dimulai[10].

4. Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan sistem dashboard berbasis web yang mengotomasi proses pemantauan data pengadaan barang dan jasa menggunakan teknik web crawling. Sistem menjawab permasalahan utama yang dihadapi PT Segara Lentera Teknologi, yaitu ketergantungan staf pada proses manual yang memakan waktu besar dan berulang kali menyebabkan peluang tender terlewat akibat keterlambatan informasi. Dengan mengintegrasikan Puppeteer dengan plugin Stealth sebagai mesin crawling untuk halaman dinamis, MongoDB dengan Mongoose sebagai database dokumen NoSQL terpadu, Agenda.js sebagai scheduler tugas latar belakang, dan Next.js dengan Server-Side Rendering sebagai antarmuka pengguna, sistem berhasil mengonsolidasikan data pengadaan dari dua sumber eksternal—Indonetnetwork sebagai direktori B2B dan LPSE sebagai portal tender pemerintah—ke dalam satu antarmuka terpusat yang terfilter otomatis berdasarkan profil bisnis pengguna. Hasil pengujian fungsional dengan metode black-box menunjukkan tingkat keberhasilan 100% dari 12 skenario yang diuji, membuktikan bahwa seluruh fitur utama sistem berjalan sesuai spesifikasi. Pengujian crawling selama lima sesi menghasilkan rata-rata 126,2 entri data bersih per sesi dari dua sumber dengan rata-rata waktu penyelesaian 49,6 detik dan tingkat akurasi penyaringan duplikat 96,6%. Angka-angka ini menunjukkan bahwa sistem mampu bekerja secara konsisten dan efisien dalam kondisi operasional nyata. Meski demikian, penelitian ini memiliki sejumlah keterbatasan yang membuka ruang untuk pengembangan lebih lanjut. Pertama, cakupan sumber data saat ini terbatas pada dua portal. Penambahan sumber ke portal seperti BELA Pengadaan atau SiRUP akan memperkaya variasi data yang tersedia. Kedua, mesin crawling belum dilengkapi mekanisme rotasi proksi atau penanganan captcha otomatis, sehingga rentan terhadap pemblokiran jika frekuensi crawling ditingkatkan secara signifikan—meskipun plugin Stealth sudah mengurangi risiko ini pada kondisi normal. Ketiga, sistem belum mengintegrasikan kecerdasan buatan untuk analisis tren harga atau rekomendasi tender berdasarkan riwayat data yang telah terkumpul. Basis data historis yang dibangun sistem ini sebenarnya sudah cukup kaya untuk dijadikan fondasi fitur tersebut. Penelitian selanjutnya disarankan mengatasi ketiga keterbatasan ini agar sistem berkembang dari agregator data menjadi platform intelijen pengadaan yang lebih komprehensif.

Referensi

- [1] dan T. S. D. Ardiyansah, O. Pahlevi, "Implementasi Metode Prototyping pada Sistem Informasi Pengadaan Barang Cetakan Berbasis Web," *J. Inform. dan Teknol.*, vol. 3, no, p. hal. 151–157, 2021, [Online]. Available: doi: 10.37034/jidt.v3i4.137
- [2] M. Rahmatika dan M. I. Asmara, "Data Visualisasi Pada Website Layanan Pengadaan Secara Elektronik (LPSE) Menggunakan Web Scraping," *Pros. Semin. Hi-Tech*, vol. 1, no, p. hal. 1–8, 2022.
- [3] Deddy dan E. Mailoa, "Implementasi Web Crawling untuk Pencarian Harga Sparepart Pada PT Asuransi Sinar Mas," *JATISI (J. Tek. Inform. dan Sist. Informasi)*, vol. 7, no, p. hal. 416–428, 2020.
- [4] dan S. S. L. Indawa, D. Darwis, "Implementasi Web Crawling Untuk Mengumpulkan Informasi Artikel Ilmiah (Studi Kasus: LPPM Universitas XYZ)," *J. Teknol. dan Sist. Inf.*, vol. 3, no, p. hal. 58–66, 2022.
- [5] A. Z. Rizquina dan C. I. Ratnasari, "Implementasi Web Scraping untuk Pengambilan Data Pada Website E-Commerce," *J. Teknol. dan Sist. Inf. Bisnis*, vol. 5, no, p. hal. 377–383, 2023, [Online]. Available: doi: 10.47233/jteksis.v5i4.913

DOI: <https://doi.org/10.31004/riggs.v5i2.9661>

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

- [6] dan I. S. S. A. Rismawan, A. P. Hartono, Y. Syahidin, "Implementasi Website Berita Online Menggunakan Metode Crawling Data dengan Bahasa Pemrograman Python," *JATISI (J. Tek. Inform. dan Sist. Informasi)*, vol. 10, n, p. hal. 167–178, 2023, [Online]. Available: doi: 10.35957/jatisi.v10i3.4902
- [7] dan A. R. Y. I. Sulistya, A. C. Wardhana, M. Istighosah, "Automated Google Scholar Crawling with a Web-Based Tool for Publication Data Management," *J. Teknol. dan Sist. Inf. Bisnis*, vol. 6, no, p. hal. 768–773, 2024, [Online]. Available: doi: 10.47233/jteksis.v6i4.1604
- [8] S. A. Maulana dan R. T. Prasetyo, "Rancang Bangun Aplikasi Pengadaan Barang dan Jasa Universitas Negeri XYZ dengan Metode Extreme Programming," *JISKA (J. Inform. Sunan Kalijaga)*, vol. 8, no, p. hal. 186–198, 2023, [Online]. Available: doi: 10.14421/jiska.2023.8.3.186-198
- [9] K. P. dan B. Maxim, "Software Engineering: A Practitioner's Approach," 9th ed. New York McGraw-Hill, 2020.
- [10] Google, "Puppeteer Documentation v21," *Online*, 2023.
- [11] Vercel, "Next.js Documentation v14," *Online*, 2023.
- [12] Agenda Inc., "Agenda.js Documentation: Job Scheduling for Node.js," *Online*, 2023, [Online]. Available: <https://github.com/agenda/agenda>
- [13] M. Inc., "Mongoose ODM Documentation v7," *Online*, 2023.