

PERANCANGAN EKSTENSI CHROME BERBASIS FIREFLY ALGORITHM UNTUK OPTIMASI SCRAPING DAN IMAGE TEXT SWAPPING PADA PLATFORM 1688

Dini Rahmawati¹, Tjahjanto², Noor Falih³

^{1,2,3}Program Studi Informatika, Fakultas Ilmu Komputer

Universitas Pembangunan Nasional “Veteran” Jakarta

Jl. RS. Fatmawati Raya, Pondok Labu, Cilandak, Jakarta Selatan

¹2210511093@mahasiswa.upnvj.ac.id*

Abstrak. Pelaku usaha impor di Indonesia banyak memanfaatkan platform 1688.com sebagai sumber produk, namun informasi penting pada gambar produk (in-image text) disajikan dalam bahasa Mandarin sehingga sulit dipahami menggunakan penerjemah konvensional. Penelitian ini merancang ekstensi Chrome yang melakukan web scraping data produk secara otomatis, menerapkan Image Text Swapping berbasis Optical Character Recognition (OCR), serta mengoptimalkan Pipeline Scheduler menggunakan Firefly Algorithm. Penelitian menggunakan metode Research and Development (R&D) dengan model V-Model, dievaluasi melalui pengujian fungsional setiap modul, pengujian optimasi Firefly Algorithm, serta User Acceptance Testing (UAT). Hasil pengujian menunjukkan modul OCR memperoleh akurasi 86,41%, modul Translation mencapai Success Rate 100% dengan akurasi 97,36%, dan Firefly Algorithm menurunkan waktu pemrosesan satu halaman produk dari 202,140 detik menjadi 20,63 detik atau berkurang 89,79% dibanding Sequential Processing. UAT memperoleh tingkat penerimaan pengguna 84,38% dengan kategori Sangat Baik. Sistem terbukti membantu proses pengambilan data, penerjemahan gambar, dan penyusunan katalog produk multibahasa secara lebih cepat, akurat, dan efisien.

Kata Kunci: Chrome Extension, Firefly Algorithm, Image Text Swapping, OCR, Web Scraping

1 Pendahuluan

Pertumbuhan e-commerce global yang pesat turut mendorong pelaku usaha impor di Indonesia memanfaatkan platform business-to-business (B2B) 1688.com sebagai sumber produk. Namun demikian, seluruh antarmuka dan informasi produk pada platform tersebut disajikan dalam bahasa Mandarin, terutama pada teks yang tertanam langsung di dalam gambar produk (in-image text) seperti spesifikasi teknis dan keunggulan produk. Teks tersebut tidak dapat dikenali oleh ekstensi penerjemah konvensional karena hanya bekerja pada teks HTML yang dapat dibaca secara struktural, bukan pada elemen visual gambar [1]. Kondisi ini menghambat proses product analysis sehingga pelaku usaha impor berisiko mengambil keputusan pengadaan yang kurang tepat.

Selama ini pelaku usaha melakukan pencarian dan penerjemahan informasi produk secara manual satu per satu, sehingga membutuhkan waktu lama ketika menganalisis puluhan hingga ratusan produk. Selain itu, konten pada 1688.com dimuat secara dinamis menggunakan Asynchronous JavaScript and XML (AJAX) dan arsitektur Single Page Application (SPA) sehingga metode pengambilan data berbasis HTTP request konvensional tidak efektif diterapkan pada platform tersebut [2].

Berdasarkan permasalahan tersebut, penelitian ini merancang ekstensi Chrome yang melakukan web scraping data produk dari 1688.com dilengkapi Optical Character Recognition (OCR) dan image text swapping untuk menerjemahkan serta mengganti teks Mandarin pada gambar produk secara langsung. Untuk meningkatkan efisiensi proses scraping dan image text swapping, penelitian ini menerapkan Firefly Algorithm (FA) sebagai mekanisme optimasi Pipeline Scheduler yang menentukan jumlah worker pada setiap tahapan pemrosesan secara adaptif berdasarkan kondisi performa sistem, meliputi makespan, throughput, dan keseimbangan beban antar tahapan (pipeline balance).

Tujuan penelitian ini adalah: (1) merancang ekstensi Chrome yang mampu melakukan web scraping data produk 1688 secara otomatis; (2) mengimplementasikan image text swapping berbasis OCR untuk menerjemahkan teks Mandarin pada gambar produk secara langsung di halaman 1688; dan (3) menerapkan Firefly Algorithm untuk mengoptimalkan penjadwalan proses scraping dan image text swapping pada sistem yang dikembangkan.

2 Tinjauan Pustaka

2.1 Web Scraping dan Document Object Model

Web scraping adalah proses otomatisasi pengumpulan dan ekstraksi data dari halaman web menggunakan perangkat lunak [3]. Pada platform e-commerce modern yang memuat konten secara dinamis, ekstraksi kode sumber HTML mentah sering terkendala karena data produk belum sepenuhnya dimuat saat request awal dilakukan [2]. Teknik web scraping kemudian berkembang menjadi DOM-based scraping yang membaca struktur Document Object Model (DOM) di sisi client setelah seluruh skrip JavaScript selesai dieksekusi [4]. Melalui manipulasi DOM setelah proses rendering selesai pada browser, ekstensi peramban dapat mengekstraksi informasi produk seperti judul, harga, variasi, dan URL gambar secara akurat sesuai elemen visual yang ditampilkan kepada pengguna [5]. Pada penelitian ini, web scraping diterapkan melalui komponen content script pada ekstensi Chrome untuk mengekstraksi data produk secara real-time dari halaman 1688.com.

2.2 OCR dan Image Text Swapping

Optical Character Recognition (OCR) berfungsi mengekstraksi teks dari gambar sehingga sistem dapat membaca informasi visual secara otomatis. Akurasi OCR dipengaruhi oleh kualitas pra-pemrosesan citra seperti binarization dan noise reduction [6], sedangkan pengenalan karakter Mandarin memiliki tingkat kesulitan yang tinggi

dan dapat ditingkatkan melalui kombinasi pra-pemrosesan citra dengan post-OCR correction [7]. Penelitian ini menggunakan EasyOCR, pustaka OCR multibahasa yang mendukung pengenalan teks Mandarin dan Inggris [8]. Setelah teks diekstraksi dan diterjemahkan, tahap Image Text Swapping (scene text editing) menghapus teks asli pada gambar melalui image inpainting dan menyisipkan teks terjemahan pada lokasi yang sama dengan tetap mempertahankan atribut visual seperti font, warna, dan background. Pendekatan berbasis inpainting dan text rendering, seperti model SwapText, terbukti mampu menghasilkan penggantian teks yang lebih natural dengan preservasi tekstur latar belakang [9].

2.3 Firefly Algorithm

Firefly Algorithm (FA) adalah algoritma optimasi berbasis populasi (population-based metaheuristic) yang terinspirasi dari perilaku kunang-kunang dalam menggunakan intensitas cahaya sebagai mekanisme komunikasi untuk menarik individu lain [10]. Tingkat kecerahan (brightness) merepresentasikan kualitas solusi (fitness), sehingga individu dengan kecerahan lebih rendah bergerak menuju individu dengan kecerahan lebih tinggi. FA termasuk kelompok swarm intelligence yang menyeimbangkan proses eksplorasi dan eksploitasi sehingga banyak diterapkan pada permasalahan optimasi seperti penjadwalan dan alokasi sumber daya [11]. Penelitian terkait menunjukkan bahwa FA yang dimodifikasi mampu meningkatkan efisiensi workflow scheduling pada lingkungan hybrid cloud-edge dengan menurunkan makespan [12]. Pada penelitian ini, setiap kandidat solusi (firefly) merepresentasikan satu konfigurasi Pipeline Scheduler yang terdiri atas kombinasi jumlah worker pada setiap tahapan pemrosesan gambar, sehingga FA digunakan untuk menentukan konfigurasi worker optimal berdasarkan kondisi pipeline yang diamati secara real-time.

3 Metodologi Penelitian

3.1 Pendekatan dan Tahapan Penelitian

Penelitian ini menggunakan pendekatan Research and Development (R&D) dengan model pengembangan sistem V-Model, karena model ini menghubungkan setiap tahap perancangan dengan tahap pengujian yang sesuai sehingga hasil pengembangan dapat diverifikasi dan divalidasi secara bertahap. Tahap pendahuluan meliputi identifikasi masalah, studi literatur, perumusan masalah, serta penetapan tujuan penelitian. Selanjutnya proses pengembangan sistem dilakukan melalui tahapan analisis kebutuhan sistem, perancangan sistem dan arsitektur, perancangan modul, implementasi (coding), serta pengujian yang meliputi Unit Testing, Integration Testing, System Testing, dan User Acceptance Testing (UAT). Setiap tahapan perancangan pada sisi kiri V-Model memiliki keterkaitan langsung dengan tahapan pengujian pada sisi kanan sehingga hasil rancangan telah melalui proses verifikasi dan validasi sebelum sistem digunakan.

3.2 Arsitektur Sistem

Sistem dirancang menggunakan pendekatan client-server yang terdiri atas tiga komponen utama, yaitu Chrome Extension berbasis Manifest V3 sebagai client yang berinteraksi langsung dengan halaman produk 1688.com, Backend Flask sebagai pusat pemrosesan dan integrasi layanan, serta Web Dashboard untuk mengelola data hasil scraping dan pengolahan gambar. Melalui Chrome Extension, pengguna dapat menjalankan fitur Import Data, Translate Image, Download Original Image, dan Settings, yang setiap permintaannya dikirim ke Backend Flask melalui REST API. Backend Flask memproses data melalui tahapan ekstraksi teks (OCR), penerjemahan menggunakan Google Translate API, penggantian teks pada gambar melalui Image Text Swapping, serta optimasi konfigurasi pipeline menggunakan Firefly Algorithm, dan terintegrasi dengan Gemini API untuk menghasilkan ringkasan deskripsi produk berbahasa Indonesia. Seluruh hasil pemrosesan disimpan pada basis data MySQL sehingga seluruh komponen sistem dapat mengakses data secara terpusat dan konsisten. Ringkasan modul utama sistem ditunjukkan pada Tabel 1.

Tabel 1. Modul Utama Sistem

Komponen	Fungsi Utama	Teknologi
Chrome Extension	DOM scraping, import data, translate image, download gambar, pengaturan bahasa	Manifest V3, JavaScript
Backend Flask	OCR, translasi, image text swapping, optimasi pipeline (Firefly Algorithm)	Flask, EasyOCR, OpenCV
Web Dashboard	Manajemen data produk, katalog, generate deskripsi produk	Flask, MySQL, Gemini API

3.3 Optimasi Firefly Algorithm pada Pipeline Scheduler

Modul Firefly Algorithm menentukan konfigurasi Pipeline Scheduler yang terdiri atas jumlah Download Worker, OCR Worker, Translate Worker, dan Image Text Swapping Worker agar setiap tahapan pipeline berjalan secara efisien sesuai beban kerja. Modul menerima masukan berupa panjang antrean (queue), jumlah active task, throughput, dan jumlah retry dari Performance Monitor, kemudian melakukan optimasi apabila terdeteksi penurunan performa seperti bottleneck, penurunan throughput, atau peningkatan retry. Setiap kandidat solusi (firefly) direpresentasikan sebagai:

$$X = [D, O, T, S]$$

dengan D, O, T, dan S masing-masing menyatakan jumlah Download Worker, OCR Worker, Translate Worker, dan Image Text Swapping Worker. Setiap kandidat solusi dievaluasi menggunakan fungsi fitness berdasarkan empat komponen, yaitu Throughput, Parameter Match, Resource Efficiency, dan Retry Rate, sehingga firefly dengan nilai fitness lebih tinggi akan menarik firefly lain untuk bergerak menuju konfigurasi tersebut hingga diperoleh konfigurasi Pipeline Scheduler yang optimal.

4 Hasil dan Pembahasan

4.1 Pengujian Modul Sistem

Pengujian sistem dilakukan melalui Unit Testing terhadap modul DOM Scraping, OCR, Translation, dan Image Text Swapping menggunakan 10 halaman produk dan 10 gambar produk pada platform 1688 dengan variasi kategori dan kompleksitas. Ringkasan hasil pengujian ditunjukkan pada Tabel 2.

Tabel 2. Ringkasan Hasil Pengujian Modul

Modul	Parameter Utama	Hasil
DOM Scraping	Success Rate ekstraksi 10 halaman produk	100%
OCR (EasyOCR)	Akurasi pengenalan 81 text region	86,41%
Translation	Success Rate / akurasi 76 permintaan	100% / 97,36%
Image Text Swapping	Rata-rata keberhasilan 4 aspek pengujian	92,5%

Modul DOM Scraping berhasil mengekstraksi seluruh atribut utama (nama produk, harga, URL gambar, informasi penjual) dari 10 halaman produk uji, meskipun ditemukan 12 dari 302 URL gambar berupa gambar non-katalog sehingga mekanisme filtering gambar masih perlu disempurnakan. Modul OCR memperoleh akurasi 86,41% dari 81 text region uji, dengan kesalahan pengenalan pada gambar berkontras rendah atau karakter yang saling berhimpitan. Modul Translation menggunakan Google Translate API berhasil memproses seluruh 76 permintaan tanpa kegagalan (Success Rate 100%) dengan rata-rata response time 0,398 detik dan akurasi makna terjemahan 97,36%. Modul Image Text Swapping memperoleh rata-rata keberhasilan 92,5%, dengan tingkat keberhasilan tertinggi pada kesesuaian posisi teks dan penyimpanan gambar hasil (masing-masing 100%), sedangkan penghapusan area teks asli (image inpainting) memperoleh 80% karena masih ditemukan sisa teks pada gambar dengan latar belakang kompleks.

4.2 Optimasi Firefly Algorithm

Pengujian Firefly Algorithm dilakukan dengan membandingkan Sequential Processing (konfigurasi statis 1,1,1,1) sebagai baseline terhadap Firefly Algorithm pada satu halaman produk berisi delapan gambar. Hasil perbandingan ditunjukkan pada Tabel 3.

Tabel 3. Perbandingan Sequential Processing dan Firefly Algorithm

Metode	Konfigurasi Worker	Total Waktu
Sequential Processing	(1,1,1,1)	202,140 detik
Firefly Algorithm	(1,1,1,1) → (1,2,1,1)	20,63 detik

$$\text{Pengurangan Waktu (\%)} = ((T_{\text{Sequential}} - T_{\text{Firefly}}) / T_{\text{Sequential}}) \times 100\%$$

Berdasarkan Persamaan (2), penerapan Firefly Algorithm mampu mengurangi waktu pemrosesan satu halaman produk sebesar 89,79% dibandingkan Sequential Processing. Selama proses berlangsung, Firefly Algorithm melakukan 11 kali evaluasi konfigurasi, dimulai dengan meningkatkan Download Worker dari 1 menjadi 5 ketika terdeteksi bottleneck pada antrean download, kemudian secara bertahap menyesuaikan jumlah

OCR Worker dan Translation Worker seiring perpindahan bottleneck antar tahapan, hingga diperoleh konfigurasi akhir (1,2,1,1). Hasil ini menunjukkan bahwa mekanisme penyesuaian worker secara dinamis berdasarkan fungsi fitness efektif mengurangi bottleneck pada pipeline pemrosesan gambar.

4.3 User Acceptance Testing (UAT)

Pengujian UAT dilakukan terhadap empat kategori responden (Owner, Purchasing, Admin, dan General User) menggunakan delapan pernyataan terkait kemudahan penggunaan dan manfaat sistem. Sistem memperoleh tingkat penerimaan pengguna sebesar 84,38% dengan kategori Sangat Baik. Aspek dengan penilaian tertinggi (95%) adalah kemudahan pengambilan data produk, hasil Image Text Swapping, dan pengurangan pekerjaan manual, sedangkan aspek fitur penerjemahan gambar memperoleh penilaian terendah (65%) karena kualitas terjemahan pada beberapa kasus masih memerlukan penyempurnaan. Ringkasan hasil UAT ditunjukkan pada Tabel 4.

Tabel 4. Ringkasan Hasil User Acceptance Testing

Aspek Pengujian	Persentase	Kategori
Kemudahan pengambilan data produk	95,00%	Sangat Baik
Hasil Image Text Swapping	95,00%	Sangat Baik
Pengurangan pekerjaan manual	95,00%	Sangat Baik
Fitur penerjemahan gambar	65,00%	Baik
Rata-rata Keseluruhan	84,38%	Sangat Baik

Secara keseluruhan, hasil UAT menunjukkan bahwa sistem telah memenuhi kebutuhan pengguna dalam mempercepat proses pengambilan data, penerjemahan gambar, dan penyusunan katalog produk multibahasa secara lebih cepat, akurat, dan efisien dibandingkan proses manual.

5 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem, dapat disimpulkan bahwa: (1) ekstensi Chrome berbasis Manifest V3 berhasil dirancang untuk melakukan web scraping data produk 1688.com secara otomatis menggunakan teknik DOM Scraping dan terintegrasi dengan Web Dashboard, dengan tingkat penerimaan pengguna (UAT) sebesar 84,38% kategori Sangat Baik; (2) fitur Image Text Swapping berbasis OCR berhasil diimplementasikan dengan akurasi OCR 86,41%, Success Rate Translation 100% dan akurasi translasi 97,36%, serta rata-rata keberhasilan Image Text Swapping sebesar 92,5%; dan (3) Firefly Algorithm berhasil diterapkan sebagai mekanisme optimasi Pipeline Scheduler yang menurunkan waktu pemrosesan satu halaman produk dari 202,140 detik menjadi 20,63 detik atau berkurang 89,79% dibandingkan Sequential Processing. Untuk pengembangan selanjutnya, disarankan memperluas dukungan platform e-commerce selain 1688.com serta menambah bahasa tujuan penerjemahan agar pemanfaatan sistem lebih luas.

Referensi

- [1] Zhao, L., & Wang, J. (2023). Automated product catalog localization system using web extraction and optical character recognition. *Journal of E-Commerce Research and Applications*, 58, 101245.
- [2] Chandra, S., & Tiwari, A. (2021). Performance evaluation of dynamic web scraping techniques on JavaScript-rendered applications. *Journal of Web Engineering*, 20(3), 745–768.
- [3] Khder, M. A. (2021). Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. *International Journal of Advances in Soft Computing and its Applications*, 13(3), 144–168.
- [4] Varghese, R., & Vijay, M. (2019). DOM-based data extraction for modern dynamic web applications. *International Journal of Computer Applications*, 178(25), 12–18.
- [5] Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
- [6] Gabor, D., Singh, R., & Mistry, A. (2018). Noise reduction and binarization for document image enhancement. *International Journal of Computer Vision Applications*, 29(4), 301–312.
- [7] Wang, L., Zhou, M., & Lin, H. (2020). Hybrid model for Chinese character recognition based on Tesseract-OCR. *Neural Computing and Applications*, 32(15), 11245–11256.
- [8] Chen, H., Zhang, Y., & Li, J. (2021). Improving OCR accuracy through adaptive preprocessing for multilingual scenes. *Pattern Recognition Letters*, 145, 12–20.
- [9] Yang, X., Chen, M., & Xu, H. (2020). SwapText: Image-Based Texts Transfer in Scenes. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, 14700–14709.
- [10] Yang, X. S. (2008). *Nature-Inspired Metaheuristic Algorithms*. Luniver Press.
- [11] Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *Stochastic Algorithms: Foundations and Applications (SAGA 2009)*, *Lecture Notes in Computer Science*, Vol. 5792 (pp. 169–178). Springer.
- [12] Alsadie, D., & Alsulami, M. (2024). Enhancing workflow efficiency with a modified Firefly Algorithm for hybrid cloud edge environments. *Scientific Reports*, 14, 24675.