

---

---

**EKSTRAKSI DATA TERSTRUKTUR PADA HALAMAN WEB  
MENGUNAKAN *MOBILE AGENT* BERBASIS  
ARSITEKTUR BDI**

*(Structured Data extraction on Web Pages Using a BDI-Based Mobile agent)*

**Ahmad Naswin<sup>1</sup>, Sulkifli<sup>2</sup>, Sitti Mawaddah Umar<sup>3</sup>**

**Universitas Megarezky Makassar, Makassar**

e-mail: [ahmadnaswin@unimerz.ac.id](mailto:ahmadnaswin@unimerz.ac.id)<sup>1</sup>, [sulkifli@unimerz.ac.id](mailto:sulkifli@unimerz.ac.id)<sup>2</sup>,  
[sittimawaddahumar@unimerz.ac.id](mailto:sittimawaddahumar@unimerz.ac.id)<sup>3</sup>

**Abstract:** *The growth of semi-structured and unstructured data on web pages has increased the need for efficient, autonomous, and structure-aware methods for structured data extraction. This study proposes a mobile agent approach based on the BDI (Belief-Desire-Intention) architecture as an alternative to conventional wrappers for structured web data extraction. The system was developed using the JADE (Java Agent Development Framework) platform and the Prometheus methodology, involving two main agents: AgenDownload, responsible for migration and retrieval of web page content, and AgenEkstraksi, which implements the Simple Tree Matching and MDR (Mining Data Regions) algorithms to identify and extract flat data records. Experiments were conducted on 40 web pages from Indonesian and English online bookstores. The results show an average recall of 73%, indicating the system capability to locate most target records, while the average precision of 15.59% indicates that further refinement is needed in post-extraction filtering. The mobile agent approach remains relevant in distributed network scenarios with bandwidth constraints, near-source processing requirements, and autonomous operation needs. This study contributes a multi-role agent system architecture that can be adapted for automated web data extraction.*

**Keywords:** *web data extraction; mobile agent; BDI architecture; Simple Tree Matching; JADE*

**Abstrak:** Pertumbuhan data semi-terstruktur dan tidak terstruktur pada halaman *web* meningkatkan kebutuhan terhadap metode ekstraksi data terstruktur yang efisien, otonom, dan dapat beradaptasi terhadap variasi struktur HTML. Penelitian ini mengajukan pendekatan berbasis *mobile agent* dengan arsitektur BDI (Belief-Desire-Intention) sebagai alternatif terhadap *wrapper* konvensional dalam proses ekstraksi data terstruktur pada halaman *web*. Sistem dibangun menggunakan platform JADE (*Java Agent Development Framework*) dan metodologi *Prometheus*, melibatkan dua agen utama: AgenDownload yang bertugas melakukan migrasi dan pengambilan konten halaman *web*, serta AgenEkstraksi yang mengimplementasikan algoritma *Simple Tree Matching* dan MDR (Mining Data Regions) untuk mengidentifikasi serta mengekstrak flat *data records*. Pengujian dilakukan terhadap 40 halaman *web* toko buku daring berbahasa Indonesia dan Inggris. Hasil menunjukkan rata-rata *recall* sebesar 73%, yang mengindikasikan kemampuan sistem dalam menemukan sebagian besar target data, sedangkan rata-rata *precision* sebesar 15,59% menunjukkan perlunya penyempurnaan mekanisme pemfilteran hasil ekstraksi. Pendekatan *mobile agent* tetap relevan pada skenario jaringan terdistribusi dengan keterbatasan bandwidth, kebutuhan pemrosesan dekat sumber data, dan operasi otonom. Kontribusi penelitian ini adalah rancangan arsitektur sistem agen multi peran yang dapat diadaptasi untuk kebutuhan ekstraksi data *web* secara otomatis.

**Kata Kunci:** *ekstraksi data web; mobile agent; arsitektur BDI; Simple Tree Matching; JADE*

## PENDAHULUAN

Perkembangan internet telah mendorong pertumbuhan data yang sangat masif pada halaman *web*. Data tersebut umumnya disajikan dalam format semi-terstruktur atau tidak terstruktur, sehingga menyulitkan sistem informasi yang membutuhkan data terstruktur untuk analisis, integrasi, temu kembali informasi, dan pengambilan keputusan. Proses pengambilan informasi terstruktur dari halaman *web* dikenal sebagai *web data extraction* atau *web information extraction*, dan telah lama menjadi topik penting dalam data mining, rekayasa perangkat lunak, serta kecerdasan buatan.

Pendekatan yang lazim digunakan adalah *wrapper*, yaitu program atau aturan ekstraksi yang dirancang untuk mengambil data dari sumber *web* tertentu. Meskipun efektif pada struktur halaman yang stabil, *wrapper* konvensional rentan terhadap perubahan DOM, variasi template, dan perubahan antarmuka situs *web*. Selain itu, praktik *web* scraping modern juga menghadapi tantangan terkait perbedaan antara konten statis dan dinamis, ketersediaan API, serta kebutuhan menjaga keberlanjutan proses ekstraksi ketika sumber *web* berubah.

Sebagai alternatif, pendekatan berbasis *mobile agent* menawarkan sejumlah keunggulan untuk sistem terdistribusi, antara lain kemampuan memindahkan komputasi menuju sumber data, mengurangi lalu lintas jaringan, dan menjalankan tugas secara otonom dalam lingkungan yang berubah. Dalam konteks jaringan sensor, IoT, *fog computing*, dan *edge computing*, paradigma agen tetap relevan karena proses komputasi dapat didekatkan ke node sumber data atau node perantara sehingga tidak seluruh data mentah harus selalu dikirimkan ke server pusat.

Penelitian ini mengimplementasikan sistem *mobile agent* berbasis arsitektur BDI (Belief-Desire-Intention) menggunakan platform JADE untuk melakukan ekstraksi flat

*data records* dari halaman *web* toko buku daring. Sistem dirancang menggunakan metodologi *Prometheus* dan menerapkan algoritma MDR (Mining Data Regions), *Simple Tree Matching*, serta *Partial Tree Alignment* sebagai komponen utama aturan ekstraksi. Rumusan masalah dalam penelitian ini adalah: (1) bagaimana merancang sistem *mobile agent* yang mampu melakukan ekstraksi data terstruktur pada halaman *web* secara otonom; dan (2) bagaimana kinerja sistem tersebut diukur menggunakan metrik *precision* dan *recall*.

Artikel ini disusun sebagai berikut. Bagian 2 memaparkan tinjauan pustaka dan landasan teori; Bagian 3 menjelaskan metodologi penelitian; Bagian 4 menyajikan implementasi sistem; Bagian 5 membahas hasil dan evaluasi; serta Bagian 6 menyampaikan simpulan dan rekomendasi penelitian lanjutan.

## TINJAUAN PUSTAKA

### **Web Data extraction dan Pendekatan Berbasis Pohon**

Ekstraksi data *web* adalah proses mengidentifikasi dan mengambil informasi terstruktur dari dokumen HTML yang pada dasarnya bersifat semi-terstruktur. Literatur umum membagi pendekatan ekstraksi *web* ke dalam beberapa kelompok, antara lain *wrapper* manual, *wrapper* induction berbasis pembelajaran mesin, pendekatan berbasis struktur DOM, pendekatan berbasis visual, dan pendekatan hibrida yang menggabungkan fitur teks, struktur, serta visual.

Pendekatan berbasis pohon HTML memanfaatkan struktur DOM untuk mendeteksi pola data yang berulang. Algoritma MDR mengidentifikasi data region melalui kemiripan struktural antar-node pada pohon tag HTML, sedangkan *Partial Tree Alignment* digunakan untuk menjajarkan record yang ditemukan agar item data dapat disusun menjadi bentuk tabel. Keunggulan pendekatan ini adalah

tidak selalu memerlukan data latih berlabel, sehingga sesuai untuk kondisi ketika skema target belum diketahui atau halaman sumber sangat beragam.

Namun, pendekatan tanpa pengawasan juga memiliki kelemahan. Algoritma yang mengutamakan kemiripan struktur cenderung menghasilkan banyak region kandidat, termasuk region non-target seperti navigasi, rekomendasi, iklan, dan metadata halaman. Hal ini dapat meningkatkan *recall* tetapi menurunkan *precision*. Untuk mengatasi persoalan tersebut, penelitian terbaru mengembangkan pendekatan berbasis deep learning, computer vision, dan self-training agar sistem ekstraksi lebih adaptif terhadap variasi layout dan kebutuhan pelabelan data dapat dikurangi.

Pada penelitian TEX, Sleiman dan Corchuelo menunjukkan bahwa pendekatan ekstraksi *web* tanpa pengawasan dapat bekerja efektif pada halaman dengan pola data berulang, namun tetap memerlukan strategi pemilihan region agar hasil ekstraksi tidak terlalu inklusif. Dengan demikian, penelitian ini menempatkan MDR dan *Simple Tree Matching* sebagai baseline berbasis struktur pohon yang kuat, tetapi tetap mengakui kebutuhan mekanisme penyaringan pasca-ekstraksi untuk meningkatkan *precision*.

### **Mobile agent dan Relevansinya dalam Sistem Terdistribusi**

*Mobile agent* merupakan entitas perangkat lunak yang mampu berpindah dari satu node jaringan ke node lainnya dengan membawa kode, status eksekusi, atau konteks komputasinya. Paradigma ini berakar pada konsep code mobility dan process migration dalam sistem terdistribusi. Perbedaannya dengan model client-server konvensional adalah agen dapat menjalankan sebagian proses di lokasi sumber data, sehingga tidak semua operasi harus dilakukan melalui permintaan terpusat dari host asal.

Pada jaringan sensor dan IoT, *mobile agent* banyak dikaji untuk

itinerary planning, data aggregation, dan information fusion karena dapat mengurangi komunikasi berulang antar-node. Sementara itu, pada fog computing, pendekatan agent-based orchestration digunakan untuk menempatkan layanan secara dinamis pada node fog dengan mempertimbangkan keterbatasan sumber daya, mobilitas, dan kebutuhan aplikasi. Temuan tersebut memperkuat relevansi *mobile agent* untuk konteks ekstraksi data ketika sumber *web* tersebar dan konektivitas tidak selalu ideal.

### **Arsitektur BDI dan Metodologi**

#### ***Prometheus***

Arsitektur BDI (Belief-Desire-Intention) adalah model agen rasional yang membedakan antara belief sebagai pengetahuan agen tentang lingkungan, desire sebagai tujuan yang ingin dicapai, dan intention sebagai komitmen tindakan atau rencana yang sedang dieksekusi. Model ini sesuai untuk agen yang harus mengambil keputusan berdasarkan kondisi lingkungan yang berubah, karena agen dapat memperbarui belief, mengevaluasi goal, dan memilih plan yang paling relevan.

Metodologi *Prometheus* adalah pendekatan *agent-oriented software engineering* yang menyediakan tahapan spesifikasi sistem, perancangan arsitektural, dan perancangan rinci. *Prometheus Design Tool* mendukung pemodelan goal, role, percept, action, interaction protocol, capability, dan plan sehingga rancangan agen dapat diturunkan secara sistematis ke implementasi. Dalam penelitian ini, *Prometheus* digunakan untuk memetakan kebutuhan sistem ke dua agen utama, yaitu AgenDownload dan AgenEkstraksi.

### **Platform JADE dalam Pengembangan Sistem Multi-Agen**

JADE (*Java Agent Development Framework*) adalah platform pengembangan sistem multi-agen berbasis Java yang mendukung standar FIPA, komunikasi berbasis ACLMessage,

manajemen *container*, agent management system, dan mekanisme mobilitas agen. Meskipun JADE bukan platform baru, ekosistem dan dokumentasinya tetap menjadi rujukan penting untuk implementasi sistem multi-agen berbasis Java. Pengembangan lanjutan seperti JADEL juga menunjukkan bahwa JADE masih digunakan sebagai fondasi untuk *model-driven development* dalam konteks *agent-oriented programming*

## METODE

### Desain Sistem

Penelitian ini menggunakan pendekatan rekayasa sistem eksperimental. Sistem yang dibangun melibatkan dua agen utama yang berinteraksi dalam platform JADE. AgenDownload berperan sebagai *mobile agent* yang bermigrasi dari main *container* ke *container* tujuan untuk mengambil konten halaman *web*, kemudian kembali ke host asal dan meneruskan lokasi konten tersebut kepada AgenEkstraksi. AgenEkstraksi merupakan agen statis yang menerima konten halaman *web* dan menjalankan proses ekstraksi data terstruktur.

Perancangan sistem mengikuti metodologi *Prometheus* melalui tiga tahap. Tahap pertama adalah spesifikasi sistem, yang mencakup identifikasi aktor, percept, action, scenario, goal, dan role. Tahap kedua adalah perancangan arsitektural, yang mencakup penentuan jenis agen, data coupling, dan protokol interaksi. Tahap ketiga adalah perancangan rinci, yang mencakup *capability overview* dan plan internal setiap agen berdasarkan struktur BDI.

### Proses Ekstraksi Data

Proses ekstraksi pada AgenEkstraksi mengikuti empat tahap. Pertama, konten HTML yang diterima dari AgenDownload diurai menggunakan parser DOM sehingga setiap tag HTML direpresentasikan sebagai node dalam

pohon tag, sedangkan teks diperlakukan sebagai properti inner text dari node yang bersangkutan. Kedua, pohon tag ditelusuri menggunakan algoritma MDR untuk menemukan generalized nodes yang berdampingan dan memiliki kemiripan struktural di atas ambang batas tertentu. Ketiga, data region yang ditemukan diproses untuk mengidentifikasi *data records* individual berdasarkan kemiripan string tag. Keempat, *data records* dijabarkan menggunakan *Partial Tree Alignment* agar item data dapat disusun menjadi bentuk tabel terstruktur.

Nilai ambang batas kemiripan yang digunakan dalam implementasi awal adalah 0,80, sedangkan nilai K atau jumlah node maksimum dalam generalized node ditetapkan sebesar 9. Target ekstraksi dibatasi pada flat contiguous *data records* karena penelitian ini belum menangani nested *data records* atau struktur data yang bersarang.

### Data Pengujian dan Metrik Evaluasi

Data pengujian terdiri atas 40 halaman *web* dari toko buku daring berbahasa Indonesia dan Inggris. Seluruh halaman yang digunakan berupa list pages kategori buku komputer. Pemilihan halaman list bertujuan memastikan keberadaan *data records* yang berulang sehingga sesuai untuk evaluasi algoritma berbasis pencocokan struktur pohon.

Kinerja sistem dievaluasi menggunakan *precision* dan *recall*. *Precision* mengukur proporsi *data records* yang diekstrak sistem dan benar-benar merupakan target, sedangkan *recall* mengukur proporsi target *data records* yang berhasil ditemukan oleh sistem. Kedua metrik ini umum digunakan pada evaluasi sistem ekstraksi informasi dan temu kembali informasi.

## IMPLEMENTASI

### Lingkungan Pengembangan

Sistem diimplementasikan menggunakan bahasa pemrograman Java dengan NetBeans IDE 7.1.1 sebagai

editor. Platform agen yang digunakan adalah JADE 4.3.0 untuk menyediakan layanan *agent container*, komunikasi antar-agen, dan migrasi agen. Perancangan agen dilakukan menggunakan *Prometheus Design Tool*. Lingkungan runtime menggunakan JDK 1.7.

### Implementasi *Mobile agent* (AgenDownload)

AgenDownload diimplementasikan sebagai subkelas dari *jade.core.Agent* dengan *behaviour* yang menangani perintah migrasi dari controller. Mekanisme migrasi memanfaatkan metode *doMove()* yang disediakan JADE untuk perpindahan agen antar-*container*. Setelah mencapai *container* tujuan, agen mengambil konten halaman *web* dan menyimpan hasilnya secara lokal sebelum kembali ke *main container*. Komunikasi antar-agen menggunakan *ACLMessage* bertipe *INFORM*.

### Implementasi AgenEkstraksi

AgenEkstraksi diaktifkan oleh pesan dari AgenDownload yang berisi *path* konten halaman *web*. Implementasi *Simple Tree Matching* menggunakan *dynamic programming* untuk menghitung tingkat kemiripan dua struktur pohon. Setelah data *region* ditemukan oleh MDR,

sistem melakukan identifikasi *data records* dan penjajaran *record* menggunakan prinsip *Partial Tree Alignment*. Hasil ekstraksi disimpan dalam berkas HTML temporer untuk keperluan evaluasi manual *precision* dan *recall*.

## HASIL DAN PEMBAHASAN

### Pengujian Mobilitas Agen

Pengujian menunjukkan bahwa AgenDownload dapat berpindah dari *main container* ke *container* tujuan, mengambil konten halaman *web*, dan kembali ke *container* asal secara otonom. Tiga karakteristik yang diverifikasi dalam pengujian ini adalah delegasi, mobilitas, dan komunikasi. Delegasi tampak ketika agen menjalankan tugas berdasarkan perintah sistem; mobilitas tampak pada perpindahan agen antar-*container*; dan komunikasi tampak pada pertukaran pesan antara AgenDownload dan AgenEkstraksi melalui *ACLMessage*.

### Hasil Evaluasi *Precision* dan *Recall*

Ringkasan hasil evaluasi terhadap 40 halaman *web* pengujian disajikan pada Tabel 1. Nilai rata-rata *recall* mencapai 73%, sedangkan nilai rata-rata *precision* sebesar 15,59%.

Tabel 1 Ringkasan evaluasi *precision* dan *recall*

Kelompok halaman uji	Jumlah halaman	<i>Precision</i> (%)	<i>Recall</i> (%)
Toko buku berbahasa Inggris	20	12,40	74,50
Toko buku berbahasa Indonesia	20	18,70	71,50
Rata-rata keseluruhan	40	15,59	73,00

Nilai *recall* yang relatif tinggi menunjukkan bahwa kombinasi MDR dan *Simple Tree Matching* mampu menemukan sebagian besar target *data records* pada halaman list yang memiliki pola berulang. Hal ini sejalan dengan karakteristik pendekatan berbasis struktur pohon yang efektif untuk mengenali

kesamaan pola DOM tanpa memerlukan data latih berlabel.

Sebaliknya, *precision* yang rendah menunjukkan bahwa sistem masih mengekstrak banyak data non-target. Fenomena ini wajar pada pendekatan *unsupervised* yang berorientasi pada deteksi *region* berulang, karena *region*

seperti menu, daftar rekomendasi, navigasi, dan blok promosi juga dapat memiliki kemiripan struktural tinggi. Oleh karena itu, mekanisme pemfilteran pasca-ekstraksi diperlukan agar sistem dapat membedakan data target dari data tambahan yang tidak relevan.

#### Contoh Pola Hasil Ekstraksi

Sebagai ilustrasi karakteristik

keluaran sistem, Tabel 2 menampilkan contoh ringkasan hasil pada beberapa toko buku yang digunakan dalam pengujian. Data pada tabel ini digunakan sebagai cuplikan evaluatif, sedangkan nilai keseluruhan kinerja sistem dirangkum pada Tabel 1.

**Tabel 2 Contoh ringkasan hasil ekstraksi pada beberapa toko buku**

Toko buku	Aktual	Benar	Salah	Gagal	Lain	Semua
Amazon.com	12,0	7,0	4,5	0,5	167	179
Barnesandnoble.com	30,0	30,0	0,0	0,0	120	150
Bookdepository.com	20,0	20,0	0,0	0,0	34	54
Inibuku.com	30,0	30,0	0,0	0,0	32	62
Garisbuku.com	24,0	24,0	0,0	0,0	117	141

Perbedaan antara jumlah data benar dan data lain menunjukkan bahwa sistem cenderung mengidentifikasi banyak region yang secara struktural mirip tetapi tidak semuanya merupakan data buku yang menjadi target. Pada situs dengan pola daftar produk yang stabil, jumlah data benar lebih tinggi dan kegagalan relatif kecil. Pada situs dengan banyak elemen tambahan, hasil salah dan data lain meningkat karena struktur non-target ikut memenuhi ambang batas kemiripan.

#### Relevansi Pendekatan *Mobile agent*

Dalam praktik industri modern, ekstraksi data *web* sering dilakukan menggunakan API, *crawler* terpusat, atau headless browser. Meskipun demikian, pendekatan *mobile agent* masih memiliki ruang relevansi pada skenario khusus, terutama ketika sumber data tersebar, bandwidth terbatas, atau pemrosesan perlu dilakukan dekat dengan lokasi data. Kajian pada wireless sensor networks dan fog computing menunjukkan bahwa pendekatan agen dapat mengurangi komunikasi berulang dan mendukung pengambilan keputusan terdistribusi.

Keterbatasan utama penelitian ini adalah cakupan ekstraksi yang masih terbatas pada flat *data records*, belum

menangani halaman *web* dinamis yang dirender JavaScript, dan belum menerapkan mekanisme autentikasi maupun anti-duplication filtering. Selain itu, *precision* yang rendah menunjukkan bahwa sistem membutuhkan modul seleksi region berbasis fitur semantik, pembelajaran mesin ringan, atau self-training untuk menekan jumlah data non-target.

#### SIMPULAN

Penelitian ini berhasil mendemonstrasikan implementasi sistem *mobile agent* berbasis arsitektur BDI sebagai alternatif *wrapper* konvensional untuk ekstraksi data terstruktur pada halaman *web*. Sistem yang dibangun menggunakan JADE dan metodologi *Prometheus* dapat menjalankan proses pengambilan konten, komunikasi antar-agen, serta ekstraksi data secara otonom melalui AgenDownload dan AgenEkstraksi.

Algoritma MDR dan *Simple Tree Matching* mampu menemukan target *data records* dengan rata-rata *recall* sebesar 73% pada 40 halaman *web* pengujian. Hasil tersebut menunjukkan bahwa

pendekatan berbasis pencocokan struktur pohon cukup efektif untuk mendeteksi pola data berulang pada halaman list. Namun, rata-rata *precision* sebesar 15,59% menunjukkan bahwa sistem masih terlalu inklusif dalam mengekstrak region non-target, sehingga diperlukan mekanisme pemfilteran tambahan.

Untuk penelitian lanjutan, disarankan pengembangan modul pemfilteran pasca-ekstraksi berbasis fitur semantik, perluasan kemampuan sistem untuk menangani nested *data records* dan halaman JavaScript-rendered, serta evaluasi performa pada skenario edge/fog computing agar relevansi pendekatan *mobile agent* dapat diuji pada lingkungan terdistribusi yang lebih realistis.

#### DAFTAR PUSTAKA

- B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, 2nd ed. Berlin, Heidelberg: Springer, 2011. doi: 10.1007/978-3-642-19460-3.
- E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, “Web data extraction, applications and techniques: A survey,” *Knowledge-Based Systems*, vol. 70, pp. 301–323, 2014. doi: 10.1016/j.knosys.2014.07.007.
- C.-H. Chang, M. Kayed, M. R. Girgis, and K. Shaalan, “A survey of *web information extraction* systems,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1411–1428, 2006. doi: 10.1109/TKDE.2006.152.
- H. A. Sleiman and R. Corchuelo, “A survey on region extractors from *web documents*,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 9, pp. 1960–1981, 2013. doi: 10.1109/TKDE.2012.135.
- D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jato, and F. Fdez-Riverola, “Web scraping technologies in an API world,” *Briefings in Bioinformatics*, vol. 15, no. 5, pp. 788–797, 2014. doi: 10.1093/bib/bbt026.
- D. S. Milojevic, F. Douglis, Y. Paindaveine, R. Wheeler, and S. Zhou, “Process migration,” *ACM Computing Surveys*, vol. 32, no. 3, pp. 241–299, 2000. doi: 10.1145/367701.367728.
- M. El Fissaoui, A. Beni-hssane, S. Ouhmad, and K. El Makkaoui, “A survey on *mobile agent* itinerary planning for information fusion in wireless sensor networks,” *Archives of Computational Methods in Engineering*, vol. 28, pp. 1323–1334, 2021. doi: 10.1007/s11831-020-09417-1.
- A. Liutkevičius, N. Morkevičius, A. Venčkauskas, and J. Toldinas, “Distributed agent-based orchestrator model for fog computing,” *Sensors*, vol. 22, no. 15, article 5894, 2022. doi: 10.3390/s22155894.
- Y. Zhai and B. Liu, “Web data extraction based on *Partial Tree Alignment*,” in *Proceedings of the 14th International Conference on World Wide Web (WWW 2005)*, Chiba, Japan, 2005, pp. 76–85. doi: 10.1145/1060745.1060761.
- Y. Zhai and B. Liu, “Structured *data extraction* from the *Web* based on *Partial Tree Alignment*,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1614–1628, 2006. doi: 10.1109/TKDE.2006.197.
- S. K. Patnaik, C. N. Babu, and M. Bhawe, “Intelligent and adaptive *web data extraction* system using convolutional and long short-term memory deep learning networks,” *Big Data Mining and Analytics*, vol. 4, no. 4, pp. 279–297, 2021. doi: 10.26599/BDMA.2021.9020012.
- R. Sarkhel, B. Huang, C. Lockard, and P. Shiralkar, “Self-training for label-efficient information extraction from semi-structured *web-pages*,” *Proceedings of the VLDB Endowment*, vol. 16, no. 11, pp.

- 3098–3110, 2023. doi: 10.14778/3611479.3611511.
- H. A. Sleiman and R. Corchuelo, “TEX: An efficient and effective *unsupervised web* information extractor,” *Knowledge-Based Systems*, vol. 39, pp. 109–123, 2013. doi: 10.1016/j.knosys.2012.10.009.
- A. Fuggetta, G. P. Picco, and G. Vigna, “Understanding code mobility,” *IEEE Transactions on Software Engineering*, vol. 24, no. 5, pp. 342–361, 1998. doi: 10.1109/32.685258.
- L. de Silva, F. Meneguzzi, and B. Logan, “BDI agent architectures: A survey,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2020, pp. 4914–4921. doi: 10.24963/ijcai.2020/684.
- M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Chichester: Wiley, 2009.
- L. Padgham and M. Winikoff, *Developing Intelligent Agent Systems: A Practical Guide*. Chichester: Wiley, 2004.
- L. Padgham, J. Thangarajah, and M. Winikoff, “The *Prometheus* Design Tool: A conference management system case study,” in *Agent-oriented software engineering VIII*, *Lecture Notes in Computer Science*, vol. 4951. Berlin, Heidelberg: Springer, 2008, pp. 197–211.
- F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Chichester: Wiley, 2007. doi: 10.1002/9780470058411.
- F. Bergenti, E. Iotti, S. Monica, and A. Poggi, “Agent-oriented *model-driven development* for JADE with the JADEL programming language,” *Computer Languages, Systems & Structures*, vol. 50, pp. 142–158, 2017. doi: 10.1016/j.cl.2017.06.001.