Web Content Extractor Menggunakan Neural Network untuk Konten Artikel di Internet

Syabith Umar Ahdan, *Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terapan Surabaya*,
Joan Santoso, *Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terapan Surabaya*,
Hendrawan Armanto, *Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terapan Surabaya*.

***Abstrak*— Berkembangnya teknologi Javascript khususnya AJAX dan CSS membuat halaman web yang dulunya statis menjadi lebih dinamis dengan tampilan yang lebih menarik dan dipenuhi iklan dan rekomendasi artikel lain. Oleh karena itu, sulit untuk mengotomatisasi proses pengambilan konten artikel pada konteks ini. Penelitian ini dibuat untuk menyelesaikan masalah otomatisasi pengambilan konten artikel di Internet. Aplikasi web yang akan dibuat terbagi menjadi empat modul, yaitu web crawler, web extractor, content classifier dan web visualizer. Penelitian ini memiliki dua desain arsitektur. Arsitektur yang pertama adalah arsitektur saat training. Arsitektur yang kedua adalah arsitektur program jadi. Proses training menggunakan 200 URL halaman web dari lima website berbeda. Metode pengujian yang akan digunakan adalah 4-Fold Cross Validation, sehingga 75% dari blok teks akan menjadi data latihan dan 25% dari blok teks akan menjadi data pengujian. Program jadi berupa Web Visualizer yang mengolah JSON file berisi hubungan antara halaman web yang didapatkan dari web crawler sehingga dapat dipresentasikan dalam sebuah grafik. Kesimpulan dari penelitian ini adalah bahwa kombinasi Scrapy, Splash, Neural Network Classifier dan D3 bekerja sangat baik untuk automasi ekstraksi konten artikel website di Internet sekaligus memvisualisasi hubungan antar halaman web. Deep Feed Forward Neural Network (DFFNN) dapat melakukan klasifikasi multi-class konten judul, penulis, dan isi artikel dengan baik selama template halaman web sudah pernah dilatih sebelumnya. DFFNN juga dapat melakukan klasifikasi binari untuk halaman web secara umum dengan F1-score 62.87%, dua kali lebih baik dari SVM yang hanya 31.28%.**

***Kata Kunci*—Content Extractor, DBSCAN, Neural Network, Web Crawler, Web Visualization.**

I. PENDAHULUAN

Bagi peneliti literatur dan bahasa web, sangatlah penting untuk mendapatkan data penelitian sebanyak-banyaknya untuk kepentingan penelitiannya. Namun akan sangat memakan waktu jika data penelitian harus didapatkan dengan membuka halaman web secara manual satu per satu. Akan sangat membantu apabila ada suatu software yang dapat membantu peneliti literatur dan bahasa mengumpulkan data penelitian. Sebuah web parser biasanya digunakan untuk tujuan ini. Sangat mudah bagi sebuah parser untuk mengekstrak konten pada suatu halaman web, asalkan parser mampu mengenali struktur dari halaman web tersebut. Namun, dengan kecepatan pertambahan jumlah website yang ada, sangatlah sulit, bahkan mustahil, untuk menganalisis setiap website secara manual satu per satu. Struktur sebuah website juga dapat berubah sewaktu-waktu, sehingga akan mustahil juga untuk mengecek dan memperbaharui struktur website yang sudah ada.

Tujuan utama dari Penelitian ini adalah untuk mengotomatisasi proses pengambilan konten artikel di internet. Selain itu, dapat menyaring konten yang ada pada halaman web dengan hanya mengambil judul, penulis, dan konten pada halaman artikel. Tujuan lainnya adalah untuk memvisualisasi berbagai halaman website dengan mempresentasikan setiap halaman web sebagai sebuah node yang memiliki banyak koneksi ke halaman web yang lain.

II. TEORI PENUNJANG

Terdapat lima teori utama yang akan dibahas pada bagian ini. Kelima teori tersebut adalah Web Framework, Web Crawler, Headless Web Browser, Neural Network, dan Web Data Visualization.

A. *Web Framework*

Web Framework adalah koleksi *packages* dan *modules* yang memungkinkan developer untuk membuat aplikasi atau layanan web tanpa harus mengurusi detail *low-level* seperti manajemen protokol, soket, dan *process/thread*. Web Framework yang akan digunakan adalah Flask.

B. *Web Crawler*

Web Crawler, atau *spider* (laba-laba), adalah tipe bot yang biasanya dijalankan oleh *search engine* (mesin pencarian) seperti Google dan Bing. Tujuan web crawler adalah untuk mengindeks konten dari website yang ada di internet, mempelajari informasi apa yang ada pada website tersebut, sehingga website tersebut dapat muncul di hasil pencarian search engine saat informasi yang sama dibutuhkan pengguna search engine. Web Crawler Framework web crawler yang digunakan adalah Scrapy.

C. *Headless Web Browser*

Headless Web Browser adalah sebuah web browser yang tidak memiliki GUI (*Graphical User Interface)*. Headless

Syabith Umar Ahdan, Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terapan Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: syabith1@mhs.stts.edu)
Joan Santoso, Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terapan Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: joan@stts.edu)
Hendrawan Armanto, Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terapan Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: hendrawan@stts.edu)

web browser menyediakan kontrol otomatis pada halaman web di lingkungan yang mirip dengan web browser pada umumnya. Namun dijalankan menggunakan CLI (Command Line Interface) atau menggunakan komunikasi jaringan. Headless web browser sangat bermanfaat untuk pengujian halaman web karena dapat menerjemahkan dan mengerti HTML. Splash akan digunakan sebagai Headless Web Browser untuk Penelitian ini.

D. *Neural Network*

Neural Network adalah salah satu tipe *Machine Learning* (Pembelajaran Mesin) di mana modelnya dibuat berdasarkan cara kerja otak manusia. Melalui berbagai algoritma, model neural network dapat mengelompokkan dan mengklasifikasi sebuah dataset berdasarkan fitur-fitur yang diberikan sebagai input. Pola input yang dikenali neural network adalah pola angka yang dimuat dalam vector, dimana panjang vector sesuai dengan jumlah fitur. Perangkat lunak dan library yang akan digunakan untuk membangun neural network antara lain: Scikit Learn [1], Numpy , Keras [2], dan TensorFlow [3].

E. *Web Data Visualization*

Web Data Visualization atau visualisasi data web adalah teknik merepresentasikan grafik dari data dan informasi melalui media web. Representasi grafik dapat mempermudah pembaca untuk mengerti arti maupun implikasi dari data seperti tren, pola, *outliers*, maupun hubungan antar data. Komunikasi ini dicapai melalui pemetaan yang sistematis antara tanda grafik dengan nilai data yang dipresentasikan dalam visualisasi yang diciptakan. Pemetaan ini menetapkan bagaimana nilai data akan dipresentasikan secara visual. Pemetaan ini juga mendeterminasikan apa saja properti tanda grafik yang mengandung nilai data, seperti ukuran dan warna. Penelitian ini akan menggunakan D3 [4] sebagai tools pembuatan visualisasi data web.

III. Analisa Penelitian Sejenis

Terdapat dua penelitian yang akan digunakan sebagai rujukan. Penelitian pertama adalah *web content extractor through machine learning* yang ditulis oleh Ziyan Zhou et al [5]. Penelitian yang kedua adalah WebSPHINXdari Computer Science of Carnegie Mellon University.

A. *Web Content Extractor Through Machine Learning*

Paper ini dipilih karena kebanyakan paper web ekstraktor berupaya untuk mengekstrak data yang terstruktur dan jumlahnya ada beberapa dalam satu halaman. Hal ini dicapai dengan menggunakan model classifier SVM atau Support Vector Machine [6]. Sedangkan tujuan dari paper tersebut adalah untuk mengekstrak konten web yang hanya dimuat sekali oleh sebuah halaman web, seperti konten artikel, berita, maupun cerita. Tujuan paper tersebut selaras dengan tujuan dari Penelitian ini.

Paper ini terdiri dari enam bagian. Bagian-bagian tersebut antara lain pengumpulan data, ekstraksi blok teks dan CSS, klasterisasi, pelabelan klaster, SVM dan cross validation, dan pemilihan fitur.

B. *WebSPHINX*

WebSPHINX adalah sebuah library Java Class dan lingkungan development yang interaktif untuk web crawler. WebSPHINX ditujukan untuk pengguna web lanjutan dan programmer Java yang ingin melakukan crawling pada sebagian kecil dari sebuah website secara otomatis. WebSPHINX dirilis secara open source, di bawah Apache-style license.

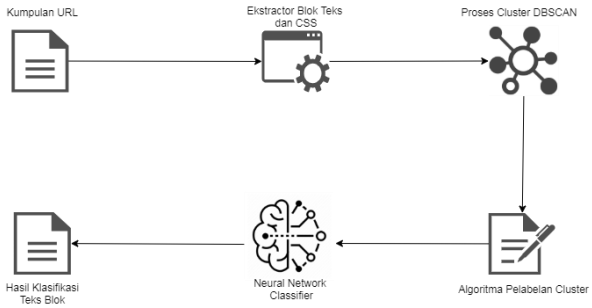
Bagian WebSHINX yang akan dirujuk adalah bagian Crawler Workbench. Crawler Workbench adalah sebuah graphical user interface yang memungkinkan penggunanya untuk mengkonfigurasi dan mengkontrol web crawler yang terkostumisasi. Dengan Crawler Workbench, pengguna dapat memvisualisasi kumpulan halaman web dalam sebuah grafik, menyimpan halaman web ke tempat penyimpanan lokal di komputer untuk browsing secara offline, menggabungkan beberapa halaman menjadi satu dokumen untuk dibaca atau dicetak, mengekstrak teks tertentu dengan mencocokkan pola dari kumpulan halaman web, dan membuat crawler yang dapat dikostumisasi menggunakan Java atau Javascript agar dapat memproses halaman web sesuai keinginan.

IV. Arsitektur Sistem

Penelitian ini memiliki dua desain arsitektur. Arsitektur yang pertama adalah arsitektur saat training. Arsitektur yang kedua adalah arsitektur program jadi.

A. *Arsitektur Saat Training*

Arsitektur saat training adalah arsitektur sistem yang digunakaan saat melatih model neural network. Model neural network sering digunakan untuk kebutuhan klasifikasi teks [7], [8]. Desain arsitektur saat training model neural network dari Penelitian ini dapat dilihat pada Gambar 1.



Gambar. 1. Desain Arsitektur Saat Training

Proses training dimulai dengan pengumpulan 200 URL halaman web dari lima website yaitu detik.com, kompas.com, iwanbanaran.com, teknojurnal.com, dan tipspintar.com. Kira-kira seperempat diantara 200 URL tersebut adalah halaman web yang tidak berartikel.

Ekstraktor blok teks dan CSS akan membuka tiap halaman web dari daftar URL seperti halnya sebuah web browser. Ekstraktor blok teks dan CSS kemudian mengekstrak setiap elemen teks blok beserta tag path dan properti CSS nya dari setiap halaman web.

Proses Cluster DBSCAN akan memproses kumpulan file json yang diproduksi oleh Ekstraktor blok teks dan CSS untuk mengumpulkan teks blok yang sejenis di satu klaster yang sama menggunakan algoritma klasterisasi DBSCAN [9].

Hasil dari klasterisasi ini dapat berupa belasan hingga ratusan klaster tergantung pada karakteristik layout website.

Algoritma pelabelan cluster kemudian menggunakan hasil dari proses cluster DBSCAN untuk melabeli klaster terbaik berdasarkan nilai tertinggi. Klaster ini dihitung menggunakan algoritma LCS yang memanfaatkan tag meta dari sebuah halaman web dan algoritma buatan Ziyan Zhou untuk mengatasi pemberian nilai yang tidak akurat pada blok teks berisi komentar. Blok teks yang ada dalam klaster dengan nilai terbaik tersebut kemudian dilabeli sebagai konten, sedangkan blok teks di dalam klaster lainnya dilabeli sebagai non konten. Klaster yang berisi judul dan penulis kemudian dilabeli secara manual.

Blok teks yang sudah dilabeli kemudian dijadikan sebagai dataset untuk melatih dan menguji model neural network. Kombinasi dari properti CSS dan tag path dari setiap blok teks dijadikan sebagai fitur atau input untuk model neural network yang akan dilatih. Metode pengujian yang akan digunakan adalah 4-Fold Cross Validation, sehingga 75% dari blok teks akan menjadi data latihan dan 25% dari blok teks akan menjadi data pengujian.

Selanjutnya akan dijelaskan tentang arsitektur dari model neural network yang akan dilatih. Matriks input untuk model neural network dapat dinotasikan sebagai $X \in \mathbb{R}^{188_x \times 35177}$. X adalah variabel matriks input berisi bilangan riil dengan ukuran 188 x 35177. Sedangkan sebuah sampel dataset ke-i dapat dinotasikan sebagai $x^{(i)} \in \mathbb{R}^{188_x}$, dimana x adalah vektor input berisi bilangan riil dengan ukuran 188.

Metode pengujian yang akan digunakan adalah 4-Fold Cross Validation, sehingga 75% dari blok teks akan menjadi data latihan dan 25% dari blok teks akan menjadi data pengujian. Pembagian data latihan dan data pengujian dilakukan per website, sebelum akhirnya dataset per website disatukan.

Untuk label kelas, dilakukan binarisasi, dimana cara kerjanya hampir sama dengan One Hot Encoding. Label yang semula berupa integer dengan kemungkinan nilai angka 0-4 diubah menjadi kumpulan vektor binari dengan kemungkinan angka 0 sampai 1. Panjang dari vektor binari ini adalah sebesar jumlah kelas klasifikasi. Penelitian ini memiliki empat kelas klasifikasi, sehingga panjang vektor binari adalah empat.

Matriks label dari model neural network pada Penelitian ini dapat dinotasikan sebagai $Y \in \mathbb{R}^{4_y \times 35177}$. Y adalah variabel matriks label berisi bilangan riil dengan ukuran 4 x 35177. Sedangkan sebuah label untuk dataset ke-i dapat dinotasikan sebagai $y^{(i)} \in \mathbb{R}^{4_y}$, dimana y adalah vektor label berisi bilangan riil dengan ukuran empat. Setiap indices dari vektor mencerminkan label dari dataset tersebut, dimana index pertama adalah non konten, index kedua adalah judul, index ketiga adalah penulis, dan index keempat adalah konten.

Matriks weight yang pertama untuk model neural network dapat dinotasikan sebagai $W^{[1]} \in \mathbb{R}^{96 \times 188}$. $W^{[1]}$ adalah variabel matriks weight berisi bilangan riil dengan ukuran 96 x 188. Matriks weight yang kedua untuk model neural network dapat dinotasikan sebagai $W^{[2]} \in \mathbb{R}^{20 \times 96}$. $W^{[2]}$ adalah variabel matriks weight berisi bilangan riil dengan ukuran 20 x 96. Matriks weight yang ketiga untuk model neural network dapat dinotasikan sebagai $W^{[3]} \in \mathbb{R}^{4 \times 20}$. $W^{[3]}$ adalah variabel matriks weight berisi bilangan riil dengan ukuran 4 x 20. Rumus fungsi aktivasi pada hidden layer pertama dapat dilihat pada formula 1.

$$a^{[1]} = \mathrm{ReLu}^{[1]}(W^{[1]}x^{(i)} + b_1) \quad (1)$$

Fungsi aktivasi yang digunakan pada hidden layer pertama adalah fungsi ReLu. Input dari fungsi ReLu adalah jumlah dari hasil perkalian matriks $W^{[1]}$ dengan vektor sampel x ke-i ditambah dengan nilai bias b_1. Rumus fungsi aktivasi pada hidden layer kedua dapat dilihat pada formula 2.

$$a^{[2]} = \mathrm{ReLu}^{[2]}(W^{[2]}h_1^{(i)} + b_2) \quad (2)$$

Fungsi aktivasi yang digunakan pada hidden layer kedua adalah fungsi ReLu. Input dari fungsi ReLu adalah jumlah dari hasil perkalian matriks $W^{[2]}$ dengan vektor sampel h_1 ke-i dari hidden layer sebelumnya ditambah dengan nilai bias b_2. Rumus fungsi aktivasi pada output layer dapat dilihat pada formula 3 di bawah ini.

$$\hat{y}^{(i)} = \mathrm{Softmax}(W^{[3]}h_2^{(i)} + b_3) \quad (3)$$

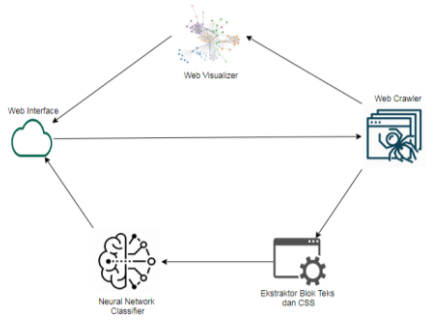
Fungsi aktivasi yang digunakan pada output layer adalah fungsi Softmax. Inputnya adalah jumlah dari hasil perkalian matriks $W^{[3]}$ dengan vektor sampel h_2 ke-i dari hidden layer sebelumnya ditambah dengan nilai bias b_3.

Optimizer yang digunakan pada model neural network adalah optimizer adam [10]. Adam merupakan singkatan dari Adaptive Momen Estimation. Adam mengkombinasikan properti terbaik dari AdaGrad dan RMSProp untuk menangani gradient yang jarang dan masalah noise. Adam sangatlah mudah dikonfigurasi dan konfigurasi defaultnya biasanya bekerja baik untuk kebanyakan masalah. Adam memiliki beberapa manfaat lain yaitu komputasinya yang efisien dan kebutuhan kapasitas memory nya yang kecil.

Loss function yang digunakan adalah categorical cross entropy. Categorical cross entropy dipakai karena jumlah kelas untuk klasifikasi adalah empat kelas, yaitu konten, judul, penulis, dan non konten. Karena terdapat ketimpangan yang tinggi antara blok teks yang berlabel non konten dengan blok teks yang berlabel konten, judul, dan penulis, maka fitur class weight saat latihan akan digunakan. Dengan fitur class weight, model neural network dapat disetting agar menyesuaikan nilai loss function setiap kelas sesuai dengan nilai class weight. Nilai dari class weight akan ditentukan sesuai dengan jumlah dataset pada setiap kelas. Dalam kata lain, kelas konten, judul, dan penulis, yang jumlah sampel datanya jauh lebih sedikit ketimbang sampel data non konten, akan memiliki nilai class weight yang jauh lebih besar ketimbang kelas non konten. Dengan ini diharapkan permasalahan dataset yang tidak seimbang dapat diatasi.

B. Arsitektur Program Jadi

Arsitektur program jadi digunakan untuk website yang akan digunakan oleh pengguna. Desain arsitektur program jadi dapat dilihat pada Gambar 2.



Gambar. 2. Desain Arsitektur Program Jadi

Sistem dimulai dari beroperasinya web interface yang bertatap muka langsung dengan pengguna website. Dari sini, pengguna website akan memberi input berupa base URL atau URL pertama yang akan dijadikan target crawling, depth atau jumlah kedalaman crawling dari base URL, dan total maksimal jumlah halaman web. Pengguna juga dapat memilih apakah crawler hanya akan menargetkan domain yang sama dari base URL, atau akan menargetkan semua out link yang ditemui.

Setelah input dan konfigurasi disubmit, maka web crawler akan bekerja sesuai nilai dari input dan konfigurasi. Web Crawler kemudian akan mengeluarkan output berupa daftar URL yang telah dicrawl dengan jumlah sesuai dengan nilai input dan juga file JSON yang menyimpan data tentang hubungan antar link URL.

Ekstraktor blok teks dan CSS pada program jadi adalah modul yang sama persis seperti yang digunakan pada saat training model neural network. Ekstraktor blok teks dan CSS akan mengekstrak blok teks dari daftar URL.

Neural Network Classifier kemudian memproses lebih lanjut kumpulan file JSON dari ekstraktor agar bisa dijadikan sebagai input untuk classifier. Hal ini dilakukan sebelum akhirnya diklasifikasi apakah sebuah data set termasuk dalam konten, penulis, judul, atau non konten. Data hasil dari klasifikasi ini kemudian diberikan kepada web interface untuk selanjutnya dipresentasikan.

Web Visualizer mengolah JSON file berisi hubungan antara halaman web yang didapatkan dari web crawler sehingga dapat dipresentasikan dalam sebuah grafik. Grafik tersebut berisi kumpulan node yang melambangkan sebuah halaman web dengan garis penghubung antar node yang melambangkan adanya link penghubung diantara keduanya. Grafik ini kemudian diberikan kepada web interface untuk selanjutnya ditunjukkan pada pengguna.

V. UJI COBA

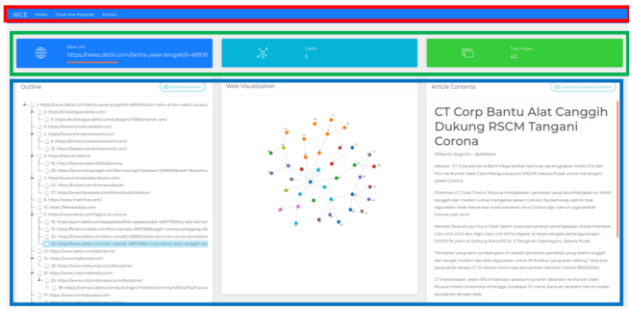
Pada bab uji coba sistem ini akan dijelaskan lebih detail mengenai hasil uji coba dan penjelasan detailnya dari website yang akan dibuat oleh Penelitian ini. Bab ini akan dibagi menjadi tujuh bagian, yaitu Web Interface, Web Crawler, Ekstraktor Blok Teks dan CSS, Proses Cluster DBSCAN, Algoritma Pelabelan Cluster, Neural Network Classifier, dan Web Data Visualizer.

A. Web Interface

Pada halaman utama, pengguna dikenalkan dengan website Penelitian ini beserta deskripsinya dan dapat langsung menggunakannya dengan mengisi form yang ada pada halaman yang sama. Form ini berisi data yang akan dijadikan input untuk modul web crawler.

Pada halaman utama website, terdapat tiga komponen utama, yaitu navbar dalam kotak merah, penjelasan tentang fungsi website dalam kotak hijau, dan form input dalam kotak biru. Navbar berhasil ditampilkan pada area atas halaman website di setiap halaman web. Setiap link pada Navbar berhasil membawa membawa pengguna kepada halamannya masing-masing, begitu juga dengan link pada logo website. Penjelasan tentang fungsi website berisi deskripsi tentang website dan penjelasan tentang cara penggunaannya. Form input berisi input base URL, depth, dan total maksimal jumlah halaman web. Form input juga berisi switch konfigurasi untuk mengkonfigurasi web crawler agar hanya menargetkan link dari domain yang sama dengan starting atau base URL. Dan terakhir, form input memiliki button submit yang mengirimkan data form ke proses selanjutnya.

Pada halaman hasil utama, pengguna dapat melihat hasil dari klasifikasi dan visualisasi website. Pengguna juga dapat melihat rekap dari input yang dimasukkan sebelumnya. Tampilan halaman hasil utama website dapat dilihat pada Gambar 3 di bawah ini.

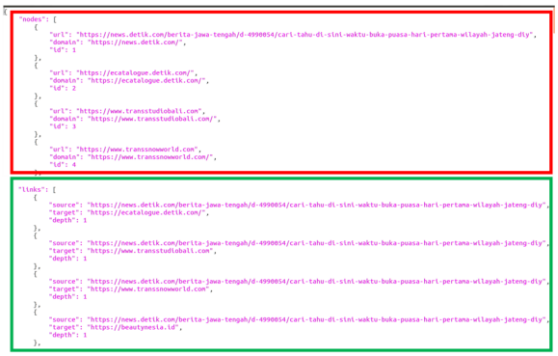


Gambar. 3. Halaman Hasil Utama Website

Pada halaman hasil utama website, terdapat tiga komponen utama, yaitu navbar dalam kotak merah di atas, rekap dari input yang dimasukkan sebelumnya dalam kotak hijau di tengah, dan hasil dari klasifikasi dan visualisasi dalam kotak biru di bawah. Navbar sudah dijelaskan pada paragraf sebelumnya. Rekap dari input yang dimasukkan sebelumnya berisi input base URL, depth, dan total maksimal jumlah halaman web. Hasil dari klasifikasi dan visualisasi berisi outline tree dan grafik hasil web crawler serta hasil klasifikasi dari neural network classifier.

B. Web Crawler

Web Crawler berhasil melakukan crawling sesuai dengan parameter dan konfigurasi yang diberikan. Web Crawler juga berhasil menghasilkan file teks daftar URL dan file JSON yang berisi daftar nodes dan daftar link antar nodes. Tampilan file JSON dapat dilihat pada Gambar 4.



Gambar. 4. File Daftar Nodes dan Link

Gambar 4 menunjukkan empat nodes pertama dalam kotak merah dan empat links pertama dalam kotak hijau dari 40 nodes dan links yang telah dihasilkan oleh web crawler. Base URL dapat dilihat pada URL node pertama pada file JSON yang dihasilkan. Setiap node yang dihasilkan memiliki tiga atribut, yaitu URL, domain, dan id. Sedangkan setiap link yang dihasilkan memiliki tiga atribut juga, yaitu source, target, dan depth. Hasil file daftar nodes yang dihasilkan membuktikan bahwa web crawler dapat menghasilkan daftar nodes sesuai dengan data dan atribut yang diinginkan. Hasil file ini juga membuktikan bahwa web crawler hanya akan memasukkan nodes yang memiliki source URL yang sudah ada pada daftar nodes saat itu. Sehingga setiap node pada daftar dapat dipastikan memiliki link dengan node lain. Dan terakhir, file daftar nodes yang dihasilkan berhasil membatasi jumlah depth dan jumlah total halaman web yang dihasilkan sesuai dengan input.

C. Ekstraktor Blok Teks dan CSS

Ekstraktor Blok teks berhasil melakukan ekstraksi blok teks dan CSS sesuai daftar URL yang diberikan. Setiap blok teks memiliki tujuh atribut, yaitu bound, computed, element, html, text, path, dan selector. Bound adalah lebar dan tinggi serta koordinat dari blok teks. Computed adalah properti CSS blok teks. Element adalah elemen tag HTML beserta atribut id dan class nya, html adalah isi raw HTML blok teks. Text adalah daftar isi teks pada blok teks. Path adalah tag path dari elemen sebelum tag body sampai pada blok teks tersebut. Selector adalah daftar CSS selector pada setiap elemen yang ada pada tag path. Blok teks di atas merupakan konten AJAX dari halaman web. Oleh karena itu, hasil uji coba ini membuktikan pula bahwa ekstraktor blok teks dan CSS dapat menangani konten AJAX.

D. Proses Cluster DBSCAN

Proses Cluster DBSCAN berhasil mengklasiterisasi kumpulan blok teks yang diberikan. Rincian hasil jumlah klaster dapat dilihat pada Tabel I di bawah ini.

TABEL I
JUMLAH KLASTER WEBSITE LATIHAN

Website	Jumlah Klaster
Detik.com	1116
Iwanbanaran.com	507
Kompas.com	859
Teknojurnal.com	141
Tipspintar.com	215
Total	200

Website dengan hasil klasterisasi yang terbaik adalah website dengan jumlah klaster yang paling sedikit. Maksud dari hasil klasterisasi terbaik disini adalah hasil klaster yang menyatukan semua blok teks yang penampilannya mirip kedalam klaster yang sama. Dalam hal ini, Teknojurnal.com dan Tipspintar mendapatkan posisi pertama dan kedua secara berurutan sebagai website dengan klaster terbaik. Namun sayangnya, hasil klaster dari kedua website tersebut masih kurang ideal. Pada Teknojurnal, blok teks yang merupakan isi artikel masih terbagi menjadi dua klaster. Pada Tipspintar.com, blok teks berisi artikel terbagi ke dalam empat klaster. Hal ini dikarenakan tidak semua blok teks yang berisi artikel memiliki properti CSS dan tag path yang 100% sama. Terdapat sedikit perbedaan yang menyebabkan blok teks berisi artikel terpisah ke beberapa klaster yang berbeda.

Masalah yang sama juga dimiliki oleh ketiga website yang lain. Detik.com dan Kompas.com memiliki hasil klasterisasi yang terburuk karena selain masalah properti CSS yang agak berbeda, kedua website tersebut memiliki template yang berbeda-beda pula. Template yang dipakai tergantung pada sub kategori artikel yang tertulis. Sehingga, blok teks yang berisi artikel terbagi menjadi beberapa klaster berdasarkan template pada halaman tersebut.

E. Algoritma Pelabelan Cluster

Telah ditentukan sebelumnya bahwa klasterisasi dari proses cluster DBSCAN kurang ideal. Oleh karena itu, diputuskan bahwa blok teks akan dilabeli secara manual untuk memastikan integrasi dan keakuratan dataset yang akan digunakan untuk melatih model neural network.

Meskipun begitu, algoritma pelabelan cluster akan tetap dicoba. Hal ini dilakukan untuk melihat apakah algoritma ini dapat digunakan pada website berbahasa Indonesia atau tidak. Dengan catatan bahwa kumpulan klaster yang akan diperiksa adalah kumpulan klaster yang baik. Dalam kata lain, algoritma pelabelan cluster ini hanya akan digunakan pada dataset website Iwanbanaran.com, Teknojurnal.com, dan Tipspintar.com yang hanya memiliki satu template.

Hasil yang didapatkan dari algoritma pelabelan cluster pada website Iwanbanaran.com, Teknojurnal.com, dan Tipspintar.com sedikit beragam. Pada Iwanbanaran.com, klaster dengan skor tertinggi adalah klaster dengan blok teks yang berisi username dan tanggal komentar. Setelah ditelaah lebih jauh, ternyata pada salah satu artikel terdapat tag meta yang berisi bulan dan tahun. Pada kolom komentar, terdapat teks bulan dan tahun pada setiap komentarnya juga. Hal ini menjelaskan kenapa algoritma LCS memberi nilai tertinggi pada klaster berisi komentar. Enam klaster terbaik jatuh pada blok teks komentar. Klaster dengan blok teks berisi konten jatuh pada posisi ketujuh. Pada Teknojurnal.com, klaster dengan skor tertinggi pertama dan kedua adalah klaster dengan blok teks yang berisi konten artikel. Pada Tipspintar.com, klaster dengan skor tertinggi pertama dan kedua adalah klaster dengan blok teks yang berisi konten artikel juga. Hasil dari pelabelan manual blok teks pada setiap website dapat dilihat pada Tabel II.

TABEL II
HASIL PELABELAN BLOK TEKS

Website	Non Konten	Judul	Penulis	Konten
Detik.com	4551	25	25	219
Iwanbanaran.com	9583	30	30	282
Kompas.com	9858	28	97	333
Teknojurnal.com	1645	29	29	383
Tipspintar.com	5724	30	58	2218
Covid19.go.id	246	3	0	32
Turnbackhoax.id	787	3	3	78
Total	32394	148	242	3545

F. Neural Network Classifier

Metode evaluasi model neural network yang akan digunakan adalah metode Precision, Recall, dan F-1 Score. Hasil uji coba yang akan ditampilkan adalah hasil yang menggunakan dataset dari website latihan dan dataset dari website baru yang belum pernah dilihat sebelumnya. Performa dari model neural network dasar dengan arsitektur dan konfigurasi yang sudah dideskripsikan sebelumnya dapat dilihat pada Tabel III di bawah ini.

TABEL III
PERFORMA MODEL DASAR

	Website Latihan			Website Baru		
Label	P	R	F-1	P	R	F-1
Non-Konten	99.66%	99.73%	99.69%	96.26%	87.12%	91.46%
Judul	97.06%	97.06%	97.06%	100%	50%	66.67%
Penulis	96.97%	91.43%	94.12%	0%	0%	0%
Konten	97.89%	97.67%	97.78%	37.75%	70%	49.04%

Model dasar neural network ideal digunakan pada website yang sudah dilatih sebelumnya. Namun model ini tidak ideal digunakan untuk website baru yang belum pernah dilatih.

Model selanjutnya adalah model dasar yang dimodifikasi hanya menggunakan satu hidden layer dengan jumlah node 96. Dalam kata lain, hidden layer kedua dengan jumlah node 20 tidak digunakan pada model ini. Performa variasi model dengan satu hidden layer dapat dilihat pada Tabel IV.

TABEL IV
PERFORMA MODEL SATU HIDDEN LAYER

	Website Latihan			Website Baru		
Label	P	R	F-1	P	R	F-1
Non-Konten	99.72%	99.69%	99.70%	90.71%	93.61%	92.14%
Judul	100%	97.06%	98.51%	0%	0%	0%
Penulis	97.06%	94.29%	95.65%	0%	0%	0%
Konten	97.45%	98.02%	97.73%	25.4%	14.55%	18.5%

Pada website yang sudah dilatih, model neural network dengan satu hidden layer memiliki performa yang hampir sama, bahkan sedikit lebih baik, dibandingkan dengan model dasar yang memiliki dua hidden layer. Namun, performa model dengan satu hidden layer jauh lebih buruk pada website yang belum pernah dilatih sebelumnya. Hal ini terjadi karena tingkat abstraksi dan generalisasi model dengan satu hidden layer jauh lebih rendah dibandingkan dengan model dengan dua atau lebih hidden layer.

Model selanjutnya adalah model dasar yang dimodifikasi dengan dua layer Dropout dengan dropout rate sebesar 0.5. Layer dropout pertama diletakkan diantara hidden layer pertama dan kedua. Layer dropout kedua diletakkan diantara hidden layer kedua dan output layer. Performa variasi model dengan dropout layer dapat dilihat pada Tabel V.

TABEL V
PERFORMA MODEL DROPOUT LAYER

	Website Latihan			Website Baru		
Label	P	R	F-1	P	R	F-1
Non-Konten	99.66%	99.61%	99.64%	95.36%	87.71%	91.38%
Judul	100%	88.24%	93.75%	0%	0%	0%
Penulis	88.57%	88.57%	88.57%	0%	0%	0%
Konten	96.99%	97.78%	97.39%	51.51%	30.91%	38.64%

Pada website yang sudah dilatih, model neural network dengan dua dropout layer memiliki performa yang sepadan dengan model neural network tanpa dropout layer. Namun hal ini berlaku hanya pada label kelas yang memiliki banyak sampel. Performa klasifikasi label kelas dengan sampel sedikit seperti judul dan penulis justru sedikit menurun. Sedangkan pada website baru yang belum pernah dilatih sebelumnya, performa model ini cenderung menurun.

Model selanjutnya adalah model dasar yang dilatih hanya menggunakan properti CSS sebagai fitur. Performa variasi model ini dapat dilihat pada Tabel VI.

TABEL VI
PERFORMA MODEL PROPERTI CSS

	Website Latihan			Website Baru		
Label	P	R	F-1	P	R	F-1
Non-Konten	99.71%	99.45%	99.58%	98.43%	85.09%	91.28%
Judul	100%	94.12%	96.97%	0%	0%	0%
Penulis	96.92%	90%	93.33%	0%	0%	0%
Konten	95.25%	98.37%	96.79%	40%	92.73%	55.89%

Pada website yang sudah dilatih, model neural network yang hanya menggunakan properti CSS sebagai fitur memiliki performa yang sepadan dengan model neural network dasar dengan tag path dan properti CSS sebagai fitur. Pada website yang belum pernah dilatih sebelumnya, performa identifikasi konten lebih baik jika hanya mengandalkan properti CSS. Hal ini mungkin dikarenakan karakteristik properti CSS yang cenderung lebih objektif dan umum pada berbagai website jika dibandingkan dengan tag path. Namun penggunaan tag path juga penting untuk mengidentifikasi suatu judul pada website secara umum. Hal ini dikarenakan kebanyakan blok teks judul pada suatu website ditandai dengan tag heading ketimbang tag paragraf.

Model selanjutnya adalah model dasar yang dilatih hanya menggunakan dataset website yang sama saja. Sehingga terdapat lima total model yang dilatih khusus untuk masing-masing Detik.com, Iwanbanaran.com, Kompas.com, Teknojurnal.com, dan Tipspintar.com. Performa variasi model per website ini dapat dilihat pada Tabel VII.

TABEL VII
PERFORMA F-1 SCORE MODEL PER WEBSITE

Website	Non-Konten	Judul	Penulis	Konten
Detik.com	99.99%	100%	88.89%	100%
Iwanbanaran.com	99.98%	100%	100%	99.13%
Kompas.com	99.61%	95.24%	90%	92.94%
Teknojurnal.com	99.88%	100%	100%	99.43%
Tipspintar.com	99.19%	100%	100%	98%

Model neural network yang dilatih hanya dengan

menggunakan dataset suatu website saja dapat mengidentifikasi judul, penulis, dan konten dengan hampir sempurna. Namun model neural network yang sudah terlatih ternyata sedikit sensitif terhadap noise sehingga mencegah performa model untuk mencapai nilai sempurna. Masalah ini tidak dimiliki oleh model Support Vector Machine yang dijadikan sebagai rujukan.

Selanjutnya model klasifikasi binari akan di uji coba. Model binary class yang pertama adalah model binary class yang hampir sama dengan model dasar, namun jumlah node pada output layernya hanya satu. Node ini merepresentasikan apakah suatu blok teks merupakan konten atau tidak. Loss Function yang digunakan juga diganti dengan Binary Crossentropy. Model ini selanjutnya akan direferensikan sebagai model dasar klasifikasi binari. Model kedua adalah model dasar klasifikasi binari yang datasetnya hanya menggunakan properti CSS sebagai fitur. Model ketiga adalah model dasar klasifikasi binari yang tidak menggunakan class weight. Model keempat adalah model dasar klasifikasi binari yang menggunakan dua dropout layer dengan dropout rate 0.2. Performa setiap model klasifikasi binari diatas dapat dilihat pada Tabel VIII.

TABEL VIII
PERFORMA MODEL KLASIFIKASI BINARI

	Website Latihan			Website Baru		
Website	**P**	**R**	**F-1**	**P**	**R**	**F-1**
Model 1	96.82%	98.23%	97.52%	16.19%	14.29%	15.18%
Model 2	95.51%	97.50%	96.5%	11.46%	9.24%	10.23%
Model 3	97.40%	97.61%	97.51%	57.24%	69.75%	62.87%
Model 4	97.5%	97.4%	97.45%	21.05%	26.89%	23.62%

Performa semua variasi model neural network pada website yang sudah dilatih sedikit lebih baik ketimbang SVM. Performa model neural network yang dilatih tanpa menggunakan class weight memberikan performa terbaik. Performa F-1 score yang dihasilkan bahkan dua kali lebih baik dari performa model Support Vector Machine yang dijadikan sebagai rujukan pada website baru yang belum pernah dilihat sebelumnya.

Model selanjutnya adalah model klasifikasi binari tanpa class weight yang dilatih hanya menggunakan dataset website yang sama saja. Sehingga terdapat lima total model yang dilatih khusus untuk masing-masing Detik.com, Iwanbanaran.com, Kompas.com, Teknojurnal.com, dan Tipspintar.com. Performa variasi model per website ini dapat dilihat pada Tabel IX.

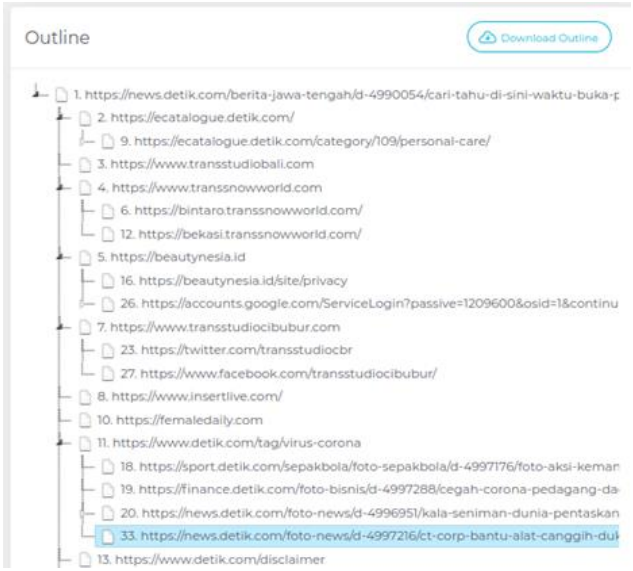
TABEL IX
PERFORMA MODEL KLASIFIKASI BINARI PER WEBSITE

Website	**Precision**	**Recall**	**F-1 Score**
Detik.com	98.57%	98.57%	98.57%
Iwanbanaran.com	98.61%	100%	99.3%
Kompas.com	95.76%	91.87%	93.78%
Teknojurnal.com	99.03%	100%	99.51%
Tipspintar.com	97.82%	98.15%	97.99%

Model neural network klasifikasi binari pada tiap website tidak dapat menghasilkan performa sempurna seperti model klasifikasi SVM. Hal ini mungkin dikarenakan tingkat kesensitifan model neural network terhadap noise sehingga mencegah performa model untuk mencapai nilai sempurna. Selain itu, dalam konteks satu model neural network per website, performa model neural network klasifikasi binari tidak jauh beda dengan performa model neural network klasifikasi multiclass. Oleh karena itu, jika terdapat suatu kebutuhan untuk melatih sebuah model neural network untuk suatu website, maka model neural network klasifikasi multiclass adalah pilihan yang lebih baik.

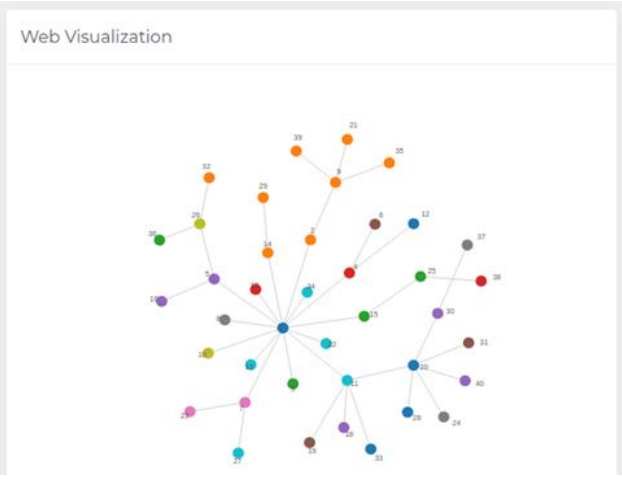
G. Web Data Visualization

Visualisasi outline tree dari hasil web crawling berhasil ditampilkan pada bagian kiri halaman hasil utama. Halaman web yang didapatkan dari halaman web lainnya akan berada di dalam turunan dari halaman web tersebut dengan posisi sedikit menjorok ke kanan. Turunan suatu halaman web juga dapat di buka tutup dengan klik. Hasil visualisasi grafik outline tree treeJS dapat dilihat pada Gambar 5 di bawah ini.



Gambar. 5. Outline Tree

Visualisasi grafik D3 dari hasil web crawling berhasil ditampilkan pada bagian tengah halaman hasil utama. Setiap node memiliki banyak koneksi ke halaman web yang lain. Setiap node juga memiliki warna yang berbeda-beda tergantung pada domain dari node tersebut. Hasil visualisasi grafik D3 dapat dilihat pada Gambar 6 di bawah ini.



Gambar. 6. Grafik

Blok teks terklasifikasi dari hasil klasifikasi neural network classifier ditampilkan pada bagian kanan halaman hasil utama. Blok teks yang diklasifikasikan sebagai judul akan ditampilkan sebagai judul dengan tag heading 1. Blok teks yang diklasifikasikan sebagai penulis akan ditampilkan sebagai penulis dengan tag paragraph dan emphasize. Blok teks yang diklasifikasikan sebagai konten akan ditampilkan sebagai konten dengan tag paragraph. Tampilan blok terklasifikasi dapat dilihat pada Gambar 7 di bawah ini.

Gambar. 7. Blok Teks Terklasifikasi

VI. KESIMPULAN

Website Web Content Extractor dapat mengotomatisasi proses pengambilan artikel di internet dengan menggunakan kombinasi web crawler dan ekstraktor blok teks dan CSS. Namun untuk mendapatkan hasil penyaringan konten murni yang ideal, model neural network yang digunakan harus dilatih dengan website yang akan diambil artikelnya. Dengan sampel yang cukup, model neural network dapat mengidentifikasi judul, penulis, dan konten artikel.

Website Web Content Extractor dapat memvisualisasi berbagai halaman website dengan mempresentasikan setiap halaman web sebagai sebuah node yang memiliki banyak koneksi ke halaman web yang lain. Website ini juga memiliki visualisasi berupa outline tree.

Model Deep Feed Forward Neural Network memiliki potensi untuk melakukan klasifikasi antara konten dan non konten secara umum selama diberi dataset latihan yang cukup dan bervariasi. Namun model ini belum dapat memberikan performa yang cukup untuk mengklasifikasikan judul dan penulis pada website secara umum.

Model Deep Feed Forward Neural Network dapat melakukan klasifikasi antara konten dan non konten untuk halaman web secara umum dengan lebih baik dibandingkan menggunakan Support Vector Machine. Performa F-1 score yang dihasilkan model neural network dua kali lebih baik ketimbang performa F-1 score model SVM. F-1 score dari model neural network adalah 62.87%. Sedangkan F-1 score dari model Support Vector Machine adalah 31.28%.

Penggunaan DBSCAN tidak cocok untuk mengklasterisasi blok teks pada halaman website yang memiliki berbagai template dan tidak menggunakan prinsip atau atribut yang sama untuk setiap elemen artikelnya. Penggunaan algoritma LCS hanya cocok digunakan ketika semua teks blok yang berisi konten berhasil diklasterisasi dalam satu klaster saja. Selain itu, meta tag yang deskriptif juga sangat diperlukan pada setiap halaman berartikel agar proses penilaian klaster berisi teks artikel dapat bekerja dengan baik. Website Indonesia yang menjadi data set sayangnya tidak menggunakan meta tag dengan standard yang disebutkan diatas.

DAFTAR PUSTAKA

[1] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[2] F. Chollet, "Keras: The Python Deep Learning library," *Keras.Io*, 2015.
[3] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv Prepr. arXiv1603.04467*, 2016.
[4] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, 2011.
[5] Z. Zhou and M. Mashuq, "Web content extraction through machine learning," *Standford Univ.*, pp. 1–5, 2014.
[6] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[7] E. Lim, E. I. Setiawan, and J. Santoso, "Stance Classification Post Kesehatan di Media Sosial Dengan FastText Embedding dan Deep Learning," *J. Intell. Syst. Comput.*, vol. 1, no. 2, pp. 65–73, 2019.
[8] M. A. Rahman, H. Budianto, and E. I. Setiawan, "Aspect Based Sentimen Analysis Opini Publik Pada Instagram dengan Convolutional Neural Network," *J. Intell. Syst. Comput.*, vol. 1, no. 2, pp. 50–57, 2019.
[9] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, and others, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *kdd*, 1996, vol. 96, no. 34, pp. 226–231.
[10] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015.