

# Membangun *Search Engine* “Caari” dengan Metode *Term Frequency Invers Document Frequency* (TF-IDF) dan Rekomendasi *Simple Multi Attribute Rating Technique* (SMART)

Siti Rosalina<sup>1</sup>, Naim Rochmawati<sup>2</sup>

<sup>1,2</sup> Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

<sup>1</sup>[siti.19073@mhs.unesa.ac.id](mailto:siti.19073@mhs.unesa.ac.id)

<sup>2</sup>[naimrochmawati@unesa.ac.id](mailto:naimrochmawati@unesa.ac.id)

**Abstrak**— *Search engine* merupakan sebuah *website* atau perangkat lunak yang dibangun dengan tujuan membantu pengguna untuk mencari informasi di internet. Tidak semua *search engine* memberikan hasil yang relevan sehingga pengguna tidak menemukan informasi secara cepat dan akurat. Kecenderungan pengguna mengklik hasil halaman pertama membuat perankingan hasil dapat meningkatkan relevansi informasi yang diinginkan pengguna. Maka dari itu dibuatlah sebuah *search engine* “Caari” dengan metode *Term Frequency Inverse Document Frequency* (TF-IDF) dan rekomendasi *Simple Multi Attribute Rating Technique* (SMART) untuk meningkatkan relevansi hasil pencarian. Melalui *User Acceptance Test* (UAT) kepada 100 responden, sebanyak 85% responden berpendapat bahwa *search engine* dapat berjalan dengan baik. Selain itu berdasarkan hasil pengujian performa menggunakan GTMetrix diperoleh persentase sebesar 97%, Sehingga dapat disimpulkan bahwa *search engine* ini relevan dan layak untuk digunakan.

**Kata Kunci**— *search engine*, *Term Frequency Inverse Document Frequency* (TF-IDF), *Simple Multi Attribute Rating Technique* (SMART), relevan, performa.

## I. PENDAHULUAN

Informasi menurut Kamus Besar Bahasa Indonesia (KBBI) memiliki arti pemberitahuan, kabar, atau berita tentang sesuatu. Keberadaan informasi akhir-akhir ini sangat melimpah. Hal ini selaras dengan kebutuhan informasi manusia yang semakin meningkat. Kita bisa memperoleh informasi diberbagai tempat, dimana saja, dan kapan saja.

Internet menjadi salah satu contoh tempat untuk memperoleh informasi. Berdasarkan penelitian dari Georgia Tech's GVU Center, Lebih dari 80% pemakai internet mengandalkan *search engine* sebagai alat pencari informasi [1]. Mesin pencari atau *search engine* adalah sebuah perangkat lunak komputer yang dibangun dengan tujuan membantu pengguna untuk mencari informasi di dunia maya [2]. Ada beberapa contoh *search engine* yang sering digunakan dikalangan masyarakat yaitu Google, yahoo!search, Bing, dll.[3].

Tidak semua *search engine* memberikan hasil pencarian yang relevan sehingga menyebabkan sulitnya masyarakat

untuk mendapat informasi yang cepat dan akurat [4]. Pengguna cenderung melihat atau mengklik item dengan peringkat atas [5], pada *search engine* Google, 28,5% pengguna klik hasil halaman pertama [6], berdasarkan uraian tersebut perankingan hasil akan meningkatkan relevansi informasi yang diinginkan pengguna.

Penelitian mengenai relevansi hasil menggunakan TF-IDF telah dilakukan oleh Arif Tirtana dkk [4] yang membahas mengenai pembuatan sistem pencarian pekerjaan, penelitian tersebut memanfaatkan TF-IDF untuk meningkatkan relevansi hasil pencarian dengan menghitung frekuensi kata kunci pada tiap data tidak hanya mengandalkan *query* atau kecocokan dengan kata kunci saja. Penelitian dengan metode yang sama juga dilakukan oleh Fitroh dkk [7] dalam rancang bangun mekanisme pencarian judul skripsi pada SISINTA dengan hasil akurasi 92 % yang diuji menggunakan pengujian *white box*. Penelitian terbaru pada tahun 2022 dilakukan oleh Rudiansyah dkk dengan judul *search engine* Menggunakan Metode *Information Retrieval* [8], dalam penelitian tersebut metode yang digunakan yaitu TF-IDF dengan terlebih dahulu membagi kata kunci yang dimasukkan menjadi per kata, kemudian dicocokkan dengan data informasi yang ada.

Selain dari kecocokan kata kunci, Rushikesh Karwa [9] dalam artikel yang berjudul *Building Search engine Using Machine Learning Technique*, menyatakan bahwa dengan menggunakan *machine learning*, *search engine* akan memberikan halaman website yang relevan berada di atas atau berada di ranking atas untuk pengguna.

Salah satu algoritma *machine learning* yang bisa digunakan yaitu algoritma sistem pengambilan keputusan. Pada beberapa penelitian yang membandingkan beberapa algoritma SPK didapatkan bahwa Algoritma SMART memiliki tingkat akurasi yang tinggi. Penelitian tentang metode SMART telah dilakukan oleh Afsha Zahara [10] dalam skripsinya yang berjudul Analisis Perbandingan Metode SAW, MOORA, SMART dalam Pemilihan Calon Mitra Statistik Pada BPS Kabupaten Serdang Bedagai, dengan hasil akurasi 100%. Penelitian lain dilakukan oleh Musri Iskandar Nasution [11] yang membandingkan algoritma SPK untuk pemilihan karyawan, metode yang dibandingkan yaitu SMART dan MAUT dengan hasil SMART lebih baik daripada MAUT. Penelitian yang sama juga dilakukan oleh Andi Elia dkk [12] dalam pemilihan lokasi toko, penelitian ini mendapatkan hasil sensitivitas yang

paling tinggi ada pada algoritma SMART dengan hasil 0,45. Selain dari akurasi yang tinggi, SMART memiliki algoritma yang sederhana sehingga mudah difahami dan fleksibel ketika ada perubahan atau penambahan kriteria baru. Dari beberapa penelitian tersebut dapat ditarik suatu kesimpulan bahwa metode SMART merupakan algoritma yang baik jika diterapkan pada *search engine*.

Berdasarkan latar belakang tersebut, penulis melakukan penelitian dengan judul Membangun *Search Engine* “Caari” Dengan Metode *Term Frequency Inverse Document Frequency* (TF-IDF) Dan Rekomendasi *Simple Multi Attribute Rating Technique* (SMART). Melalui penelitian ini diharapkan penulis dapat membangun *search engine* yang menyajikan hasil relevan untuk pengguna.

## II. METODE PENELITIAN

Metode *experimental* yang diterapkan pada penelitian ini terdiri dari beberapa tahapan sebagai berikut :



Gbr. 1 Flowchart Alur Penelitian.

### A. Identifikasi Masalah

Identifikasi masalah merupakan tahapan pertama dalam penelitian, dalam tahap ini ditentukan latar belakang penelitian itu dilakukan. Selain itu pada tahap ini ditentukan poin-poin permasalahan yang akan diselesaikan dalam penelitian.

### B. Pengumpulan Data

Pengumpulan data adalah prosedur yang sistematis dan standar untuk memperoleh data. Metode pengumpulan data yang digunakan sebagai berikut:

1) Metode Observasi: Dilakukan observasi atau pengamatan terhadap *search engine* yang ada untuk mengamati cara kerja dan rekomendasi hasil pencarian yang diberikan kepada pengguna.

2) Studi Literatur: Studi literatur memperoleh data dari berbagai sumber pustaka seperti hasil penelitian, skripsi, tesis, jurnal, buku, dan juga hasil pencarian internet sebagai referensi penelitian.

### C. Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan untuk menganalisis kebutuhan yang diperlukan dalam berlangsungnya penelitian. Adapun kebutuhan dalam pembuatan *search engine* “Caari” sebagai berikut :

#### 1) Perangkat keras (Hardware)

- Processor : Intel Core i7-6600U 2.80 GHz
- Memori : 16GB DDR4
- Penyimpanan : 512GB SSD

#### 2) Perangkat Lunak (Software)

- Sistem Operasi Windows 10 64-bit
- Visual Studio Code
- Web Browser Google Chrome
- Apache HTTP Server versi 2.4
- PHP versi 8.2

### D. Desain dan Perancangan

Desain dan perancangan merupakan pemodelan konseptual pengembangan sistem, tahap ini digunakan untuk meminimalisir kesalahan konsep pada saat pembuatan sistem atau implementasi.

1) Database: Database dalam *search engine* memiliki peran yang sangat penting sebagai penyimpanan situs atau dokumen yang akan ditampilkan di hasil pencarian. Dalam *search engine* ini menyediakan hasil pencarian berupa situs dan gambar, sehingga terdapat 2 tabel, untuk penyimpanan situs seperti pada Tabel I dan tabel untuk penyimpanan gambar seperti pada Tabel II.

TABEL I  
RANCANGAN TABEL SITES

| Nama Field  | Type Field   | Keterangan   |
|-------------|--------------|--|
| Id          | Integer (11) | Id situs   |
| url         | Varchar(512) | url situs  |
| Title       | Varchar(512) | Judul situs  |
| Description | Varchar(512) | Deskripsi situs  |
| Keyword     | Varchar(512) | Kata kunci situs   |
| Cliks       | Integer(11)  | Jumlah klik yang sudah diterima situs                          |
| Tf-idf      | Float        | Hasil TF –IDF berdasarkan kata kunci yang dimasukkan pengguna. |
| wtfIdf      | float        | Pembobotan TF-IDF  |

|          |       |                   |
|----------|-------|-------------------|
| w-clicks | float | Pembobotan clicks |
|----------|-------|-------------------|

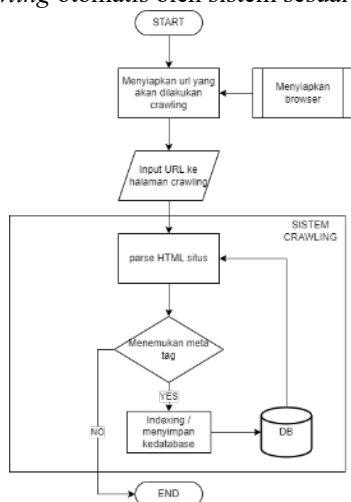
TABEL II  
RANCANGAN TABEL IMAGES

| Nama Field | Type Field   | Keterangan  |
|------------|--------------|---|
| Id         | Integer (11) | Id gambar   |
| siteUrl    | Varchar(512) | Url situs tempat gambar berada                                |
| imageUrl   | Varchar(512) | url gambar  |
| Alt        | Varchar(512) | Alt/deskripsi gambar  |
| Title      | Varchar(512) | Judul gambar  |
| Cliks      | Integer(11)  | Jumlah klik yang sudah diterima gambar                        |
| Broken     | tinyint(4)   | Untuk menampilkan gambar ukuran besar                         |
| Tf-idf     | Float        | Hasil TF-IDF berdasarkan kata kunci yang diinputkan pengguna. |
| wtfidf     | Float        | Pembobotan TF-IDF   |
| w-clicks   | Float        | Pembobotan clicks   |

## 2) Alur sistem *search engine*

### a) Penambahan data

Proses penambahan data menggunakan metode *hybrid* dengan *human-powered directories* untuk memasukkan url situs, kemudian dari url yang dimasukkan tersebut akan dilakukan *crawling* otomatis oleh sistem sesuai dengan Gbr. 2.



Gbr. 2 Flowchart penambahan data.

Proses dimulai dengan menyiapkan URL yang akan dilakukan *crawling* atau penambahan data, kemudian url tersebut dimasukkan ke field input URL di halaman *crawling*. Setelah URL tersebut dimasukkan, maka sistem kan merayapi

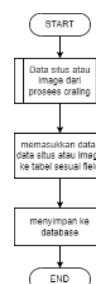
situs tersebut dengan parsing HTML yang kemudian mengambil meta tag yang ada pada situs tersebut. Apabila ditemukan meta tag yang sesuai maka akan dilakukan *indexing* ke dalam database, proses pencarian meta tag terus berulang hingga tidak ditemukan lagi meta tag dalam situs tersebut.

Untuk tag yang diambil dalam proses *crawling* sebagai berikut :

- Tag “a” untuk mendapatkan alamat atau url suatu situs;
- Tag “title” yang digunakan untuk memperoleh judul suatu situs atau gambar.
- “meta” tag dengan atribut “keywords” untuk memperoleh kata kunci dari suatu situs;
- “meta” tag dengan atribut “description” untuk mendapatkan deskripsi dari situs yang didapatkan;
- Tag “img” untuk mendapatkan gambar dalam situs yang dilakukan *crawling*;
- Tag “alt” untuk memperoleh alt, alternatif atau deskripsi dari suatu gambar;
- Tag “src” yang digunakan untuk mendapatkan alamat situs dari suatu gambar.

### b) Indexing

Proses *indexing* diawali dengan diperolehnya data situs maupun pada proses *crawling*, sesuai dengan *flowchart* pada Gbr. 3. kemudian data tersebut akan ditentukan tabelnya sesuai dengan kriteria masing-masing yaitu tabel situs untuk data-data situs dan tabel images untuk menampung data-data gambar, beserta *field* yang ada di dalamnya. Setelah menemukan tabel yang sesuai data akan tersimpan dalam database *search engine* Caari.



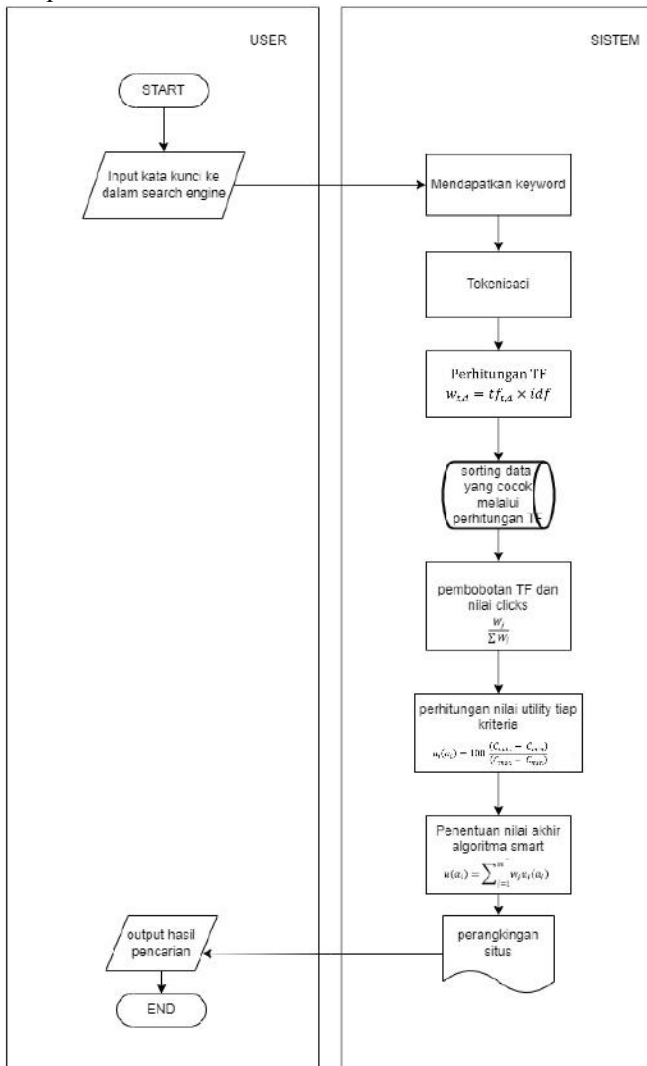
Gbr. 3 Flowchart indexing

### c) Searching

Proses *searching* diawali oleh pengguna memasukkan kata kunci, kemudian kata kunci tersebut akan dilakukan tokenisasi untuk memecah kata kunci yang telah diterima sistem, sesuai dengan *flowchart* pada Gbr. 4.

Proses selanjutnya yaitu dilakukan perhitungan TF-IDF untuk mencari situs-situs yang memiliki kecocokan dengan kata kunci yang telah diterima oleh *search engine* Caari. Kemudian situs-situs yang memiliki kecocokan akan dilakukan perhitungan metode SMART yang diawali dengan pembobotan, kemudian dihitung nilai *utility* dari masing-masing kriteria yaitu TF-IDF dan Clicks. Setelah nilai *utility* didapatkan dilakukan dengan perhitungan nilai akhir SMART, setelah itu situs yang memiliki kecocokan dengan kata kunci yang telah dimasukkan pengguna akan dilakukan pengurutan atau *sorting*. Proses pengurutan akan mengurutkan situs dengan nilai akhir tertinggi di urutan paling atas hingga situs dengan nilai akhir

terendah memiliki urutan paling bawah, kemudian untuk situs yang tidak memiliki kecocokan tidak akan ditampilkan dalam hasil pencarian.



Gbr. 4 Flowchart pencarian

### 3) Desain antarmuka

*Search engine* Caari membutuhkan 3 desain antar muka utama yaitu untuk *crawling* yang dilakukan administrator dan halaman untuk input kata kunci yang dilakukan pengguna juga penampilan hasil pencarian.

#### a) Rancangan desain antarmuka *crawling*

Rancangan desain pada Gbr. 5 terdapat form untuk memasukkan situs yang akan dilakukan. Setelah memasukkan alamat situs ada tombol “Crawling” yang digunakan untuk memproses alamat situs tersebut. Kemudian ada tulisan interaktif yang menunjukkan proses *crawling* sedang berlangsung ataupun proses *crawling* telah sukses.



Gbr. 5 Rancangan UI *crawling*

#### b) Rancangan desain antarmuka *searching*

Pada Gbr. 6 antar muka *searching* dirancang sesederhana mungkin agar pengguna tidak terganggu. Terdapat *branding* logo *search engine* dan juga *form* untuk memasukkan kata kunci yang akan dicari oleh pengguna.



Gbr. 6 Rancangan UI *searching*

#### c) Rancangan desain antarmuka hasil pencarian

Rancangan hasil pencarian dibuat seperti *search engine* pada umumnya, berupa daftar website atau situs yang sesuai dengan kata kunci seperti pada Gbr. 7, selain itu tetap disediakan *branding* logo dan juga *form* untuk memasukkan kata kunci untuk mempercepat pencarian jika ingin mencari hal baru tanpa harus kembali ke landing *page search engine* Caari.



Gbr. 7 Rancangan UI hasil pencarian

### E. Implementasi

Hasil perancangan yang didapatkan pada tahap sebelumnya, kemudian diimplementasikan dalam program / kode untuk membangun sistem *search engine* yang dapat berjalan dengan baik. Implementasi algoritma menggunakan bahasa pemrograman PHP yang dikombinasikan dengan javascript dan CSS dengan penyimpanan / database Mysql. Algoritma yang digunakan disesuaikan dengan metode *Term Frequency Invers Document Frequency* (TF-IDF) dan rekomendasi hasil menggunakan *Simple Multi Attribute Rating Technique* (SMART)

1) Perhitungan TF-IDF: Rumus *Term Frequency* (TF) dapat dilihat pada persamaan berikut :

$$tf_{t,d} = \frac{f_d(t)}{\max_{t \in d} f_d(t)} \quad (1)$$

Dengan  $tf$  adalah kemunculan kata  $t$  pada dokumen  $d$ , kemudian  $\max_{t \in d} f_d(t)$  adalah jumlah kata pada dokumen  $d$ . Sedangkan IDF memiliki rumus sebagai berikut :

$$idf_{t,d} = \log \left( \frac{N}{df_t + 1} \right) \quad (2)$$

Dimana :

$idf_{t,d}$  : nilai IDF dari suatu dokumen

$N$  : total atau jumlah dokumen

$df_t$  : jumlah dokumen yang terdapat kata  $t$

Sehingga untuk rumus TF- IDF adalah :

$$w_{(t,d)} = tf \times idf \quad (3)$$

2) Perhitungan SMART: Proses perhitungan SMART memiliki beberapa langkah sebagai berikut:

- a) Menentukan kriteria, bobot masing-masing kriteria, dan normalisasi kriteria. Rumus perhitungan normalisasi kriteria sebagai berikut

$$\frac{w_j}{\sum w_j} \quad (4)$$

Dimana :

$w_j$  : nilai bobot suatu kriteria

$\sum w_j$  : total jumlah bobot dari seluruh kriteria

- b) Perhitungan nilai *utility*: Menghitung nilai *utility* dengan mengonversikan nilai dari atribut pada tiap-tiap kriteria menjadi atribut data baku, menggunakan persamaan sebagai berikut:

$$u_i(a_i) = 100 \frac{(C_{out\ i} - C_{min})}{(C_{max} - C_{min})} \quad (5)$$

Dimana :

$u_i(a_i)$  : nilai *utility* atribut ke-1 untuk atribut ke-i

$C_{max}$  : nilai kriteria maksimal

$C_{out\ i}$  : nilai atribut atau kriteria ke-i

$C_{min}$  : nilai kriteria minimal

- c) Perhitungan nilai akhir: untuk perhitungan nilai akhir dari masing-masing atribut menggunakan persamaan

$$u(a_i) = \sum_{j=1}^m w_j u_i(a_i) \quad (6)$$

Dimana :

$u(a_i)$  : nilai total alternatif

$w_j$  : hasil normalisasi bobot kriteria

$u_i(a_i)$  : hasil penentuan nilai *utility*

$i$  : 1,2, ...  $m$

## F. Hasil

Tahap ini menjelaskan tentang keluaran dari hasil implementasi algoritma TF dan SMART berupa *search engine*. *Search engine* ini dapat menerima masukkan kata kunci dari pengguna yang kemudian akan ditampilkan hasil pencarian dengan rekomendasi sesuai algoritma SMART.

## G. Pengujian

Proses pengujian berfokus pada logika *search engine* untuk menemukan kesalahan-kesalahan dan memastikan *search engine* berjalan dengan baik. Proses ini juga dilakukan untuk menghitung nilai akurasi algoritma SMART pada hasil rekomendasi hasil pencarian terhadap relevansi yang diharapkan oleh pengguna. Pengujian akan dilakukan dengan

metode UAT dengan metode penyebaran kuesioner yang melibatkan 100 masyarakat umum dengan rentang usia lebih dari (>) 15 tahun. Dilakukan juga pengujian menggunakan GTMetrix untuk menganalisis performa *search engine* sekaligus dibandingkan dengan performa *search engine* "Google".

1) *User Acceptance Test (UAT)*: Dalam pelaksanaan UAT terdapat beberapa langkah yang perlu dilakukan :

- a) Perencanaan: Proses ini berkaitan dengan menguraikan strategi untuk tes UAT yang harus dilakukan.
- b) Skenario Tes: Pada tahap ini mulai menyusun strategi yang sudah direncanakan mengacu pada semua yang dapat dilakukan atau kemungkinan situasi. Untuk pengujian *search engine* yang dapat digunakan oleh masyarakat umum, dibuatlah kuesioner pengujian dengan kriteria responden adalah masyarakat umum usia >15 tahun dan terbiasa melakukan pencarian dengan mesin pencari/ *search engine* (Google, Yahoo, Bing, Duck-duck Go, dll). Kuisisioner yang digunakan sebagai berikut :
  - Apakah Anda dapat mengakses <http://caari.rf.gd/> ?
  - Apakah Anda dapat memasukkan kata kunci dalam pencarian ?
  - Apakah *search engine* "Caari" dapat menampilkan hasil pencarian Anda?
  - Apakah tampilan mudah digunakan dan mudah dalam memahami fungsi dalam *search engine*?
  - Apakah secara keseluruhan, *search engine* dapat berjalan dengan baik?
  - Jika Tidak berjalan dengan baik, dimana letak proses tersebut ?
  - Menurut Anda, apa yang perlu ditambahkan atau diperbaiki dalam *search engine* ini?
- c) Eksekusi: Setelah kuesioner tersusun, dilanjutkan dengan pengujian UAT yaitu penyebaran kuesioner dan pastikan semua pertanyaan terjawab sehingga semua proses dalam *search engine* sudah dicoba.
- d) Keputusan Akhir : Dari data yang diperoleh saat eksekusi, diolah untuk menentukan apakah sistem dapat digunakan atau mungkin harus dilakukan pengujian lagi setelah perbaikan.

2) GTMetrix: Untuk menganalisis *website* menggunakan GTMetrix terdapat beberapa tahapan yang perlu dilakukan:

- a) Akses web GTmetrix yaitu <https://gtmetrix.com/> ;
- b) Pada halaman beranda GTmetrix terdapat kotak *input* "Enter URL to Analyze" masukkan link *website* yang akan dianalisis;
- c) Kemudian klik tombol "Test Your Site";
- d) Proses analisis akan dilakukan pada link *website* yang telah dimasukkan dan akan membutuhkan waktu beberapa saat;
- e) Setelah selesai maka akan tampil hasil dari analisis berupa ringkasan performa dari link *website* dari skor PageSpeed, waktu muat halaman, ukuran halaman, serta dapat melihat faktor-faktor yang mempengaruhi performa dari *website*;

- f) Untuk membandingkan dengan website lain klik tombol “Compare” di pojok kanan atas;
- g) Kemudian masukkan alamat/URL website yang akan dibandingkan;

### III. HASIL DAN PEMBAHASAN

#### A. Hasil dan Analisa

*Search engine* “Caari” dapat diakses dengan alamat <http://caari.rf.gd/>

1) Dataset: Dataset diperoleh melalui 2 proses penambahan data, yang pertama *crawling* secara otomatis menggunakan metode *parsing* HTML dan metode *human-powered directories* yang dilakukan oleh admin dengan memasukkan data *website* secara manual. Dalam proses tersebut dihasilkan data *website* dan juga gambar yang menjadi informasi dalam proses pencarian.

##### a) *Crawling*:

Pada *landing page crawling* terdapat *form* untuk memasukkan alamat situs yang akan dilakukan *crawling* seperti pada Gbr. 8. Pada gambar tersebut akan dilakukan *crawling* terhadap situs <https://unesa.ac.id>. Setelah memasukkan alamat situs ada tombol submit yang digunakan untuk memproses alamat situs tersebut.

Setelah memasukkan alamat situs maka pengguna diminta untuk menunggu. Pada tahap ini situs dilakukan *crawling* menggunakan *parsing* HTML oleh sistem. HTML dari situs tersebut diidentifikasi dan di ambil data-data yang sesuai untuk dapat dimasukkan ke dalam database *search engine* Caari.

Gbr. 8 UI *Crawling*

Pada saat proses *crawling* selesai, maka akan ditampilkan “The system has received the url” yang mengindikasikan bahwa proses *crawling* telah selesai dan data telah di *index* ke dalam database.

Dari proses *crawling* terhadap <https://unesa.ac.id> didapatkan beberapa situs, total lebih dari 100 data didapatkan dari proses *crawling* situs tersebut. Data tersebut nantinya yang akan ditampilkan apabila ada yang mencari informasi mengenai Unesa

##### b) *Human Powered-Directories*:

Pada *landing page crawling* akan didapati “Add Manually”. Link tersebut akan membawa pengguna kepada *form* untuk memasukkan data *website* secara manual. Seperti pada Gbr. 9 dibawah ini.

Pengguna dapat memasukkan url atau alamat situs di *field* url, kemudian memasukkan judul di *field* Title, memasukkan deskripsi situs yang telah dimasukkan ke dalam *field* description, dan yang terakhir memasukkan di *field* keyword mengenai kata kunci yang sesuai dengan situs yang telah dimasukkan. Setelah mengisi semua *field* yang telah tersedia, dilanjutkan melakukan klik di tombol submit. Data *website* akan langsung masuk ke dalam database *search engine* Caari.

#### Add Data Manually

Gbr. 9 Form *human powered-directories*

#### 2) Desain antarmuka

- a) Desain antarmuka *crawling*: Pada tampilan *crawling* terdapat *form* untuk memasukkan situs yang akan dilakukan *crawling* seperti pada Gbr. 10, setelah memasukkan alamat situs ada tombol *submit* yang digunakan untuk memproses alamat situs tersebut. Selain itu terdapat *Add manually* yang dapat digunakan untuk memasukkan data situs langsung tanpa *crawling*.

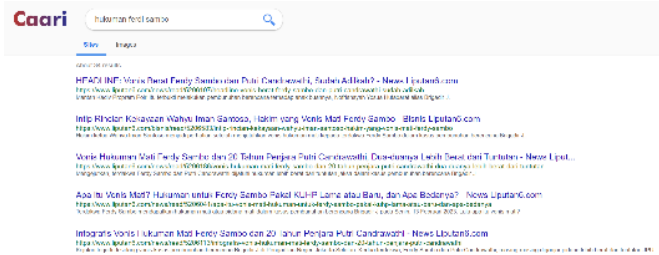
Gbr. 10 UI Landing Page *Crawling*

- b) Desain antarmuka *searching*: Pada *landing page* Caari terdapat logo atau *branding* nama Caari disertai *field* untuk pengisian kata kunci untuk pencarian seperti yang ada pada Gbr. 11. Selain itu pada halaman ini disertai tombol search untuk memulai pencarian, selain menekan tombol, untuk meningkatkan efisiensi menekan enter setelah mengetikkan kata kunci juga dapat dilakukan untuk memproses pencarian.

Gbr. 11 UI Landing Page *Searching*



- c) Desain antarmuka Hasil Pencarian: Setelah memasukkan kata kunci, *search engine* akan menampilkan hasil pencarian seperti pada Gbr. 12. pada tampilan ini juga terdapat *field* untuk memasukkan kata kunci, kemudian terdapat situs yang dirangking dari atas ke bawah secara berurutan, juga terdapat daftar halaman dibagian bawah.



Gbr. 12 UI Hasil Pencarian

### 3) Perhitungan TF-IDF dan SMART

- a) Perhitungan TF-IDF: Menggunakan rumus perhitungan TF-IDF yang dilakukan terhadap 1000 situs pertama hasil *crawling* sebagai sampel, yang kemudian didapatkan hasil seperti pada Tabel III berikut:

TABEL III  
TABEL PERHITUNGAN TF-IDF

| No | Url                                   | Title  | Description  | Keyword   | TF |
|----|---------------------------------------|--|--|---|----|
| 1  | https://www.niagahoster.co.id/        | Hosting No. 1 di Indonesia - Niagahoster               | Niagahoster provider hosting di Indonesia dengan pelayanan terbaik. Nikmati layanan Unlimited Hosting, Cloud Hosting, VPS, dan Domain dengan harga terjangkau! | hosting, hosting murah, vps murah, membuat website, web hosting, hosting indonesia, hosting terbaik, hosting indonesia, hosting terbaik indonesia, hosting dan domain gratis, hosting indonesia terbaik, web hosting terbaik, web hosting murah, hosting murah indonesia, hosting murah indonesia, cloud hosting indonesia, cloud vps indonesia | 0  |
| 2  | https://panel.niagahoster.co.id/cart  | Cart   Web Hosting Indonesia Gratis SSL - Niagahoster  | Web hosting murah Indonesia dengan disk space dan bandwidth unlimited, support 24 jam, cPanel, serta server dan data center terbaik di Indonesia.              |   | 0  |
| 3  | https://panel.niagahoster.co.id/login | Login Akun   Hosting, Domain dan Website - Niagahoster | Login dengan menggunakan alamat email dan password Anda untuk masuk ke member area. Kelola hosting, domain dan website Anda dalam 1 akun.                      |   | 0  |

|     |   |  |  |   |      |
|-----|---|--|--|---|------|
| 4   | https://panel.niagahoster.co.id/client/logout   | Hosting No. 1 di Indonesia - Niagahoster                           | Niagahoster provider hosting di Indonesia dengan pelayanan terbaik. Nikmati layanan Unlimited Hosting, Cloud Hosting, VPS, dan Domain dengan harga terjangkau!   | hosting, hosting murah, vps murah, membuat website, web hosting, hosting indonesia, hosting terbaik, web hosting indonesia, hosting terbaik indonesia, hosting dan domain gratis, hosting indonesia terbaik, web hosting terbaik, web hosting murah, hosting murah indonesia, hosting ... | 0    |
| 5   | https://esportsku.com/build-item-nana-marksmn-ml-tersakit-mobile-legends-2020/          | Build Item Nana Marksmn an ML Tersakit Mobile Legends Baru!        | Kali ini build item Nana Marksman tersakit di meta season 16 Mobile Legends. Bukan Marksman kaleng-kaleng  | mobile legends nana build item marksmn  | 5,8  |
| ... | ...   | ...  | ...  | ...   | ...  |
| ... | https://zathong.com/nana-build-mobile-legends   | Best Nana build 2023 :: Items, Emblems & Strategy [Mobile Legends] | Best Nana build guides for Mobile Legends 2023. I works hard to keep my's Mobile Legends builds and guides updated, and will help you craft the best Nana build for the meta. Learn more about Nana's abilities, Items, Emblems & Strategy | nana mobile legends item emblems strategy   | 4,37 |
| ... | https://www.getdroidtips.com/mobile-legends-nana-guide-2022-nana-best-build-and-emblem/ | Mobile Legends Nana Guide 2023   Nana Best Build and Emblem        | Nana has skills that deal with damage to the animals especially in group fights, heres Mobile Legends Nana Guide 2023 Best Build and Emblem  | mobile legends nana guide   | 4,37 |
| ... | dst   |  |  |   |      |

Berdasarkan hasil perhitungan TF-IDF, terdapat 10 situs yang memiliki nilai tidak 0 yang akan dilanjutkan untuk perhitungan SMART seperti pada Tabel IV berikut:.

TABEL IV  
SITUS DENGAN NILAI TF-IDF TAK 0

| No | Situs  | Nilai TF |
|----|--|----------|
| A  | Build Item Nana Marksman ML Tersakit Mobile Legends Baru!          | 5,8      |
| B  | Cara Menjadikan Nana Sebagai Marksman Support di Mobile Legends    | 5,8      |
| C  | Best Nana build 2023 :: Items, Emblems & Strategy [Mobile Legends] | 4,37     |

|   |  |      |
|---|--|------|
| D | Mobile Legends Nana Guide 2023   Nana Best Build and Emblem                              | 4,37 |
| E | Panduan Hero Mobile Legends: Nana – METACO   | 2,9  |
| F | Serba-serbi Nana Mobile Legend, Karakter Imut Paling Ditakuti                            | 1,73 |
| G | Daftar Nama Hero Mobile Legends Lengkap di Season 27                                     | 1,7  |
| H | Segala Hal tentang Ixia, Hero Baru Mobile Legends: Bang Bang yang Akan Rilis 8 Juli 2023 | 1,7  |
| I | Belajar Pemrograman Javascript untuk Pemula  | 0,25 |
| J | REBORN iMOBA APK 2023 PART 119 v2.20 [Download Skins]                                    | 0,25 |

diasumsikan 7 klik dan seterusnya sehingga diperoleh data pada Tabel VII.

TABEL VII  
NILAI UTILITY CLICKS

| Situs | Nilai Clicks | Utility Clicks | Situs | Nilai Clicks | Utility Clicks |
|-------|--------------|----------------|-------|--------------|----------------|
| A     | 730          | 100            | F     | 424          | 58,08219       |
| B     | 627          | 85,89041       | G     | 371          | 50,82192       |
| C     | 656          | 89,86301       | H     | 318          | 43,56164       |
| D     | 617          | 84,52055       | I     | 229          | 31,36986       |
| E     | 513          | 70,27397       | J     | 245          | 33,56164       |

#### b) Perhitungan SMART

- Menentukan kriteria, bobot masing-masing kriteria, dan normalisasi kriteria: berdasarkan rumus perhitungan normalisasi kriteria didapatkan hasil pada Tabel V sebagai berikut:

TABEL V  
KRITERIA DAN BOBOT SMART

| Kriteria | Bobot | Normalisasi |
|----------|-------|-------------|
| TF       | 70    | 0,7         |
| Clicks   | 30    | 0,3         |

Penentuan bobot didasarkan pada hasil ranking pada hasil survey yang dilakukan kepada 100 responden. Dari hasil tersebut dilakukan pembobotan dengan hasil atau keakuratan tertinggi.

- Perhitungan nilai *utility*: dari masing-masing kriteria yang sudah ditentukan, dihitung masing-masing nilai *utility*-nya. Berdasarkan nilai TF-IDF yang sudah didapatkan, kemudian dihitung menggunakan rumus *utility* dan didapatkan nilai pada Tabel VI sebagai berikut:

TABEL VI  
NILAI UTILITY TF-IDF

| Situs | Nilai TF-IDF | Utility TF-IDF | Situs | Nilai TF-IDF | Utility TF-IDF |
|-------|--------------|----------------|-------|--------------|----------------|
| A     | 5,8          | 99,9           | F     | 1,73         | 29,8           |
| B     | 5,8          | 99,9           | G     | 1,7          | 29,3           |
| C     | 4,37         | 75,4           | H     | 1,7          | 29,3           |
| D     | 4,37         | 75,4           | I     | 0,25         | 8,72           |
| E     | 2,9          | 50,4           | J     | 0,25         | 8,72           |

Untuk kriteria Clicks, nilai clicks berasal dari hasil kuisioner yang telah dibagikan, setiap urutan pertama dari responden diasumsikan dengan 10 nilai klik, di urutan kedua diasumsikan dengan 9 klik, urutan ke 3

- Perhitungan nilai akhir: Nilai *utility* yang didapatkan dikalikan dengan bobot masing-masing kriteria kemudian dijumlahkan sesuai rumus nilai akhir SMART. Diperoleh hasil seperti pada Tabel VIII berikut:

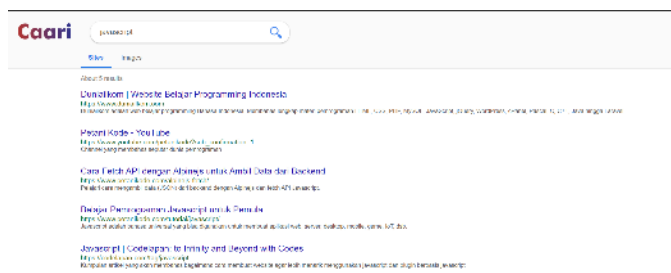
TABEL VIII  
PERHITUNGAN NILAI AKHIR SMART

| Situs | Nilai TF-IDF | Utility TF-IDF | Nilai Clicks | Utility Clicks | Nilai Akhir | Rank |
|-------|--------------|----------------|--------------|----------------|-------------|------|
| A     | 5,8          | 99,9           | 730          | 100            | 99,93       | 1    |
| B     | 5,8          | 99,9           | 627          | 85,89041       | 95,69712    | 2    |
| C     | 4,37         | 75,4           | 656          | 89,86301       | 79,7389     | 3    |
| D     | 4,37         | 75,4           | 617          | 84,52055       | 78,13617    | 4    |
| E     | 2,9          | 50,4           | 513          | 70,27397       | 56,36219    | 5    |
| F     | 1,73         | 29,8           | 424          | 58,08219       | 38,28466    | 6    |
| G     | 1,7          | 29,3           | 371          | 50,82192       | 35,75658    | 7    |
| H     | 1,7          | 29,3           | 318          | 43,56164       | 33,57849    | 8    |
| I     | 0,25         | 8,72           | 229          | 31,36986       | 15,51496    | 10   |
| J     | 0,25         | 8,72           | 245          | 33,56164       | 16,17249    | 9    |

#### 4) Pola Pencarian

- Pencarian satu kata: Pada Gbr. 13, ditampilkan hasil pencarian dengan jumlah kata kunci satu kata yaitu "javascript". Pada gambar tersebut ditampilkan situs hasil pencarian yang memiliki judul atau deskripsi yang terdapat kata "javascript". Situs yang tidak memiliki judul, URL, deskripsi, maupun kata kunci "javascript" tidak akan ditampilkan untuk pengguna.





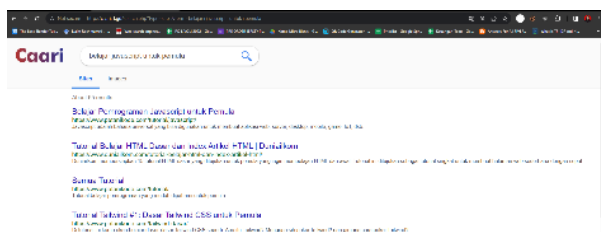
Gbr. 13 Pencarian 1 kata

- b) Pencarian 2 kata: Pencarian dalam *search engine* juga dapat dilakukan dengan memasukkan lebih dari satu kata. Seperti pada Gbr. 14 yang menampilkan hasil untuk pencarian dengan kata kunci “Ferdin Sambo”. Berdasarkan kata kunci tersebut, *search engine* menampilkan semua situs yang memiliki judul atau nama situs, deskripsi, dan kata kunci yang memuat kata “Ferdin” dan situs yang memuat kata “Sambo” maupun keduanya. Situs yang memuat kedua kata tersebut akan ditampilkan lebih atas daripada situs yang hanya memuat satu kata kunci pencarian, karena situs dengan kecocokan kata kunci terbanyak akan dinilai lebih relevan.



Gbr. 14 Pencarian 2 kata

- c) Pencarian lebih dari 2 kata : Sangat mungkin pengguna melakukan pencarian lebih dari 2 kata untuk memaksimalkan pencarian mereka. Pada Gbr. 15 dibawah ini ditampilkan hasil pencarian dengan 4 kata yaitu “belajar javascript untuk pemula”. Berdasarkan kata kunci tersebut maka *search engine* akan menampilkan situs yang sesuai dengan kata kunci tersebut.



Gbr. 15 Pencarian lebih dari 2 kata

Pada gambar 4.15 merupakan hasil pencarian dengan kata kunci “belajar javascript untuk pemula” pada rangking bawah dihalaman 10 pencarian. Pada

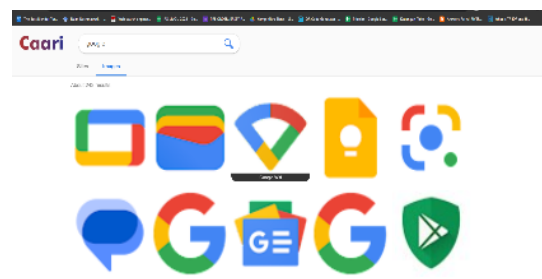
beberapa hasil pencarian jika ditelaah dari kata kuncinya ada beberapa situs yang ditampilkan tidak sesuai dengan maksud pengguna, akan tetapi situs tersebut tetap ditampilkan kata ada kesesuaian diantara salah satu kata kunci. Hal tersebut bisa dilihat pada hasil situs “Tutorial Docker : Cara Install dan Setup Docker dengan Benar!”, situs tersebut tidak ada hubungannya dengan belajar *javascript* akan tetapi di deskripsinya terdapat kata “belajar” sehingga situs tersebut muncul pada hasil pencarian.

- d) Pencarian tanpa hasil: Adakalanya dalam suatu pencarian, kata kunci yang dimasukkan oleh pengguna tidak memiliki kesesuaian dengan situs manapun. Pada Gbr. 16 merupakan tampilan hasil pencarian tanpa hasil pada mesin pencari, yang kemudian menampilkan “Maaf kami tidak menemukan apa yang Anda cari”.



Gbr. 16 Pencarian tanpa hasil

- e) Pencarian gambar: *Search engine* “Caari” memiliki fitur pencarian gambar seperti pada Gbr. 17. Sama seperti pada pencarian situs, pada pencarian gambar mesin pencari akan menyesuaikan kata kunci dengan judul gambar, alt,situs asal, dan juga deskripsi gambar. Kemudian gambar yang memiliki kecocokan akan ditampilkan oleh mesin pencari kepada pengguna



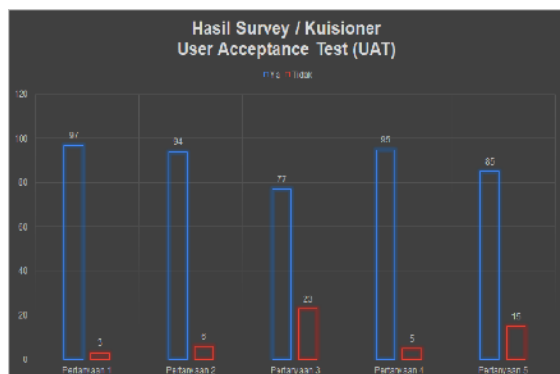
Gbr. 17 Pencarian Gambar

## B. Pengujian

Pengumpulan data adalah prosedur yang sistematis dan standar untuk memperoleh data. Metode pengumpulan data yang digunakan sebagai berikut:

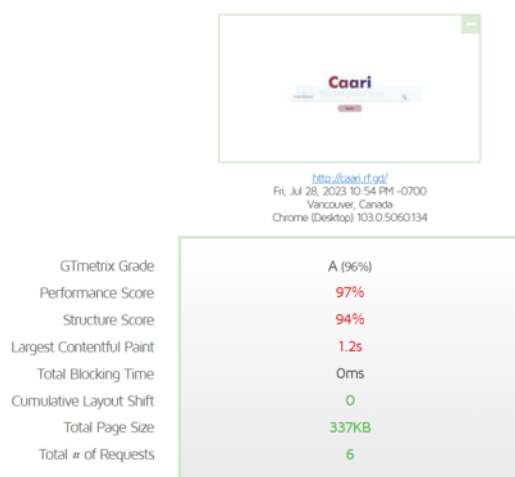
- 1) *User Acceptance Test (UAT)*: Berdasarkan kuisioner yang telah diisi oleh responden, penulis memperoleh hasil seperti pada Gbr. 18. Pada diagram tersebut menyatakan 97% orang dapat mengakses *search engine* “Caari” dan 85% orang berpendapat bahwa *search engine* tersebut berjalan dengan baik. Akan tetapi 23% responden tidak mendapatkan hasil dalam pencarian mereka, hal tersebut dikarenakan database atau *resource search engine* yang masih sedikit, prosentase tersebut akan terus menurun apabila *resources* yang dimiliki

*search engine* ini terus ditingkatkan. Berdasarkan uraian tersebut dapat disimpulkan bahwa *search engine* “Caari” dapat berjalan dengan baik sesuai dengan keinginan pengguna.



Gbr. 18 Diagram UAT

2) GTMetrix: merupakan salah satu tools yang dapat digunakan untuk menganalisa performa pada suatu website. Dalam pengujian ini *search engine* caari dibandingkan dengan Google, hasilnya ditampilkan pada Gbr. 19.



Gbr. 19 Hasil Pengujian GTMetrix

Pada pengujian menggunakan GTMetrix terdapat beberapa aspek yang dinilai, yaitu:

- GTmetrix Grade*, yang merupakan skor keseluruhan *website* menurut analisa yang telah dilakukan oleh GTMetrix. Pada aspek ini skor yang diperoleh Caari sebesar 96%.
- Performance* atau performa situs. Aspek ini menunjukkan seberapa cepat suatu *website* ketika diakses oleh pengunjung, semakin tinggi nilai maka semakin cepat *website* tersebut. Pada aspek performa, *search engine* Caari mendapatkan skor 97%.
- Structure Score*, penilaian ini berdasarkan struktur program situs yang menunjukkan seberapa baik *website* dibangun untuk performa yang optimal. Semakin tinggi nilainya berarti semakin bagus dan semakin efektif pula kode dalam *website* tersebut. Pada aspek ini, *search engine* Caari mendapatkan skor 94%. Jika dari segi struktur diperbaiki lagi dan

dikembangkan lebih lanjut, maka *structure score* Caari akan lebih baik.

- Largest Contentful Paint* (LCP). Aspek ini mengarah pada kecepatan *loading* suatu *website*, hal ini sangat mempengaruhi ranking *website* dalam mesin pencari. LCP menghitung waktu *loading* atau memuat sebagian besar elemen ditampilkan ke pengguna. Standar maksimal dari GTMetrix adalah 1,2 s dan Caari memperoleh skor tersebut, hal ini dikarenakan logo dan semua elemen yang ada dalam *search engine* Caari masih belum optimal, ukuran yang dipakai masih terlalu besar.
- Total Blocking Time* (TBT) yang menghitung jumlah waktu *website* terkunci sebelum dapat diakses oleh pengguna. Untuk aspek ini Caari memiliki nilai 0, karena *website* selalu dapat diakses pengguna tanpa terkunci.
- Cumulative Layout Shift* (CLS), mengarah pada pergeseran elemen yang tidak terduga pada saat *website* dimuat atau *loading*. Pada aspek ini Caari memiliki nilai yang baik, dimana Caari memiliki nilai 0 yang berarti tidak adanya kemungkinan terjadi pergeseran elemen dalam proses pemuatan atau *loading*.
- Total Page Size* yang menampilkan ukuran keseluruhan dari halaman *website* yang dicek. Semakin kecil maka semakin cepat pula *Fully Time Loaded Time* dan semakin tinggi skor. *Total Page Size* yang didapatkan Caari sebesar 337KB.
- Total of Request* yang didapatkan Caari sebesar 6. *Total of Request* mempengaruhi ukuran transfer, semakin tinggi *request* ukuran transfer semakin besar pula sehingga kinerja situs akan menurun. Hal ini dipengaruhi adanya *linking* atau pengalihan halaman yang cukup banyak.

#### IV. KESIMPULAN

Metode TF-IDF (*Term Frequency Inverse Document Frequency*) dan SMART (*Simple Multi Attribute Rating Technique*) dapat digunakan dalam proses pencarian untuk memberikan hasil yang relevan untuk pengguna dengan berbagai kombinasi kata kunci. Selain itu menurut kuisioner yang telah dibagikan, 85% berpendapat bahwa *search engine* “Caari” dapat berjalan dengan baik;

Pengujian menggunakan GTMetrix menunjukkan bahwa *search engine* Caari berdasarkan performa dan strukturnya memiliki nilai yang baik yaitu 97%. Selain itu secara keseluruhan, Caari mendapatkan skor GTMetrix sebesar 96%.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan YME atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan artikel ini. Penulis juga berterima kasih kepada keluarga yang senantiasa mendukung secara penuh. Terima kasih kepada Dosen pembimbing dan Dosen penguji yang banyak memberikan masukan sehingga penelitian ini dapat selesai dengan baik.

## REFERENSI

- [1] N. T. Cahyono, J. Triyono, dan S. Raharjo, "PENERAPAN TEKNIK SEO (SEARCH ENGINE OPTIMIZATION) PADA BLOG (STUDI KASUS: NOVA13.COM)", [Daring]. Tersedia pada: <http://www.hostinggokil.com>
- [2] I. Handayani, E. Febriyanto, dan M. Shofwatullah, "Optimalisasi visibilitas situs iLearning Journal Center (iJC) Pada Mesin Pencari Berbasis Search Engine Optimization (SEO) On Page," *SATIN - Sains dan Teknologi Informasi*, vol. 5, no. 1, hlm. 27–35, Jun 2019, doi: 10.33372/stn.v5i1.448.
- [3] Indira, D. Sulisty Kusumo, dan I. Lukmana Sardi, "Perancangan Website News Generator Menggunakan Rest API dan Search Engine Internal."
- [4] A. Tirtana, A. , Zulkarnain, dan Y. Dwi Listio, "Pembuatan Sistem Pencarian Pekerjaan Menggunakan TF-IDF," *Jurnal Ilmiah Teknologi Asia*, 13(2), hlm. 91–100, 2019.
- [5] F. Germano, V. Gómez, dan G. Le Mens, "The few-get-richer: A surprising consequence of popularity-based rankings," dalam *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, Association for Computing Machinery, Inc, Mei 2019, hlm. 2764–2770. doi: 10.1145/3308558.3313693.
- [6] H. Werthner, E. Prem, E. A. Lee, dan C. Ghezzi, *Perspectives on Digital Humanism*. Springer International Publishing, 2021. doi: 10.1007/978-3-030-86144-5.
- [7] M. Ayu, A. N. Fitroh, A. Prasetya Wibawa, dan U. Pujianto, "Metode term frequency-invers document frequency pada mekanisme pencarian judul skripsi," 2018. [Daring]. Tersedia pada: <http://journal2.um.ac.id/index.php/tekno>
- [8] R. Ariyansyah, R. Nanda, dan O. Wiranda, "Search Engine Menggunakan Metode Information Retrival," 2022.
- [9] R. Karwa dan V. Honmane, "Building search engine using machine learning technique," dalam *2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019*, Institute of Electrical and Electronics Engineers Inc., Mei 2019, hlm. 1061–1064. doi: 10.1109/ICCS45141.2019.9065846.
- [10] A. Zahara, "ANALISIS PERBANDINGAN METODE SAW, MOORA, SMART DALAM PEMILIHAN CALON MITRA STATISTIK PADA BPS KABUPATEN SERDANG BEDAGAI SKRIPSI Diajukan untuk memenuhi syarat mencapai gelar Sarjana," Sumatra Utara, 2021.
- [11] M. Iskandar Nasution dan A. Fadlil, "PERBANDINGAN METODE SMART DAN MAUT UNTUK PEMILIHAN KARYAWAN PADA MERAPI ONLINE CORPORATION," vol. 8, no. 6, hlm. 1205–1214, 2021, doi: 10.25126/jtiik.202183583.
- [12] A. Elia, A. Fadilah, M. Fitriani, dan P. Suryani, "MALCOM: Indonesian Journal of Machine Learning and Computer Science Comparison of SMART, SMARTER AND TOPSIS Methods in Selection Location of Serba Murah Store Pulau Kijang Perbandingan Metode SMART, SMARTER dan TOPSIS dalam Pemilihan Lokasi Toko Serba Murah Pulau Kijang," vol. 1, hlm. 170–176, 2021.