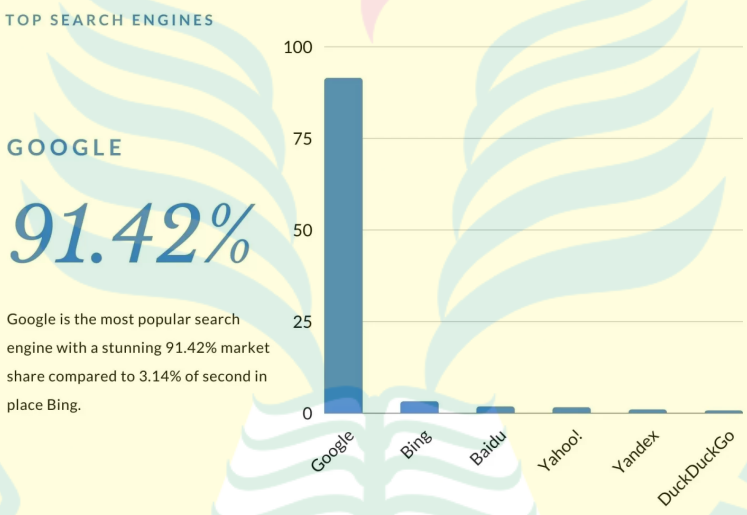
BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Saat ini, penggunaan *search engine* atau mesin pencari telah digunakan oleh khalayak umum untuk mencari berbagai informasi yang ada. Berdasarkan data dari (Christ, 2022), Google merupakan *search engine* yang menempati urutan pertama terpopuler, selanjutnya diikuti dengan Bing, Baidu, Yahoo!, Yandex, dan DuckDuckGo.

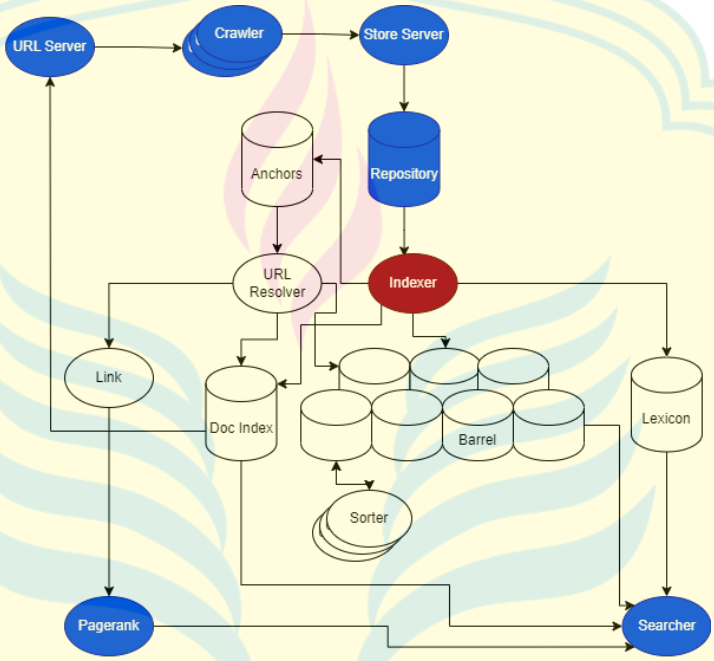


Gambar 1.1: Pengguaan search engine terpopuler (Christ, 2022)

Web Search Engine atau mesin pencari web merupakan suatu perangkat lunak yang digunakan untuk mencari sesuatu di internet berdasarkan kata-kata yang diberikan oleh pengguna sebagai *search terms*. Pembuatan *search engine* pertama kali dilakukan oleh Alan Emtage, Bill Heelan, dan J. Peter Deutsch pada tahun 1990. Mereka menamai *search engine* tersebut yaitu Archie (Seymour et al., 2011).

Pekerjaan utama dari *search engine* ada tiga yaitu *web crawling, indexing,* dan *searching*. *Search engine* bekerja dengan cara mengirimkan informasi tentang halaman web, Halaman tersebut di dapat dari *web crawler* suatu *automated web browser* yang mengikuti seluruh pranala yang ada di situs. Pengecualian situs yang

dicari dapat dilakukan melalui "*robots.txt*". Kemudian, konten dari setiap halaman akan dianalisis untuk menentukan urutan index. Data tentang halaman web dikirim ke dalam *index database* yang nantinya akan dilakukan *query*. *Query* bisa satu kata atau lebih. Tujuan pengindeksan adalah untuk menemukan informasi secepat mungkin (Seymour et al., 2011).



Gambar 1.2: High Level Google Architecture (Brin & Page, 1998)

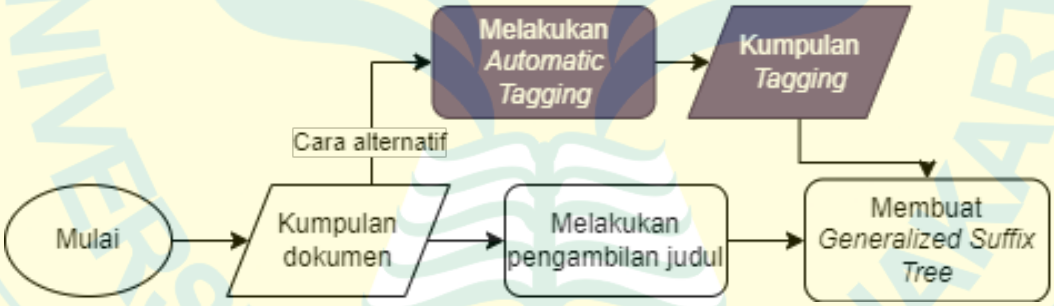
Pada Gambar 1.1, telah diperlihatkan bahwa Google merupakan *search engine* terfavorit. Google ditemukan oleh Larry Page dan Sergey Brin pada tahun 1998. Salah satu keunggulan Google adalah pengaplikasian PageRank yaitu mengatasi *underspecified queries*. Sebagai contohnya, jika kita mencari kata Real Madrid, maka situs pertama kali yang terlihat adalah situs resmi Real Madrid.

Gambar 1.2 merupakan struktur arsitektur Google. Warna biru menunjukkan hasil penelitian yang dilakukan oleh Lazuardy Khatulistiwa dalam penelitian yang berjudul "Perancangan Arsitektur Search Engine dengan Mengintegrasikan *Web Crawler, Algoritma Page Ranking*, dan *Document Ranking*" dan warna merah menunjukkan proses penelitian dari Zaidan Pratama dalam judul "Perancangan Modul Pengindeks pada *Search Engine* Berupa *Induced Generalized Suffix Tree* untuk Keperluan Perangkingan Dokumen". Salah satu komponen dalam *search engine* milik Google adalah *Indexer*. Dalam penelitian Zaidan Pratama, di sana

menjelaskan tentang melakukan pengindeksan melalui algoritma *General Suffix Tree* (*GST*) yang termodifikasi. (Pratama (2022))

Proses pengindeksan dimulai dengan memasukkan kumpulan dokumen yang dibuat menjadi *GST*. Kemudian, membuat *GST* dari kumpulan dokumen tersebut. Dari *GST* tersebut, nantinya akan dilakukan reduksi untuk node yang redundan atau node yang mengalami perulangan yang tidak diperlukan sehingga akan terbentuk pohon yang terinduksi untuk frekuensi f yang bernama *Induced Generalized Suffix Tree-f*.

IGST-f ini menjadi komponen utama dalam pengindeksan. Setelah itu, program menerima masukan berupa pola kata dan batas k untuk dicari pada kumpulan dokumen. Dari sinilah kita akan mencari nilai *count* dari setiap *node*. Kemudian, mencatat jumlah dokumen dalam *sublist* yang tereduksi untuk setiap *node*. Selanjutnya, mencari nilai *counter lowest common ancestor* dari setiap *node*. Terakhir, mengenai Indeks Efisien. Untuk *Top-k Document Retrieval Problem* dilakukan pengurutan terhadap representasi *array IGST-f* yang sudah memiliki nilai *count* dan mengembalikan hasil *top-k* yang meiliki pola *P*. Pratama (2022)



Gambar 1.3: Bagian *Automatic Tagging* pada *Indexer*

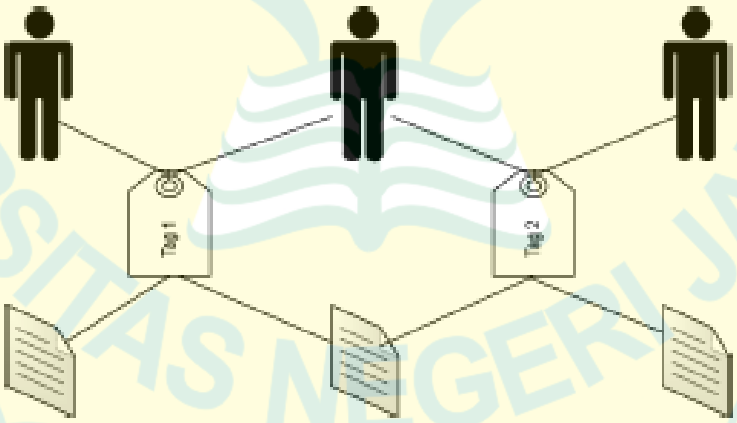
Akan tetapi, pengindeksan tersebut masih hanya melalui judul dan belum melalui *tag*. Oleh karena itu, salah satu alternatifnya adalah melakukan pengindeksan melalui *tag*. Pada gambar 1.3 Warna ungu adalah salah satu alternatif yang dapat dilakukan yaitu melalui *automatic tagging* yang nantinya peneliti akan lakukan. Hal ini dapat dimanfaatkan agar bisa melakukan pengindeksan lebih akurat.

Tagging merupakan hal yang biasa dilakukan untuk menggambarkan suatu kata kunci yang relevan atau frasa kunci pada suatu dokumen, gambar, atau video. Dalam merambatnya perkembangan Web 2.0 aplikasi seperti Del.icio.us dan Flickr, pelayanan *tagging* mulai populer dan menarik perhatian pihak akademis dan

industri. Penelitian tentang cara *automatic tag* membuahkan hasil. Cara melakukannya dengan algoritma *Poisson Mixture Model*. Dengan cara ini, kecepatan untuk membuat *automatic tag* bisa lebih cepat dibandingkan SimFusion dan VS+IG. Contohnya pada saat *Delicious Test Time*, *PMM* mampu menghasilkan 1,23 detik saat proses *automatic tag*, sedangkan SimFusion membutuhkan waktu 6,4 detik dan *VS+IG* membutuhkan waktu 77,43 detik. Selain kecepatan, *PMM* juga mampu di atas *SimFusion* serta *VS+IG* secara signifikan dalam hal akurasi, presisi, dan *recall*. (Song et al., 2008)

Selain itu, *tagging* juga digunakan untuk membantu pengorganisasian, *browsing*, dan pencarian. Seperti *image tagging* yang digunakan oleh Flickr, *web page tagging* yang digunakan oleh Del.ico.us, dan *social tagging* yang digunakan oleh Facebook, semua sistem tersebut menjadi populer dan dipergunakan di penjuru Web. (Sood, 2007)

Secara umum, sumber yang memiliki *tag* biasanya berasosiasi *tag* yang lain. Selain itu, sumber yang memiliki *tag* berasosiasi terhadap *user*. Sebagai contoh, *tagging* terhadap dokumen d yang dilakukan oleh *user* u dengan *tag* t dapat direpresentasikan sebagai tiga kesatuan (u, d, t). Dengan menggunakan pendekatan itu, dapat terbentuk suatu graf yang digambarkan sebagai berikut.



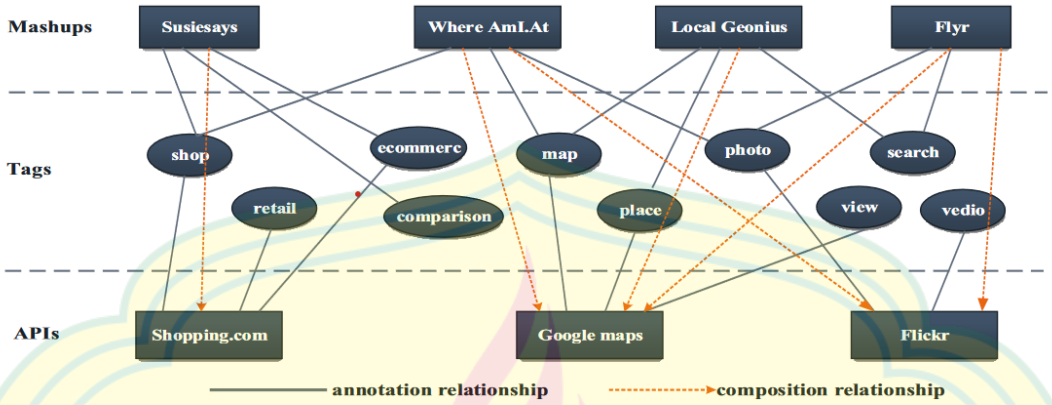
Gambar 1.4: Relasi antara user, tag, dan dokumen (Song et al., 2011)

Dengan relasi pada gambar 1.4 , rekomendasi *tag* dapat dilakukan dengan dua jenis menurut Song yaitu jenis pendekatan melalui pengguna dan jenis pendekatan melalui dokumen. Rekomendasi *tag* merupakan suatu sistem yang di mana sistem tersebut menampilkan *tag* yang relevan pada suatu dokumen agar pengguna bisa memperhitungkan apakah *tag* yang ditampilkan itu ingin dipakai atau

tidak. Melalui pendekatan *user*, sistem ini akan mengolah rekomendasi *tag* berdasarkan *tag-tag* yang telah dilakukan *user* sebelumnya dan merekomendasikan tag yang mirip dengan *user* ini atau kelompok dari *user* tersebut. Berbeda halnya dengan pendekatan dokumen, cara ini dilakukan dengan cara mengklasterisasikan dokumen-dokumen tersebut ke dalam topik-topik yang berbeda. Topik yang sama pada suatu dokumen akan memiliki *tag* yang diasumsikan lebih mirip dibandingkan dokumen yang berbeda topik. Namun, di antara kedua cara ini, yang dinilai kurang efektif adalah melalui pendekatan *user*. Pertama, berdasarkan penelitian dari Farooq et al. (2007), distribusi dari *user* vs *tag* mengikuti *long tail power law distribution*. Itu artinya, hanya sebagian kecil porsi dari *user* yang melakukan *tag* dengan panjang atau meluas. Sebagai tambahan, penggunaan *tag* yang berulang juga terbilang rendah, tetapi pertumbuhan perbendaharaan *tag* terus berkembang. Dengan sedikit pengguna relatif yang didapat, pendekatan *user* akan sulit untuk mencari model mana yang cocok buat untuk melakukan rekomendasi *tag* yang efektif. Berbanding terbalik dengan pendekatan dokumen yang lebih kokoh karena kekayaan informasi yang ada di dokumen. Bahkan, *tag* dan kata akan menciptakan relasi yang potensial antara topik dan konten di suatu dokumen yang di mana tag dianggap sebagai kelas label untuk dokumen dalam skenario *supervised learning* atau kesimpulan dari dokumen dalam skenario *unsupervised learning*. (Song et al., 2011)

Namun, percobaan ini hanya terbatas pada CiteULike dan del.icio.us. Untuk saat ini, kedua situs tersebut sudah tidak dapat diakses dengan semestinya. CiteULike beralih menjadi situs judi, sedangkan del.icio.us tidak dapat diakses oleh umum.

Beberapa tahun kemudian, suatu penelitian membahas mengenai *automatic mashup tag*. Secara sederhana, *mashup* adalah suatu *web service* yang di mana merupakan kumpulan dari kombinasi beberapa Web *API* dan konten dari berbagai sumber. Berbeda dengan rekomendasi *tag* yang menggunakan pendekatan dengan konten tekstual, di dalam *Web services* terdapat banyak sekali relasi seperti komposisi relasi antara *mashup* dengan *API* dan anotasi berelasi antara *API* dan *tag*. (Shi et al., 2016)



Gambar 1.5: Skema *Mashup*, *Tag*, dan *API* (Shi et al., 2016)

Selain teks, *tag* juga digunakan dalam hal yang bersifat non teks seperti video, musik, dan gambar. Dalam suatu video, *tag* sangat diperlukan untuk menentukan relevansi antara pencarian yang diinginkan dengan isi video. Meskipun beberapa platform video seperti Youtube menyediakan judul dan deskripsinya, bisa saja judul tersebut tidak ada keterkaitannya dengan video dan deskripsinya yang sangat panjang sehingga orang malas untuk membaca. Manfaat dalam *tag* video ada dua yaitu bisa menemukan daftar video yang representatif dan *tag* dapat mendukung untuk melakukan penemuan tentang video yang kontennya berhubungan dengan video yang telah ditonton. (Parra et al., 2018)

Untuk kasus *automatic tag* pada musik, *Automatic music tagging* adalah *multi label binary classification* yang bertujuan untuk memprediksi *tag* yang relevan pada suatu lagu. *Tag* tersebut membawa informasi musik semantik yang nantinya dapat digunakan untuk membuat aplikasi seperti rekomendasi musik. (Won et al., 2020)

Dengan demikian, peneliti ingin membuat penelitian terkait *automatic tagging* dengan menggunakan penelitian dari Song et al. (2008) yang terdapat dua algoritma utama yaitu *Bipartite Graph Partition* dan *Two Way Poisson Mixture Model*.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, maka perumusan masalah pada penelitian ini adalah "Bagaimana cara melakukan Automatic Tagging dengan menggunakan algoritma *Bipartite Graph Partition* dan *Two Way Poisson Mixture Model*?".

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini yaitu:

1. Data latih dan data uji yang digunakan berasal dari satu sumber *web*.
2. Untuk *crawling* data, akan digunakan *crawler* dari sistem yang telah dibuat oleh Khatulistiwa (2022) berjudul "Perancangan Arsitektur Search Engine dengan Mengintegrasikan Web Crawler, Algoritma Page Ranking, dan Document Ranking".
3. Banyaknya K yang digunakan adalah dua.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membuat program *automatic tagging* dengan menggunakan algoritma *Bipartite Graph Partition* dan *Two Way Poisson Mixture Model* sesuai penelitian Song et al. (2008).

1.5 Manfaat Penelitian

Dalam penelitian ini, manfaat yang bisa diperoleh yaitu:

1. Bagi Peneliti

   Menambah pengetahuan penulis tentang *automatic tagging* terhadap dokumen.

2. Bagi Peneliti Selanjutnya

   Diharapkan metode yang diusulkan pada penelitian ini dapat membantu penelitian selanjutnya dalam mengembangkan sistem yang lebih kompleks dan bermanfaat.