

Rancang Bangun Fitur Pencarian Kontak Tamu Yang Tidak Diketahui Pada Webinvito Design Dengan Web Crawling Social Media Facebook

Chandra Satria Nugraha¹, Citra Lestari²

Abstrak— Invitation atau undangan adalah salah satu media yang sering digunakan dalam pembentukan suatu acara yang bersifat formal maupun informal. Untuk dapat mengirim undangan yang tidak tercetak melalui email maupun social media, kita memerlukan kontak untuk dapat mengirimkan undangan tersebut. Masalah muncul ketika seorang ingin mengirim undangan kepada temannya ataupun orang lain namun tidak memiliki kontak email dari temannya tersebut. Salah satu kontak yang dapat digunakan selain email adalah Social media contohnya adalah Facebook. Untuk dapat berkomunikasi dengan teman online di Facebook, perlu relasi antara satu pengguna dengan teman – temannya yang disebut dengan friend. Selain itu pengguna Facebook akan memiliki email yang secara otomatis akan dibuat yaitu menggunakan ekstensi .Facebook.com. Rancang bangun fitur ini menggunakan Snoopy sebagai web crawler dan Facebook API sebagai penyedia data Facebook untuk mendapatkan data user. Fitur dibagi dalam lima tahap, mulai dari Query Input untuk memasukkan nama yang ingin dicari, Facebook User Search untuk mencari nama user dari Facebook API, Name Checking untuk mencocokkan nama dari teman Facebook user, Friend of Friend Crawling untuk mencari lebih lanjut teman dari Facebook user ke level berikutnya, dan Name Listing untuk mengurutkan nama dari Facebook user yang ditemukan. Pengujian dilakukan dua macam, pertama adalah pengujian page load untuk mengetahui apakah halaman tidak akan menjadi lambat ketika diakses oleh banyak pengguna, pengujian kedua adalah pengujian pencarian nama teman yang dilakukan untuk mengetahui faktor apa yang mempengaruhi waktu dalam pencarian. Pengujian page load memberikan hasil waktu diantara nol hingga dua detik. Pada pengujian pencarian teman ditemukan bahwa upload speed berpengaruh dalam pembuatan crawler, sedangkan download speed berpengaruh dalam kecepatan crawler melakukan friend of friend crawling. Dengan hasil pengujian, fitur pencarian teman dapat menerapkan web crawling untuk dapat melakukan pencarian kontak teman yang tidak diketahui.

Kata Kunci: Facebook, web crawling, aplikasi, friend of friend crawling, Snoopy, Facebook user, Tree, Breadth-First

Search.

Abstract— Invitation is one of many media that is often used in an event making that is both formal and informal. To be able to send a paperless invitation by email or social media, we need the contacts to be able to send the invitation.

A problem arises when a person doesn't have the contact of people he wants to invite. One of the contact that can be used besides email is social media which one of the example is Facebook. Social media that is often used by people is Facebook. To be able to communicate with the other friend on Facebook, a relation is needed between one user and the others that is called friend. Besides that, Facebook users will have an automatically generated email that have Facebook.com extension. This feature design uses Snoopy as its crawler and Facebook API as data source to get user data. Feature consisted in five steps, from Query Input to define names to search for, Facebook User Search to find username from Facebook API, Name Checking to check the name of the Facebook user's friend, Friend of Friend Crawling to find more about the Facebook user's friend of friends from the next level, and Name Listing to make a list consist of the found Facebook user. There are two types for the testing, first is page load testing to find out whether the page will be slow when accessed by many users, second is friend search testing to find out which factor affects in search time. Page load testing gives result in between zero and two seconds. In friend search testing, upload speed affect the declaration of the crawler, while download speed affect the crawler speed doing friend of friend crawling. With the result of testing, friend search can use web crawling to find unknown friend contact.

Keyword: Facebook, web crawling, application, friend of friend crawling, Snoopy, Facebook user, Tree, Breadth-First Search.

I. PENDAHULUAN

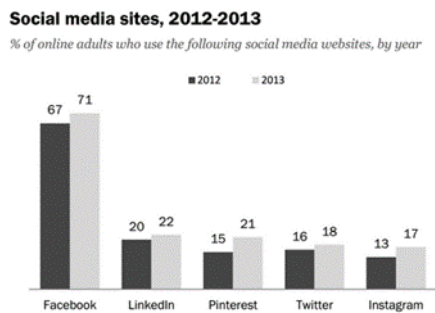
Perusahaan WebInvito Design adalah perusahaan yang bergerak pada bidang undangan online. Undangan online ini disebarkan kepada masyarakat melalui Internet. Berbeda dengan undangan yang menggunakan kertas, undangan online ini tidak dicetak namun langsung disebarkan melalui media online. Media utama yang digunakan pada pengiriman adalah email.

Hambatan yang sering terjadi pada pengiriman undangan online ini muncul ketika pengguna tidak mengetahui email dari orang yang hendak diundang. Hal ini berarti pengundang harus mengetahui email yang ingin diundang dahulu atau tidak dapat mengundang.

¹ Mahasiswa, Program Studi Teknik Informatika Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Surabaya 60291 (telp: 031-555 5555; fax: 031-876 54321; e-mail: csatria@student.ciputra.ac.id)

² Dosen, Program Studi Teknik Informatika Fakultas Industri Kreatif Universitas Ciputra, Jln. UC Town, Surabaya 60291 INDONESIA (telp: 031-745-1699; fax: 031-745-1698; e-mail: caecilia.citra@ciputra.ac.id)

Saat ini social media adalah sarana komunikasi yang umum digunakan. Social media memungkinkan orang – orang dapat berteman satu sama lain tanpa harus bertemu dengan orang secara langsung atau bertemu dengan teman lama. Social media yang banyak digunakan saat ini adalah Facebook. Menurut Pew Research Center pada Gambar 1.1, jumlah pengguna Facebook pada tahun 2013 mencapai 71% dari jumlah orang dewasa di dunia.



Gambar 1.1 Persentase Pengguna Social Media

Untuk dapat berkomunikasi dengan teman online di Facebook, perlu relasi antara satu pengguna dengan teman – temannya yang disebut dengan friend. Namun, tanpa menjadi temanpun, orang dapat mengirimkan pesan pada pengguna Facebook yang lain. Facebook memberikan fasilitas email pada setiap pengguna dengan akhiran @facebook.com. Ketika ada orang yang mengirimkan pada email yang dibuat oleh Facebook, maka email tersebut akan masuk ke email pengguna Facebook yang digunakan ketika dia melakukan login pada Facebook. Sehingga undangan dapat dikirimkan ke email pengguna Facebook dengan menggunakan @facebook.com asalkan diketahui siapa yang menggunakan akun Facebook tersebut.

Facebook menyediakan Facebook API yang berguna untuk mendapatkan data – data tentang penggunaannya. Di sisi lain, terdapat teknologi yang digunakan untuk mendapatkan data untuk dapat diolah yang bernama web crawling. Dengan teknologi Facebook API dan web crawling, maka sudah dapat dimungkinkan untuk mencari pengguna Facebook yang ingin dikirimkan undangan.

Batasan – batasan pada penelitian ini adalah social media yang digunakan adalah Facebook dengan dasar pencarian berupa friend dari pengguna Facebook, lalu akan dilakukan crawl untuk mengambil friend list dari friend. Kata kunci yang digunakan untuk melakukan pencarian adalah nama. PHP adalah bahasa pemrograman yang digunakan dengan web crawler menggunakan Snoopy. Pencarian bersifat online dan dibatasi agar hanya menjelajahi tiga level dari user pertama yang dicari.

Pencarian dibatasi sebanyak ini karena lebih dari level tiga, maka hubungan satu orang dengan orang yang lain akan menjadi terlalu jauh dan tidak bisa menjadi target untuk orang yang diundang.

Tujuan dari penelitian ini adalah merancang bangun modul untuk fitur pencarian tamu atau teman yang tidak diketahui pada Facebook pada WebInvito Design. Manfaat dari penelitian ini adalah untuk dapat mencari kontak Facebook yang tidak diketahui tanpa membuka Facebook secara langsung dan dapat mendapatkan kontak *email* dari hasil pencarian Facebook.

II. LANDASAN TEORI

A. Data & Web Mining

Data Mining adalah proses menganalisa data dari berbagai perspektif dan menyimpulkannya menjadi informasi – informasi yang dapat digunakan untuk meningkatkan pendapatan, menurunkan pengeluaran, atau keduanya. Data Mining adalah proses menganalisa data dari berbagai perspektif dan menyimpulkannya menjadi informasi – informasi yang dapat digunakan untuk meningkatkan pendapatan, menurunkan pengeluaran, atau keduanya.

Web Mining adalah salah satu cabang dari data mining, namun data yang ditelusuri ada di web. Tujuan dari web mining adalah mengumpulkan informasi dari penggalan struktur web hyperlinks. Salah satu contoh pemanfaatan web mining adalah penggunaan Google, yang menggunakan pengurutan ranking dari halaman web.

B. Web Crawling

Web crawling adalah aktivitas penelusuran halaman – halaman web yang dilakukan oleh program komputer yang disebut crawler. Crawler akan melihat suatu halaman web serta menelusuri semua link yang didapatkan dari halaman tersebut. (Najork: 2009.) Semua hasil yang ditelusuri oleh Web crawling akan disimpan pada suatu tempat penyimpanan yang disebut dengan Repository.

Hasil dari Web crawling adalah halaman yang diperiksa semua atribut – atributnya, mulai dari kata – kata seperti judul, header, dan meta tag. Setelah penelusuran halaman web selesai dilakukan, maka informasi yang didapatkan berguna seperti contoh search engine, di mana user akan memasukkan inputan yang ingin dicari oleh crawler, lalu akan mendapatkan list halaman yang paling berhubungan dengan inputan. Bentuk dari hasil pencarian inputan adalah daftar halaman web yang paling sesuai dengan pertanyaan beserta suatu ringkasan singkat yang terdapat judul dokumen dan sebagian dari teks. Gambar sebaiknya diletakkan pada

awal atau akhir halaman jika memungkinkan, dan sedekat mungkin dengan acuan pertama pada naskah.

Proses dari web crawling terbagi menjadi dua tahap yaitu *Queueing and Caching Documents* dan *Indexing Documents*. (Oracle: 2014.) Tahapan yang dilakukan dalam proses *Queueing and Caching Documents* adalah sebagai berikut:

1. Crawler akan dibentuk untuk melakukan crawling pada URL utama. URL utama akan masuk ke dalam queue sebagai penjelajahan pertama. Setelah terinisialisasi, URL akan menghasilkan seed untuk crawler dapat menjelajahi HTML dari URL

2. Crawler mengambil dokumen dari web, bentuk dari dokumen biasanya adalah file HTML yang berisi teks dan hyperlink atau link yang terdapat pada web. Jika bentuk dokumen bukan HTML, maka crawler akan mengubah dahulu dokumen menjadi HTML sebelum dilakukan proses caching.

3. Crawler mencari link dari hyperlink untuk dapat dimasukkan ke queue baru, jika terjadi duplikasi, dokumen tersebut tidak diambil.

4. Crawler akan memasukkan HTML ke dalam cache.

5. Crawler akan mengulangi langkah no 2 dari URL queue yang baru.

Proses *Indexing Documents* dilakukan setelah cache penuh, proses ini dilakukan dengan cara memasukkan semua atribut dari dokumen – dokumen ke dalam index. Konten dari dokumen – dokumen yang akan menjadi index dari kumpulan dokumen tersebut.

C. Snoopy

Snoopy adalah PHP yang bersifat open-source yang dapat mengambil konten dari suatu halaman web ataupun mengirimkan suatu form.

Fitur – fitur yang dimiliki dari Snoopy adalah sebagai berikut:

1. Dapat mengambil konten dari suatu halaman web dengan mudah
2. Memisahkan teks dari halaman web tanpa menyertakan HTML tag.
3. Mengambil link yang terdapat pada halaman web.
4. Dapat bekerja pada host yang bersifat proxy.
5. Dapat melewati autentikasi dari user dan password.

Snoopy bekerja dengan cara melakukan scrap terhadap seluruh elemen pada halaman yang diberikan kepada Snoopy untuk dijadikan patokan. Setelah seluruh elemen terambil, maka pada class Snoopy diproses dengan mengambil bagian – bagian yang diperlukan pada crawler untuk dijadikan suatu string yang dikembalikan kepada pengguna. Pada gambar 2.1 terdapat salah satu potongan program dengan dokumentasi berupa fungsi dari metode utama dari Snoopy, yaitu *fetch*.

```

Function:  fetch
Purpose:  fetch the contents of a web page
          (and possibly other protocols in the
          future like ftp, nntp, gopher, etc.)
Input:    $URI the location of the page to fetch
Output:   $this->results the output text from the fetch
function fetch($URI)

```

Gambar 2.1 Metode fetch pada Snoopy

Fetch adalah metode yang akan mengambil konten dari suatu webpage berupa teks yang membutuhkan parameter awal berupa lokasi dari webpage yang ingin diambil. Metode Fetch pada Snoopy memiliki beragam bentuk, mulai dari *Fetchlinks* yang berfungsi untuk mendapatkan link dari satu halaman yang diinginkan dalam bentuk array, *Fetchtext* yang mengambil teks dari satu halaman yang tidak termasuk dalam elemen HTML, dan *Fetchform* dimana seluruh form dari satu halaman web akan terambil. Penggunaan Snoopy dapat diterapkan pada halaman PHP dengan cara harus melakukan deklarasi terlebih dahulu pada crawler. Setelah deklarasi selesai, Snoopy dapat ditambahkan parameter tertentu untuk dapat masuk ke halaman web yang membutuhkan autentikasi. Snoopy akan dapat melakukan fetch dengan cara memanggil crawler yang ada lalu menggunakan parameter untuk memberitahu URL yang ingin diambil.

D. Facebook API

Facebook API adalah *Application Programming Interface* yang disediakan oleh Facebook untuk dapat mengambil data maupun menambah data sesuai dengan ketentuan yang dimiliki. API utama dari Facebook adalah *Graph API*, yang merupakan pengganti dari *Old Rest API*. *Graph API* memungkinkan untuk pengembang perangkat lunak untuk melakukan analisa maupun pengambilan data dan relasi data – data tersebut, seperti *friend list*, *post*, *likes*, dan berbagai fitur dari Facebook.

Graph API ini dapat diakses dengan cara pertama kali menjadi *Developer* di Facebook. *Graph API* ini dapat dimasukkan ke dalam salah satu aplikasi Facebook untuk mengetes terlebih dahulu bagaimana request yang dimasukkan dapat berjalan, aplikasi tersebut bernama *Graph API Explorer*. Query yang dimasukkan dapat juga dimasukkan ke dalam browser, karena *Graph API* berbasis HTTP, jadi dapat digunakan dalam berbagai bahasa yang membutuhkan HTTP. Contoh request untuk mengambil gambar pada Facebook adalah sebagai berikut.

```
graph.facebook.com/facebook/picture?redirect=f
alse
```

Pada contoh request tersebut, API akan bekerja untuk masuk ke dalam gambar yang dimiliki oleh Facebook, dalam kasus ini adalah yang diambil adalah gambar logo dari Facebook. Jika kata pada setelah

”graph.facebook.com” diganti dengan username yang lain, maka akan didapat foto dari profile username tersebut.

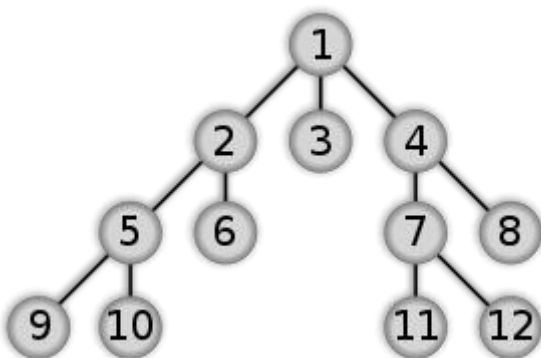
E. Tree

Tree adalah salah satu tipe dari struktur data yang menyerupai pohon, dengan memiliki *root node* sebagai *node* utama dan cabang untuk menjadi *node* turunan dari *root node* atau *node* lainnya. Sebuah *node* dapat merepresentasikan sebuah nilai tertentu dan memiliki sebuah *node parent*. Selain *root*, Tree juga memiliki satu bentuk *node* unik lain yang bernama *leaf*. *Leaf* adalah *node* yang tidak memiliki anak *node*. Tidak seperti Array atau Linked List yang merupakan struktur data yang linier, Tree adalah struktur data yang memiliki hirarki.

F. Breadth-First Search

Algoritma Breadth-First Search adalah algoritma dianggap salah satu yang paling simpel dalam melakukan pencarian pada suatu Graph ataupun Tree. Breadth-First Search menginspirasi beberapa algoritma terkenal seperti Minimum-Spanning-Tree milik dari Prim dan pencarian Shortest Path yang dimiliki oleh Dijkstra (Cormen: 2009). Keunggulan dari algoritma ini adalah dia dapat lebih cepat menemukan hasil pencarian untuk tree yang memiliki lebih dari dua cabang. Selain itu, hasil pencarian akan didapatkan pada level yang paling dekat dengan root dari Tree. Depth-First Search tidak digunakan pada aplikasi ini karena jika menggunakan Depth-First Search maka akan membuat penelusuran Tree harus dilakukan hingga level terdalam terlebih dahulu dan dapat mengakibatkan waktu yang dibutuhkan semakin lama.

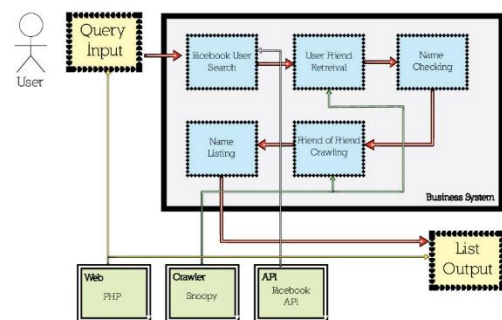
Pertama dari node utama 1, algoritma ini akan mencari dari semua node yang didapatkan dari 1, yaitu 2, 3, 4. Nomor 2, 3, dan 4 akan ditelusuri terlebih dahulu. Setelah selesai, maka anak dari node 2 akan ditelusuri, yaitu 5 dan 6. Setelah 5 dan 6 selesai ditelusuri, terlihat bahwa node nomor 3 tidak ditelusuri lebih lanjut karena tidak memiliki cabang berupa node, maka penelusuran berikutnya adalah pada anak node 4, yaitu 7 dan 8. Setelah 7 dan 8 selesai ditelusuri, maka anak dari node 5 akan ditelusuri. Terakhir anak dari node 7 akan ditelusuri. Gambar 2.2 menunjukkan alur dari Breadth-First Search.



Gambar 2.2 Alur Breadth-First Search

III. PENCARIAN KONTAK TAMU DENGAN WEB CRAWLING PADA FACEBOOK

Sistem bekerja pertama kali saat ada query dari user yang berupa nama dari pengguna Facebook yang ingin dicari yang disebut dengan Query Input. Query Input ditangkap dalam PHP, diteruskan dalam proses Crawling yang menggunakan Snooply sebagai web crawler dan Facebook API sebagai tempat crawler mendapatkan data user yang dicari. Setelah hasil ditemukan, maka akan dibentuk suatu list yang berisikan semua Facebook user yang dicari. List akan ditampilkan dalam webpage untuk pengguna dapat memilih Facebook user untuk mendapatkan email. Pada Gambar 3.1 terdapat sistem arsitektur yang digunakan dalam penelitian ini. Sistem dibuat terbagi menjadi enam tahap, mulai dari Query Input, Facebook User Search, User Friend Retrieval, Name Checking, Friend of Friend Crawling, dan Name Listing.



SYSTEM ARCHITECTURE - WEBINVITO DESIGN - FACEBOOK CONTACT RETRIEVAL

Gambar 3.1 Gambar Sistem Arsitektur

A. Query Input

Langkah pertama dari sistem ini adalah Query Input. Langkah ini adalah proses dimana user diminta untuk memasukan username yang dia miliki pada Facebook. Hal ini diperuntukan untuk mencocokkan nama yang dicari dengan Facebook dari username yang dimasukkan. Selanjutnya, diperlukan untuk memasukan nama dari orang yang dicari, berupa nama lengkap ataupun potongan nama pada field yang sudah disediakan. Setelah kedua field selesai dimasukkan, maka dapat dilakukan proses pencarian.

B. Facebook User Search

Pada proses ini dilakukan pencarian semua user Facebook yang memiliki nama yang sama dengan apa yang

diinputkan dengan user pada Query Input. Nama yang dimasukkan dihasilkan berupa Facebook User yang merupakan personal page, bukan group page maupun post dari orang – orang. Langkah pertama yang dilakukan adalah membuat crawler yang digunakan dalam mengambil text yang dihasilkan oleh Facebook API. Facebook API yang digunakan adalah Search API. Pada API, sistem mengambil data berupa user, gambar profil dan username dari nama yang dimasukkan. Pada gambar 3.2 ini tampak salah satu query yang digunakan dalam melakukan pencarian untuk user yang memiliki nama “john”. Pada query tersebut, nama dan ID dari user akan ditampilkan dalam bentuk JSON, lalu untuk pengambilan gambar, hanya perlu memasukkan query seperti berikut.

GET graph.facebook.com/john/picture?redirect=false

[/graph.facebook.com/search?q=john&type=user&access_token=CAAJKoiu6t00BAaEaTaZAZCX](https://graph.facebook.com/search?q=john&type=user&access_token=CAAJKoiu6t00BAaEaTaZAZCX)

Gambar 3.2 Contoh dari Query Facebook API

C. User Friend Retrieval

Proses ini adalah mengambil facebook friends yang dimiliki oleh Facebook user disimpan pada hasil fetch pada proses pencarian pertama. Pertama, untuk setiap hasil fetch diambil username yang digunakan crawler untuk mendapatkan facebook friends. Kedua, dari username tersebut, crawler memasuki halaman Facebook dari username lalu memasuki pada halaman Friends yang dimiliki oleh semua pengguna Facebook.

Crawler yang bertugas pada pengambilan facebook friends mendapatkan list dari teman dari user yang dimasuki berupa link. Link ini yang merupakan nama dan username dari pengguna Facebook yang masuk ke dalam halaman Friends. Di situlah crawler bekerja, tugas crawler adalah mengambil seluruh teman pada user tersebut. Tujuan dari pengambilan nama dan username ini adalah untuk mengetahui apakah user yang dimasuki oleh crawler ini memiliki hubungan pertemanan dengan user yang melakukan pencarian pada sistem.

D. Name Checking

Proses ketiga ini adalah proses dimana sistem melakukan pengecekan pada node yang didapatkan pada proses sebelumnya untuk mengetahui apakah node sama dengan user yang melakukan pencarian ini. Pertama, setelah crawler mendapatkan semua node dari proses sebelumnya, tree ditelusuri untuk mengecek apakah root dan node sesuai dengan user yang mencari.

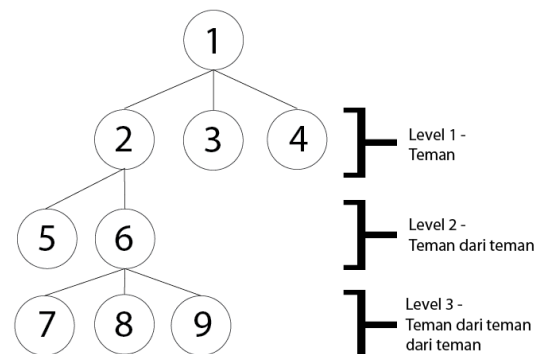
Ketika sesuai, maka tree tidak ditelusuri lebih lanjut, dan Facebook user yang menjadi root dari tree masuk ke kategori “teman” dari user yang mencari pada tingkatan level 1. Ketika tidak ada yang sesuai, maka dilanjutkan pada proses pencarian berikutnya yaitu Friend of Friend Crawling. Proses ini menggunakan Algoritma Pencarian Friend of Friend yang dibahas pada bagian selanjutnya. Algoritma dilakukan bersesuaian dengan proses Name Checking yang sama dengan sebelumnya.

Ketika kesesuaian antara node dan user yang melakukan pencarian ditemukan pada proses tepat setelah level 1 berakhir, maka Facebook user yang menjadi root dari tree masuk ke kategori teman dari user yang mencari pada tingkatan level 2. Pada pencarian Friend of Friend ini dilakukan hingga level 3, ketika pada saat level 3 masih belum ditemukan kesesuaian, maka pencarian dihentikan, dan root dari tree tersebut masuk ke kategori yang bernama “bukan teman”. Maka terdapat kategori “teman” dan “bukan teman” pada saat menyelesaikan proses pencarian ketiga dan keempat ini. Pada kategori yang dibentuk ini, kategori “teman” memiliki penanda level yang menunjukkan bahwa user ini ditemukan pada level beberapa.

E. Algoritma Pencarian Friend of Friend

Struktur data yang digunakan berupa Tree dengan metode pencarian Breadth-First Search. Satu Tree akan merepresentasikan satu Facebook user, dengan isi dari root node adalah username yang dimiliki oleh Facebook user. Cabang dari root node adalah node yang berisi username dari friend yang dimiliki oleh Facebook user tersebut. Dalam algoritma Pencarian Friend of Friend, cabang dari root node itulah yang ditelusuri untuk melakukan pencarian berikut. Setiap cabang ditelusuri untuk mendapatkan friend ke level berikutnya.

Berikut adalah ilustrasi dari algoritma pencarian yang tertera pada Gambar 3.3. Pencarian dimulai dari node no 2 dimana dari nomor 2 hingga 4 tidak ditemukan username dari pengguna. Node no 2 memiliki anak node 5, 6, dan 7 untuk ditelusuri. Setelah menelusuri node no 5, 6, dan 7, ketika masih belum menemukan username dari pengguna maka akan dilanjutkan ke anak node no 3. Pencarian ini akan dilakukan secara rekursif dengan mengganti patokan dari parent yang ditelusuri. Contohnya adalah ketika menelusuri no 2, 3, dan 4, maka username yang menjadi patokan adalah node no 1. Pada penelusuran node no 5, 6, dan 7, username yang menjadi patokan adalah node no 2. Pencarian ini akan berhenti ketika sudah mencapai anak dari root pada kedalaman tiga level. Node no 2, 3, dan 4 terhitung pada level 1, sedangkan node 5, 6, dan 7 terhitung pada level 2.



Gambar 3.3 Ilustrasi Pencarian

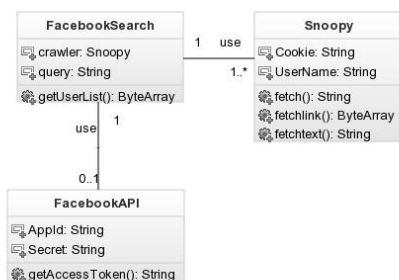
F. Name Listing

Proses pencarian yang terakhir adalah Name Listing. Pada proses ini dilakukan pengurutan yang dilakukan untuk mengetahui siapa user yang ditemukan yang paling dekat dengan user yang mencari teman pada sistem ini. Proses Name Listing dilakukan pada setelah mendapatkan kategori pada proses sebelumnya, lalu dibuat dalam bentuk list.

List terisi dengan nama yang menjadi teman yang paling dekat dengan user, lalu nama yang memiliki angka level pencarian yang paling rendah, lalu menuju ke angka yang lebih besar, hingga mencapai angka level 3 karena batas level maksimal adalah 3. Sedangkan untuk kategori “bukan teman” tidak mendapatkan angka level karena berarti pada user yang bersangkutan tidak ditemukan adanya user yang mencari.

IV. IMPLEMENTASI

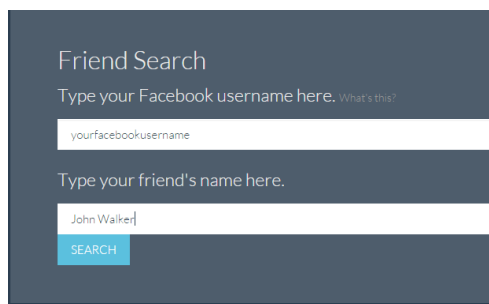
Pada gambar 4.1 terdapat class diagram untuk penelitian ini, terdiri dari satu FacebookSearch class yang berisikan algoritma dari pencarian. FacebookSearch class menggunakan Snoopy dan Facebook API untuk membantu dalam proses melakukan pencarian.



Gambar 4.1 Class Diagram

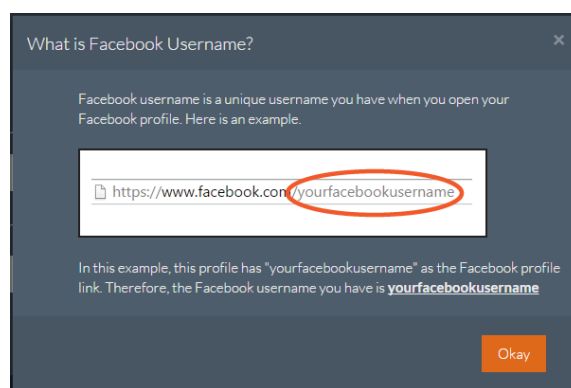
A. Query Input

Pada sistem bagian Query Input, hal pertama yang dilakukan oleh sistem adalah mengambil nilai pada pengetikan nilai field yang dituliskan oleh user pada tampilan utama. Pada gambar 4.2, ditunjukkan untuk salah satu contoh dari tampilan yang didapatkan setelah melakukan pengisian pada field yang disediakan. Pertama user diminta untuk mengisikan username yang dia miliki pada Facebook. Lalu setelah username sudah diisi, maka user perlu mengisi nama teman yang dicari.



Gambar 4.2 Tampilan Query Input

Ketika user tidak tahu mengenai apa yang dimaksud dengan *username* pada Facebook, maka user dapat menggunakan *link what's this* di sebelah kalimat *Type your Facebook Username here*. Setelah *link* ditekan, maka keluar tampilan seperti pada Gambar 4.3 yang menjelaskan apa itu *Facebook Username*.



Gambar 4.3 Tampilan Query Input untuk Help

Maka user mendapat penjelasan yang didapat untuk mengetahui apa yang dimaksud dengan *Facebook Username*. Setelah selesai memasukkan *Facebook Username* dengan nama teman yang ingin dicari, maka terjadi proses pencarian berikutnya, yaitu *Facebook User Search*.

B. Facebook User Search

Saat tombol Search ditekan, maka sistem mengecek apakah terjadi kesalahan pada input. Kesalahan yang dimaksud adalah apakah ketika ada yang dari kedua field yang kosong atau mengandung karakter yang ilegal. Setelah sistem memverifikasi kedua field sudah benar isi dan ketentuannya, maka dilanjutkan pada proses pencarian Facebook user yang sesuai dengan nama yang didapatkan pada field nama yang ingin dicari. Pada sistem dilakukan pelembaran isi dari kedua field ke FacebookSearch class yang bertugas untuk melakukan pencarian. Setelah kedua field didapatkan pada class tersebut, hal pertama yang dilakukan oleh sistem adalah membuat crawler yang digunakan untuk mengambil nilai

dari Facebook API yang berisi Facebook user yang dicari dengan nilai kedua field.

Crawler diberikan tiga parameter yang digunakan untuk memberi akses agar dapat memasuki Facebook API. Parameter yang pertama adalah cookieCrawler yang bertugas sebagai cookie yang dimiliki oleh crawler agar koneksi yang didapat pada crawler ketika melakukan crawling tidak terputus. Kedua parameter berikutnya yaitu emailCrawler dan keyCrawler adalah dua parameter yang digunakan untuk verifikasi masuk ke Facebook agar memberikan akses pada Facebook API itu sendiri. Setelah itu, maka dilakukan fetch ke Facebook API untuk dapat melakukan pengambilan data yang berupa siapa saja yang memiliki nama yang sesuai dengan input dari user. Setelah itu, hasil API diambil oleh crawler, lalu diproses untuk melakukan pengambilan nama lengkap, username, dan detail foto.

C. User Friend Retrieval

Pada proses pencarian ini maka diambil semua *Facebook friend* yang dimiliki oleh *Facebook user* dari hasil sebelumnya. Pada tampilan yang dilihat oleh *user* hanya berupa *loading page* yang menunjukkan untuk menunggu sampai *crawler* selesai melakukan pencarian. Pada gambar 4.4, berikut adalah tampilan untuk *user* ketika menunggu hasil dari *crawling* selesai.



Gambar 4.4 Tampilan *Loading Bar*

Crawler melakukan *fetch* ke dalam halaman *friend* yang dimiliki oleh setiap *Facebook user*, sehingga dapat mengumpulkan semua *friend* yang dimiliki oleh *Facebook user* tersebut. *Friend* yang didapat pada halaman tersebut adalah berupa *link* ke profil *Facebook user* masing-masing, maka *crawler* mendapatkan nama lengkap dan username untuk *user* tersebut.

Ketika semua *Facebook user* sudah diambil semua *friend*, maka sistem melanjutkan ke proses pencarian berikutnya, yaitu *Name Checking*. Setiap *Facebook user* yang diambil pada proses ini akan dibuat dalam bentuk *tree*.

D. Name Checking

Pada proses *Name Checking*, sistem mengecek dari data-data yang dibawa dari crawler apakah sesuai dengan

nama user yang mencari. Pertama, diambil list nama – nama teman yang dimiliki oleh seorang *Facebook user*. Dari list yang berisi teman milik *Facebook user*, maka username teman dari *Facebook user* dimasukkan ke dalam *tree* milik *Facebook user* yang ada.

Dari nama - nama itu dicek apakah user tersebut adalah teman dari *Facebook user* yang didapat dari crawling dengan cara mencocokkan username yang dimasukkan oleh user pertama kali dengan username yang dimiliki oleh teman dari *Facebook user* yang bersangkutan. Karena jika termasuk dalam golongan teman, maka *Facebook user* tersebut tidak masuk ke dalam proses *Friend of Friend Crawling* dan *Facebook User* tersebut masuk ke dalam tampilan teratas untuk pembuatan list hasil. Setelah ditemukan semua teman dari user, maka sisa *Facebook user* dimasukkan kedalam list yang dianggap belum diketahui, sehingga perlu untuk dimasukkan ke proses berikutnya, yaitu *Friend of Friend Crawling*. Pada proses inilah dilakukan pengecekan yang bersifat *friend of friends*.

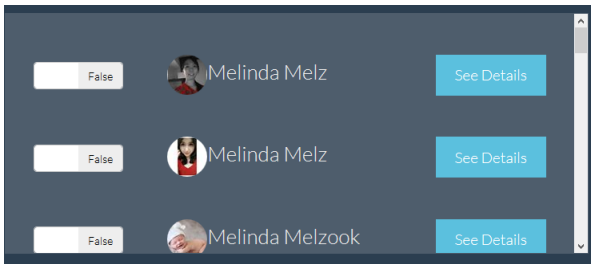
E. Friend of Friend Crawling

Proses pencarian inilah yang bertugas untuk mencari hubungan *friend of friend* yang dimiliki antara user dengan *Facebook user*. Dari *tree* yang dikepalai oleh *Facebook user* yang bukan teman dari user yang mencari, dilakukan proses crawling lebih lanjut ke halaman *friend* yang dimiliki oleh *friend* dari *Facebook user*, maka disebut dengan *Friend of Friend Crawling*. Hal yang dimiliki oleh hubungan *Friend of Friend* antara 1 user dengan user yang lain, adalah dimana mereka bukan *friend* di *Facebook*, namun memiliki *mutual friends*, sehingga dari situlah, kita dapat mencari apakah *Facebook user* yang sedang kita telusuri merupakan *Friend of Friend*.

Ketika pencarian *mutual friend* tidak mendapatkan jumlah atau tidak memiliki satupun, maka crawling dilanjutkan untuk *tree* tersebut secara *Breadth-First Search* hingga kedalaman *tree level 3*. Ketika pencarian *tree* sudah mencapai level maksimal, maka penelusuran *tree* pasti berhenti dan mengeluarkan hasil apakah *Facebook user* tersebut termasuk dalam *friend* yang ditemukan pada *tree level 1*, *friend of friend* yang ditemukan pada *tree level 2*, *friend of friend* yang ditemukan pada *tree level 3*, atau dia tidak memiliki hubungan sama sekali dengan *user* yang melakukan pencarian. Dari kriteria itulah, sistem mulai mengurutkan menjadi suatu *list* yang dapat dilihat oleh *user*.

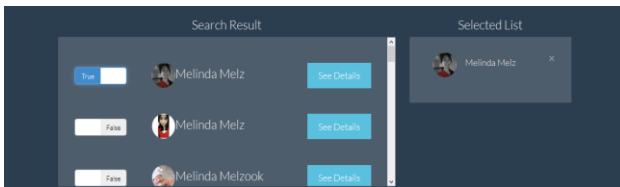
F. Name Lising

Proses pencarian ini adalah dimana sistem membawa hasil crawling yang telah dilakukan menjadi suatu list, yang dapat dilihat oleh user dan dapat dicek hubungan lengkapnya. Gambar 4.5 menunjukkan salah satu contoh list yang berhasil dicari dalam proses crawling.

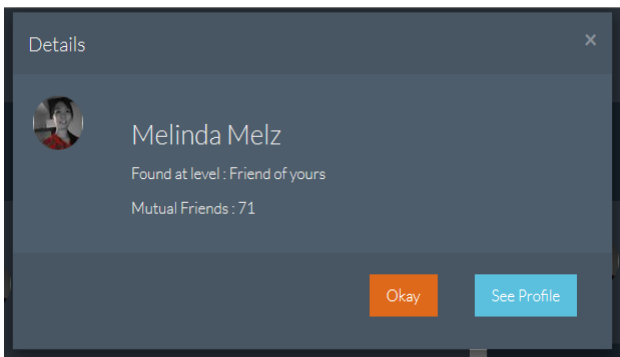


Gambar 4.5 List yang Dibuat Setelah Crawling

User dapat memilih salah satu atau lebih Facebook user yang telah dicarinya dengan menekan toggle button di sebelah kiri nama dari Facebook user atau dapat melihat detail dari Facebook tersebut dengan menekan tombol See Details. Gambar 4.6 menunjukkan contoh ketika memilih Facebook user, sedangkan Gambar 4.7 menunjukkan apakah yang terjadi ketika user menekan tombol See Details.



Gambar 4.6 List yang Dipilih Salah Satu Facebook User



Gambar 4.7 Contoh Tampilan See Details

Selain dapat melihat nama lengkap yang dimiliki oleh Facebook user, pada tampilan Gambar 4.7, kita dapat melihat bagaimana Facebook user ini memiliki hubungan pertemanan dengan user. Pertama, dapat dilihat bahwa Facebook user ini ditemukan pada level friend. Kedua, dapat dilihat jumlah mutual friend yang dimiliki antara user dengan Facebook user. Ketiga, user dapat melihat Facebook Profile dari tampilan ini, dengan menekan tombol “See Profile”. Setelah tombol tersebut ditekan, maka tertuju pada halaman Facebook yang dimiliki oleh Facebook user tersebut. Halaman Facebook yang terbuka

adalah halaman profil dari Facebook user yang telah dipilih.

V. PENGUJIAN

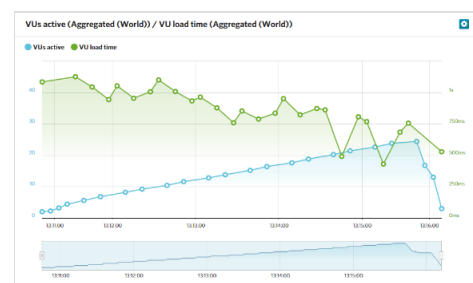
Penulis melakukan pengujian kepada sistem untuk melihat apakah sistem menjadi berat ketika ada request yang dibuat dengan jumlah yang besar. Tujuan berikutnya untuk pengujian sistem adalah untuk mengetahui apakah ketika melakukan pencarian di tempat yang memiliki koneksi yang berbeda, waktu yang dibutuhkan untuk menyelesaikan pencarian juga berbeda. Pengujian ini dibutuhkan karena perlunya untuk mengetahui apakah sistem dapat berjalan dengan lancar ketika sistem digunakan oleh orang banyak. Selain itu sistem perlu diuji apakah memiliki dependensi terhadap koneksi. Pengujian akurasi tidak perlu dilakukan, karena pada pencarian Facebook user, sistem menggunakan Facebook API yang mengambil langsung dari data Facebook. Pengujian dibagi menjadi dua tahap, pengujian Page Load dan pengujian Pencarian Teman.

A. Pengujian Page Load

Pada pengujian pertama ini, sistem diuji dengan adanya request dari virtual user. Pengujian ini dilakukan dengan cara menggunakan virtual user untuk melakukan request kepada sistem untuk menguji apakah ada kelambatan dalam melakukan proses. Request yang dilakukan berjumlah lebih dari 1000 dengan menggunakan Virtual User yang dibuat dengan menggunakan LoadImpact.com.

Virtual User yang dihasilkan oleh LoadImpact.com melakukan request pada sistem dan hasilnya akan tercatat. Jumlah dari request yang dibuat oleh LoadImpact.com adalah berjumlah 5310 request.

Pada Gambar 5.1, ditunjukkan hasil ketika LoadImpact.com selesai mengetes URL yang berisikan sistem tersebut. Pada grafik yang diberikan, maka terlihat pada VU load time yang berwarna hijau, sistem masih memberikan respon disekitar satu detik meskipun request yang dilakukan sebanyak 1314. Respon yang diberikan oleh sistem juga dibawah dua detik sehingga menunjukkan sistem tidak memerlukan waktu yang lama dalam memasuki sistem.



Gambar 5.1 Hasil Pengujian Page Load

Grafik ini juga menunjukkan bahwa selama pengetesan sistem, *Virtual user* tidak mengalami masalah dalam membuka sistem karena garis biru yang dibuat adalah garis lurus, garis lurus menunjukkan bahwa tidak ada *Virtual user* yang menjadi tidak aktif atau gagal dalam membuka sistem. Respon yang dianggap pada batas wajar untuk *user* tidak mengalihkan perhatian pada *website* yang diunduh adalah satu hingga dua detik (Nielsen: 1993).

B. Pengujian Pencarian Teman

Pada pengujian ini, dilakukan pencarian pada sistem berupa memasukkan nama yang diuji sebanyak sepuluh nama berbeda yang dicari dengan sistem, lalu dicatat waktu sistem bekerja dalam melakukan pencarian. Setelah itu pengujian dilakukan pada kondisi koneksi yang berbeda dan dengan sepuluh nama lagi. Hasil yang dicatat adalah nama yang dilakukan untuk pencarian, bagaimana kondisi koneksi yang digunakan, waktu yang diperlukan untuk melakukan pencarian. Waktu yang diperlukan untuk melakukan pencarian sendiri dibagi lagi menjadi lima,

1. Waktu yang dicatat pertama adalah ketika sistem mulai melakukan pencarian
2. Waktu kedua adalah ketika sistem sudah selesai dalam membuat crawler dan memasukan parameter ke dalam crawler tersebut.
3. Waktu ketiga yang dicatat adalah ketika sistem sudah melakukan pencarian friend pada level 1.
4. Waktu keempat adalah ketika sistem sudah melakukan pencarian friend of friend untuk level 2
5. Waktu kelima adalah ketika sistem sudah selesai melakukan pencarian friend of friend untuk level 3.

Pengujian pertama dilakukan pada kondisi koneksi seperti berikut, Ping sebesar 57 ms, Download Speed sebesar 0.58 Mbps, dan Upload Speed sebesar 1.57 Mbps. Pengujian kedua dilakukan pada kondisi koneksi seperti berikut, Ping sebesar 59 ms, Download Speed sebesar 3.96 Mbps, dan Upload Speed sebesar 0.46 Mbps. Sebagai pengujian terakhir, pengujian ketiga dilakukan pada kondisi koneksi seperti berikut, Ping sebesar 35 ms, Download Speed sebesar 4.15 Mbps, dan Upload Speed sebesar 1.90 Mbps.

Setelah ketiga pengujian selesai, maka dibuat penggabungan dari hasil rata – rata yang telah dibuat yang tergambar pada Tabel 5.1.

TABEL 5.1
TABEL RATA – RATA PENGUJIAN

	Ping	Upl oad	Downl oad	I	II	III	IV	V
1	57	1.57	0.58	0	15.2	18.1	166.6	316.2
2	59	0.46	3.96	0	17.5	20.1	74.2	97.6
3	35	1.90	4.15	0	14.2	15.3	51.9	77.8

Dari ketiga hasil pengujian di atas, maka yang paling berperan pada sistem untuk membuat crawler adalah upload speed dari suatu sistem. Pernyataan ini didapatkan ketika proses II di ketiga pengujian, pada pengujian nomor dua yang memiliki rata – rata waktu paling rendah, yaitu 17.5 detik, berselisih 2 detik dengan pengujian pertama, dan berselisih 3 detik untuk pengujian ketiga. Dalam melakukan friend of friend crawling, koneksi pada hal download speed yang paling berperan, karena pada proses dari III ke V, pengujian nomor satu yang memiliki rata – rata waktu yang paling lama untuk melakukan friend of friend crawling. Hal ini disebabkan karena halaman yang diambil oleh crawler, berjumlah sangat banyak, sehingga sangat bergantung kepada download speed dari suatu koneksi.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan bab - bab sebelumnya, penelitian ini dapat menerapkan web crawling untuk dapat mencari friend dari friend list dari suatu pengguna Facebook yang tidak diketahui. Sistem yang baik adalah sistem yang memiliki page load diantara 0 hingga 2 detik (Nielsen: 1993). Sistem yang telah dibangun oleh penulis telah diuji dan mendapatkan hasil pengujian untuk page load yang baik yaitu diantara 1 hingga 2 detik. Download speed mempengaruhi kecepatan crawling menelusuri level 2 dan 3 karena pada pengujian yang menggunakan download speed 4.15 Mbps, waktu untuk crawling adalah yang paling cepat, yaitu 62 detik dari crawling level 1 hingga level 3. Upload speed akan mempengaruhi kecepatan terbentuknya crawler, karena pada pengujian yang menggunakan upload speed 1.90 Mbps, crawler paling cepat dibentuk oleh sistem selama 14.2 detik.

B. Saran

Saran yang dianjurkan untuk penelitian ini adalah mengembangkan program crawler yang digunakan karena dalam melakukan crawling dari level 2 ke 3 membutuhkan waktu yang cukup signifikan jika menggunakan Snoopy.

DAFTAR PUSTAKA

- Acquisti (2006). *Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook*. Pittsburgh.
- Average Cost Of Wedding Invitations/ 2013*. (n.d.). Retrieved May 13, 2014, from

- <http://www.weddingstats.org/average-cost-of-wedding-invitations.html>
- Beaird, J. (2008). *The principles of beautiful web design*. Collingwood: Sitepoint.
- Cooley, R. (n.d.). *Web Mining: Information and Pattern Discovery on the World Wide Web*.
- Data Mining: What is Data Mining?* (n.d.). Retrieved from <http://www.anderson.ucla.edu/faculty/jason.frand/techer/technologies/palace/datamining.htm>
- Ellison, B., Steinfield, & Lampe (2007). *The Benefits of Facebook "Friends: Social Capital and College Students' Use of Online Social Network Sites*.
- Gross, R., & Acquisti, A. (2005). *Information revelation and privacy in online social networks*.
- Hartanti, A. A. (2007). *Perilaku Konsumen*.
- Introduction to Facebook APIs*. (n.d.). Retrieved from <http://www.ibm.com/developerworks/library/x-androidFacebookapi/>
- Jubille Enterprise (2009). *Facebook Untuk UKM (Usaha Kecil dan Menengah)*. Elex Media Komputindo: Jakarta.
- Julia, R. A. (2012). *Pemanfaatan Situs Jejaring Sosial Facebook Sebagai Media Pembelajaran pada Mata Pelajaran Teknologi Informasi dan Komunikasi (TIK) Bagi Siswa Kelas XI SMAN 1 Depok Sleman Yogyakarta*.
- Knuth, D. E. (1997). *The art of computer programming: Volume 1*. Reading (Mass.: Addison-Wesley.
- Mangold, W. G., & Faulds, D. J. (2009). *Social media: The new hybrid element of the promotion mix*. Indiana: Kelley School of Business.
- Najork, Marc. (2009). *Web Crawler Architecture*.
- Nielsen, Jakob. (1993). *Usability Engineering*.
- Pusat Pembinaan dan Pengembangan Bahasa., Indonesia, & Balai Pustaka, P. (1991). *Kamus besar bahasa Indonesia*. Jakarta: Departemen Pendidikan dan Kebudayaan.
- Rohani, L. (2007). *Perilaku Masyarakat Dalam Pengelolaan Sampah di Desa Medan Senembah Kabupaten Deliserdang dan di Kelurahan Asam Kumbang Kota Medan Tahun 2007*.
- Skinner, B. F. (1966). *The behavior of organisms*. Englewood Cliffs, NJ: Prentice-Hall.
- Understanding the Crawling Process*. (n.d.). Retrieved from http://docs.oracle.com/cd/E21698_01/admin.1122/e21605/crawler006.htm
- V. Patel (2013). *Distributed High Performance WEB CRAWLERS*.