

Rancang Bangun Agregator Toko Aplikasi Mobile Berbasis Web untuk Menyediakan Informasi Kompatibilitas Aplikasi Multi Platform

Pri Rezki Destrianto, Rully Agus Hendrawan, dan Andre Parvian Aristio

Departemen Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: eraha@is.its.ac.id

Abstrak—Layanan konten digital berkembang seiring meningkatnya jumlah aplikasi yang melimpah pada layanan konten digital. Para pengguna aplikasi menginginkan cara yang efektif dan efisien dalam melakukan evaluasi berbagai aplikasi yang tersedia dengan cepat dan sesuai dengan kebutuhannya masing-masing. Namun saat ini, tidak jarang pengguna aplikasi yang mengalami kesulitan memilih layanan konten yang tepat dalam mengevaluasi aplikasi tertentu. Sehingga pengguna memerlukan usaha lebih dalam menemukan informasi aplikasi yang ingin dievaluasi tersebut. Selain itu, keterbatasan media publikasi karya perangkat lunak saat ini, membatasi eksistensi dan penghargaan bagi Indie Developer. Oleh karena itu dibutuhkan direktori perangkat lunak yang dapat menyajikan daftar perangkat lunak dari berbagai sumber layanan konten digital secara terintegrasi untuk memudahkan pengguna dalam mengevaluasi dan melakukan pencarian perangkat lunak secara efektif dan efisien. Teknik yang digunakan dalam penelitian ini yaitu dengan menggunakan agregasi detail informasi produk dari banyak toko aplikasi mobile yang multi platform kedalam satu media. Luaran yang ada dalam penelitian ini berupa sebuah prototype aplikasi berbasis website berbasis PHP/HTML yang mampu menyajikan software direktori aplikasi multi platform yang terstruktur dengan metode prototyping. Hasil yang diharapkan dalam penelitian ini adalah untuk membantu pengguna untuk mengevaluasi dan mendapatkan informasi aplikasi multi platform secara detail dan lengkap serta dapat menjadi bahan pertimbangan dalam pemilihan device di masa depan. Hasil yang didapat adalah fungsi levenshtein mencapai keakuratan 90.7% untuk mencari similaritas. Angka ini tidak cukup mencapai titik keakuratan yang dibutuhkan, sedangkan penggabungan antara fungsi levenshtein dan fungsi similar text mencapai keakuratan 94.5% sudah mencapai tingkat keakuratan dalam mencari persamaan aplikasi di play store dan app store. Sehingga, metode gabungan dua fungsi yang diimplementasikan ke aplikasi. Aplikasi yang dibuat diharapkan dapat membantu pengguna untuk mengevaluasi dan mendapatkan informasi aplikasi multi platform secara detail dan lengkap serta dapat menjadi bahan pertimbangan dalam pemilihan device di masa depan.

Kata Kunci— *Product Aggregation, Software Directory, Software Development.*

I. PENDAHULUAN

TEKNOLOGI internet di masa modern ini semakin meningkat penggunaannya. Sampai saat ini, penggunaan internet mulai beralih kearah mobile phone. Pada tahun 2016, jumlah pengguna smartphone di dunia tercatat hingga 2.1 miliar [1]. Hal ini menyebabkan akses ke layanan konten digital seperti play store dan app store meningkat. Play store

dan app store adalah jenis platform distribusi digital untuk mobile dan aplikasi desktop. App store terus tumbuh dengan tingkat pertumbuhan yang tinggi dengan ribuan pengembang, ribuan aplikasi, dan keuntungan jutaan dollar yang didapatkan [2]. Google play sebagai sebagai app store terbesar telah memiliki aplikasi yang tersedia sebanyak 2.6 juta aplikasi di 2016 [3]. Google play store pada saat ini telah mencatat perhitungan kumulatif aplikasi yang sudah di unduh sebanyak 65 miliar kali dari Agustus 2010 hingga penghujung akhir Mei 2016 [4]

Seiring bertambahnya jumlah aplikasi yang melimpah pada layanan konten digital, proses pencarian aplikasi yang spesifik akan semakin sulit. Para pengguna aplikasi menginginkan cara yang efektif dan efisien dalam melakukan evaluasi berbagai aplikasi yang tersedia dengan cepat dan sesuai dengan kebutuhannya masing-masing. Namun saat ini, tidak jarang pengguna aplikasi yang mengalami kesulitan memilih layanan konten yang tepat dalam mengevaluasi aplikasi tertentu. Sehingga pengguna memerlukan usaha lebih dalam menemukan informasi aplikasi yang ingin dievaluasi tersebut.

Selain itu, banyaknya aplikasi yang tersedia di berbagai layanan konten digital, tidak secara langsung memberikan dukungan bagi indie developer dalam mempublikasikan karyanya. Hal ini membuat para developer harus mampu mengembangkan produk-produknya dengan dana seadanya disertai publikasi sendiri. Publikasi mandiri yang dilakukan oleh para developer tentu tidak dapat dinikmati oleh berbagai kalangan masyarakat secara luas karena tidak terintegrasinya informasi aplikasi dalam suatu wadah yang dapat dinikmati banyak orang. Media ini dibutuhkan untuk memfasilitasi kedua belah pihak antara pengguna aplikasi dan developer.

Untuk menangani permasalahan tersebut, penelitian ini akan berfokus pada pengembangan sebuah prototype direktori software untuk melakukan penyajian katalog software kepada pengguna perangkat lunak dengan agregasi produk dari berbagai layanan konten digital seperti play store dan app store. Software direktori adalah media yang mengumpulkan berbagai konten perangkat lunak secara terintegrasi dari google play store dan app store. Metode yang dilakukan dalam prototype aplikasi ini yaitu prototyping. Prototyping adalah metodologi pengembangan perangkat lunak di mana model sistem dikembangkan dan dievaluasi semakin banyak digunakan untuk membantu menentukan persyaratan sistem dan kemampuan fungsional dengan cepat dan akurat. Metode prototyping digunakan karena

fleksibilitas dalam pembangunan dari desain aplikasi yang akan dibangun.

Dengan adanya prototype ini diharapkan dapat mempermudah dalam mendapatkan informasi aplikasi multi platform secara detail dan lengkap serta dapat menjadi bahan pertimbangan dalam pembelian device di masa yang akan datang. Serta bagi developer dapat membantu mempublikasikan produknya dan dapat memasarkan produknya.

II. TINJAUAN PUSTAKA

Bab ini akan menjelaskan mengenai dasar teori yang dijadikan acuan atau landasan dalam pengerjaan jurnal ini. Landasan teori akan memberikan gambaran secara umum dari landasan penjabaran studi ini.

A. Web Scraping

Web scraping adalah mengambil konten website dari sebuah situs secara otomatis tanpa harus menyalinnya secara manual untuk kemudian dipilah-pilah sesuai dengan informasi yang dibutuhkan. Cara yang bisa digunakan adalah dengan meng-copy paste sebuah website yang dilakukan seseorang secara manual, atau dengan cara otomatis yaitu memarsing seluruh isi website dengan tools atau aplikasi [5]. Pada penelitian ini, tujuan dari *web scraping* adalah untuk mencari informasi tertentu dan kemudian mengumpulkannya kedalam format yang diinginkan.

B. Levenshtein Distance

Levenshtein Distance (LD) merupakan sebuah ukuran dari kemiripan atau kecocokan antar dua String, di mana kita akan mengacu sebagai sumber atau disebut String (s) dan target atau disebut String t [6]. Distance yang dimaksud adalah jumlah dari penghapusan (deletion), penambahan (insertions) atau penggantian (substitution) yang dibutuhkan untuk merubah String s menjadi String t [6]. Contohnya adalah sebagai berikut:

- Jika s adalah "test" dan t adalah "test", maka $LD(s,t) = 0$, karena tidak ada transformasi yang dibutuhkan, karena $s = t$.
- Jika s adalah "test" dan t adalah "tent", maka $LD(s,t) = 1$, karena dibutuhkan 1 substitusi untuk membuat $s = t$, sehingga cost dari Levenshtein adalah 1.

Secara umum, algoritma dari Levenshtein Distance sendiri berdasarkan [6] dapat dijelaskan dari tabel berikut:

Tabel 1.
Pseudocode Levenshtein Distance

	Description
1	Set n to be the length of s. Set m to be the length of t. If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..m rows and 0..n columns.
2	Initialize the first row to 0..n. Initialize the first column to 0..m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 1.
6	Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately above plus 1: $d[i-1,j] + 1$. b. The cell immediately to the left plus 1: $d[i,j-1] + 1$. c. The cell diagonally above and to the left plus the cost: $d[i-1,j-1] + cost$.

7 After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell $d[n,m]$.

III. METODOLOGI PENELITIAN

Metodologi merupakan sebuah tahapan dalam penyelesaian permasalahan pada studi ini. Metodologi digunakan sebagai panduan pengerjaan tugas akhir



Gambar 1. Metodologi Penelitian

A. Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan cara melakukan kuesioner kepada 30 orang yang memiliki preferensi untuk mengevaluasi perangkat lunak yaitu orang yang memiliki kedua konten digital baik *app store* dan juga *google play* sehingga pernah memiliki pengalaman dalam mengevaluasi perangkat lunak yang ada pada kedua layanan konten digital tersebut. Selain itu, wawancara dilakukan kepada 2 orang indie developer untuk mengetahui kebutuhannya terhadap system terkait publikasi karya berupa perangkat lunak dalam suatu media. Hasil wawancara nantinya akan digunakan sebagai bahan untuk menentukan kebutuhan fungsional sistem yang akan dirancang agar sesuai dengan kebutuhan pengguna.

B. Pengambilan Data

Setelah melakukan analisis kebutuhan, lalu dilakukan pengambilan data yang dilakukan dengan cara *scraping*. Data yang diambil berasal dari *google play* dan *app store*. Data yang didapatkan meliputi nama aplikasi dan detail deskripsi aplikasi yang ada dalam layanan konten digital. Pengumpulan data dilakukan dengan memanfaatkan *Application Programming Interface (API)* yang ada berasal dari penelitian yang dilakukan oleh pihak laboratorium *system enterprise*. Data yang telah dikumpulkan disimpan dalam database system.

C. Pengujian Prototype

Setelah menyelesaikan pengambilan data, maka tahap selanjutnya adalah pembuatan prototype website agregasi produk dengan fungsi yang spesifik terhadap direktori aplikasi (produk). Tujuan utama dari proses ini adalah untuk memberikan pengalaman terbaik dalam melakukan evaluasi produk yang mudah dan efisien melalui pembuatan aplikasi web berbasis PHP/HTML yang menyajikan software direktori secara terstruktur. Prototype dibuat dengan membuat draft antar muka pengguna yang merepresentasikan berbagai fungsi yang telah direncanakan pada tahap selanjutnya. Luaran dari tahapan perancangan prototype adalah mockup web aplikasi berbasis PHP/HTML yang mampu menyajikan software direktori yang terstruktur kepada pengguna perangkat lunak. Langkah selanjutnya dilakukan analisis proses bisnis dengan cara membuat desain UML diagram yaitu use case diagram dan class diagram.

D. Pengujian Aplikasi

Pada tahapan ini aplikasi akan diuji usability desain, luaran, dan fiturnya. Kategori pengujian yang dilakukan yaitu pengujian usability menggunakan *SUS matrix*. Pengujian

usability dilakukan kepada 10 orang responden. Parameter pengujian dilakukan dengan membuat parameter ketercapaian. Apabila dari hasil pengujian didapatkan nilai dibawah dari ketentuan maka desain dan fitur diubah sesuai dengan harapan pengguna, yang dalam konteks studi ini akan menjadi bahan evaluasi di masa yang akan datang.

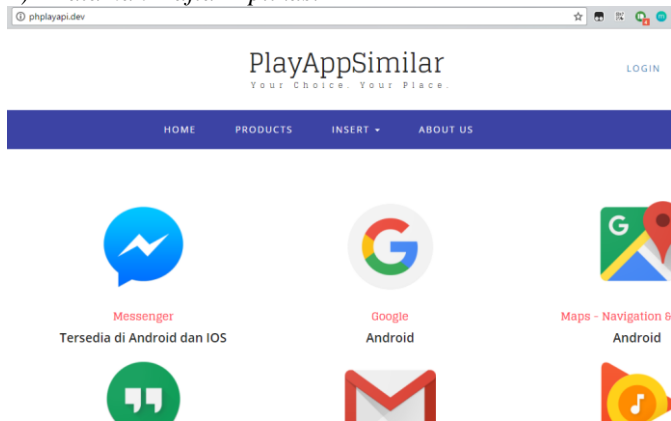
IV. HASIL DAN PEMBAHASAN

Pada tahapan ini berisi hasil implementasi penelitian yang telah dilaksanakan dan hasil pengujian prototype yang dibuat berdasarkan fitur yang ada dalam aplikasi.

A. Hasil Prototype Aplikasi

Hasil prototype aplikasi berupa antar muka pengguna yang siap dipakai dengan kapabilitas terbatas hanya pada menampilkan daftar aplikasi, melihat detail produk yang dipilih oleh pengguna, memasukkan data aplikasi yang dibuat oleh developer serta login dan register untuk user. Terdapat enam tampilan antar muka antara lain halaman daftar aplikasi, halaman detail aplikasi, halaman masukan data aplikasi berdasarkan appId, Halaman masukan data secara rinci, halaman login, dan alaman registrasi Berikut adalah penjelasan mengenai masing-masing fitur halaman.

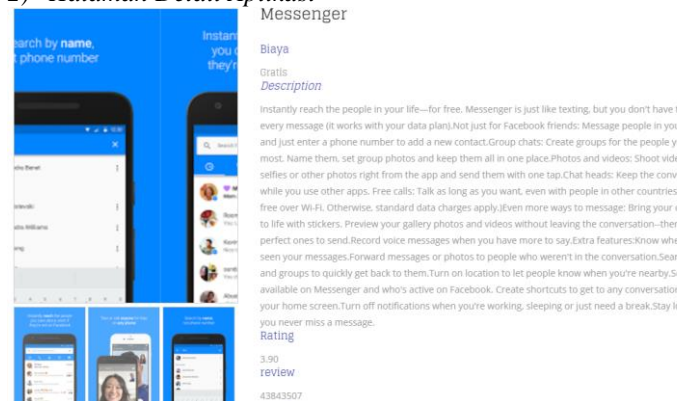
1) Halaman Daftar Aplikasi



Gambar 2. Halaman Daftar Aplikasi

Halaman pada gambar 2 adalah tampilan awal dari playappsimilar, nama prototype aplikasi untuk menampilkan daftar aplikasi dari hasil scraping yang berhasil diambil. Pengguna bisa mengetahui aplikasi tersebut tersedia di play store dan app store maupun play store saja atau sebaliknya berasal dari indikator yang ada dibawah judul.

2) Halaman Detail Aplikasi



Gambar 3. Halaman Detail Aplikasi

Halaman pada gambar 3 adalah tampilan untuk menampilkan informasi aplikasi secara rinci. Pengguna bisa mengetahui informasi aplikasi secara lengkap dan disediakan screenshot aplikasi dan hyperlink yang menuju toko aplikasi yang tersedia.

3) Halaman Ulasan Aplikasi

Halaman pada gambar 4 adalah tampilan untuk menampilkan ulasan aplikasi dari aplikasi yang dipilih. Pengguna bisa mengetahui berbagai ulasan aplikasi secara lengkap.



Gambar 4. Halaman Detail Aplikasi

4) Halaman Masukan Data Secara Rinci



Gambar 5. Halaman masukan data secara rinci

Halaman pada gambar 5 adalah tampilan untuk memasukkan data aplikasi secara rinci. Pengguna yang belum pernah memasukkan karyanya ke play store dan app store dapat memasukkan aplikasi yang dibuat ke formulir di halaman ini dengan mengisi data-data yang ada di formulir tersebut.

5) Halaman Masukan Data Berdasarkan Appid

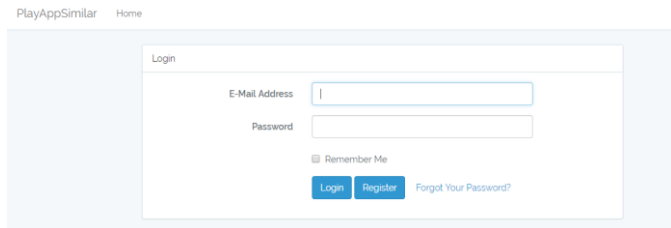


Gambar 6. Halaman masukan data berdasarkan appId

Halaman pada gambar 6 adalah tampilan untuk memasukkan appId aplikasi yang tersedia di play store maupun app store. Pengguna yang mempunyai karya berupa aplikasi dan sudah diunggah ke play store atau app store dapat memasukkan appId aplikasi dan sistem secara otomatis akan melakukan scraping ke

http://gooplayapi.herokuapp.com/app dan
 http://applestoreapi.herokuapp.com/app sesuai appId.

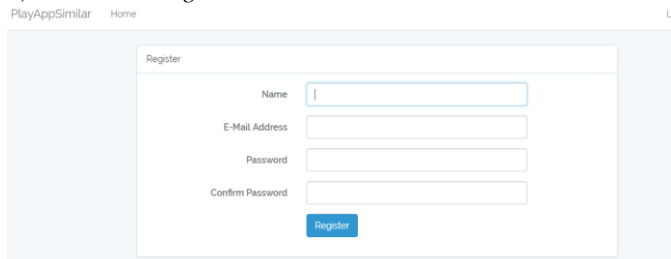
6) Halaman login



Gambar 7. Halaman login

Halaman pada gambar 7 adalah tampilan untuk masuk akun ke prototype aplikasi yang didalamnya digunakan untuk memasukkan data aplikasi. Pengguna memasukkan email address yang dimasukkan saat registrasi dan password.

7) Halaman Registrasi



Gambar 8. Halaman registrasi

Halaman pada gambar 8 adalah tampilan untuk registrasi akun untuk masuk ke dalam prototype aplikasi. Pengguna dapat mengisi formulir yang disediakan, yaitu nama, alamat email, password, dan konfirmasi password.

B. Perbandingan Antara Levenshtein Dengan Gabungan Levenshtein dan Similar Text

Dalam menentukan perbandingan, mula-mula ditentukan jumlah sampel data yang diperlukan. Cara menentukannya menggunakan metode slovin dengan toleransi kesalahan 5% dan tingkat presisi 95%. Dalam data ini menggunakan jumlah populasi 2500 data.

$$n = \frac{2635}{1 + 2635 \cdot (0.05)^2}$$

Nilai n didapatkan hasil 347.2 yang dibulatkan menjadi 347 data. Dalam hal ini digunakan 347 sampel data untuk mencakup 2635 data.

Tahapan setelah itu mencari nilai kesalahan di levenshtein dan didapatkan hasil 32 kesalahan, maka untuk mencari nilai keakuratan didapatkan hasil 90.7%.

$$n = \frac{347 - 32}{347} \times 100\%$$

Lalu untuk mencari nilai kesalahan di gabungan fungsi antara levenshtein dan similar text didapatkan hasil 19 kesalahan, maka untuk mencari nilai keakuratan didapatkan hasil 94.5%

$$n = \frac{347 - 19}{347} \times 100\%$$

Dari perhitungan tersebut, gabungan fungsi levenshtein dan fungsi similar text lebih baik dengan tingkat akurasi 94.5%.

C. Usability Testing

Pada pengembangan aplikasi, untuk mendapatkan banyak masukan, aplikasi diuji dengan membuat *Minimum Viable Product* (MVP) dari seluruh *use case* yang ada. *Usability testing* dilakukan dengan satu iterasi uji coba. Seluruh hasil dari *usability testing* yang dilakukan didokumentasikan pada dokumen protokol dan hasil uji coba.

Pada akhir setiap sesi *usability testing* yang dilakukan, responden diminta untuk mengisi kuesioner *System Usability Scale* (SUS) yang digunakan untuk mengukur kemudahan dari penggunaan aplikasi.

Tabel 2. Daftar Test Case

ID Test Case	Test Case
TC01	Memilih aplikasi
TC02	Melihat detail aplikasi
TC03	Daftar akun developer
TC04	Login ke halaman developer
TC05	Input data aplikasi
TC06	Posting aplikasi

Hasil dari kuesioner *System Usability Scale* menunjukkan aplikasi yang dikembangkan memiliki kemudahan dalam penggunaan sebesar 74.5%. Pada kuisioner tersebut menunjukkan bahwa banyak pengguna yang setuju dengan pernyataan yang diajukan kepada mereka.

Tabel 3. Hasil Justifikasi System Usability Scal

Pernyataan	Justifikasi
Pengguna berpikir bahwa saya ingin menggunakan system ini sesering mungkin	Setuju
Pengguna berpikir sistem ini sangat rumit dan tidak perlu	Tidak Setuju
Pengguna berpikir sistem ini mudah digunakan	Setuju
Penggguna berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini	Tidak Setuju
Pengguna menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik	Setuju
Pengguna berpikir terlalu banyak inkonsistensi dari sistem ini	Tidak Setuju
Pengguna berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat	Setuju
Pengguna berpikir bahwa sistem ini sangat sulit untuk digunakan	Tidak Setuju
Pengguna merasa sangat percaya diri dalam menggunakan sistem ini	Setuju
Pengguna perlu belajar banyak hal sebelum dapat menggunakan sistem ini	Tidak Setuju

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Pada tahapan ini dijelaskan mengenai kesimpulan dari hasil penelitian dan saran untuk pengembangan penelitian di kemudian hari.

Berdasarkan dengan pengerjaan studidengan judul “rancang bangun agregasi toko aplikasi mobile berbasis web untuk meningkatkan pendayagunaan aplikasi multi platform” yang telah dilakukan dapat disimpulkan:

1. Proses pengerjaan dimulai dari mengidentifikasi website API. Lalu, pembuatan entity relationship diagram (ERD) sehingga membentuk basis data yang efisien dan tidak redundan. Setelah itu pengambilan data menggunakan cURL yang pengisiannya sesuai tabel-tabel yang telah dibuat meliputi playapplication, permissions, comments, histograms, screenshots, reviews, dan

appstoreapplications. Lalu proses terakhir adalah pembuatan alur kerja website direktori aplikasi menggunakan use case.

2. Hasil pengambilan data dari play store didapatkan 44100 aplikasi dan dari app store didapatkan 18318 aplikasi. Hasil pengambilan data itu dicari kesamaan aplikasi dan didapatkan 761 aplikasi dari total 2635 aplikasi yang dicari kesamaannya menggunakan fungsi levenshtein. Sedangkan setelah dilakukan penggabungan fungsi levenshtein dan fungsi similar text, didapatkan kesamaan 1696 aplikasi dari total 2635. Sehingga, metode penggabungan dua fungsi diimplementasikan ke aplikasi karena hasilnya yang lebih baik.
3. Hasil analisis perbandingan yang telah dijabarkan pada bab 6 diketahui bahwa terdapat perbedaan hasil similarity pada kedua metode. Hal ini terjadi karena fungsi levenshtein yang mencapai keakuratan 90.7% tidak cukup mencapai titik keakuratan yang dibutuhkan sedangkan penggabungan antara fungsi levenshtein dan fungsi similar text yang mencapai keakuratan 94.5% sudah mencapai tingkat keakuratan dalam mencari persamaan aplikasi di play store dan app store.
4. Hasil dari usability testing yang dilakukan ke sebanyak 10 responden menunjukkan aplikasi yang dikembangkan memiliki kemudahan dalam penggunaan sebesar 74.5%. Angka ini tidak kecil tetapi tidak juga besar, sehingga bisa dilakukan pengembangan lebih lanjut untuk mendapatkan usability testing yang lebih besar.

B. Saran

Berdasarkan penelitian pada studi ini dapat diberikan saran ataupun masukan untuk pengembangan selanjutnya, antara lain:

1. Penggunaan fungsi levenshtein kurang cocok dalam mencari kesamaan data aplikasi yang prosesnya membandingkan per huruf, disarankan untuk membandingkan dengan blok-blok kata di appId aplikasi.
2. Prototipe ini sebaiknya lebih ke arah memperkaya metadata aplikasi secara otomatis daripada menambahkan fitur memasukkan data manual untuk pengembang aplikasi.
3. User Interface pada prototipe ini dapat dikembangkan lebih lanjut untuk meningkatkan kepuasan pengguna dalam menggunakan prototipe ini.

DAFTAR PUSTAKA

- [1] A. Firat, S. Madnick, N. A. Yahaya, C. W. Kuan, and S. Bressan, "Information Aggregation using the Caméléon# Web Wrapper," no. August, 2005.
- [2] R. H. Walsh and T. A. Musgrove, "Content aggregation method and apparatus for an on-line product catalog," vol. 2, no. 12, 2006.
- [3] facundoolano, "google-play-scraper."
- [4] facundoolano, "app-store-scraper."
- [5] Boyke Dian Triwahyudhi, "Web Scraping, Trend ancaman Bisnis dan Teknologi Selanjutnya."
- [6] Michael Gilleland and Merriam Park Software, "Levenshtein Distance."