

Aplikasi Android untuk Mencari Harga Tiket Pesawat Termurah dari Beberapa Situs Travel di Indonesia

Bobby Irawan¹, Henry Novianus Palit², Justinus Andjarwirawan³

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236

Telp (031) – 2983455, Fax. (031) - 8417658

E-mail: irawanbobby09@gmail.com¹, hnpalit@petra.ac.id², justin@petra.ac.id³

ABSTRAK

Situs-situs *travel* di Indonesia menawarkan harga tiket pesawat yang bervariasi. Sehingga untuk mencari harga yang ramah kantong menjadi susah, konsumen harus membuka setiap situs atau aplikasi dari situs *travel* untuk membandingkan harga tiket pesawat. Hal ini memakan waktu yang tidak sedikit, dan terkadang ketika menemukan harga yang sesuai, namun ketika ingin memesan, tiket sudah habis terjual. Oleh karena itu, diperlukan sebuah aplikasi untuk mencari dan membandingkan harga dari situs-situs *travel* yang ada.

Aplikasi yang dibuat melakukan perbandingan harga dari situs-situs *travel*. Untuk melakukan perbandingan, diperlukan sebuah *server* yang dapat mengambil data setiap harga dan detail dari penerbangan pada setiap situs *travel*. Setelah proses pengambilan, dilanjutkan dengan proses pengolahan data yang didapat untuk ditampilkan pada aplikasi yang dibuat.

Berdasarkan hasil pengujian yang sudah dilakukan, struktur data masing-masing situs berbeda-beda, sehingga diperlukan proses pengolahan data tambahan sebelum dibandingkan dengan situs lainnya. Selain itu, pada beberapa situs, data perlu diambil dari beberapa sumber, seperti dari API dan *page HTML*. Sementara pada situs yang lain, data hanya ada pada API saja.

Kata Kunci: *Web Scraping, CURL, Android, Situs Travel*

ABSTRACT

Travel sites in Indonesia offer varied flight ticket prices. So, to find a cheap price becomes difficult, consumers must open every site or application from a travel site to compare the ticket price each site offered. This takes a lot of time, and sometime when you find the right price and want to order it, the ticket sold out. Therefore, an application is needed to search and compare prices from each travel sites.

Application was made to do the comparison of price that each travel sites offered. To do a comparison, a server is needed to retrieve each price and details of flights on each travel site. After that, the process continue to processing all data and displayed it to the application.

Based on the result of test that been done on program, program managed to collecting, processing, and showing price as well as flight information from travel sites.

Keywords : *Web Scraping, CURL, Android, Travel Site*

1. PENDAHULUAN

Perkembangan zaman membuat hidup manusia semakin mudah,

termasuk dalam bepergian menggunakan pesawat. Sebelum masuknya era *e-ticket* ke Indonesia, proses pemesanan tiket pesawat masih secara manual. *Customer* atau pembeli harus pergi ke kantor maskapai untuk melakukan pemesanan atau pergi ke tempat *travel agent*. Berbeda jauh dengan keadaan sekarang, pemesanan bisa dilakukan melalui *smartphone* atau situs. Sebut saja situs terkenal seperti *traveloka* dan *tiket.com* yang turut serta dalam membentuk kebiasaan masyarakat sekarang dalam memesan tiket pesawat. Tidak hanya *traveloka* dan *tiket.com*, sekarang sudah banyak situs-situs serupa yang bermunculan. Selain situs, layanan yang diberikan juga tersedia pada *smartphone* berupa aplikasi, sehingga lebih mudah dijangkau oleh *customer*.

Setiap situs-situs *travel* yang ada sekarang, memiliki daya tarik masing-masing, salah satunya adalah dari segi harga. Harga yang ditawarkan berbeda-beda pada setiap situs. Mereka juga berlomba-lomba memberikan *discount* pada *customer* berupa kode *voucher*. Selain kode *voucher*, setiap situs juga memberikan *deals*. Dengan banyaknya pilihan yang ada, membuat *customer* bingung untuk memilih. Tentunya, *customer* ingin membeli dengan harga termurah. Untuk mencapai tujuan tersebut, banyak yang melakukan *search manual* ke situs-situs yang ada. Hal tersebut memakan waktu yang tidak sedikit, dan kurang efisien. Oleh karena itu, diperlukan layanan perbandingan harga untuk mempermudah *customer* dalam mencari harga yang sesuai dengan kebutuhan.

Layanan perbandingan harga yang diusulkan pada skripsi ini akan berbentuk aplikasi berbasis sistem operasi Android. Android sendiri menguasai pasar sistem operasi di dunia, dengan persentase sebanyak 72,9% pada Q3 tahun 2017 [4]. Selain faktor tersebut, di Indonesia banyak masyarakat yang menggunakan *smartphone* berbasis Android. Mengingat sistem operasi ini banyak ditemukan di *smartphone* dengan harga murah yang menjadi pilihan masyarakat Indonesia.

2. TINJAUAN PUSTAKA

2.1 *Application Programming Interface*

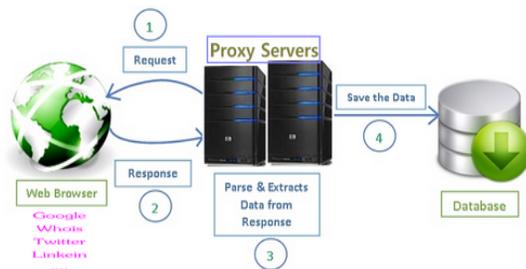
Application Programming Interface merupakan kepanjangan dari API, yang merupakan bagian dari *server* [2]. API juga menjembatani hubungan antara *server* dengan pengguna. Beberapa perusahaan menyediakan API secara gratis. Tetapi, ada juga beberapa situs yang tidak menyediakan API untuk diakses secara gratis. Salah satu cara untuk mendapatkan informasi yang diinginkan adalah dengan *web scraping*. Sebelum melakukan *scraping*, terlebih dahulu harus memahami struktur dari *target* situs. Karena beberapa situs tidak membuat orang lain untuk mengakses secara langsung. Hasil yang diterima dari API bisa digunakan untuk keperluan lain, seperti mengolah data lagi, atau langsung menampilkan kepada pengguna.

2.2 DOMXPath

DOMXPath merupakan salah satu *class* dari *library* DOM PHP [5]. *Library* DOM pada PHP memungkinkan untuk mengolah XML *document* melalui DOM API. Sedangkan XPath merupakan sintaks yang digunakan untuk mendefinisikan bagian-bagian dari XML *document* [8]. Jika ingin menggunakan *library* dari DOMXPath, terlebih dahulu mengubah data kedalam bentuk DOM, baru kemudian, data yang sudah dalam bentuk DOM tersebut diubah ke dalam bentuk DOMXPath.

2.3 Web Scraping

Secara umum web scraping adalah proses pengambilan dan menggabungkan konten yang ingin diambil dari suatu *web* dengan cara tertentu [1]. Gambar 1 merupakan contoh ilustrasi dari proses *web scraping* secara umum. Dalam proses tersebut, melibatkan robot yang bertindak sebagai pengganti manusia untuk melakukan *searching* konten yang diinginkan pengguna. Pada umumnya, untuk mendapatkan informasi, seperti cuaca, bisa menggunakan API yang ada.



Gambar 1. Ilustrasi Cara Kerja *Web Scraping* [6]

2.4 CURL

CURL merupakan program yang berfungsi untuk *scanning* data dari server, tanpa perlu adanya interaksi pengguna dengan *server* [7]. Seperti pengganti pengguna untuk melakukan koneksi dengan *server*. Program yang dijalankan akan mengambil data atau konten dari situs tujuan dan menampilkan ulang ke layar pengguna. Informasi yang didapatkan umumnya akan digunakan untuk tujuan lain. Semisal untuk melakukan *check* harga *handphone*, terlebih dahulu harus mengambil seluruh situs. Kemudian, cari *tag* yang berisikan harga dari *handphone* tersebut. Informasi dari *tag* tersebutlah yang akan digunakan.

2.5 DOMParser

DOM Parser menyediakan kemampuan untuk melakukan *parsing* dari HTML atau XML kedalam format *document* DOM [3]. Data yang diambil umumnya akan digunakan kembali untuk kepentingan pengguna. DOM Parser mempermudah untuk menemukan *tag* yang berisi informasi yang diperlukan. Jika dibandingkan dengan regex, DOM Parser memiliki struktur code yang lebih mudah dimengerti.

2.6 Penelitian Sebelumnya

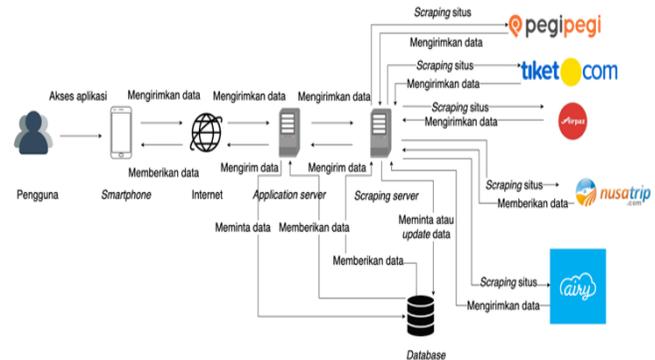
Aplikasi PriceMPT merupakan aplikasi yang membandingkan harga *handphone* dari berbagai situs *e-commerce* yang ada [9]. Situs *e-commerce* yang di-*scraping* untuk mengambil data harga dari *handphone* adalah, Blibli, Dinomarket, Erafone, Lazada, dan MatahariMall. Selain dari situs *e-commerce* yang sudah disebutkan, situs GSMarena juga di-*scraping*. Hal ini dilakukan untuk mendapatkan spesifikasi dari *handphone* yang dicari. Metode

scraping yang digunakan pada pembuatan skripsi ini menggunakan CURL PHP dan untuk mengambil data dari situs yang di-*scraping* menggunakan *library* DOMParser. Proses *scraping* dilakukan secara berkala untuk melakukan *update* pada data yang ada pada *database*.

3. ANALISA dan DESAIN SISTEM

3.1 Grafik Hirarki Aplikasi

Aplikasi yang dibuat bernama Aticket, merupakan singkatan dari *Affordable Ticket*. Pada Gambar 2 menjelaskan desain sistem secara keseluruhan dari aplikasi Aticket.



Gambar 2. Desain Sistem Secara Keseluruhan

Pengguna harus menggunakan *smartphone* berbasis sistem operasi Android untuk bisa mengakses aplikasi Aticket. Hal ini karena aplikasi dibuat menggunakan Android Studio, yang mana, nantinya akan menghasilkan aplikasi Android *native*. Pertama-tama, pengguna memasukkan data penerbangan melalui aplikasi. Jika selesai, pengguna akan menekan tombol untuk melakukan pencarian. Pada saat menekan tombol tersebut, aplikasi akan memanggil API melalui internet. API digambarkan dengan *Application server* pada Gambar 2. Kemudian, API menyalurkan data penerbangan dari pengguna, ke *server*. *Server* dilambangkan sebagai *Scraping server*. Selanjutnya, *Scraping server* akan melakukan *scraping* ke situs *travel* yang bisa dilihat pada Gambar 2. Apabila proses *scraping* berhasil, maka situs akan mengirimkan data ke *server* untuk diproses. Tidak hanya melakukan *scraping* ke situs *travel*, *server* juga mengambil data dari *database*. Selain mengambil, *server* juga akan melakukan *update* ke *database* jika ada perubahan data. Setelah semua proses pada *server* selesai, data tersebut akan dikirim ke API melalui internet. API juga akan mengambil data tertentu yang ada pada *database* dan melakukan *update* pada *database*. Data yang didapat dari *server* kemudian disalurkan ke aplikasi untuk ditampilkan ke pengguna. Dari aplikasi yang digunakan, pengguna bisa melihat harga yang ditawarkan dari masing-masing situs *travel*.

3.2 Use Case Diagram

Pada Gambar 3, terdapat dua aktor, yaitu *admin* dan pengguna. Pengguna tidak perlu melakukan login, tetapi tetap tersedia pilihan untuk melakukan *login* atau *signup* pada aplikasi. Pengguna bisa melakukan pencarian penerbangan yang diinginkan, dengan terlebih dahulu memasukkan data penerbangan. Beberapa data tersebut diantaranya, data asal dan tujuan bandara, tanggal kebergian, kelas kabin, dan jumlah penumpang. Pengguna juga bisa melihat promo yang *available* dari situs-situs *travel* yang menyediakan. Selain itu, ketika masuk ke *detail* dari promo, Pengguna dapat *copy* kode promo, untuk selanjutnya digunakan ke situs *travel* yang menyediakan promo tersebut. Pengguna dapat

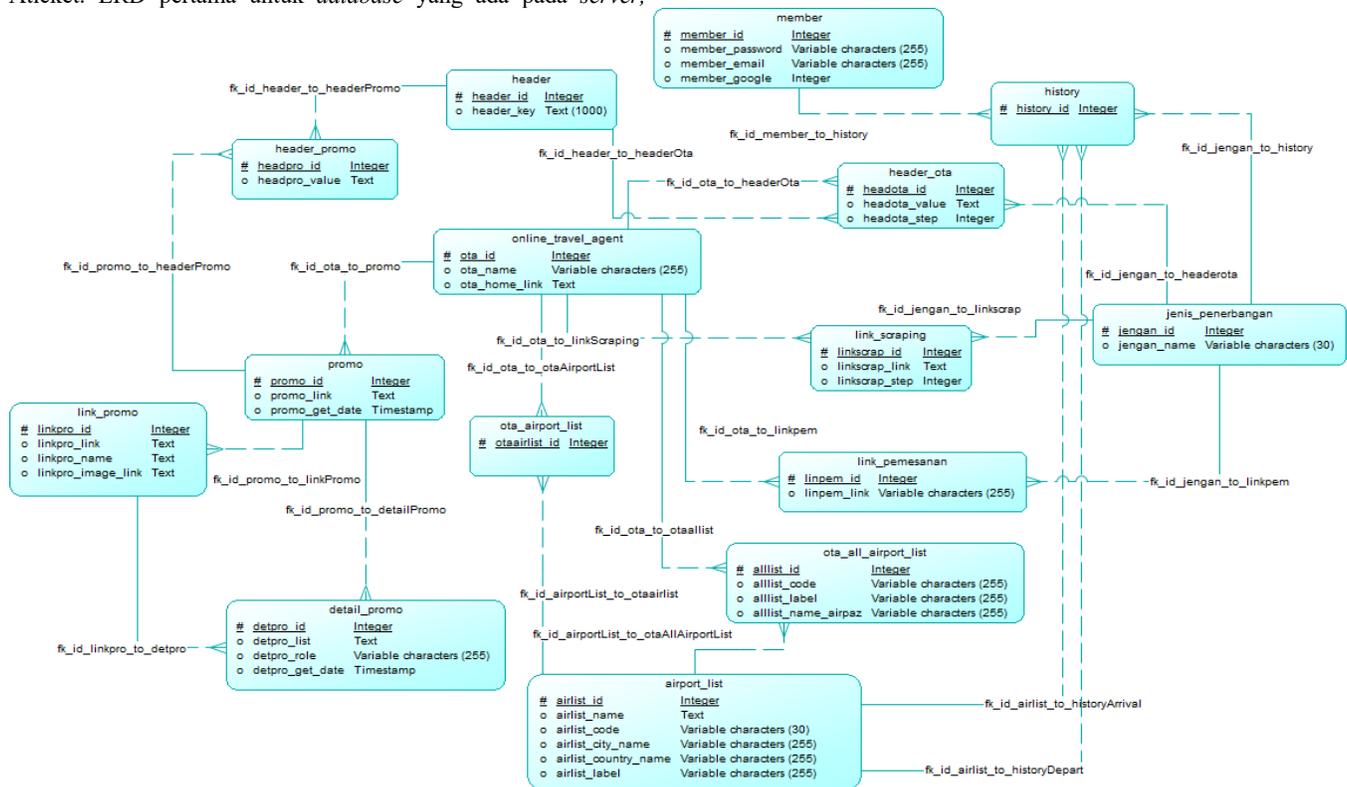
melihat hasil dari pencarian, dan melihat detail dari penerbangan yang dipilih. Total harga yang ditawarkan juga bisa dilihat oleh pengguna pada saat pencarian berhasil atau pada saat melihat detail penerbangan yang dipilih. Untuk aktor *admin*, hanya bisa *maintenance* data pada *database*. *Maintenance* yang dilakukan lebih kearah melakukan *update* kepada data-data yang berubah. Contoh data tersebut adalah *link* atau alamat URL yang digunakan untuk melakukan *scraping* ke situs *travel*. Sebelumnya, *admin* harus melakukan *login* terlebih dahulu.



Gambar 3. Use Case Diagram Aplikasi

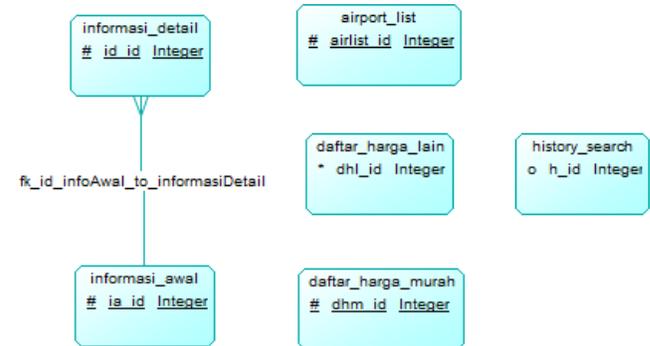
3.3 Entity Relationship Diagram

Terdapat dua *Entity Relationship Diagram* (ERD) pada aplikasi Aticket. ERD pertama untuk *database* yang ada pada *server*,



Gambar 5. ERD Server Conceptual Model

sedangkan yang kedua untuk *database* yang ada pada aplikasi Aticket. Desain ERD untuk *database server* dapat dilihat pada Gambar 5, sedangkan ERD untuk *database aplikasi*, dapat dilihat pada Gambar 4. ERD aplikasi tidak memiliki hubungan antar tabel, karena pada awalnya digunakan hanya untuk menyimpan data yang tidak kompleks seperti yang ada pada *server*.



Gambar 4. ERD Aplikasi Conceptual Model

4. PENGUJIAN SISTEM

Pengujian dilakukan untuk mengetahui apakah program yang telah dibuat dapat berjalan dengan baik atau tidak. Terdapat tiga pengujian, yaitu pengujian *scraping* harga ke masing-masing situs *travel*, pengujian terhadap harga yang didapat dari hasil *scraping*, *up to date* atau tidak. Pengujian ketiga adalah pengujian terhadap tampilan dari aplikasi yang dibuat. Pengujian lainnya yang juga dilakukan adalah terhadap data promo yang di-*scraping* dari masing-masing situs *travel* yang menyediakan. Data promo disesuaikan dengan situs penyedia promo tersebut. Apakah data promo sesuai atau tidak.

4.1 Pengujian Aplikasi *Mobile*

Pengujian aplikasi difokuskan untuk mengetahui tampilan pada *smartphone* dengan spesifikasi yang berbeda-beda. Ada tiga *smartphone* yang digunakan untuk pengujian aplikasi. Spesifikasi masing-masing *smartphone* dapat dilihat pada Tabel 1.

Tabel 1. Tabel Spesifikasi *Smartphone* untuk Pengujian Aplikasi

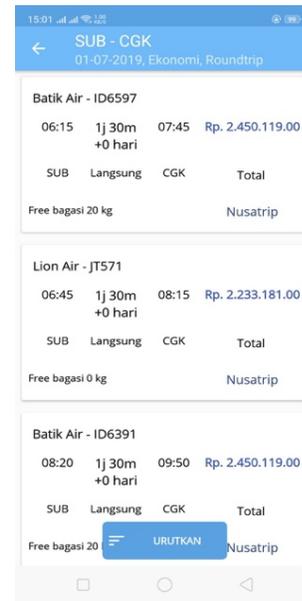
<i>Smartphone</i>	<i>Operating System</i>	<i>Display Size</i>	CPU	<i>Memory</i>
Samsung Galaxy A5 (2017) (SM-A520F)	Android 8.0 (Oreo)	1080 x 1920 pixels, (~423 ppi density)	Exynos 7880 Octa-core (1.9 GHz Cortex-A53)	32 GB, 3 GB RAM
Oppo A5S (CPH1909)	Android 8.1.0 (Oreo)	720 x 1520 pixels, (~269 ppi density)	ARM Cortex-A53 (2.30 GHz)	32 GB, 3 GB RAM
OnePlus 3 (ONEPLUS A3000)	Android 8.0.0 (Oreo)	1080 x 1920 pixels, (~400 ppi density)	Qualcomm Snapdragon 820 (2.15 GHz)	64 GB, 6 GB RAM

Jenis penerbangan yang diuji adalah jenis penerbangan *round-trip*. Untuk bandara asal adalah bandara Juanda, Surabaya, dan bandara tujuan adalah bandara Soekarno Hatta, Jakarta. Untuk tanggal pergi adalah 1 Juli 2019, tanggal pulang adalah 9 Juli 2019. Jumlah penumpang dewasa satu orang, anak-anak satu orang, dan bayi satu orang. Untuk kelas penerbangan yang dipilih adalah ekonomi. Gambar 6 merupakan tampilan awal apabila memilih jenis penerbangan *round-trip*. Jika *click* pada *button* bertuliskan “CARI”, maka akan muncul tampilan seperti pada Gambar 7.

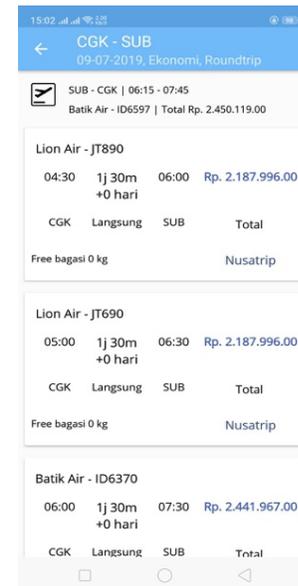


Gambar 6. Tampilan Penerbangan *Round-trip*

Informasi yang diberikan diantaranya, nama maskapai beserta kode penerbangan, jam keberangkatan dan jam tiba, harga termurah dari situs *travel* yang di-*scraping*, serta situs yang memberikan harga tersebut.



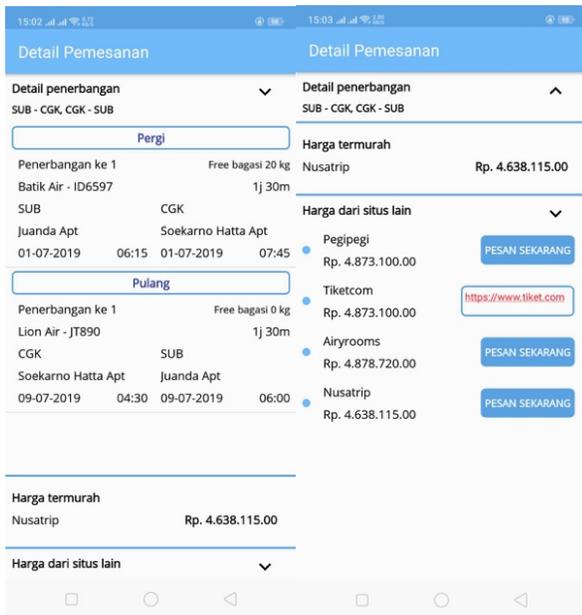
Gambar 7. Tampilan Hasil Pencarian Pergi



Gambar 8. Tampilan Hasil Pencarian Pulang

Jika pengguna memilih salah satu dari hasil pencarian pergi, aplikasi akan memanggil API pada *server*, untuk melakukan *scraping* penerbangan pulang. Hasil dari *scraping* tersebut akan ditampilkan seperti pada Gambar 8. Informasi sama seperti pada Gambar 7, hanya saja, terdapat informasi dari penerbangan yang dipilih sebelumnya. Informasi ini bertujuan agar pengguna mengetahui penerbangan apa yang dipilih sebelumnya, dan menentukan penerbangan pulang, agar waktu tidak bertabrakan dengan waktu sampai dari penerbangan pergi.

Pengguna juga bisa mengurutkan hasil pencarian yang muncul. Untuk melakukan pengurutan, pengguna perlu *click* tombol bertuliskan “URUTKAN”, yang ada pada Gambar 7. Urutan tersebut berupa harga termurah, jam tiba awal, jam tiba akhir, jam pergi awal, dan jam pergi akhir. Secara *default* hasil pencarian akan menampilkan data dari harga termurah semua penerbangan dari hasil pencarian penerbangan tersebut.



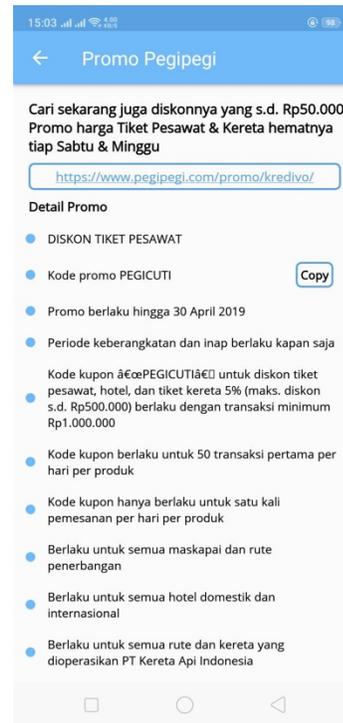
Gambar 9. Tampilan *Detail Penerbangan Yang Dipilih*

Apabila pengguna memilih salah satu penerbangan pulang, maka akan muncul tampilan seperti pada Gambar 9. Dua gambar tersebut merupakan tampilan layar. Karena *screenshot* tidak bisa *full*, maka gambar dipisah menjadi dua. Terdapat informasi *detail* dari penerbangan pergi dan pulang. Selain itu, ada informasi mengenai harga yang ditawarkan oleh situs lain, tetapi dengan total harga yang lebih mahal. Untuk penerbangan *round-trip* harga yang ditampilkan adalah akumulasi dari harga penerbangan pergi, dan harga penerbangan pulang. Syarat tambahan adalah, situs harus menyediakan kedua penerbangan yang dipilih. Jika tidak, maka harga lain dari situs tersebut tidak akan muncul.



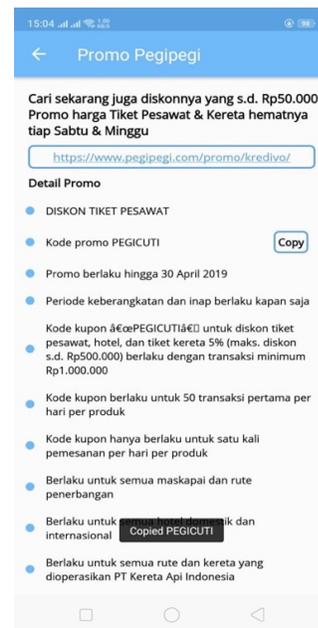
Gambar 10. Tampilan Halaman Promo

Semua pengguna bisa melihat promo yang tersedia pada aplikasi. Tampilan dari halaman promo aplikasi dapat dilihat pada Gambar 10. Terdapat informasi nama situs, gambar promo dari situs, serta informasi singkat mengenai situs.



Gambar 11. Tampilan *Detail Promo*

Gambar 11 merupakan tampilan dari *detail* promo. *Detail* promo akan muncul apabila pengguna *click* salah satu promo. Terdapat informasi *detail* promo, informasi singkat mengenai promo, serta URL dari promo tersebut. Jika *click* URL, maka akan membuka tampilan promo di aplikasi, jika pengguna memiliki aplikasi dari situs, atau pada tampilan *web* jika pengguna tidak memiliki aplikasi dari situs yang menyediakan promo tersebut.



Gambar 12. Pemberitahuan Bila *Copy* Kode Promo

Salah satu fitur dari aplikasi yang dibuat adalah bisa *copy* kode promo. Pengguna hanya perlu menekan informasi bertuliskan "Copy", maka akan muncul notif seperti pada Gambar 12.

5. KESIMPULAN

Dari hasil pengujian sistem yang telah dilakukan, dapat diambil beberapa kesimpulan, diantara lain:

- Berdasarkan kesimpulan yang diperoleh dari pengujian, penggunaan metode *scraping* untuk mengambil data harga dan penerbangan dari situs-situs *travel* berhasil dilakukan. Meskipun terdapat data yang tidak sesuai antara data *scraping* dengan data pada situs, seperti harga pada situs Nusatrip.
- Proses pengolahan data dari hasil *scraping* dilakukan sebanyak dua kali. Pertama untuk menyamakan format data antar situs, yang kedua untuk membandingkan harga dari masing-masing situs yang sudah memiliki format yang sama.
- Beberapa situs *travel* memiliki sumber data yang berbeda-beda. Data yang disimpan bisa saja terbagi-bagi. Seperti pada *page* HTML situs, informasi tersebut merupakan tambahan, dan terkadang penting, seperti promo potongan harga. Padahal, pengambilan data utama, berasal dari *scraping* ke URL yang berisi data penerbangan, bukan *page* HTML. Sehingga memerlukan proses pengolahan data lagi.

6. DAFTAR PUSTAKA

- [1] Daniel, G., Anália, L., Hugo, L., Miguel, R., Florentin, F. 2014. Web scraping technologies in an API world. *Briefings in Bioinformatics*, 15(5), 788–797. DOI= <https://doi.org/10.1093/bib/bbt026>.
- [2] Gazarov, P. 2016. What is an API? In English, please. URI= <https://medium.freecodecamp.org/what-is-an-api-in-english-please-b880a3214a82>.
- [3] Goel, T. 2018. DOMParser. URI= <https://developer.mozilla.org/en-US/docs/Web/API/DOMParser>.
- [4] Katadata. 73% Perangkat *Mobile* Global Menggunakan Android. 2017. URI= <https://databoks.katadata.co.id/datapublish/2017/10/09/73-perangkat-mobile-global-menggunakan-android>.
- [5] PHP Group. DOM Intorduction. URI= <https://www.php.net/manual/en/intro.dom.php>.
- [6] Singh, S. 2018. Web Scraping with BeautifulSoup. URI= <https://towardsdatascience.com/web-scraping-with-beautifulsoup-d119a5b7f35c>
- [7] Syahtriatna, D., Zamzami, Asril, E., Supriatna, J. 2017. Development of A Content Grabbing using php curl to Read News Online. URI= http://www.estech.org/index.php/IJSAT/article/view/53/pdf_1.
- [8] W3Schools.com. XPath Syntax. URI= https://www.w3schools.com/xml/xpath_syntax.asp.
- [9] Wijaya, R. C.. 2016. Aplikasi pencarian produk jual mobile devices dari berbagai situs e-commerce. (TA No. 03021490/INF/2016). Surabaya: Universitas Kristen Petra.