JUSTIN
Jurnal Sistem dan Teknologi Informasi
p-ISSN : 2460-3562 / e-ISSN : 2620-8989
Vol. 7, No. 4, Oktober 2019

Implementasi *Web Scraping* untuk Pengambilan Data pada Situs *Marketplace*

Dhita Deviacita Ayani[#1], Helen Sasty Pratiwi[#2], Hafiz Muhardi[#3]

[#]*Program Studi Informatika Universitas Tanjungpura*
Jl. Profesor Dokter H. Hadari Nawawi, Bansir Laut, Pontianak Tenggara, Kota Pontianak, Kalimantan Barat 78115
[1]ddeviacita192013@gmail.com
[2] helensastypratiwi@informatika.untan.ac.id
[3]hafizmuhardi@informatika.untan.ac.id

Abstrak— **Perdagangan elektronik atau *e-commerce* merupakan proses penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet. *E-commerce* dapat melibatkan transfer dana elektronik, pertukaran data elektronik, sistem manajemen inventori otomatis, dan sistem pengumpulan data otomatis. Di Indonesia, terdapat beberapa perusahaan yang bergerak dalam bidang *e-commerce,* yang beberapa diantaranya adalah Bukalapak, Elevenia, dan JD.id. Setiap perusahaan tersebut memiliki situs web dimana calon pembeli dapat mencari, memilih, dan membeli produk yang diinginkan. Agar dapat mengambil keputusan yang terbaik ketika melakukan proses perbelanjaan, selain seorang calon pembeli yang harus melakukan pencarian di beberapa situs *marketplace* mengenai produk yang hendak dibeli secara manual juga mereka ingin mencari produk yang terbaik berdasarkan jumlah produk dengan penjualan terbanyak atau terlaris. Dengan bantuan aplikasi web pencarian terbaik dan teknik *web scraping* ini akan mampu melakukan pencarian di beberapa situs *marketplace* dan menampilkan hasil pencarian secara bersamaan. Menurut hasil pengujian dengan menggunakan *white box testing* dan *black box testing*, sistem telah mampu memberikan hasil terbaik produk dari gabungan hasil pencarian di tiga situs web *marketplace* sesuai kata kunci yang dimasukkan oleh pengguna.**

Kata kunci— **Web Scraping, E-Commerce, Marketplace, White Box Testing, Black Box Testing**

I. PENDAHULUAN

Perkembangan teknologi informasi dari hari ke hari semakin cepat, terutama internet yang memiliki kelebihan untuk menyediakan informasi yang cepat. Pengguna (*user*) dapat dengan mudah mengoperasikan dan menggunakan internet sepanjang waktu dengan *smartphone*, *tablet*, *notebook* dan komputer yang dimiliki. Kemudahan yang ditawarkan layanan internet memberikan dampak positif bagi para penggiat bisnis untuk menawarkan atau menjual produk melalui internet. Salah satu manfaat yang diperoleh adalah kita dapat mengetahui atau mencari berbagai macam produk yang dibutuhkan melalui Internet. Hal ini mendorong banyaknya toko *online* yang bermunculan di Indonesia. Dari segi bahasa, toko online berasal dari dua suku kata, Toko dan Online..

Dalam rangka memudahkan pengguna (*user*) untuk mendapatkan produk yang diinginkan, dengan hasil penjualan terbaik , serta perbandingan harga produk antar situs maupun antar produk serta ulasan dari *buyer*, maka diperlukan sebuah situs yang mampu mengumpulkan berbagai informasi yang terdapat dalam situs toko *online* di antaranya Bukalapak.com, Elevenia, dan JD.id. Informasi tersebut berupa harga, nama produk, banyak produk terjual, *rating* produk, deskripsi produk dan asal situs.

Web Scraping adalah proses pengambilan sebuah dokumen semi-terstruktur dari internet, umumnya berupa halaman-halaman *web* dalam bahasa *markup* seperti HTML atau XHTML, dan menganalisis dokumen tersebut untuk diambil data tertentu dari halaman tersebut untuk digunakan bagi kepentingan lain, serta banyak penelitian yang menggunakan tools scraping untuk mengumpulkan datanya dari web.[9,14,15] Dalam penelitian ini, akan dikembangkan situs untuk memberikan saran produk yang *terbaik* kepada calon pembeli sebagai acuan membeli pada toko *online*, baik pada situs Bukalapak.com, Elevenia, dan JD.id serta membandingkan harga antar produk di ketiga toko *online* tersebut. Sistem yang dibangun menggunakan *Web Scraping* untuk mengambil data berupa nama, harga, banyak produk terjual, deskripsi, gambar dan situs asal dari produk tersebut. Disini, penulis hanya mengambil tiga situs *Marketplace* yang telah disebutkan, dikarenakan selain dari pada tiga situs tersebut, situs *Marketplace* lainnya tidak dapat dilakukan pengambilan Data (*web scraping*) dengan alasan sistem keamanan dari situs lainnya yang dapat mencegah tindakan *scraping* terhadap konten-konten maupun produknya.

Submitted 09-01-2019; Revised 22-01-2019; Accepted 22-01-2019

II. TINJAUAN PUSTAKA

A. *Web Scraping*

Web scraping adalah teknik untuk mendapatkan informasi dari website secara otomatis tanpa harus menyalinnya secara manual. Tujuan dari web scraper adalah untuk mencari informasi tententu dan kemudian mengumpulkannya dalam web yang baru. Web scraping berfokus dalam mendapatkan data dengan cara pengambilan dan ekstraksi. Manfaat dari web scraping ialah agar informasi yang dikeruk lebih terfokus sehingga memudahkan dalam melakukan pencarian sesuatu. Aplikasi Web Scraping hanya fokus pada cara memperoleh data melalui pengambilan dan ekstraksi data dengan ukuran data yang bervariasi. Web scraping memiliki sejumlah langkah meliputi:

1. Create Scraping Template: Pembuat program mempelajari dokumen HTML dari website yang akan diambil informasinya untuk tag HTML yang mengapit informasi yang akan diambil.
2. Explore Site Navigation: Pembuat program mempelajari teknik navigasi pada website yang akan diambil informasinya untuk ditirukan pada aplikasi web scraper yang akan dibuat.
3. Automate Navigation and Extraction: Berdasarkan informasi yang didapat pada langkah 1 dan 2 diatas, aplikasi web scraper dibuat untuk megotomatisasi pengambilan informasi dari website yang ditentukan.
4. Extracted Data and Package History: Informasi yang didapat dari langkah 3 disimpan dalam tabel database.

B. *HTML*

HTML digunakan sebagai dokumen yang mendetailkan elemen-elemen yang digunakan untuk membangun halaman web. Banyak dari elemen tersebut digunakan untuk mendeskripsikan konten dari halaman web seperti heading, paragraf dan list. HTML yang terbaru saat ini adalah HTML5. HTML5 adalah perubahan dari versi sebelumnya dari HTML dan HTML5 berusaha untuk merefleksikan kebutuhan dari website saat ini dan masa depan.[1]

C. *Unified Modelling Language* (UML)

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. [8]

1. *Use Case Diagram*
   Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dengan kata lain, use case diagram digunakan untuk mengetahui fungsi-fungsi apa saja yang terdapat di dalam sistem dan siapa saja yang berhak mengakses fungsi tersebut.[8]
2. *Class Diagram*
   Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas.[8]
3. *Sequence Diagram*
   Sequence Diagram memperlihatkan sekumpulan objek yang berinteraksi dan urutannya.[11] Dalam menggambarkan *sequence diagram* perlu memperhatikan objek-objek yang terlibat di dalam *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.[8]
4. *Activity Diagram*
   Activity Diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem.[8]

III. PERANCANGAN APLIKASI

A. Arsitektur Sistem

Desain arsitektur akan ditunjukkan pada Gambar 1,

Gambar 1. Desain Arsitektur

1. *Halaman Utama*

Merupakan halaman utama yang menampilkan pencarian produk berdasarkan kata kunci, pengguna dapat mencari produk yang diinginkan pada halaman ini.

2. *Halaman Pencarian*

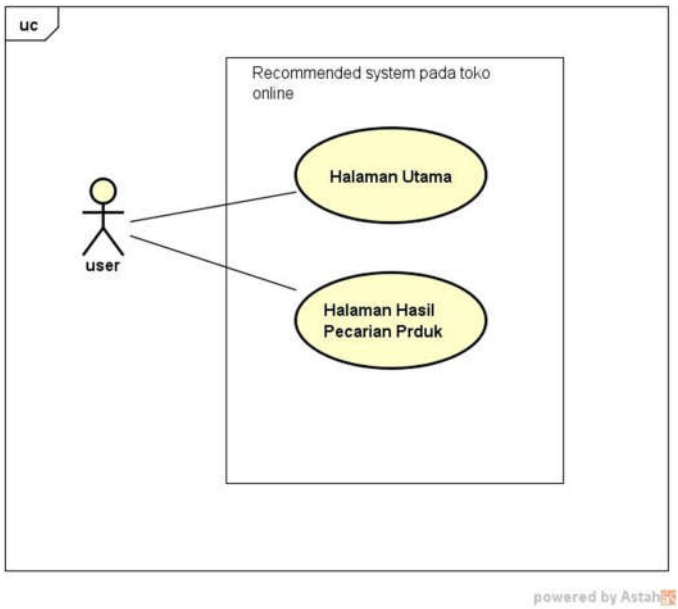
Merupakan halaman yang menampilkan produk dari hasil pencarian berdasarkan kata kunci yang telah diinput oleh pengguna pada halaman utama.

3. *Tombol Detail*

Merupakan tombol yang akan mengarahkan detail lengkap produk pada website aslinya.

B. Use Case Diagram

Use case diagram aplikasi diperlihatkan pada Gambar 2,



Gambar 2. Use Case Diagram

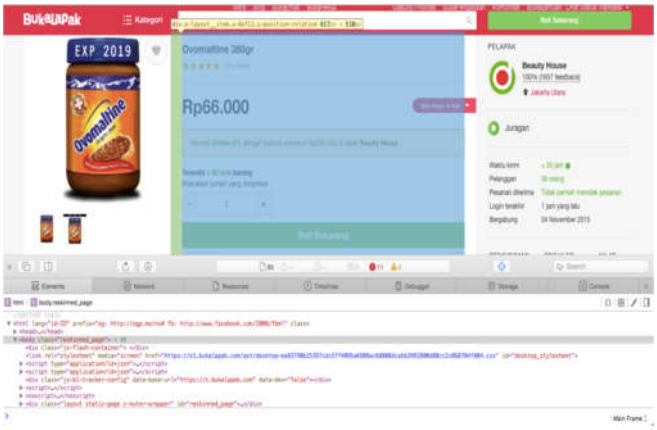
C. Pengujian Sistem

Pengujian sistem yang akan digunakan pada penelitian ini adalah *White Box Testing* dan *Black Box Testing*.

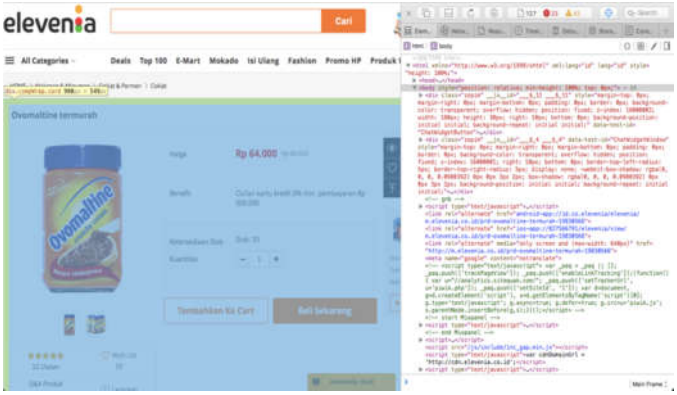
D. Proses Web Scraping, pengambilan data dari ketiga situs Marketplace. (kata kunci produk : Ovomaltine)

1. Pada Bukalapak.com, Elevenia, dan JD.id

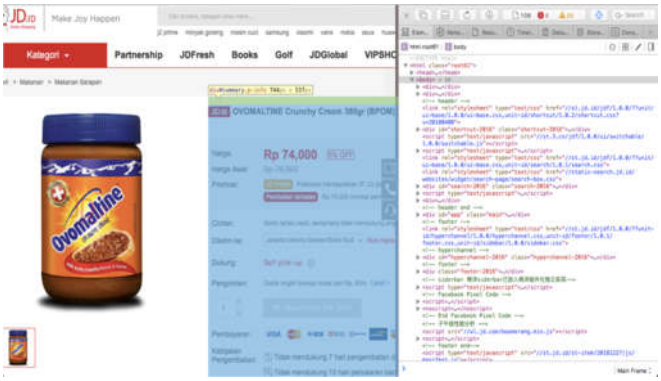
Proses melakukan pengambilan data (web scraping) adalah dengan cara mengunjungi situs Marketplace, yaitu Bukalapak.com, kemudian mencari produk apa saja, lalu melakukan inspect element guna mengambil class yang terdapat didalamnya berupa data-data penting yang akan di ekstraksi, contohnya dapat dilihat pada Gambar blablabla, dengan langkah-langkahnya sebagai berikut:



Gambar 3. Proses *Web Scraping* pada Bukalapak.com



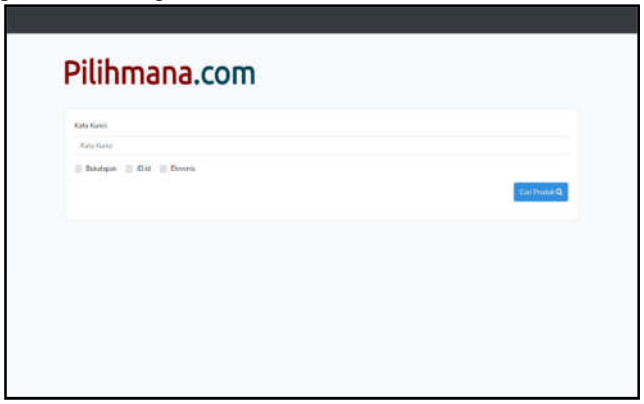
Gambar 4. Proses *Web Scraping* pada Elevenia



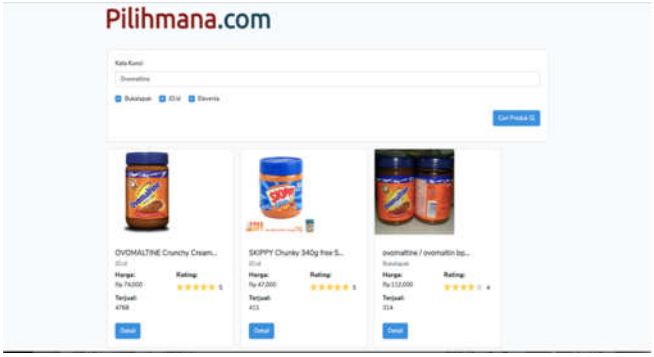
Gambar 5. Proses *Web Scraping* pada JD.id

E. Hasil Aplikasi

Aplikasi yang dibangun adalah Impelementasi *Web Scraping* Untuk Pengambilan Data Pada Situs *Marketplace*. Berikut adalah beberapa tampilan hasil perancangan aplikasi. Gambar 6 merupakan tampilan dari menu utama yang berisi Kolom pencarian produk, dan pilihan *Marketplace*.

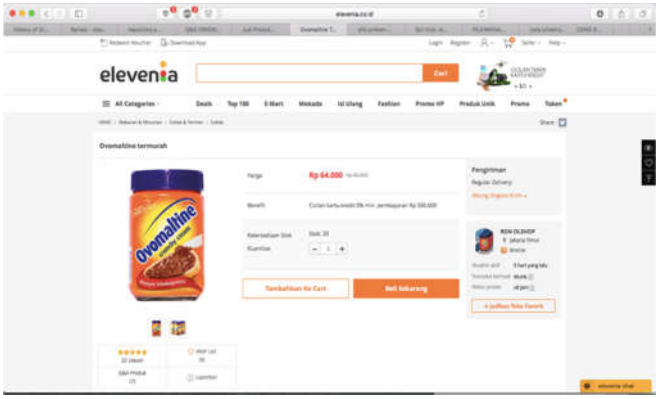


Gambar 6. Menu Utama



Gambar 7. Tampilan halaman hasil pencarian produk

Gambar 7 merupakan tampilan dari halaman hasil pencarian produk yang berisi data informasi produk yaitu : nama produk, harga, jumlah terjual, rating, dan tombol detail.



Gambar 8. Tampilan Tentang Halaman Situs *Marketplace* aslinya.

Gambar 8 merupakan tampilan tentang halaman dari salah satu situs *marketplace.*

IV. HASIL PENGUJIAN

1. Pengujian *White Box Testing*

Pengujian ini digunakan untuk meyakinkan semua perintah dan kondisi pada sistem dieksekusi secara minimal. Pengujian *white box* menggunakan dua tools yaitu *flow graph* yang digunakan untuk menggambarkan alur dari algoritma dan *graph matrix* yang digunakan untuk menggenerasi flow graph. Lalu perhitungan *cyclomatic complexity* dan algoritma *graph matrix.* [3]

a. Source code *web scraping*

Dapat dilihat pada Gambar 9 merupakan source code yang merupakan *flowchart* cara melakukan *scraping.*

```
    public function getFromElevenia($keyword = '', $limit = 5)
    {
        # Bagian 1: Mulai
# Bagian 2: Penarikan data dari situs web
        $crawler = $this->non_js_scraper->request(
            'GET',
            "http://www.elevenia.co.id/search?q=$keyword&lCtgrNo="
        );
        $products = collect();

        # Bagian 3: Loop / Iterasi terhadap daftar produk
        $crawler->filter('li.itemList')->each(function($product_card, $i)
use($products, $limit) {

            # Bagian 4: Keputusan untuk melakukan / tidak melakukan
scraping berdasarkan limit
            if ($i > $limit) { return; }

            # Bagian 5: Selesai
            $name = $product_card->filter('a.pordLink')->text();
            $short_name = str_limit($name, $this->name_char_limit);
            $url = $product_card->filter('a.img')->link()->getUri();
            $img_url = $product_card->filter('a.img > img')->attr('src');
            $rating_node = $product_card->filter('span.rating');
            $rating = $rating_node->count() != 0 ? (int) substr($rating_node-
>attr('class'), -1) : 0;
            $id = (string) Str::orderedUuid();
            $products->push(compact('id', 'name', 'short_name', 'url', 'img_url',
'rating'));
        });
        $products->transform(function($product) {
            $crawler = $this->non_js_scraper->request('GET', $product['url']);
            $product['price'] = $crawler->filter('span.price')->text();
            $sales_node = $crawler->filter('span#reviewCount');
            $product['sales'] = $sales_node->count() != 0 ? (int) $sales_node->text() :
0;
            $product['source'] = 'Elevenia';
            return $product;
        });
        return $products;
# Bagian 6: Selesai
```

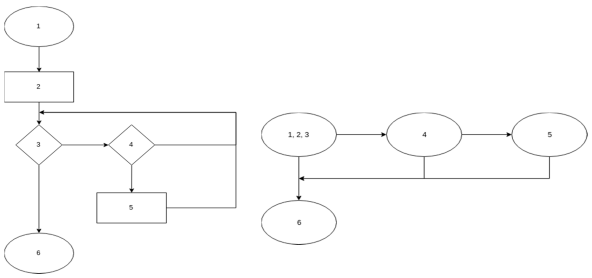
Gambar 9. Source code *Web Scraping*

b. Perubahan *flowchart* menjadi *flow graph*

flow graph dapat dibuat dari grafik flowchart ataupun dari pseudocode/program design language/source code yang telah dibuat sebelumnya. Hasil dapat dilihat pada Gambar 10. [12]

c. Perhitungan *cyclomatic complexity*

cyclomatic complexity dapat digunakan untuk menentukan berapa minimal *test case* yang harus dijalankan untuk menguji sebuah program dengan menggunakan teknik *basis path testing*.[2] Hasil dapat dilihat pada Tabel 4.2



Gambar 10. Perubahan *flowchart* menjadi *flow graph*

V(G) = E – N + 2
= 5 – 4 + 2
= 3

Dimana :
E = jumlah *edge* pada grafik alir
N = jumlah *node* pada grafik alir

TABEL 4.1
GRAPH MATRIX ALGORITMA

No.	Kasus Uji	Hasil yang Diharapkan	Hasil Uji Kasus	Keterangan
1.	$crawler->filter('div.product-card')->each(function ($product_card, $i) use($products, $limit) { }	Perulangan proses *scraping* akan terjadi pada seluruh produk yang ada	Perulangan terjadi pada seluruh produk yang ada	Alur terlewati
2.	if ($i + 1 > $limit) { return; }	Tindakan *scraping* tidak akan dilakukan ketika jumlah produk melebihi limit	*Tindakan scraping* tidak dilakukan ketika jumlah produk melebihi limit	Alur terlewati
3.	$crawler->filter('div.product-card')->each(function ($product_card, $i) use($products, $limit) { }	Perulangan proses *scraping* tidak akan terjadi jika daftar produk kosong	Perulangan proses *scraping* tidak akan terjadi jika daftar produk kosong	Alur terlewati

TABEL 4.2
CYCLOMATIC COMPLEXITY

Path 1	1, 2, 3, 4, 6
Path 2	1, 2, 3, 4, 5, 6
Path 3	1, 2, 3, 6

d. Pengujian Algoritma dilakukan untuk mengetahui angka keberhasilan setiap sistem yang berjalan sampai dengan 0 (Zero) error. Lihat pada Table 4.1 [13]

e. Tabel *Graph Matrix*

Menentukan tabel *Graph Matrix* pada Tabel 4.3 dengan melihat kepada setiap *edge (E)* yang dilalui pada *Flow Graph* penomoran ulang, sebagai berikut:

node (N) :

1 ke 1 = 0, 1 ke 2 = 1, 1 ke 3 = 0, 1 ke 4 = 1
2 ke 1= 0, 2 ke 2 = 0, 2 ke 3 = 1, 2 ke 4 = 1
3 ke 1= 0, 3 ke 2 = 0, 3 ke 3 = 0, 3 ke 4 = 1
4 ke 1 = 0, 4 ke 2 = 0, 4 ke 3 = 0, 4 ke 4 = 0

TABEL 4.3
TABEL *GRAPH MATRIX*

N	1	2	3	4	n(E) - 1
1		1		1	1
2			1	1	1
3				1	0
4					0
Sum(E) + 1					3

2. Pengujian *Black Box Testing*

Pengujian Implementasi *Web Scraping* Untuk Pengambilan Data pada Situs *Marketplace* dilakukan pada fitur-fitur dalam aplikasi menggunakan metode *Blackbox* yang akan memeriksa apakah aplikasi yang dibangun berjalan dengan benar sesuai dengan yang diharapkan atau tidak. Data pengujian dipilih berdasarkan sepesifikasi masalah tanpa memperhatikan detail internal dari program. Hasil pengujian dapat dilihat pada Tabel 4.4.

TABEL 4.4
PENGUJIAN SISTEM

Jenis pengujian	Hasil yang diharapkan	Hasil uji
Melakukan pencarian produk dengan memasukan kata kunci pada kolom pencarian kemudian mencentang pilihan dibawah kolom kata kunci dan menekan tombol cari produk	Masuk ke halaman hasil pencarian produk	Berhasil
Menekan tombol detail pada kolom produk	Masuk ke halaman web asli produk	Berhasil

V. ANALISIS HASIL PENGUJIAN

Rincian hasil analisis pengujian Implementasi *Web Scraping* Untuk Pengambilan Data pada Situs *Marketplace* yang telah dilakukan adalah sebagai berikut:

1. Berdasarkan hasil pengujian *black box testing* sistem dapat berjalan sesuai fungsionalitas yang diturunkan dari analisis kebutuhan.
2. Berdasarkan hasil pengujian *white box testing* sistem berhasil dijalankan dengan kondisi data yang diambil sesuai dengan script yang dijalankan, dimana setiap satu rangkaian pernyataan proses pada program telah dieksekusi paling tidak satu kali selama pengujian dan semua kondisi logis telah diuji dan berhasil.

VI. KESIMPULAN

A. Kesimpulan

Berdasarkan hasil analisis dan pengujian terhadap *Implementasi Web Scraping Untuk Pengambilan Data pada Situs Marketplace* didapatkan hasil tujuan penelitian yang tercapai sesuai kebutuhan pengguna (*user*), dimana :

1. Pengambilan data (*Web Scraping*) yang dilakukan *Marketplace* berhasil dilakukan dan dapat menyajikan hasil yang sesuai dengan kebutuhan.
2. Perancangan dan hasil telah sesuai dilakukan dengan pengujian *black box testing* maupun *white box texting* yang memberikan hasil sesuai dengan kriteria kebutuhan pengambilan data yang difokuskan.

B. Saran

Adapaun hal yang perlu ditambahkan dalam pengembangan implementasi *web scraping* pada situs *marketplace* ini adalah sistem ini dapat dikembangkan lagi dengan menambahkan parameter ulasan produk atau *reviews* dari pembeli, dan range harga yang dapat disesuaikan.

REFERENSI

[1] Castro, E., & Hyslop, B. 2012. *HTML 5 and CSS 3 (7th ed.)*. Berkeley: Peachpit Press.

[2] Du, Q. & Dong, X. 2011. An improved algorithm for basis path testing. BMEI 2011 - Proceedings 2011 International Conference on Business Management and Electronic Information, 3, pp.175–178.

[3] Farzan, A., Holzer, A. & Veith, H. 2015. Perspectives on *white-box* testing: Coverage, concurrency, and concolic execution. 2015 IEEE 8th International Conference on Software Testing, Verification and Validation, ICST 2015 - Proceedings.

[4] Josi, A., Abdillah, L. A., & Suryayusra. 2014. P*enerapan Teknik Web Scraping pada Mesin Pencari Artikel Ilmiah*. Jurnal Sistem Informasi, 5(2), 259-164.

[5] Nathania, Y., Andjarwirawan, J., & Rostianingsih, S. 2016. *Pembuatan Web Pembanding Fashion Wanita*. Surabaya: Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra.

[6] Papazoglou, M.P. 2012. *Web services & SOA Principles and Technology (2nd ed.)*. England: Pearson Education, Inc.

[7] Riyadi, D. 2013. Rancang Bangun Rest Web Service Untuk Perbandingan Harga Pengiriman Dengan Metode Web Scrapping dan Pemanfaatan API. Yogyakarta: Sekolah Tinggi Manajemen Informatika dan Komputer Amikom Yogyakarta.

[8] Sukamto, Rosa Ariani dan M. Shalahudin. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

[9] Turland, M. 2010. *php| architect's Guide to Web Scraping with PHP.* Introduction-Web Scraping Defined, str, 2.

[10] Welling, L., & Thomson, L. 2003. *PHP and MySql Web Development (2nd ed).* Sams Publishing, Indianapolis.

[11] Wijaya, R.C., Andjarwirawan. J., & Palit, H.N. 2016. *Aplikasi Pencarian Produk Jual Mobile Devices dari Berbagai Situs E-Commerce*. Surabaya: Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra.

[12] Pressman, Roger, S. 2001. *Software Engineering: A Practitioner's Approach, Fifth Ed*. New York, McGraw-Hill Book Company Ganesh, K., Mohapatra, S., Anbuudayasankar, S. P., & Sivakumar, P. (2014). User Acceptance Test. In *Enterprise Resource Planning* (pp. 123-127). Springer, Cham.

[13] Pressman, Roger, S. 2005. *Software Engineering: A Practitioner's Approach, Fifth Ed*. New York, McGraw-Hill Book Company.

[14] R. A. Chrismanto, W. S. Raharjo dan Y. Lukito, "Firefox Extension untuk Klasifikasi Komentar Spam pada Instagram Berbasis REST Services,", *Jurnal Edukasi dan Penelitian Informatika (JEPIN),* 5(2).

[15] Mitra, V., Sujaini, H., Bijaksana PN, Arif. (2017). Rancang Bangun Aplikasi *Web Scraping* Untuk Korpus Paralel Indonesia - Inggris Dengan Metode HTML DOM. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN),* 5(1).