

Rancang Bangun Web Service API Aplikasi Jual Beli Barang Bekas Menggunakan Metode REST

Anang Bagus Prakoso^{a1}, I Wayan Santiyasa^{a2}

^aProgram Studi Informatika, Universitas Udayana
Jimbaran, Badung, Bali, Indonesia

¹anangbagus666@gmail.com

²santiyasa@unud.ac.id (Corresponding author)

Abstrak

The development of Information Technology plays a very important role in human life and has made great progress in various aspects such as business. With the development of IT in the business aspect, various applications and services have emerged electronically, one of which is e-commerce. To support the process of data transactions in e-commerce, it is necessary to apply web service technology with safer data transactions, high interoperability, and users can access anytime and anywhere. The architecture used in the design of this research is REST. With the implementation of the REST architecture in the implementation of this research, the percentage of success is 100%. These results are obtained from the Blackbox test on conditions that are in accordance with the functional requirements of each function.

Kata Kunci: Web Service, E-Commerce, REST API

1. Pendahuluan

Dalam beberapa dekade terakhir, Teknologi Informasi (TI) telah memainkan peran yang sangat penting dalam kehidupan manusia dan telah membuat kemajuan besar. Hampir semua aktivitas kita sehari-hari didukung oleh IT, terutama dalam bisnis yang berlangsung di dalam perusahaan, baik di sektor ekonomi maupun perbankan. Hampir semua perusahaan besar maupun kecil menggunakan teknologi informasi untuk mendukung kegiatan yang diperlukan untuk meningkatkan layanan bisnis yang dikelolanya. Salah satu peran yang paling jelas dari teknologi informasi untuk bisnis adalah bahwa semuanya akan dilakukan lebih cepat dan lebih akurat [1].

Perkembangan TI tersebut menciptakan berbagai aplikasi dan layanan secara elektronik seperti: *e-business*, *e-commerce*, *e-banking* dan lainnya. *E-commerce* adalah transaksi perdagangan menggunakan jaringan komunikasi internet [1]. Barang yang diperdagangkan tidak hanya meliputi produksi-produksi baru dari pabrik, melainkan juga barang bekas yang sudah tidak memiliki manfaat bagi pemiliknya.

Untuk mendukung proses transaksi data yang dapat digunakan oleh berbagai perangkat, diperlukan penerapan teknologi *web service*. *Web service* adalah mekanisme interaksi sistem yang mendukung interoperabilitas untuk tujuan mengintegrasikan data yang dapat diakses oleh berbagai pihak melalui internet menggunakan perangkat yang berbeda dari setiap pengguna [2]. Dengan menggunakan teknologi *web service* memungkinkan transaksi data yang lebih aman, interoperabilitas yang tinggi, dan penggunaanya dapat melakukan akses kapanpun dan dimanapun selama perangkat mereka terkoneksi dengan internet [3].

Dalam perancangan *web service* pada penelitian ini akan menggunakan arsitektur *Representational State Transfer* (REST) yang akan menghasilkan sebuah *Application Programming Interface* (API). REST API menggunakan protokol HTTP yang sudah umum digunakan dalam komunikasi antar data [4]. Penggunaan REST API juga memiliki banyak keuntungan, termasuk kecepatan, skalabilitas, dan memungkinkan data dapat diakses oleh banyak *platform* [5]. Beberapa keuntungan tersebut yang mendasari digunakannya arsitektur REST API pada penelitian ini. Dengan memanfaatkan REST API yang cepat dan bersifat skalabilitas, arsitektur ini akan cocok untuk digunakan dalam suatu sistem *e-commerce* yang

mana pada e-commerce tersebut membutuhkan suatu sistem yang dapat diakses dimana saja serta harus cepat dan dapat dikembangkan lagi sesuai dengan perkembangan teknologi.

Berdasarkan uraian latar belakang di atas, maka pada penelitian ini akan dibangun suatu “*Web Service API* untuk Aplikasi Jual Beli Barang Bekas Menggunakan Metode REST” yang diharapkan dapat digunakan sebagai tambahan wawasan bagi masyarakat.

2. Metode Penelitian

Dalam penelitian ini, metode pengembangan sistem yang akan digunakan adalah metode *waterfall*. Model *waterfall* merupakan salah satu *System Development Life Cycle* (SDLC) yang sudah banyak diketahui dan digunakan dalam pengembangan perangkat lunak. Fitur khusus dari model ini adalah langkah-langkah eksekusinya yang berurutan. Ini melewati fase analisis kebutuhan, desain, pengkodean, pengujian dan pemeliharaan. Model ini bekerja dengan baik untuk proyek-proyek di mana kontrol kualitas menjadi perhatian utama karena perencanaan dan dokumentasi yang intensif [6].

2.1 Analisis Kebutuhan

Pada tahap ini, penulis akan melakukan analisis kebutuhan yang meliputi analisa data dan informasi, serta identifikasi masalah. Analisis tersebut dilakukan dengan tujuan agar penulis dapat menetapkan alternative solusi dari hasil identifikasi masalah.

a. Analisis Data dan Informasi

Sebelum melakukan perancangan sistem baru, diperlukan adanya pengumpulan informasi yang jelas terkait kebutuhan yang diperlukan pada sistem. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah studi literatur.

b. Identifikasi Masalah

Berdasarkan analisis yang dijelaskan di atas, ditemukan beberapa hal yang ditetapkan sebagai permasalahan, antara lain:

- Media Promosi
- Pencarian Produk
- Dapat melakukan *Bid* produk
- Dapat digunakan oleh berbagai *platform*

c. Alternatif Solusi

Berdasarkan hasil identifikasi masalah tersebut, maka solusi yang dapat dilakukan untuk mengatasi permasalahan tersebut antara lain:

- Diperlukan sebuah sistem yang dapat digunakan sebagai media promosi.
- Diperlukan sebuah sistem yang dapat melakukan pencarian produk berdasarkan beberapa kriteria.
- Diperlukan sebuah sistem yang dapat melakukan *bid* pada produk yang diinginkan.
- Diperlukan sebuah sistem yang dibangun dengan arsitektur REST API *Web Service* agar sistem dapat digunakan oleh berbagai *platform*.

2.2 Desain Sistem

Perancangan desain sistem pada penelitian ini akan menerapkan *Unified Modelling Language* (UML). UML merupakan salah satu metode visualisasi model yang digunakan dalam melakukan pemodelan dan produksi sebuah perangkat lunak yang berorientasi pada objek. UML adalah standar penulisan atau jenis model yang mencakup proses bisnis [7]. Dalam penelitian ini penulis akan menggunakan dua jenis diagram UML yaitu *Use Case Diagram* dan *Class Diagram*.

2.3 Implementasi Sistem

Pada tahap implementasi, penulis akan membangun sebuah sistem yang baru berdasarkan perancangan desain sistem yang dibuat. Penulis akan membangun sistem berupa *Web Service API* menggunakan *Javascript* dengan memanfaatkan *framework* NodeJs. *Database* yang akan

digunakan pada penelitian ini ialah *PostgreSQL*. Sebagai pembantu dalam melakukan dokumentasi API, penulis akan menggunakan *Swagger* dalam penelitian ini.

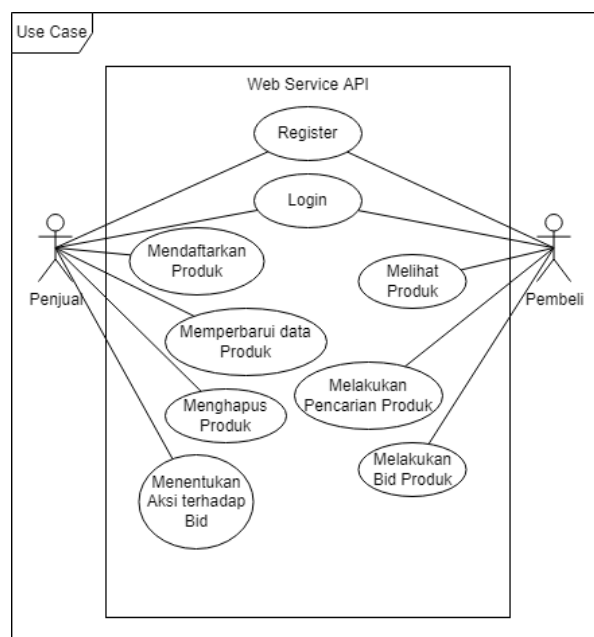
2.4 Uji Coba

Pada tahap uji coba ini, penulis akan melakukan pengujian terhadap sistem baru yang telah dibangun. Dalam proses pengujian tersebut penulis akan menggunakan metode *Blackbox* dalam melakukan pengujian. *Blackbox testing* merupakan salah satu teknik pengujian perangkat lunak yang lebih berfokus pada bagian fungsional dari perangkat lunak tersebut. Metode tersebut memungkinkan pengguna untuk mengumpulkan berbagai macam kondisi input yang akan diujikan pada perangkat lunak untuk melatih syarat-syarat fungsional yang diperlukan [8].

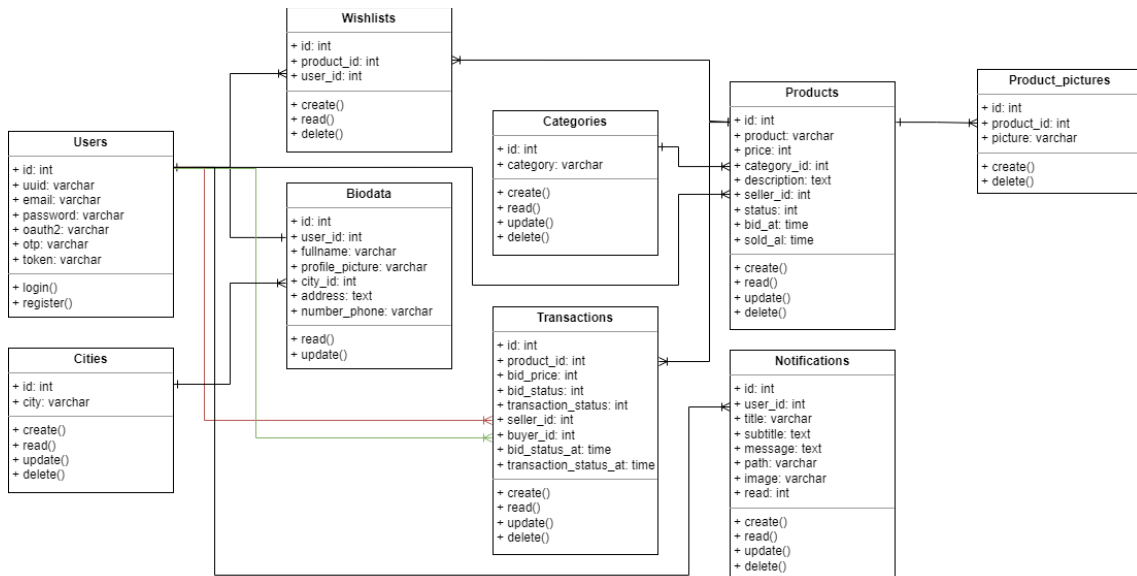
3. Hasil dan Pembahasan

3.1 Perancangan Sistem

Untuk dapat menghasilkan sistem yang sesuai dengan apa yang dibutuhkan dan diharapkan, dilakukan pemodelan sesuai dengan yang dibutuhkan. Pada Gambar 1 diperlihatkan rancangan dari *use case diagram*. Diagram tersebut menggambarkan interaksi-interaksi yang dapat dilakukan masing-masing actor dalam hal ini penjual dan pembeli. Penggambaran *class diagram* seperti Gambar 2 bermaksud untuk menjelaskan struktur sistem yang akan dipakai mulai dari kelas, atribut, metode, dan hubungan antar kelas.



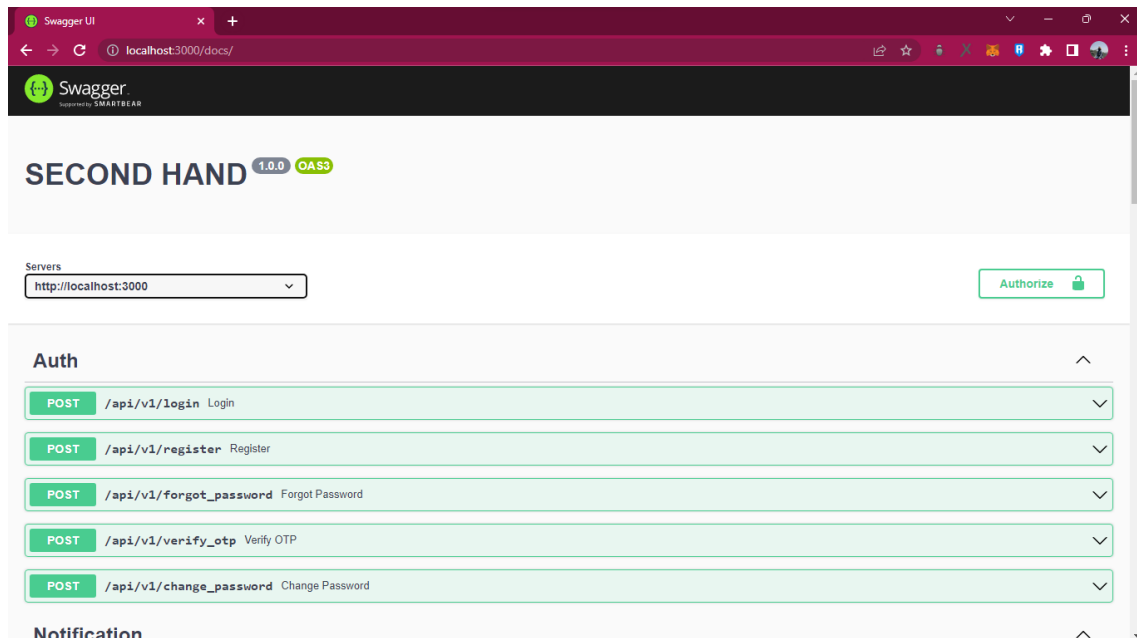
Gambar 1 Use Case Diagram



Gambar 2 Class Diagram

3.2 Implementasi Sistem

Untuk mengimplementasikan rancangan yang sudah dibuat pada perancangan sistem, pada penelitian ini digunakan bahasa pemrograman *Javascript* dengan memanfaatkan *framework Nodejs* dalam pembuatannya. Seluruh kelas yang dibutuhkan dibuat menggunakan *database PostgreSQL*. Hasil dari pembuatan sistem, didokumentasikan menggunakan *Swagger* agar mudah dalam melakukan akses. Pada Gambar 3 diperlihatkan hasil dari sistem berupa dokumentasi *Swagger*.



Gambar 3 Dokumentasi menggunakan Swagger

3.3 Pengujian

Pada tahap ini, penulis akan melakukan uji coba terhadap *web service API* yang sudah dibuat menggunakan metode *Blackbox*. Kondisi-kondisi input yang akan diujikan pada sistem beserta

penarikan kesimpulannya ditampilkan pada Tabel 1 – Tabel 9. Kondisi-kondisi tersebut dibuat berdasarkan fungsi yang akan diuji. Pengujian dilakukan pada dokumentasi API berupa *Swagger*.

a. Registrasi

Request	Hasil Diharapkan	Hasil	Kesimpulan
Body: { "name": "Anang", "email": "anang@gmail.com", "password": "rahasia" }	True	True	Success
Body: { "email": "anang@gmail.com", "password": "rahasia" }	False	False	Success
Body: { "name": "Anang", "email": "anang@gmail.com", "password": "rhs" }	False	False	Success

Tabel 1 Pengujian Registrasi

b. Login

Request	Hasil Diharapkan	Hasil	Kesimpulan
Body: { "email": "anang@gmail.com", "password": "rahasia" }	True	True	Success
Body: { "email": "anang ", "password": "rahasia" }	False	False	Success

Tabel 2 Pengujian Login

c. Mendaftarkan Produk

Request	Hasil Diharapkan	Hasil	Kesimpulan
Body: { "product": "Test Barang", "price": "5000", "category_id": "1", "description": "ini product 6", "status": 1, "pictures": ["uploads/product/pictures-1655900295570-610286145.jpg"] }	True	True	Success

<pre> } Body: { "product": "Test Barang", "price": "5000", "category_id": "1", "description": "ini product 6", "status": 1 } </pre>	False	False	Success
---------------------------------------------------------------------------------------------------------------------------------------------------------------	-------	-------	---------

Tabel 3 Pengujian Mendaftarkan Produk

d. Memperbarui Data Produk

Request	Hasil Diharapkan	Hasil	Kesimpulan
<pre> Parameters: { id: 3 }, Body: { "product": "Test Barang Diubah", "price": "5000", "category_id": "1", "description": "ini product diubah", "status": 1, "pictures": ["uploads/product/pictures-1655900295570-610286145.jpg", "uploads/product/pictures-1655900555616-390879801.jpg"] } </pre>	True	True	Success
<pre> Parameters: { id: 9 }, Body: {} </pre>	False	False	Success

Tabel 4 Pengujian Memperbarui Data Produk

e. Menghapus Produk

Request	Hasil Diharapkan	Hasil	Kesimpulan
<pre> Parameters: { id: 3 }, </pre>	True	True	Success
<pre> Parameters: { id: 9 }, </pre>	False	False	Success

Tabel 5 Pengujian Menghapus Produk

f. Menentukan Aksi terhadap Bid

Request	Hasil Diharapkan	Hasil	Kesimpulan
<pre> Parameters: </pre>	True	True	Success

<pre>{ id: 1 }, Body: { "bid_status": "2" }</pre>			
<pre>Parameters: { id: 9 }, Body: { "bid_status": "2" }</pre>	False	False	Success

Tabel 6 Pengujian Menentukan Aksi terhadap Bid

g. Melihat Produk

Request	Hasil Diharapkan	Hasil	Kesimpulan
<pre>Parameters: { product: "", page: "", limit: "", order: "", category: "" }</pre>	True	True	Success

Tabel 7 Pengujian Melihat Produk

h. Melakukan Pencarian Produk

Request	Hasil Diharapkan	Hasil	Kesimpulan
<pre>Parameters: { product: "", page: "", limit: "", order: "id:DESC", category: "1" }</pre>	True	True	Success
<pre>Parameters: { product: "test", page: "", limit: "", order: "", category: "" }</pre>	True	True	Success

Tabel 8 Pengujian Melakukan Pencarian Produk

i. Melakukan Bid Produk

Request	Hasil Diharapkan	Hasil	Kesimpulan
Body: { "product_id": "1", "bid_price": "100000" }	True	True	Success
Body: { "product_id": "1", "bid_price": "500000" }	False	False	Success

Tabel 9 Pengujian Melakukan Bid Produk

Berdasarkan hasil yang didapatkan dari Tabel 1 – Tabel 9, seluruh kondisi yang diujikan menunjukkan kesimpulan berupa *Success*. Jika dihitung persentase keberhasilannya, maka pada proses pengujian tersebut menghasilkan persentase keberhasilan sebesar 100%.

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, didapatkan hasil persentase keberhasilan sebesar 100%. Dari hasil tersebut dapat dikatakan bahwa aplikasi yang sudah dibuat berjalan dengan baik sesuai dengan perancangan sistem. Sistem yang dibuat juga sudah menerapkan arsitektur REST sehingga sistem dapat diakses oleh berbagai *platform* lainnya.

Referensi

- [1] Business Management Laboratory, "TEKNOLOGI INFORMASI DAN PERKEMBANGAN BISNIS," 2019. <https://bbs.binus.ac.id/bbslab/2019/11/teknologi-informasi-dan-perkembangan-bisnis/> (accessed Sep. 30, 2022).
- [2] F. Surahman, S. H. Al Ikhsan, F. Satrya, and F. Kusumah, "RANCANG BANGUN WEB SERVICE UNTUK TRANSAKSI DATA PADA APLIKASI SAHABAT JASA DENGAN METODE REST," 2018.
- [3] Universitas Dian Nuswantoro, "Web Service," 2018. https://repository.dinus.ac.id/docs/ajar/Sister12_-_Web_services.pdf
- [4] R. Rizal and A. Rahmatulloh, "Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *J. Ilm. Inform.*, vol. 7, no. 01, p. 54, 2019, doi: 10.33884/jif.v7i01.1004.
- [5] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019, doi: 10.30812/matrik.v18i2.407.
- [6] A. Adel and B. Abdullah, "A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model," *IJCSI Int. J. Comput. Sci. Issues*, vol. 12, no. 1, pp. 106–111, 2015, [Online]. Available: https://www.academia.edu/10793943/A_Comparison_Between_Three_SDLC_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model
- [7] M. T. Prihandoyo, "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web," vol. 03, no. 01, pp. 126–129, 2018.
- [8] T. S. Jaya, "Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis," vol. 03, no. 02, pp. 45–48, 2018.