BUILDING API MARKETPLACE FOR STUDENT AT IRIS LABS UNIKOM

Fadli Zul Fahmi

Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia (UNIKOM)

email: xi\_fadli@email.unikom.ac.id

Abstract

Website Developer is one of the jobs that many people are interested in. The number of workers compared to their employment is evidence of many enthusiasts in this work. The number of new innovations in the world of website development emerged along with the habits of people who use this technology. Businesses in this world also appear one by one; ranging from the sale of a static website's source code to the sale of its data / API only. Sales data / API in the world of website development is usually used to add content or functionality on a website someone who can then be given free or paid to the user's website. Usually in the manufacture of this API a place to show its backend developers. In the manufacture of the API is not required focus in terms of appearance (for the user), so that developers can focus to establish only the function of the service.

**Keywords:** api, marketplace, store, marketplace api, information system.

1. PENDAHULUAN

*Website Developer* adalah salah satu pekerjaan yang banyak diminati oleh orang-orang. Banyaknya pekerja dibanding dengan lapangan pekerjaannya menjadi bukti banyaknya peminat dalam pekerjaan ini.

Banyaknya inovasi-inovasi baru dalam dunia pengembangan *website* muncul seiring dengan kebiasaan orang-orang pengguna teknologi ini. Bisnis pada dunia ini pun jenisnya muncul satu per satu; mulai dari penjualan *source code* sebuah *website* statis hingga penjualan *data/ API* nya saja.

Penjualan *data/ API* di dunia pengembangan *website* biasanya digunakan untuk menambahkan konten ataupun fungsi pada sebuah *website* seseorang yang kemudian dapat diberikan gratis ataupun berbayar kepada pengguna *website* nya.

Biasanya dalam pembuatan API ini menjadi tempat unjuk gigi nya para *backend developer*. Dalam pembuatan API tidak diperlukan fokus dalam segi penampilan (untuk pengguna)-nya, sehingga pengembang dapat fokus untuk memantapkan fungsi layanannya saja.

Dalam ranah pembelajaran di suatu universitas, peneliti merasa bahwa dapat meningkatkan minat calon *backend developer* (mahasiswa) untuk memantapkan *skill* mereka ketika melaksanakan studinya. Salah satu caranya dengan membawakan sedikit kompetisi (secara tidak langsung) dan *feel* lapangan pekerjaan kedalam kehidupan mahasiswanya. Oleh karena itu, peneliti ingin membuat sebuah tempat yang dapat dipergunakan mahasiswa untuk menjadi *showcase* dan *store* hasil pekerjaan mereka.

Seperti yang sudah di ceritakan, peneliti ingin membangun sebuah *platform* yang bernama “API-STURE” dengan demikian peneliti tertarik untuk membuat pembahasan dengan judul “*Building API MarketPlace for Student at IRIS Labs UNIKOM”* yang sesuai judulnya lokasi penelitian ini di tujukan pada salah satu organisasi kemahasiswaan dibawah Program Studi Sistem Informasi Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia.

Dalam Kutipan; “Aplikasi Web adalah sebuah halaman Web yang bekerja dengan Bahasa HTML dilengkapi dengan JavaScript” (Grigorik)

Dalam Kutipan; di definisikan oleh Roy T. Fielding “REST adalah gaya arsitektur software untuk aplikasi berbasis jaringan (terkoneksi internet)” (Jeliazkova and Jeliazkov)

Dalam Kutipan; “sebuah spesifikasi masuk kedalam desain RESTful API yang menjelaskan format *representation* yang di izinkan untuk setiap *resource* atau *operation pair*.” (Jeliazkova and Jeliazkov)

Dalam Kutipan; “Sistem Informasi adalah suatu kombinasi yang berasal dari orang, *software*, *hardware*, *network*, dan sumber dara yang terorganisasi dan dapat mengumpulkan, mengubah, dan membagikan informasi dalam suatu organisasi.” (AO’Brien et al.)

* 1. Rumusan Masalah

Jika kita memperhatikan latar belakang pada pendahuluan yang sudah di ceritakan sebelumnya, peneliti dapat menetapkan rumusan masalah sebagai berikut :

1. Bagaimana cara menganalisa hal-hal yang dibutuhkan sistem agar dapat membentuk *platform* API-STURE ?
2. Bagaimana cara pembuatan proses bisnis sistem untuk *showcase* dan *store* hasil kerja mahasiswa dalam membuat API ?
   1. Batasan Masalah
3. API-STURE ini khusus digunakan untuk menampilkan dan memperjual-belikan API buatan mahasiswa UNIKOM.
4. Penelitian ini hanya menjelaskan alur proses bisnis dalam penerapan API-STURE nantinya.
   1. Tujuan
5. Menentukan kebutuhan sistem dengan Analisa kebutuhan agar dapat membentuk *platform* “API Student Store” (API-STURE).
6. Merancang proses bisnis *platform* API-STURE sebagai persiapan dalam pengembangan kedepannya.
   1. Manfaat

Dengan penelitian ini, peneliti harap dapat mendapatkan manfaat untuk :

1. Peneliti

Mendapatkan pengetahuan baru yang sebelumnya tidak diketahui peneliti.

1. Pembaca

Mendapatkan pengetahuan baru dan dapat membayangkan inovasi-inovasi lainnya yang dapat berdampak positif ke masyarakat.

1. Universitas

Dapat merealisasikan penelitian agar membuat lapangan pekerjaan (*freelance*) untuk mahasiswa sehingga *output* dari pekerjaan ini tidak hanya penilaian perkuliahan dan pengalaman mengerjakan tugas, melainkan juga mendapatkan uang hasil dari penjualan pekerjaannya. Selain itu, ini juga dapat menjadi nilai jual Universitas di mata masyarakat umum (maupun perusahaan) dalam memaparkan bukti kemampuan mahasiswanya.

1. METODE

Dengan tujuan memperlancar kegiatan penelitian, peneliti membutuhkan informasi, data penunjang, serta teori pelengkap, maka dibutuhkan beberapa Teknik untuk pengumpulan data sebagai berikut :

1. Pengamatan

Teknik pengumpulan data ini dengan cara mengamati proses bisnis pada beberapa penjualan *online* agar dapat membuat proses bisnis perdagangan yang serupa.

Contoh pengamatan yang dapat dilakukan untuk mendapatkan *sample* proses bisnis penjualan API ini mirip dengan proses bisnis dalam penjualan sebuah *Web Hosting Server* yang dimana *hosting* disewakan kepada penyewa dengan memberikan aksesnya setelah *invoice* dibayar lunas.

Selain itu, contoh yang lebih mendekati adalah seperti proses bisnis pada fasilitas premium yang didapat pada situs forum – forum tempat anak muda nongkrong pada biasanya. Ketika durasi *premium* sudah habis, maka sistem akan otomatis menonaktifkan benefit *premium*nya.

Jadi pada penelitian kali ini, proses bisnis dapat berupa akses ke *proxy* API yang kemudian akan di *authentication* yang bila mana akses di terima server akan meneruskan permintaan ke server pusat.

1. Wawancara

Teknik pengumpulan data ini dilakukan dengan menanyakan ke mahasiswa yang membuat API sebagai contoh untuk mengetahui apa saja yang dibutuhkan oleh orang sepertinya.

Seperti halnya menjadi seorang konsultan IT, pada wawancara ini mencari hal-hal yang dibutuhkan dalam menawarkan sebuah API kepada suatu orang/ perusahaan yang kemudian mereka gunakan sesuai dengan kebutuhan pelanggan.

Apabila benefit yang *platform* ini rencanakan sudah sesuai dengan apa yang mereka (para *developer*) butuhkan, maka dapat dipastikan bahwa *platform* ini akan sukses dan bahkan di cari-cari oleh orang yang jauh sekalipun (meskipun target dari penelitian ini adalah internal kampus unikom itu sendiri).

1. Studi Literatur

Teknik pengumpulan data ini dilakukan dengan cara mencari pembahasan poin-poin penelitian ini yang serupa dan sudah pernah dibahas sebelumnya. Kemudian mengutip pembahasan tersebut dengan tujuan agar memperlancar penelitian kali ini.

Sebagai contohnya literature yang membahas mengenai HTTP dan WEB. Pembahasan mereka akan dikutip untuk memperjelas pembahasan yang ada pada penelitian kali ini.

1. HASIL DAN PEMBAHASAN

Setelah memaparkan alasan mengapa *platform* ini bagus untuk diadakan, pembahasan mengenai hasil akan di paparkan sebagai berikut. Adapun Hasil dan Pembahasan dalam penelitian kali ini sebagai berikut :

* 1. Hasil Penelitian

Melihat tempat penelitian kali ini di Universitas Komputer Indonesia dan terlebih lagi pada Organisasi yang berkategori *study club* dibawah Program Studi Sistem Informasi, pasti adanya calon-calon *backend developer* yang lahir dari sini; sehingga penerapan *platform* ini tidak akan menjadi ‘salah jurusan (tempat)’.

Dari hasil pantauan peneliti, masih banyaknya perusahaan-perusahaan yang belum meng*explore* ke kampus-kampus untuk mencari tahu tingkat kesiapan lulusan setiap kampus. Masih banyaknya perusahaan yang merekrut pegawai berdasarkan dari kampus yang ternama atau dari kampus lulusan atasan yang sama.

Hal ini membuat masalah universitas yang ‘kurang’ dikenal oleh perusahaan-perusahaan yang besar. Karena akan sedikit bahkan tidak ada lulusan yang kerja di perusahaan ternama hingga lulusan akan sangat mengusahakan untuk selalu terjaga dalam mengambil peluang hingga akhirnya perusahaan ternama dapat melihat bakat dari seorang lulusan ini.

Dengan adanya aplikasi ini dan apalagi jika perusahaan ternama melakukan recruitment pegawai menggunakan metode seleksi online atau merekrut orang yang membuat aplikasi perusahaan bekerja (meskipun secara tidak langsung), dengan adanya *platform* ini akan membuat mahasiswa menjadi terkenal oleh orang-orang yang menggunakan API-nya.

Sebagai contoh, mahasiswa “X” membuat sebuah API yang memiliki fungsi untuk *autopost* pada beberapa social media secara bersamaan. Lalu ada sebuah perusahaan penjualan barang *dropshipper* yang melihat *platform* ini dan menemukan produk API *autopost* ini lalu menggunakannya. Secara tidak langsung *developer* yang men*setup* agar aplikasinya berjalan dengan memanfaatkan API dari mahasiswa “X” ini akan mengenali nama mahasiswa ini. Sehingga ketika mahasiswa ini mendaftar di perusahaan pengguna API nya, perusahaan akan langsung mengenali dan akan lebih tertarik dengannya disbanding dengan pelamar yang lainnya.

Selain itu, pada *platform* API-STURE ini memiliki *main function* sebagai berikut :

1. Adanya sistem *login* mahasiswa (penyedia produk) dan *administrator* menggunakan SSO Unikom.
2. Adanya sistem manajemen Produk dan *User* untuk pengelolaan *platform* API-STURE.
3. Adanya sistem pendaftaran dan *login* untuk calon pelanggan (pengguna API).
4. Adanya sistem *listing* API (Produk) yang tersedia untuk di beli (aksesnya).
5. Adanya sistem pembelian API beserta pembayarannya menggunakan API Bank terpusat (di bahas ke satu titik/ bukan per bank).
6. Adanya sistem penghubungan API yang dibuat oleh mahasiswa dengan sistem *proxy* API oleh *platform* API-STURE dengan tujuan *main server* tidak diketahui oleh pelanggan API serta menjadikan manajemen *server* tujuan akses API menjadi satu.
   1. Pembahasan

Dalam pembahasan pada penelitian kali ini, peneliti akan menbahas dengan *use case diagram*.

1. Use Case Diagram



1. Scenario Use Case

Tabel 3.1 Skenario Use Case Login Pembeli

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Login Pembeli** | |
| **Deskripsi Singkat** | | Aktor masuk kedalam sistem. | |
| **Aktor** | | Pembeli | |
| **Kondisi Awal** | | Aktor belum melakukan autentikasi dan sudah mendaftar. | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | Mengakses halaman Login |  |  |
| 2 | Mengisikan kombinasi *username* dan *password* |  |  |
| 3 | Menekan tombol Login | 4 | Mengkonfirmasi kombinasi benar atau salah. Jika benar, maka akan mengalihkan ke halaman *dashboard* pembeli. |
| **Kondisi Akhir** | | Aktor telah *login*. | |

Tabel 3. 2 Skenario Use Case Daftar Sebagai Pembeli

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Daftar Sebagai Pembeli** | |
| **Deskripsi Singkat** | | Aktor mendaftar keanggotaan kedalam sistem. | |
| **Aktor** | | Pembeli | |
| **Kondisi Awal** | | Aktor belum pernah melakukan pendaftaran. | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | Mengakses halaman Pendaftaran. |  |  |
| 2 | Mengisikan Form Pendaftaran |  |  |
| 3 | Menekan tombol Daftar | 4 | Memverifikasi user sudah terdaftar sebelumnya atau belum. Jika belum, akan diarahkan ke halaman pendaftaran berhasil. |
| **Kondisi Akhir** | | Aktor telah berhasil mendaftar. | |

Tabel 3.3 Skenario Use Case Manajemen Akun

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Manajemen Akun** | |
| **Deskripsi Singkat** | | Aktor melakukan pengaturan terhadap Akun yang sudah terdaftar. | |
| **Aktor** | | Admin | |
| **Kondisi Awal** | | Aktor sudah login. | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | Menekan tombol Manajemen Akun |  |  |
| 2 | Memilih dan menekan tombol *action* yang di inginkan; seperti rubah data pelanggan | 3 | Menampilkan Form rubah data *user* yang di pilih. |
| 4 | Mengisikan *Form* rubah data |  |  |
| 5 | Menekan tombol Ubah Data | 6 | Memverifikasi data; Jika sesuai format, maka akan memproses perubahan dan mengalihkan ke halaman ubah data sukses. |
| **Kondisi Akhir** | | Aktor telah berhasil memanajemen akun. | |

Tabel 3.4 Skenario Use Case Daftar Sebagai Penjual

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Daftar Sebagai Penjual** | |
| **Deskripsi Singkat** | | Aktor mendaftar dirinya untuk menjadi Penjual di *platform* ini. | |
| **Aktor** | | Mahasiswa | |
| **Kondisi Awal** | | Aktor belum pernah mendaftar. | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | menekan tombol Daftar Sebagai Penjual. | 2 | Mengalihkan ke halaman *authentication* dengan akun SSO UNIKOM |
| 3 | memasukkan kombinasi *username* dan *password* |  |  |
| 4 | menekan tombol *auth now.* | 5 | memverifikasi akses. Jika benar mahasiswa, akan memproses pendaftaran dan di alihkan ke halaman pendaftaran berhasil. |
| **Kondisi Akhir** | | Aktor telah berhasil mendaftarkan dirinya sebagai penjual. | |

Tabel 3.5 Skenario Use Case Pembayaran

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Pembayaran** | |
| **Deskripsi Singkat** | | Aktor membayar tagihan pembelian akses API. | |
| **Aktor** | | Pembeli | |
| **Kondisi Awal** | | Aktor sudah memilih Produk yang hendak di beli aksesnya. | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | menekan tombol *checkout* | 2 | memverifikasi produk yang sudah dipilih dan mengalihkan ke halaman *invoice* dan detil rekening. |
| 3 | mengirimkan pembayaran ke rekening yang disuruh. |  |  |
| 4 | menekan tombol *check* pembayaran. | 5 | memverifikasi pembayaran dan memproses pembayaran beserta pengiriman akses *credential* API |
| **Kondisi Akhir** | | Aktor telah mendapatkan akses API. | |

Tabel 3.6 Skenario Use Case Membeli Produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Membeli Produk** | |
| **Deskripsi Singkat** | | Aktor memilih Produk yang ingin di beli sebelum dibayar. | |
| **Aktor** | | Pembeli | |
| **Kondisi Awal** | | Aktor sudah *login* | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | Memilih produk yang hendak dibeli |  |  |
| 2 | menekan tombol *Add to Cart* | 3 | Menyimpan data produk yang dipilih. |
| **Kondisi Akhir** | | Aktor telah menambahkan Produk kedalam *cart* akunnya. | |

Tabel 3.7 Skenario Use Case Mengakses Produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Mengakses Produk** | |
| **Deskripsi Singkat** | | Aktor mengakses API dengan Kredential yang sudah diberikan. | |
| **Aktor** | | Pembeli | |
| **Kondisi Awal** | | Aktor sudah membeli akses ke API yang digunakan. | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | Mengirimkan sinyal berupa kombinasi kredensial yang diberikan ketika *order*. | 2 | Sistem memastikan kombinasi kredensial terdaftar di sistem dan dalam keadaan aktif. |
|  |  | 3 | Sistem memberikan token yang berlaku dalam beberapa waktu. |
| **Kondisi Akhir** | | Aktor mendapatkan Token yang dapat digunakan untuk mengakses API pusat *via proxy*. | |

Tabel 3.8 Skenario Use Case Melihat Produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Melihat Produk** | |
| **Deskripsi Singkat** | | Aktor melihat-lihat produk yang tersedia untuk dibeli. | |
| **Aktor** | | Pembeli | |
| **Kondisi Awal** | | Aktor mengakses halaman utama *platform.* | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | Menekan tombol *Product Listing*. | 2 | Menampilkan halaman daftar Produk yang tersedia untuk di beli. |
| **Kondisi Akhir** | | Aktor mendapatkan daftar Produk yang tersedia untuk di beli. | |

Tabel 3.9 Skenario Use Case Manajemen Produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama *Use Case*** | | **Manajemen Produk** | |
| **Deskripsi Singkat** | | Aktor melakukan Penambahan/ Pengurangan/ Perubahan/ Penghapusan Produk. | |
| **Aktor** | | Mahasiswa | |
| **Kondisi Awal** | | Aktor sudah *login.* | |
| **Skenario Normal** | | | |
|  | **Aksi Aktor** |  | **Reaksi Sistem** |
| 1 | Menekan tombol Manajemen Produk | 2 | Menampilkan halaman Daftar Produk Yang Dikelola. |
| 3 | Memilih Produk yang ingin di manajemen. |  |  |
| 4 | Memilih tombol *action* yang ingin dilakukan; seperti tombol ubah. | 5 | Menampilkan halaman ubah data produk yang di pilih. |
| 6 | Mengisikan *Form* perubahan data produk. |  |  |
| 7 | Menekan tombol Ubah Data Produk | 8 | Memproses perubahan data Produk. |
| **Kondisi Akhir** | | Aktor telah melakukan manajemen Produk. | |

1. KESIMPULAN

Berdasarkan apa yang sudah dibahas pada penelitian kali ini, dapat ditarik kesimpulan pada penelitian kali ini bahwa :

1. Dengan pembuatan *platform* ini, dapat meningkatkan minat dan ketertarikan mahasiswa untuk terus meng*explore* pengetahuan terlebih pengetahuan yang membahas pembuatan, pengubahan, dan penggunaan serta dokumentasi API keseluruhan.
2. Hasil kerja mahasiswa dapat terdokumentasi dengan baik dan memiliki nilai jual.
3. Dapat meningkatkan nilai jual universitas di khalayak umum terlebih perusahaan-perusahaan yang sedang mencari calon pegawai *fresh graduate*.
4. REFERENSI

AO’Brien, J., et al. “Management Information Systems.” *International Encyclopedia of Educational Technology*, vol. 4, no. 2, 2007, pp. 308–12, doi:10.1108/eb000831.

Grigorik, I. “High Performance Browser Networking.” *Journal of Chemical Information and Modeling*, vol. 53, 2013, pp. 1–383, doi:10.1017/CBO9781107415324.004.

Jeliazkova, Nina, and Vedrin Jeliazkov. “AMBIT RESTful Web Services: An Implementation of the OpenTox Application Programming Interface.” *Journal of Cheminformatics*, vol. 3, no. 1, 2011, doi:10.1186/1758-2946-3-18.

.