PEMBANGUNAN *WEB SERVICE RESTFUL PARTNER* API TERINTEGRASI DENGAN *EPP API* DARI PANDI UNTUK REGISTRASI *CCTLD* INDONESIA DI QWORDS

Saepudin Mulyono[1], Eko Budi Setiawan[2],……..

[1,2] Program Studi Teknik Informatika, Universitas Komputer Indonesia
Jl. Dipatiukur No 112-116 Bandung. 40132
E-mail : saepudinmulyono@gmail.com[1], eko@email.unikom.ac.id[2]

ABSTRAK

PT Qwords Company International atau Qwords merupakan perusahaan yang menjual nama domain/ccTLD Indonesia kepada *reseller*. Selama ini proses registrasi nama domain/ccTLD Indonesia dilakukan secara manual dengan cara *staff reseller* Qwords meminta kepada *staff registrar* Qwords melalui *email* kemudian *staff registrar login* dan memproses registrasi setiap ccTLD Indonesia di aplikasi PANDI satu persatu sehingga prosesnya lama dan juga bergantung pada jam kerja *staff registrar* Qwords.

Web service mendukung interaksi dan interoperabilitas antar sistem pada suatu jaringan dan EPP API merupakan *web service* untuk memproses registrasi nama domain/ccTLD Indonesia yang disediakan oleh *registry* PANDI hanya kepada *registrar*. Dalam perancangan *web service* ini, penulis menggunakan EPP API untuk menghubungkan aplikasi *reseller* dengan *server* PANDI sebagai solusi masalah dalam memudahkan dan mempercepat proses registrasi nama domain/ccTLD Indonesia.

Hasil dari penelitian ini adalah dengan menggunakan aplikasi ini maka *staff reseller* dan *staff registrar* dapat memproses registrasi nama domain/ccTLD Indonesia lebih mudah, lebih cepat dan tidak bergantung lagi pada jam kerja *staff registrar* Qwords.

Kata kunci : *Application Programming Interface(*API), *Extensible Provisioning Protocol*(EPP), *Country Code Top-Level* Domain(ccTLD), *Restful*, *Web Service*

1. PENDAHULUAN

Nama Domain Indonesia atau ccTLD Indonesia sudah tidak asing lagi bagi pengguna internet, dapat kita temukan banyak nama domain Indonesia yang terindeks di mesin pencari google, ada yang dipakai di *website* pemerintah, toko *online*, company *profile* bahkan *blog* pribadi.

Salah satu perusahaan yang menjual nama domain Indonesia adalah PT Qwords Company International dengan nama *Registrar* Qwords. Pada tahun 2016 lalu baru saja mendaftar menjadi salah satu *Registrar* dari institusi yang menjadi *server* pusat untuk registrasi ccTLD Indonesia yaitu PANDI, salah satu manfaat dari menjadi *registrar* adalah *registrar* Qwords mendapatkan akses ke EPP API server pusat PANDI untuk melakukan registrasi domain langsung dari website yang dimiliki oleh *registrar* Qwords supaya proses registrasi menjadi lebih cepat.

Namun *registrar* Qwords ternyata belum merasakan manfaat tersebut karena *registrar* Qwords masih melakukan proses registrasi nama domainnya secara *manual*, dimana *staff registrar* Qwords diharuskan *login* ke *server* pusat lalu meng*input*kan kembali satu persatu data pemesan dan pesanannya ke *server* pusat PANDI. Berdasarkan uji coba yang dilakukan penulis selama 10 kali, rata-rata lama waktu pemrosesan yang saat ini diperlukan untuk satu nama domain adalah 6 menit, padahal domain bersifat "siapa cepat dia dapat" dan "satu nama domain untuk satu orang" sehingga setiap orang seperti berlomba untuk mendapatkan domain dengan nama yang mereka sukai atau mereka butuhkan oleh karena itu adalah hal yang berbahaya jika pemesan yang sudah membayar tidak berhasil mendapatkan domain yang dipesannya dikarenakan lambatnya proses tersebut dapat mengakibatkan domain telah di*register* oleh perusahaan lain.

Selain itu PT. Qwords Company International disini juga bertindak sebagai *eeseller* Domain dari dirinya sendiri sebagai *registrar* Qwords, sehingga pada dasarnya Qwords membeli domain kepada *registrar* Qwords kemudian menjualnya kepada *client* nya sambil menjual produk utamanya yaitu *web hosting*. *Reseller* Qwords dapat memasarkan domain Indonesia kemudian menginformasikannya ke *Registrar* Qwords sehingga meningkatkan penjualan domain di *Registrar* Qwords. Sayangnya dengan proses yang masih manual, penjualan Qwords tidak dapat maksimal karena proses yang panjang dan bergantung pada waktu dan ketersediaan staff, dimana setelah ada pemesan melakukan pemesanan ke website *Reseller* Qwords kemudian *staff Reseller* Qwords melakukan pemesanan ke Registrar Qwords melalui email lalu *staff Registrar* Qwords perlu *login* ke *server* pusat PANDI untuk mengisi *detail* pesanan yang ada pada *email* tersebut, apabila *staff registrar* Qwords sudah berakhir jam kerjanya atau sedang tidak dalam jam kerja padahal *staff reseller* Qwords

saat itu sudah mengirim *email* pesanan maka proses registrasi domain akan diproses keesokan harinya.

Berdasarkan latar belakang diatas, diperlukan sistem yang mendukung komunikasi antara aplikasi *client* dalam hal ini adalah *Reseller* Qwords dan *server* dalam hal ini adalah *server* pusat PANDI. Selain itu juga dibutuhkan aplikasi untuk pembelian online berupa *website* untuk mempermudah pemesan/*Reseller* Qwords dalam memesan nama domain.

Adapun tujuan dari penelitian ini adalah :

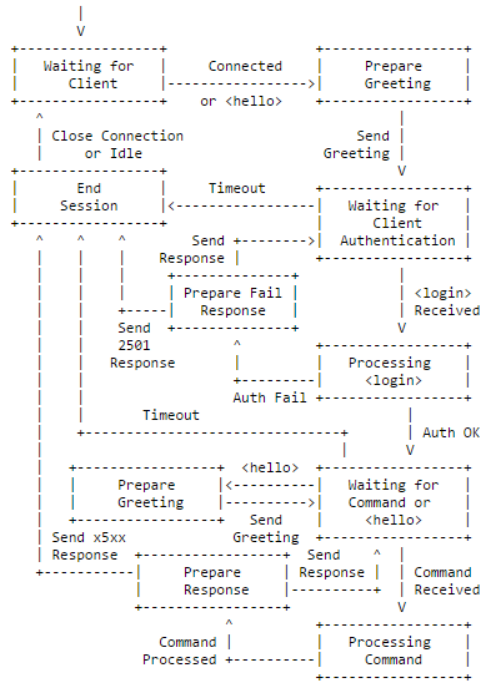
1. Mempercepat dan mempermudah proses registrasi ccTLD Indonesia
2. Menyediakan Sistem yang dapat melayani proses registrasi ccTLD kapan saja dibutuhkan tanpa bergantung pada ada atau tidaknya *staff Registrar* Qwords.

2. ISI PENELITIAN

2.1 *Extensible Provisioning Protocol* (EPP)

EPP atau *Extensible Provisioning Protocol* adalah protokol XML *stateful* yang dapat berlapis terdiri atas beberapa protokol *transport*. Dilindungi dengan menggunakan protokol keamanan-lapisan bawah, klien bertukar identifikasi, otentikasi, dan pilihan informasi, dan kemudian terlibat dalam serangkaian perintah-perintah klien yang diprakarsai pertukaran respon. Semua perintah EPP adalah *atomic* (tidak ada parsial keberhasilan atau parsial kegagalan) dan dirancang sehingga mereka dapat dibuat idempoten (mengeksekusi perintah lebih dari sekali memiliki net yang sama berpengaruh pada keadaan sistem seperti berhasil mengeksekusi perintah sekali).

Server EPP harus menanggapi komunikasi yang diinisiasi oleh *client* (yang dapat berupa permintaan sambungan-lapisan bawah atau *service discovery*) dengan me-*response* proses *greeting* kepada client. Sebuah *server* harus segera menanggapi setiap perintah EPP dengan respon yang terkoordinasi yang menggambarkan hasil dari pengolahan perintah [1]. Berikut ini diagram *Server* mesin yang menggambarkan proses pertukaran pesan secara terperinci pada Gambar 1:



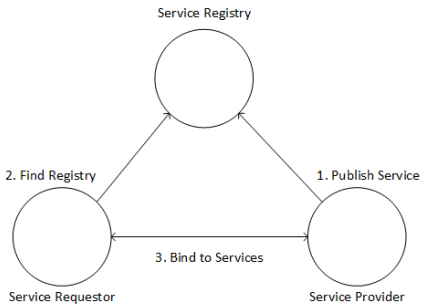
Gambar 1. EPP *Server State Machine*

2.2 *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interaction and *interoperability* antar sistem pada suatu jaringan.

Web service digunakan sebagai suatu fasilitas yang menyediakan layanan (dalam bentuk informasi atau data) kepada sistem lain, sehingga dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan. web service menyimpan data informasi dalam format JSON atau XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, dan bahasa pemrograman [2].

Secara umum, arsitektur *web service* dapat dilihat pada gambar 2 berikut ini:



Gambar 2. Arsitektur *Web Service*

Pada gambar diatas, ada tiga komponen utama dari *web service* yaitu:

1. *Service Provider* adalah penyedia *web service* yang berfungsi menyediakan kumpulan *web service* yang dapat diakses oleh pengguna.
2. *Service Requestor* adalah aplikasi yang bertindak sebagai pengguna yang melakukan permintaan layanan (berupa *web service*) ke *service provider*

3. *Service Registry* adalah tempat dimana *service provider* mempublikasikan layanannya. Pada arsitektur *web service*, *Service registry* bersifat opsional. [3]

2.3 Restful

REST adalah filosofi desain yang mendorong kita untuk menggunakan protokol dan fitur yang sudah ada pada *web* untuk memetakan permintaan terhadap sumber daya pada berbagai macam representasi dan manipulasi data di Internet (Scribner, 2009).

REST adalah gaya arsitektural yang memiliki aturan seperti antar muka yang seragam, sehingga jika aturan tersebut diterapkan pada *web service* akan dapat memaksimalkan kinerja *web service* terutama pada performa, skalabilitas, dan kemudahan untuk dimodifikasi. Pada arsitektur REST data dan fungsi dianggap sebagai sumber daya yang dapat diakses lewat *Uniform Resource Identifier* (URI), biasanya berupa tautan pada *web*.

REST menggunakan protokol HTTP yang bersifat *stateless*. Perintah HTTP yang bisa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE*. Hasil yang dikirimkan dari *server* biasanya dalam bentuk format XML atau JSON sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan di*parsing* disisi client.

Dalam penerapannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada sumber daya adalah orientasi yang menyediakan sumber daya sebagai layanannya dan bukan kumpulan-kumpulan dari aktifitas yang mengolah sumber daya itu. Bentuk *web service* menggunakan REST *style* sangat cocok digunakan sebagai *backend* dari aplikasi berbasis *mobile* karena cara aksesnya yang mudah dan hasil data yang dikirimkan berformat JSON sehingga ukuran *file* menjadi lebih kecil [3].

2.4 *Application Programming Interface*(API)

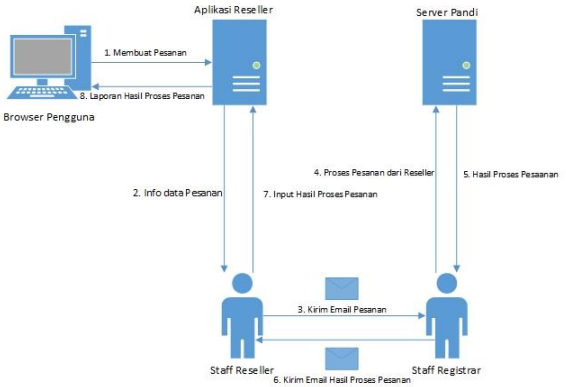
API adalah singkatan dari *application programming interface*. API dapat memberikan keterkaitan antara rekan kerja, mitra, atau pengembang pihak ketiga untuk mengakses data dan layanan untuk membangun aplikasi seperti aplikasi iPhone dengan cepat. Twitter dan Facebook API adalah contoh yang terkenal. Ada API yang terbuka untuk pengembang mana pun, API yang hanya terbuka untuk mitranya, dan API yang digunakan secara *internal* untuk membantu menjalankan bisnis dengan lebih baik dan memudahkan kolaborasi antar tim. [4].

Dalam pendekatan *partner* API, penyedia API menggunakan API untuk mengintegrasikan dan mengotomatisasi interaksi dengan mitra-mitranya. Pendekatan ini populer terutama di kalangan bisnis yang beroperasi di lingkungan pasar *business*-to-*business*. Dokumentasi yang terdefinisi dengan baik akan memudahkan integrasi proses bisnis saat hubungan bisnis baru sedang dibangun. API mengurangi kebutuhan akan penyesuaian integrasi sistem-ke-sistem dan memudahkan komunikasi antara mitra yang telah dibangun dan didokumentasikan terlebih dahulu dan diimplementasikan menggunakan teknologi yang standar. [5]

2.5 Analisis Sistem yang Sedang Berjalan

Analisis sistem merupakan gambaran tentang sistem yang saat ini sedang berjalan di PT Qwords Company International. Analisis yang penulis lakukan ini terdiri dari analisis objek sistem yang berjalan, mencari kelemahan sistem yang berjalan sesuai standar yang telah ditentukan, serta mengevaluasi sistem yang sedang berjalan sehingga menghasilkan usulan rancangan sistem yang baru. Analisa sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian komponen-komponennya dengan maksud mengidentifikasi dan mengevaluasi kekurangan-kekurangan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan – perbaikannya.

Sistem yang sedang berjalan dapat dilihat pada gambar 3 berikut ini



Gambar 3. Arsitektur sistem yang sedang berjalan

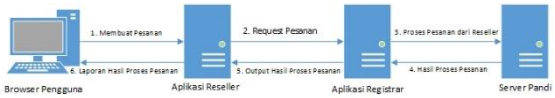
Berdasarkan hasil wawancara dengan *staff Registrar* Qwords dan *Reseller* Qwords, prosedur pemesanan nama domain yang sedang berjalan di PT Qwords Company International adalah sebagai berikut :

1. Pengguna mengakses aplikasi *reseller* kemudian melakukan pesanan nama domain
2. Informasi data pesanan dicek dan dicatata oleh *staff reseller* pada dokumen excel
3. *Staff Reseller* mengirimkan email yang berisi detail pesanan nama domain ke alamat email milik *Staff Registrar*
4. *Staff Registrar* memproses pesanan dari *Reseller* di *Server* Pandi
5. *Staff Registrar* mendapat hasil pemroses pesanan dari *Server* Pandi
6. *Staff Registrar* mengirim hasil pemrosesan pesanan ke alamat *email Staff Reseller*
7. *Staff Reseller* menginputkan hasil pemroses pesanan ke aplikasi *Reseller*
8. Laporan mengenai pesanan dapat diterima/baca oleh Pengguna

2.6 Rancangan Usulan Sistem

Sesuai dengan analisis sistem yang sedang berjalan saat ini, maka sistem baru yang akan dibangun merupakan sistem pemesanan ccTLD Indonesia pada sisi *reseller* Qwords yang terhubung ke *registrar* Qwords dan terintegrasi dengan EPP API dari PANDI, terdiri dari: aplikasi *reseller* Qwords yang digunakan oleh pengguna, aplikasi *registrar* Qwords, dan *Server* PANDI

Berikut adalah deskripsi dari aplikasi pemesanan nama domain ccTLD Indonesia di PT Qwords Company International, seperti yang terlihat pada Gambar 4 berikut ini

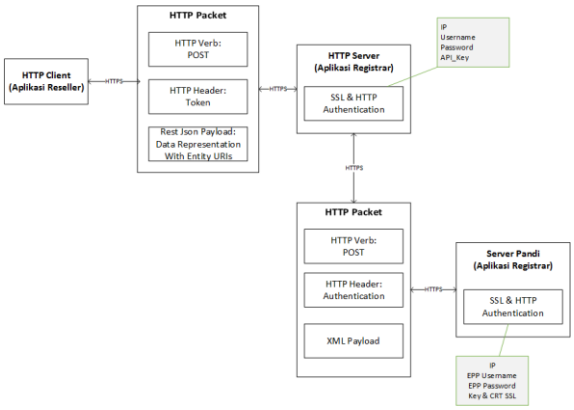


Gambar 4. Arsitektur Sistem yang akan dibangun

Berikut penjelasan dari analisis arsitektur sistem.

1. Pemesanan nama domain akan dilakukan menggunakan aplikasi *reseller* Qwords yang digunakan oleh pengguna
2. Pemesanan nama domain oleh Pengguna dikirimkan ke aplikasi *registrar* Qwords melalui *web service resful*
3. Aplikasi *registrar* Qwords mengolah request *web service* dengan mengecek saldo deposit yang ada pada akun *reseller* Qwords, jika mencukupi *registrar* Qwords mengirimkan *request* ke *server* PANDI melalui *web service*/EPP API dari PANDI, jika tidak maka r*egistrar* Qwords mengembalikan hasil *parsing* data berupa informasi kurangnya saldo pada akun *reseller* Qwords
4. *Server* PANDI akan memproses permintaan pesanan domain kemudian mengeluarkan *response* berupa informasi terkait nama domain yang dipesan
5. Aplikasi *registrar* Qwords menyimpan informasi terkait nama domain kemudian mengirimkannya ke aplikasi r*eseller* Qwords
6. *Client* mendapat notifikasi hasil proses pemesanan

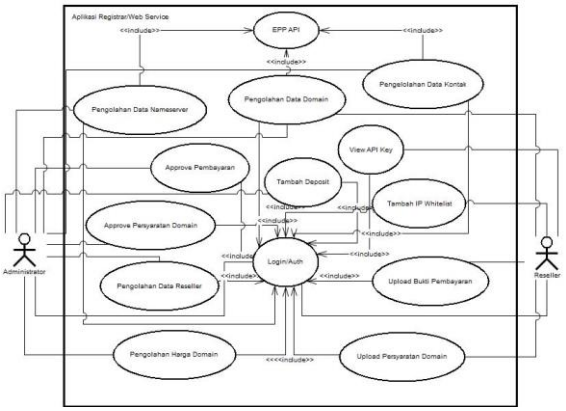
Sistem yang akan dibangun merupakan sistem pemesanan ccTLD Indonesia. Untuk aplikasi berbasis *web* milik *reseller* Qwords agar dapat melakukan pemesanan dan meng*input*kan pesanan oleh Pengguna, maka harus disediakan perangkat yang dapat men*transfer* data antara aplikasi *web* milik *reseller* Qwords dengan aplikasi s*erver* Pandi dalam hal ini *web based*, dalam hal ini *web service* yang dapat dilihat arsitekturnya pada gambar 5 berikut ini



Gambar 5. Arsitektur *Web Service*

2.7 *Use Case Diagram*

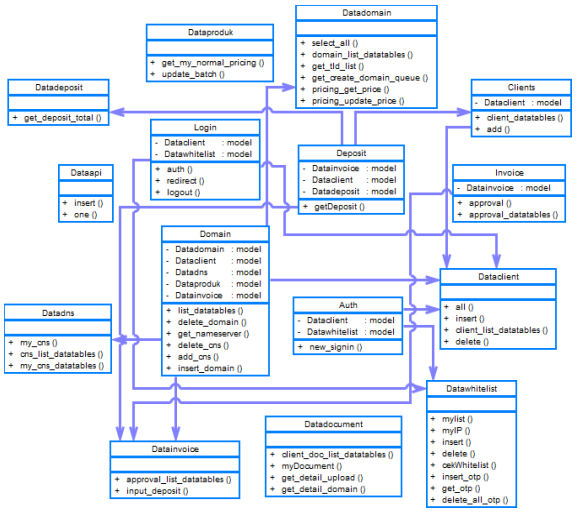
Use case diagram digunakan untuk memodelkan proses bisnis dan merepresentasikan sebuah interaksi antara aktor dengan sistem. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, *use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebauh sistem dipakai, *use case diagram* dapat dilihat pada gambar 6 berikut ini



Gambar 6. *Use Case Diagram* Aplikasi *Registrar*/*Web Service*

2.8 *Class Diagram*

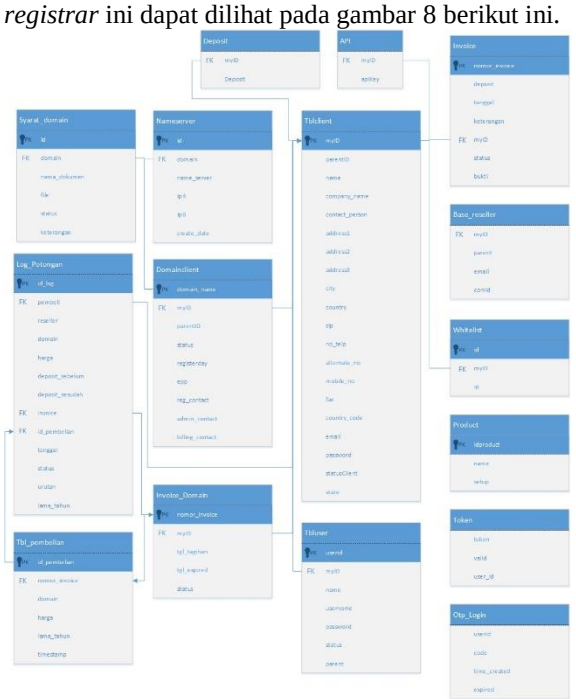
Class diagram mendeskripsikan jenis–jenis obyek dalam sistem dan berbagai macam hubungan statis yang terjadi. *Class diagram* juga menunjukkan *property* dan operasi sebuah *class* dan batasan yang terdapat dalam hubungan dengan objek. *Class* menggambarkan keadaan (attribut atau *property*) suatu sistem sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). *Class* d*iagram* dari sistem yang akan dibangun dapat dilihat pada gambar 7 berikut ini.



Gambar 7. Class Diagram

2.9 Skema Relasi

Skema relasi atau proses relasi antar *file* merupakan gabungan antar *file* yang mempunyai kunci utama yang sama, sehingga *file-file* tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci tersebut. Pada proses ini elemen-elemen data dikelompokan menjadi satu *file database* beserta entitas dan hubungannya. Skema relasi aplikasi *registrar* ini dapat dilihat pada gambar 8 berikut ini.



Gambar 8. Skema Relasi Sistem yang Akan Dibangun

2.10 Implementasi Perangkat Keras

Perangkat keras yang digunakan pada saat proses implementasi dari sistem web service / aplikasi Registrar terdapat 2 bagian yaitu komputer lokal dan komputer server:

1. Perangkat Keras Komputer Lokal

Spesifikasi perangkat keras komputer lokal dapat dilihat pada table 1 berikut ini.

Tabel 1. Implementasi Perangkat Keras Komputer Lokal

No	Perangkat	Spefisikasi
1	Prosesor	Intel® Core™ CPU i5 @ 2.60GHz
2	*Memory*	RAM 16GB
3	VGA *Card*	2 GB
4	*Harddisk*	1,5 TB
5	*Monitor*	Layar 14 Inch
6	Koneksi Internet	10 Mbps

2. Perangkat Keras Komputer Server

Spesifikasi perangkat keras komputer *server* dapat dilihat pada table 2 berikut ini.

Tabel 2. Implementasi Perangkat Keras Komputer *Server*

No	Perangkat	Spefisikasi
1	Prosesor	Intel® Core™ Xeon® CPU E5-2620 0 @ 2.00GHz
2	*Memory*	RAM 2GB
3	VGA *Card*	256 MB
4	*Harddisk*	200 GB
5	Koneksi Internet	1 Gbps

2.11 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan pada saat proses implementasi dari sistem *web service* / aplikasi *registrar* terdapat 2 bagian yaitu komputer lokal dan komputer *server*:

1. Perangkat Lunak Komputer Lokal

Spesifikasi perangkat keras komputer lokal dapat dilihat pada table 3 berikut ini.

Tabel 3. Implementasi Perangkat Lunak Komputer Lokal

No	Perangkat	Spefisikasi
1	Sistem Operasi	Windows 7 Ultimate
2	*Browser*	*Browser* Google Chrome
3	*Source Code Editor*	Notepad ++

2. Perangkat Lunak Komputer Server

Spesifikasi perangkat keras komputer lokal dapat dilihat pada table 4 berikut ini.

Tabel 4. Implementasi Perangkat Lunak Komputer Server

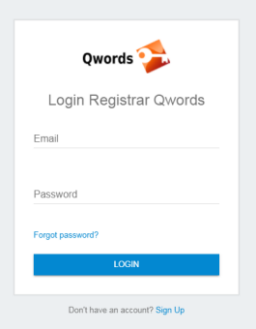
No	Perangkat	Spefisikasi
1	Sistem Operasi	Linux Centos 6.8 64 Bit
2	*Control Panel*	Centos Web Panel 0.9.8.249 (Untuk aplikasi *registrar*)

No	Perangkat	Spefisikasi
		Cpanel (Pada aplikasi *reseller*)
3	*Web Server*	Apache
4	*Database Server*	MySQL

2.11 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap halaman aplikasi *registrar* yang dibangun menggunakan *web*. Berikut ini beberapa tampilan antar muka yang telah di implementasikan.

Login adalah halaman aplikasi registrar/*dashboard* dapat dilihat pada gambar 9 berikut ini



Gambar 9. Halaman *Login*

Pada halaman *Login* terdapat 3 fungsi yaitu:

a. Login

Button Login di klik akan melakukan proses auth yang jika isian *email* dan *password* dan IP yang digunakan sesuai maka akan membawa pengguna ke halaman *home*/beranda *dashboard*

b. Forgot Password

Forgot Password disediakan untuk pengguna yang lupa *password* akun nya, jika di klik akan membawa pengguna ke *form* yang berfungsi untuk mengirim ulang *password* ke *email* pengguna.

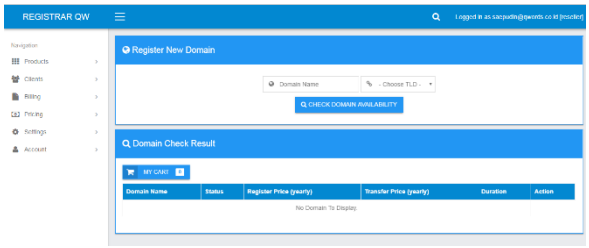
c. *Sign Up*

Sign Up disediakan untuk pengguna baru yang belum mempunyai akun. Jika di klik akan membawa pengguna ke *form* pendaftaran pengguna.

Register domain adalah halaman aplikasi/*dashboard* untuk *web service* /rest/register_domain yang terdiri dari 3 halaman yaitu

a. *Check Availability Domain*

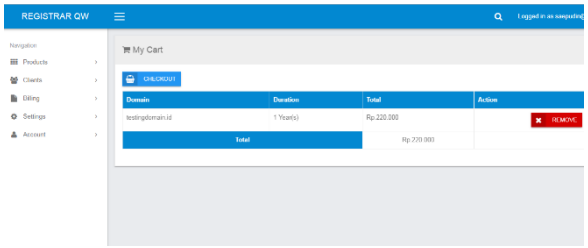
Halaman ini digunakan untuk mengecek ketersediaan domain sehingga pengguna dapat memilih domain yang tersedia untuk dipesan, halaman *check availability domain* dapat dilihat pada gambar 10 berikut ini



Gambar 10. Halaman *Check Availability Domain*

b. *My Cart*

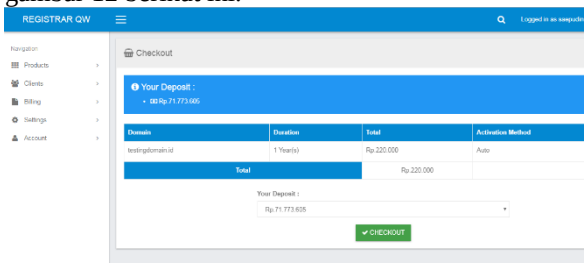
Halaman ini digunakan untuk mengecek pesanan domain dan harganya sehingga pengguna dapat mengetahui total biaya yang diperlukan dan memilih domain yang akan dihilangkan dari pesanan, halaman *my cart* dapat dilihat pada gambar 11 berikut ini



Gambar 11. Halaman *My Cart*

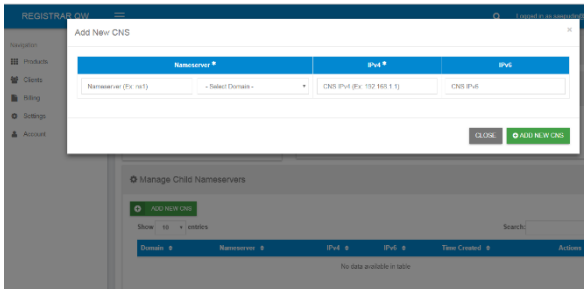
c. *Checkout*

Halaman ini digunakan untuk proses konfirmasi pengaktifan domain sehingga pengguna dapat mengaktifkan domain yang telah di cek di halaman sebelumnya. halaman *checkout* dapat dilihat pada gambar 12 berikut ini.



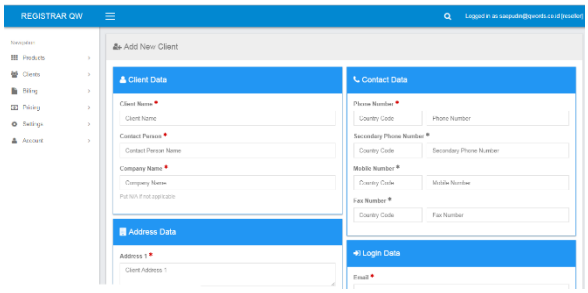
Gambar 12. Halaman *Checkout*

Add child nameserver adalah halaman aplikasi/*dashboard* berupa *form* yang dapat dilihat pada gambar 13 berikut ini



Gambar 13. Halaman *Add New* CNS

Add clients adalah halaman aplikasi/*dashboard* berupa *form* yang dapat dilihat pada gambar 14 berikut ini



Gambar 14. Halaman *Add Clients*

2.12 Pengujian Kecepatan Proses Domain

Pengujian kecepatan proses registrasi domain dilakukan untuk mengetahui apakah sistem yang sudah terintegrasi dengan *web service*/aplikasi *registrar* Qwords dapat mengatasi masalah lambat dan sulitnya proses registrasi ccTLD Indonesia pada sistem yang berjalan sebelumnya.

Proses pengujian dilakukan 10 kali dengan cara melakukan proses pendaftaran dan pemesanan nama domain dengan metode pembayaran melalui deposit/saldo pada akun, pengujian diukur menggunakan *stopwatch* sehingga ditemukan waktu rata-rata yang diperlukan untuk melakukan proses registrasi ccTLD Indonesia. hasil pengujian dapat dilihat pada tabel 5.

Tabel 5. Hasil Pengujian Kecepatan Proses Registrasi Domain

Percobaan Ke	Sebelum Implementasi	Setelah Implementasi
1	6 Menit 1 Detik	2 Menit10 Detik
2	6 Menit 5 Detik	2 Menit 1 Detik
3	6 Menit 10 Detik	2 Menit 5 Detik
4	6 Menit 4 Detik	2 Menit 6 Detik
5	6 Menit 11 Detik	2 Menit 5 Detik
6	6 Menit 9 Detik	2 Menit 4 Detik
7	6 Menit 10 Detik	2 Menit 6 Detik
8	6 Menit 2 Detik	2 Menit 5 Detik
9	6 Menit 5 Detik	2 Menit 3 Detik
10	6 Menit 3 Detik	2 Menit 4 Detik
Rata-rata	**6 Menit 1 Detik**	**2 Menit 5 Detik**

2.13 Pengujian Ketersediaan Sistem

Pengujian ketersediaan sistem dilakukan untuk mengetahui apakah sistem dapat mengatasi masalah tertunda nya registrasi ccTLD Indonesia yang diakibatkan oleh jam kerja *staff registrar* Qwords yang tidak 24 Jam sehingga adanya kebutuhan akan sistem yang dapat berjalan selama 24 Jam sehari dan 7 hari seminggu.

Proses pengujian dilakukan selama 10 kali yang terdiri dari 5 kali di jam kerja dan 5 kali diluar jam kerja *staff registrar* Qwords, hasil pengujian dapat dilihat pada tabel 6 berikut ini

Tabel 6. Hasil Pengujian Ketersediaan Sistem

Nama Domain	Tanggal dan Jam ***Register***
sulseljaya.id	18-Jul-2017 23:45:20 UTC+7
smpalwafa.sch.id	19-Jul-2017 14:36:31 UTC+7
no-ex.id	18-Jul-2017 18:57:09 UTC+7
sipbefishery.web.id	17-Jul-2017 08:16:36 UTC+7
jualanlarisdarigoogle.id	15-Jul-2017 17:55:43 UTC+7
alfadhli.id	16-Jul-2017 20:12:49 UTC+7
pribumifutsal.web.id	13-Jul-2017 21:21:00 UTC+7
sokko.co.id	15-Jul-2017 12:18:30 UTC+7
skmkicau.web.id	14-Jul-2017 15:53:40 UTC+7
neoplasproduct.co.id	18-Jul-2017 19:57:03 UTC+7

2.14 Pengujian *Web Service*

Pengujian *web service* dilakukan untuk mengetahui performa *web service* dalam memproses setiap *request* dari aplikasi *reseller* Qwords. Proses pengujian ini dilakukan 20 kali untuk masing-masing *web service* menggunakan *tool* Apache JMeter dengan *parameter* sesuai analisa *web service* sehingga didapat nilai *average* (rata-rata waktu yang dibutuhkan), *min* (waktu minimal yang dibutuhkan), *max* (waktu maksimal yang dibutuhkan), *received data* (total data diterima) per detik dan *sent data* (ukuran data dikirim) per detik nya.

1. POST /rest/auth

Hasil pengujian *web service* /rest/auth dapat dilihat pada tabel 7 berikut ini

Tabel 7. Hasil Pengujian *Web Service* /rest/auth

	Total
Samples	20
Average (milisec)	168
Min (milisec)	147
Max (milisec)	218
Received KB/sec	14.45
Sent KB/sec	5.23

2. POST /rest/create_registrant

Hasil pengujian *web service* /rest/create_registrant dapat dilihat pada tabel 8 berikut ini

Tabel 8. Hasil Pengujian *Web Service* rest/create_registrant

	Total
Samples	20
Average (milisec)	178
Min (milisec)	137
Max (milisec)	247
Received KB/sec	10.4
Sent KB/sec	13.84

3. POST /rest/register_child_nameserver

Hasil pengujian *web service* /rest/ register_child_nameserver dapat dilihat pada tabel 9 berikut ini

Tabel 9. Hasil Pengujian *Web Service* /rest/register_child_nameserver

	Total
Samples	20
Average (milisec)	163
Min (milisec)	143
Max (milisec)	203
Received KB/sec	10.17
Sent KB/sec	9.54

4. POST /rest/register_domain

Hasil pengujian *web service* /rest/ register_child_nameserver dapat dilihat pada tabel 10 berikut ini

Tabel 10. Hasil Pengujian *Web Service* /rest/register_domain

	Total
Samples	20
Average (milisec)	704
Min (milisec)	294
Max (milisec)	1286
Received KB/sec	8.96
Sent KB/sec	9.42

3. PENUTUP

3.1. Kesimpulan

Berdasarkan hasil penelitian, analisis serta pengujian sistem, dapat diambil kesimpulan sebagai berikut :

1. *Web service RESTful partner* API ini dapat mempermudah *staff registrar* dan *reseller* Qwords dalam memproses registrasi ccTLD Indonesia
2. *Web service RESTful partner* API ini dapat mempercepat proses registrasi ccTLD Indonesia

3.2. Saran

Web service RESTful partner API ini masih ada kekurangan. Oleh karena itu perlu dilakukan pengembangan dan penyempurnaan lebih lanjut. Adapun beberapa saran yang mungkin dapat diterapkan untuk pengembangan selanjutnya adalah:

1. Penambahan fitur fungsional untuk *transfer* domain baik dari *registrar* lain ataupun sesama *reseller*.
2. Penambahan fitur fungsional untuk sub-*reseller* sehingga *reseller* dapat memiliki *reseller* lain dibawahnya
3. Penambahan fitur fungsional untuk mengubah url halaman aplikasi *registrar* sesuai keinginan *reseller* sehingga aplikasi dapat menjadi *white-label*.

DAFTAR PUSTAKA

[1] S. Hollenbeck, “RFC 5730: Extensible Provisioning Protocol (EPP)”, https://tools.ietf.org/html/rfc5730. 22 Agustus 2017 20.30

[2] M.A. Rahman, I. Kuswardayan and R.R. Hariadi. "Perancangan dan Implementasi RESTful Web Service untuk Game Sosial Food Merchant Saga pada Perangkat Android", J. Teknik POMITS. vol.2, no.1, 2013.

[3] E. Kurniawan. "Implementasi Rest Web Service Untuk Sales Order dan Sales Tracking Berbasis Mobile", J.EKSIS. vol.07, no.01, Mei. 2014.

[4] D. Woods, D. Jacobson, and G. Brail “APIs: A Strategy Guide, Creating Channels with Application Programming Interfaces, vol 1. 2011.

[5] A. Hatvala, “Open innovation opportunities and business benefits of web APIs: A case study of Finnish API providers”. Vol 1. 2016.