

PENGEMBANGAN WEBSITE DINAMIS MENGGUNAKAN ASP.NET MVC DAN SQL SERVER DENGAN METODE RAD (STUDI KASUS: PT X)

Fityan Aula Juyuspan¹, Prihatin Oktivasari²

¹ Politeknik Negeri Jakarta

Jl. Prof. Dr. G.A Siwabessy, Kampus Baru UI, Beji, Kukusan, Beji, Kota Depok, Jawa Barat, Indonesia.

fityanaula@gmail.com¹, ti2n_oktivasari@yahoo.com²

Abstrak— Pembangunan *website* dinamis merupakan salah satu kebutuhan bagi beberapa perusahaan. Kebutuhan tersebut dapat berupa sebuah komunikasi dengan pelanggan, sebagai sarana informasi, promosi produk, *statistic* pengunjung, dan pengelolaan *quotation*. PT X merupakan sebuah perusahaan yang menjadikan *website* dinamis sebagai salah satu kebutuhan mereka. *Website* dinamis bertujuan untuk mengelola konten serta mempublikasikan promosi-promosi perusahaan sehingga mempermudah pengunjung dalam memahami dan berinteraksi aktif dengan perusahaan. *Framework ASP.NET MVC* merupakan salah satu metode yang dikembangkan oleh Microsoft yang digunakan untuk pembangunan *website* dinamis. Pada studi kasus ini bahasa pemrograman yang digunakan adalah C# dan menggunakan database SQL Server 2016. Tujuan dari pengembangan ini adalah untuk mengimplementasikan *MVC Pattern*, *bundling and minification*, *cookie*, *AJAX (Asynchronous JavaScript and XML)*, dan *Action Filter* pada ASP.NET. Pengembangan tersebut menghasilkan *website* dinamis yang interaktif, menyediakan informasi dan layanan bagi pengunjung baru ataupun pelanggan dari perusahaan tersebut, serta dikembangkan dengan cepat (*rapid*) dengan menggunakan metode RAD.

Keywords— ASP.NET MVC, *Website Dinamis*, C#, *MVC Pattern*, *Bundling and Minification*.

I. PENDAHULUAN

Pada dunia perusahaan, terdapat beberapa kebutuhan-kebutuhan yang harus dipenuhi seperti, meningkatkan promosi produk-produk, mempermudah pengelolaan *quotation*, mempermudah akses artikel medis, mempermudah akses produk yang ditawarkan serta bagaimana cara pengunjung menghubungi perusahaan tersebut.

Dalam rangka pemenuhan kebutuhan-kebutuhan tersebut, diperlukan suatu *website* dinamis yang dapat menjembatani antara perusahaan dengan pelanggannya. Sehingga, ketika pelanggan ingin mengetahui informasi tentang perusahaan, pelanggan akan dengan mudah mengaksesnya melalui *website* tersebut. Salah satu jenis implementasi teknologi dalam hal meningkatkan persaingan bisnis dan penjualan produk produk adalah dengan menggunakan *website* perusahaan untuk memasarkan berbagai macam produk atau jasa, baik dalam bentuk fisik maupun digital. Oleh karena itu, disarankan disamping promosi pada media sosial, bekerjasama dengan beberapa perusahaan terkait, perusahaan membangun *website* agar mendatangkan peluang lainnya.

Framework ASP.NET MVC merupakan salah satu cara yang dapat digunakan untuk membangun *website* dinamis. Karena, arsitektur ini memiliki kelebihan yaitu SEO (*Search Engine Optimizer friendly*) [1]. Artinya arsitektur ini memiliki kelebihan dasar yaitu salah satunya dapat ter-index dengan mudah oleh search engine seperti

Google, Yahoo, dan Bing. *Website* dinamis ini memiliki database yang dibuat dan dijalankan pada SQL Server 2016. Kemudian dikembangkan dengan menggunakan metode prototyping RAD.

II. TINJAUAN PUSTAKA

A. Visual Studio 2015

Visual Studio merupakan salah satu *Integrated Development Environment (IDE)* yang digunakan untuk mengembangkan dan membangun sebuah aplikasi atau software. IDE memudahkan dalam menulis code dan men-debug sebuah software. Microsoft Visual Studio merupakan salah satu contoh dari IDE yang dikembangkan oleh Microsoft. Visual Studio dapat digunakan untuk membangun program komputer yang berbasis Microsoft Windows seperti *website*, *web application*, *web service*, *desktop*, bahkan *mobile apps*. Visual Studio menyediakan berbagai macam bahasa pemrograman, seperti bahasa C#, C++, VB.NET, C, F#, Python, Ruby, Node.js, dan lainnya. Visual Studio juga mendukung XML, HTML/XHTML, JavaScript, dan CSS. Akan tetapi, Visual Studio pun memiliki beberapa lisensi yang terbagi menjadi Community Edition yang merupakan versi gratis dari Visual Studio dan versi berbayar yaitu edisi Professional dan Enterprise. Visual studio juga memiliki versi untuk pelajar yaitu Express Edition dan dapat digunakan secara gratis oleh pelajar dengan persyaratan dan fitur tertentu.

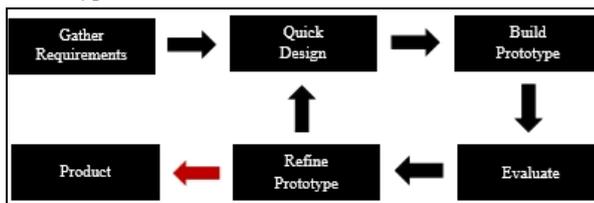
B. SQL Server 2016

SQL Server adalah *Relational Database Management System* (RDBMS) yang dikembangkan oleh Microsoft. SQL Server 2016 hanya dapat digunakan pada arsitektur *processor 64bit* [2].

C. Bahasa Pemrograman C#

C# merupakan bahasa berorientasi objek yang elegan dan aman yang memungkinkan pengembang untuk membangun berbagai aplikasi yang aman dan kuat yang beroperasi pada *.NET Framework* [3]. Bahasa pemrograman ini telah digunakan oleh para pengembang untuk membangun aplikasi berbasis *website* dinamis, *desktop*, bahkan *mobile*. Bahasa pemrograman ini juga mendukung teknologi *cloud service* seperti *Windows Communication Foundation (WCF)*, *ASP.NET Core*, dan lain sebagainya. Sedangkan *.Net Framework* itu sendiri adalah *platform* pengembangan yang digunakan untuk membangun aplikasi berbasis *Windows*, *Windows Phone*, *Windows Server*, dan *Microsoft Azure*. Sehingga, dalam pembuatan suatu aplikasi ataupun *website*, bahasa pemrograman C# sudah dirancang untuk cocok dengan *.Net Framework*.

D. Prototype Model



Gbr 1 Prototype Flow

Pada siklus pembangunan sistem (*System Development Life Cycle* [SDLC]), ada beberapa model yang dapat digunakan untuk membangun sistem perangkat lunak. Model tersebut dapat digunakan dengan kondisi-kondisi tertentu. *Prototype model* digunakan ketika client tidak begitu yakin dengan syarat-syarat (*requirements*) yang diperlukan dalam kebutuhan pengerjaan suatu proyek, sehingga dengan model ini client dapat merubah sesuai kebutuhan ketika *interface* tidak sesuai.

Prototyping disebut juga *Rapid Application Development* (RAD) karena menyederhanakan dan mempercepat desain sistem [4]. Kemudian model ini dapat mengurangi kemungkinan kesalah pahaman dari penafsiran kebutuhan sistem karena calon pengguna berinteraksi langsung dengan pengembang tentang kesesuaian dari rancangan pembangunan sistem tersebut.

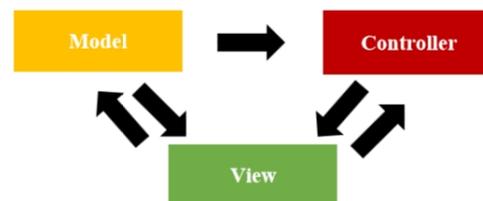
Model prototype memiliki enam tahap yaitu tahap pertama mengumpulkan data-data yang dibutuhkan selama pengembangan dan harapan pengguna untuk aplikasi yang akan dikembangkan, kemudian melakukan perancangan secara cepat, lalu hasil dari rancangan

tersebut akan dievaluasi oleh perancang, pengguna dan akan terus berulang sampai menemui titik hasil (*refine prototype*) lalu ketika hasil sudah sesuai, *prototype* siap menjadi produk dan akan dikembangkan dengan proses selanjutnya.

E. Rapid Application Development

Metode RAD merupakan pengembangan suatu sistem informasi dalam waktu yang relatif singkat [5]. Metode RAD sangat sesuai untuk perusahaan sejenis usaha kecil menengah [6]. Dengan menggunakan metodologi ini, *prototype* akan dibuat untuk menggambarkan suatu produk. Metodologi ini melibatkan pengguna dalam pengembangannya sehingga dapat meningkatkan kepuasan dalam penggunaan sistem [7]. Alasannya, mengganti *requirements* pada saat pengembangan dibolehkan mengingat metodologi ini menggunakan *prototype* dan jika *prototype* telah sesuai maka hasil ini akan dilanjutkan ke tahap pengembangan selanjutnya.

F. Model View Controller (MVC)



Gbr 2 MVC Pattern

Arsitektur MVC digunakan untuk membagi-bagi pengerjaan aplikasi menjadi bagian kecil yaitu dibagi menjadi *model*, *view*, dan *controller* [8]. Dengan demikian proses pengerjaan menjadi lebih fokus karena pengembang mengerjakan aplikasi tersebut secara terpisah (terbagi atas *front-end* dan *back-end*) serta proses *maintenance* menjadi lebih fleksibel.

Business Layer (Model) memiliki peran untuk merepresentasikan keadaan pada aspek-aspek tertentu, maksudnya *model* mengimplementasikan *logic* untuk *data domain* pada aplikasi, sebagai contoh ketika dalam pengembangan membuat suatu objek Pegawai yang terdiri dari atribut seperti Nama Pegawai, Alamat Pegawai dan lain sebagainya. Model juga memiliki peran lain yaitu memproses data seperti koneksi *database*, *querying database*, dan mengimplementasikan *business rules* [9]. Data pada lapisan *logic* ini dapat digunakan untuk diakses (*pass data*) dari *controller* ke *view*. Kemudian *Display Layer (View)* dapat menggunakan model untuk menampilkan data pada sebuah halaman.

Display Layer (View) adalah data yang dipresentasikan secara grafis terlepas dari pemrosesan real data [10]. Pada lapisan inilah komponen-komponen *User Interface (UI)* seperti *textbox*, *dropdown*, *button*, dan lain sebagainya digunakan untuk berinteraksi dengan pengguna.

Input Control (Controller) memiliki peran untuk menerima *input* dari pengguna yang kemudian akan menjadi *action*. Contohnya ketika pengguna berpindah dari satu halaman ke halaman yang lainnya, pengguna secara tidak langsung akan memberi *input* dari *view* kepada suatu *controller* lalu *controller* akan menampilkan halaman yang diminta. Sementara *controller* bertindak sebagai kordinator antara *view* dan *model*.

G. ASP NET MVC

ASP.NET MVC adalah arsitektur yang digunakan untuk mengembangkan *website* dinamis. Untuk mengembangkan *website* berbasis arsitektur ini membutuhkan versi Microsoft .NET Framework 3.5 atau lebih tinggi. Bahkan *website* yang dikembangkan menggunakan ASP.NET MVC memiliki kelebihan seperti SEO (*Search Engine Optimizer*) *friendly* [1]. Artinya arsitektur ini memiliki kelebihan dasar yaitu salah satunya dapat di-*index* dengan mudah oleh *search engine* seperti Google, Yahoo, dan Bing. ASP.NET dikembangkan oleh Microsoft yang di-*release* pertama kali pada Januari 2002 dan berlisensi *open source*. *Framework* ini dibangun menggunakan CLR (*Common Language Runtime*) dan dapat menulis *code* untuk ASP.NET menggunakan bahasa pemrograman berbasis .NET seperti C# dan Visual Basic.

H. Razor View Syntax

Razor view syntax digunakan untuk membuat *template* dan konten berbasis *Windows web servers* seperti ASP.NET MVC. *Syntax* ini mempermudah dalam proses pembuatan *website* dinamis. Razor mengkombinasikan antara *code* dan konten dalam suatu *syntax* [11]. Pada gambar 3 merupakan contoh dari *razor view syntax*:

```
using SPHEMIDFinalwebsite.Models
model IEnumerable<DomainPopular>

<h5>POPULAR POST</h5>
<table>
    <foreach (var item in Model)
    <tr>
        <td class="number">@item.Counter /td>
        <td>
            <h6>
                <a href="article.html" class="black-text">
                    @item.ActionLink(item.Title, "Article", "News",
                        new { Id = item.IDArticle
                            new { @class = "teal-text"
                                </a>
                            </td>
                    </tr>
                </table>
```

Gbr 3 Razor Syntax Example

I. Action Filter

Action Filter adalah atribut yang digunakan untuk mengeksekusi sebuah *method* pada sebuah *controller* atau seluruh *controller*. Pada Microsoft ASP.NET MVC, *Action Filter* dapat dibuat secara *custom* dan tersedia pada *framework* ASP.NET MVC itu sendiri seperti *Handle Error*, *Authorize Attribute*, *Anti Forgery Token*, *Output Cache* dan lain sebagainya. Kode pada *Action Filter* dapat dieksekusi pada situasi-situasi tertentu, seperti pada saat sebelum *controller action* dieksekusi, setelah *controller action* dieksekusi, sebelum *action result* dieksekusi, dan setelah *action result* dieksekusi [12]. *Action filter* juga

dapat digunakan untuk lokalisasi pada suatu *method*, dengan demikian *method* tersebut dapat dieksekusi pada *controller* tertentu dengan situasi tertentu.

J. Bundling and Minification

Bundling and Minification merupakan dua teknik yang dapat digunakan pada ASP.NET 4.5. Dengan menggunakan teknik ini, ukuran (*size*) pada *file assets* seperti CSS dan JavaScript dapat dikurangi dan mengoptimalkan waktu pemuatan suatu page dikarenakan adanya pengurangan permintaan pada *server*. Teknik ini dapat digunakan dengan cara menggunakan System.Web.Optimization dan WebGrease.

Jika didefinisikan secara terpisah, *bundling* merupakan fitur yang ada pada ASP.NET 4.5 yang memudahkan dalam pengintegrasian file CSS maupun JavaScript. Semakin sedikit file berarti semakin sedikit pula HTTP *request* dan performa mengalami peningkatan ketika halaman dimuat pertama kali [13].

Sedangkan *minification* merupakan fitur yang digunakan untuk optimisasi kode, seperti menghapus spasi yang tidak perlu, menghapus komentar, dan mengganti nama *variable* menjadi satu karakter [13].

K. Cookie

Cookie merupakan teks yang pendek yang ditulis oleh *web server* pada perangkat yang digunakan (seperti laptop dan smartphone) ketika *browser* mengakses sebuah *website* yang menggunakan *cookie*. *Cookie* digunakan untuk mengidentifikasi sebuah pengguna, artinya pengguna yang pernah berkunjung ke *website* tertentu akan diidentifikasi kembali oleh *website* tersebut untuk keperluan statistik atau keperluan lainnya. *Cookie* merupakan salah satu cara yang digunakan untuk mengatur keadaan (*state*) dari permintaan pengguna (*client request*) terhadap suatu *website* [14].

L. AJAX

Dengan menggunakan AJAX pada pengembangan suatu *website* dinamis, maka JavaScript yang ada pada sebuah halaman *website* tersebut dapat berkomunikasi langsung ke *server* menggunakan objek JavaScript XMLHttpRequest. Dengan objek ini, kode JavaScript dapat mengakses data di *server* tanpa harus memuat halaman *website* sepenuhnya [15]. AJAX menggunakan XHTML untuk mempresentasikan data pada lapisan *view* dan menggunakan DOM (*Document Object Model*) untuk memanipulasi *object* pada HTML secara dinamis.

III. METODE PENELITIAN

A. Pengumpulan Data

Pengumpulan data dilakukan melalui wawancara langsung dengan *stakeholder*. Wawancara berisi tentang pertanyaan-pertanyaan terkait dengan pengembangan seperti menanyakan visi dan misi, produk apasajakah yang akan ditawarkan atau ditampilkan pada *website*, jenis artikel apasajakah yang akan dipublikasikan, dan fitur apa sajakah yang akan digunakan pada *website* yang akan dibangun. Pemodelan yang digunakan saat pembangunan *website* adalah RAD. Karena dalam pembangunan memerlukan komunikasi dengan beberapa *stakeholder* untuk memastikan bahwa *User Interface (UI)* dan fitur-fiturnya telah sesuai dengan permintaan. Kemudian dari sisi pengembang, pemodelan ini dapat membuat tim bekerja secara *parallel* mengingat arsitektur MVC memiliki keunggulan untuk melakukan hal tersebut itu. Setelah proses tersebut selesai, pengembang mengumpulkan sampel artikel dan produk yang akan di tampilkan untuk keperluan tahap pengembangan selanjutnya.

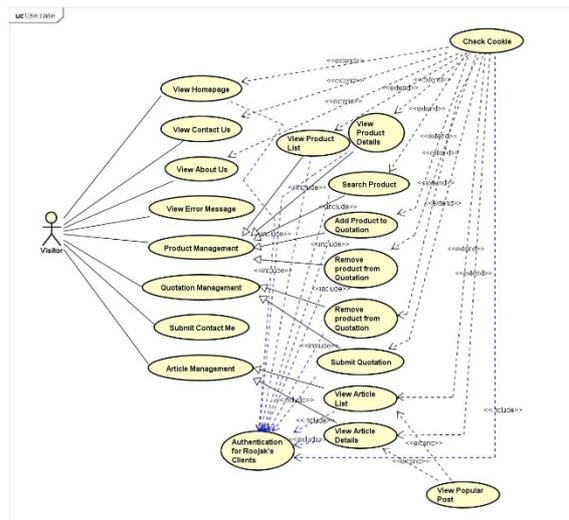
B. Analisa dan Perancangan

Prototype dibuat menggunakan Adobe Photoshop CC 2015 dengan fitur *Artboard* untuk keperluan perancangan *website*. Kemudian hasil dari UI tersebut diunggah di www.invisionapp.com sehingga *prototype* tersebut dapat diakses dan direvisi oleh *stakeholder*. Lalu secara bersama-sama melakukan penentuan jadwal yang akan berguna untuk kepentingan pembangunan *website*. Kemudian bekerja secara *parallel* yaitu bagian *front-end* dan *back-end* bekerja bersama-sama dengan mengacu pada *prototype* yang telah disetujui.

IV. HASIL DAN PEMBAHASAN

A. Usecase Diagram

Usecase diagram digunakan untuk menggambarkan interaksi antara pengguna sistem (*role*) dan sistem itu sendiri untuk mendapatkan hasil tertentu. Pada pengembangan *website* di PT. X, terdapat fitur-fitur tertentu yang dikerjakan. Fitur-fitur tersebut digambarkan dalam *use case* dan dapat dilihat pada gambar 4.



Gbr 4 PT. X Usecase Diagram

B. Perancangan Database

Hasil dari tahap pengumpulan data, akan dijadikan sebuah entitas yang akan digunakan untuk pembuatan tabel-tabel database. Dalam perancangan database dan sebelum pembuatan tabel-tabel, pengembang akan lebih mudah jika membuat *Entity Relationship Diagram (ERD)* terlebih dahulu. Pembuatan ERD ini akan memudahkan pengembang dalam merancang entitas yang akan berhubungan satu sama lainnya. Dalam studi kasus ini dapat dilihat pada gambar 5 hasil dari pembuatan ERD dari *website* PT. X.

Kemudian, ERD akan diterjemahkan dalam bentuk query yang nantinya akan digunakan untuk pembuatan tabel database pada SQL Server 2016. Sebelum membuat tabel-tabel yang diperlukan, pengembang diharuskan untuk membuat database terlebih dahulu. Pada gambar 6 terlihat query yang digunakan dalam pembuatan database baru.

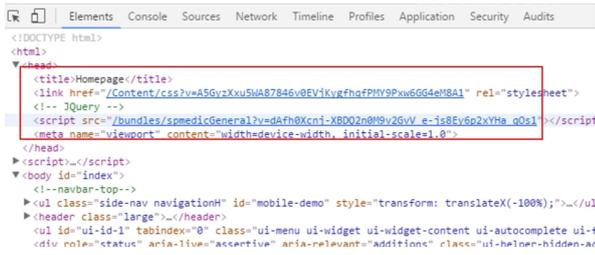
```
CREATE DATABASE SPMedicDB
```

Gbr 5 Create new Database

Kemudian ketika *database* telah dibuat, barulah *query* dari tabel-tabel tersebut di-*execute*. Sehingga database tersebut berisi tabel-tabel yang diperlukan untuk selanjutnya akan diakses saat pengembangan *website* dinamis dengan ASP.NET MVC.

C. Implementasi Bundling and Minification

Dapat dilihat implementasi dari *bundling* yang diterapkan pada *website* PT. X pada gambar 7 yang diamati melalui fitur *inspect element* yang tersedia pada google chrome browser.



Gbr 6 Bundling

Sedangkan implementasi dari *minification* dapat dilihat pada gambar 8 yang diamati melalui fitur *inspect element* yang tersedia pada google chrome browser.

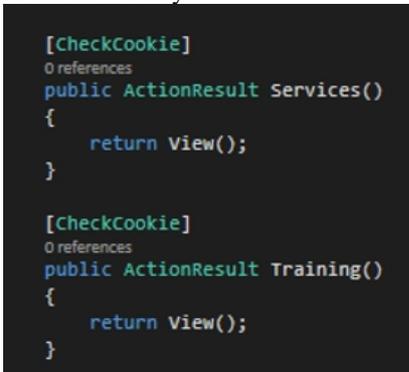


Gbr 7 Minification

Pada gambar 8 terlihat bahwa *minification* memadatkan JavaScript/CSS. Sehingga ukuran dari *file* tersebut akan menjadi lebih kecil

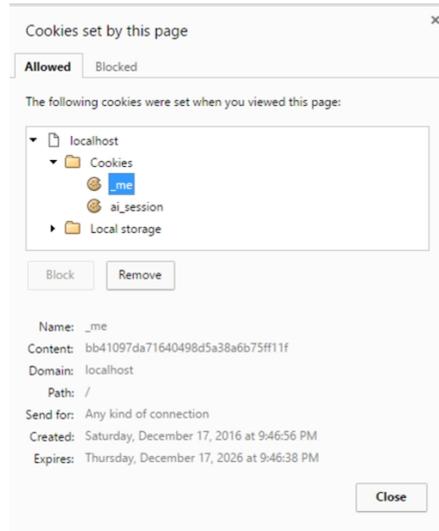
D. Implementasi Cookie

Implementasi *cookie* pada PT. X menggunakan *action filter*. Sehingga, *action filter cookie* dapat digunakan kembali pada *method* lainnya. Pada gambar 9 menunjukkan bahwa *method cookie* dapat digunakan kembali pada *method* lainnya.



Gbr 8 Action Filter Cookie

Ketika pengguna mengakses suatu laman pada *website* pertama kali, maka *cookie* akan otomatis membaca pengguna tersebut sebagai pengguna baru pada *website*. Sehingga, ketika pengguna tersebut mengunjungi *website* kembali, maka *website* akan mencatat aktivitas kunjungan dari pengguna tersebut sehingga perusahaan memiliki informasi tentang *statistic* dari pengunjung. Gambar 10 adalah implementasi dari *cookie* yang menunjukkan bahwa pengguna tersebut dapat dikenali ketika melakukan kunjungan kembali pada laman tertentu.

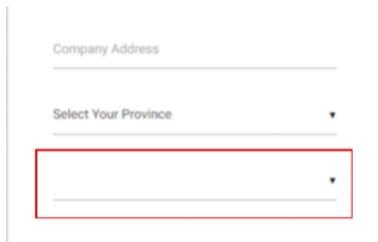


Gbr 9 Cookie Operation

Cookie _me akan menandai setiap pengunjung yang mengunjungi setiap halaman *website*. *Cookie* ini akan menandai pengunjung jika pengunjung mengunjungi *website* pertama kali dan memiliki masa berlaku selama 10 tahun dan akan kadaluarsa jika sudah melebihi dari masa berlaku. Sedangkan *cookie ai_session* memiliki fungsi sebagai penanda batas waktu seorang pengunjung selama 10 menit dan akan mencatat aktivitas yang dilakukan pengunjung setiap 10 menit. Contohnya ketika pengunjung mengunjungi halaman X selama 1 menit kemudian pengunjung berpindah ke halaman Y dan diam selama 9 menit maka perusahaan memiliki informasi bahwa pengunjung selama waktu 10 menit di hari tertentu mengunjungi 2 halaman.

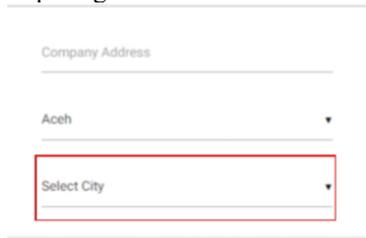
E. Implementasi AJAX

Implementasi *AJAX* yang digunakan dalam pembangunan *website* PT. X tidak diterapkan secara menyeluruh melainkan hanya beberapa keperluan saja seperti ketika pengunjung memilih produk dan menghapus produk dari halaman *product details* dan *quotation list*. Kemudian ketika pengunjung memilih provinsi dan kota ketika ingin memasukan data untuk *submit quotation* dan *contact us*. Berdasarkan rincian tersebut, penggunaan *AJAX* sangat diperlukan mengingat hasil dari permintaan data harus *real-time* dan tanpa pengunjung memuat kembali halaman. Selain itu, *AJAX* digunakan untuk menampilkan pesan *status* seperti “Data successfully removed from quotation list” atau “Data successfully added to quotation list”. Pada gambar 11 dan 12 merupakan contoh implementasi dari *AJAX* pada saat pemilihan provinsi dan kota.



Gbr 10 Select Province

Ketika pengguna memilih provinsi maka pada kotak merah pada gambar 11 akan muncul list kota pada suatu provinsi seperti pada gambar 12.



Gbr 11 Select City

V. KESIMPULAN

Pengembangan *website* dinamis merupakan salah satu kebutuhan bagi beberapa perusahaan. Pada studi kasus yang diangkat pada jurnal ini (PT X) merupakan salah satu contoh perusahaan yang membutuhkan *website* dinamis sebagai jembatan antara perusahaan dan pelanggannya. Pengunjung baru ataupun pelanggan dari suatu perusahaan, dibuat mudah dalam mengakses informasi dan layanan yang disediakan oleh perusahaan pada *website* dinamis. Oleh karena itu, pengembangan *website* dengan mengimplementasikan metode-metode tertentu seperti *cookie*, *AJAX*, *action filter*, *database*, dan pemilihan *pattern* seperti MVC akan membuat *website* dinamis lebih *functional* dari segi pengembangan dan pengunjung serta membuat *website* lebih interaktif dan memiliki performa yang lebih baik dalam *monitoring* statistik pengunjung dan fungsi permintaan *quotation*.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Risanuri Hidayat yang telah meluangkan waktu untuk membuat template ini. Dan terimakasih sekali untuk tim editor dan semua yang berkontribusi pada jurnal ini.

REFERENSI

- [1] Mistri, Mijal. 2014. ASP.NET MVC with Derivative Patterns and Its Comparative Study. International Journal of Information and Computing Technology. 0976-5999.
- [2] MSDN. 2017, Januari. Requirements for Installing SQL Server 2016. [https://msdn.microsoft.com/en-us/library/ms143506\(v=sql.130\).aspx](https://msdn.microsoft.com/en-us/library/ms143506(v=sql.130).aspx) [26 Februari 2017]
- [3] MSDN. 2015, Juli. Introduction to the C# Language and the .NET Framework. <https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx> [26 Februari 2017]
- [4] Perkasa, Muhammad Dhika. 2015. Pengembagnan Sistem Keuangan AZZAHRA Salon dan SPA Muslimah. Skripsi. Universitas Islam Sunan Kalijaga Yogyakarta.
- [5] Kosasi, Sandy. 2015, Oktober. Penerapan Rapid Application Development Dalam Sistem Perniagaan Elektronik Furniture. Citec Journal Vol.2 No. 4, 24604259.
- [6] Ziman, I., 2011. RAD Applied in the Context of Investment Banking Trading Systems Development, Informatica Economică, Vol. 15, No. 4, hal 134146.
- [7] Lee, S. 2012. Test Cases Design for Software Database Provisioning Development. International Journal of Advanced Science and Technology. Vol. 49, Dec, hal 95-104.
- [8] Qureshi, M. Rizwan Jameel dan Sabir, Fatima. 2013. A Comparison of Model View Controller and Model View Presenter. Sci.Int, 1013-5316.
- [9] Pressman, R. S. 2010. Software Engineering. Boston: McGraw Hill.
- [10] Sommerville, I. 2011. Software Engineering. Boston: Pearson.
- [11] Chadwick, Jess. 2011. Programming Razor: Tools for Templates in ASP.NET MVC or WebMatrix. Sebastopol: O'Reilly Media, Inc.
- [12] Walther, Stephen. 2010. ASP.NET MVC Framework Unleashed. Pearson Education, Inc.
- [13] Anderson, Rick. 2012. Bundling and Minification. <https://www.asp.net/mvc/overview/performance/bundling-and-minification>. [12 Desember 2016]
- [14] Singh, Rahul Rajat. 2012. A Beginner's Guide for Understanding and Implementing Cookies in ASP.NET. <https://goo.gl/ksXmZe> [12 Desember 2016]
- [15] Khanna, SamratVivekanand Omprakash dan Mistry, Mijal. 2012. Impact of AJAX in Web Applications. International Journal of Advanced Engineering Technology, 0976-3945.