

1. Pendahuluan

1.1 Latar belakang

Perkembangan teknologi terus menawarkan banyak hal menarik kepada para pengguna web. Apalagi ketika teknik pemrograman Ajax (*Asynchronous JavaScript and XML*) diterapkan pada suatu aplikasi web. Dengan Ajax, dimungkinkan meningkatnya interaktifitas komunikasi antara *user* dan *server*.

Dengan menerapkan Ajax, suatu *web-client* dapat berkomunikasi dengan *web-server* tanpa adanya interupsi ketika *user* melakukan suatu aksi. Hal ini dikarenakan pertukaran data antara *web-client* dan *web-server* dilakukan secara *background*. Selain itu, *reload* terhadap suatu halaman dilakukan pada bagian yang diperlukan saja, tidak seluruhnya. Hal ini tentu menghemat waktu dan meminimalisir pertukaran data yang tidak perlu.

Ajax diterapkan dengan menggunakan beberapa komponen secara bersamaan. *Pertama*, HTML, yang merupakan *markup language* untuk membuat halaman web. *Kedua*, DOM (Document Object Model), yang memungkinkan terjadinya perubahan isi dan presentasi secara visual, sehingga antara tampilan dan interaksi dengan informasi yang disampaikan menjadi dinamis. DOM diakses dengan menggunakan javascript. *Ketiga*, javascript, yang merupakan bahasa (*client-side*) inti untuk menuliskan kode Ajax. *Keempat*, Objek *XMLHttpRequest*, yang digunakan untuk melakukan pertukaran data secara Asynchronous dengan *server* web. Dan terakhir, XML (*Extensible Markup Language*), yang merupakan format dalam transfer data antara *client* dan *server*.

Jika menerapkan Ajax secara manual, tentu akan menghabiskan banyak waktu dan tenaga. Untuk menjawab kebutuhan tersebut, saat ini hadir berbagai macam *framework* yang mendukung implementasi Ajax. Dua di antaranya adalah PRADO dan Ruby on Rails (RoR).

PRADO adalah singkatan dari *PHP Rapid Application Development Object-oriented*. PRADO merupakan *framework* yang mengedepankan RAD (*Rapid Application Development*) yang bersifat *Component-Based* dan *Event-Driven*. Dengan menggunakan PRADO, maka dimungkinkan terjadinya *re-useability* pada setiap komponennya sehingga mudah digunakan kembali.

RoR juga mendukung konsep RAD. Bedanya, dukungan tersebut dilakukan dengan konsep *Model-View-Controller* (MVC). Konsep MVC ini berguna untuk memisahkan antara *database*, *method*, dan *interface*. Dengan pemisahan ketiga komponen tersebut, proses pembangunan suatu web aplikasi dapat dilakukan dengan lebih rapi.

Kedua *framework* ini, baik PRADO maupun RoR, sama-sama mendukung penerapan Ajax pada aplikasi web, tapi dengan cara yang berbeda. PRADO dengan *component-based* dan *event-driven*-nya. Sedangkan RoR dengan MVC-nya. Menarik untuk mengetahui lebih dalam tentang pengaruh konsep tersebut pada proses pembangunan suatu aplikasi web Ajax.

1.2 Perumusan masalah

Berikut ini akan dijabarkan rumusan masalah yang akan menjadi pembahasan dalam tugas akhir ini :

1. Bagaimana cara membangun dan menerapkan web berbasis Ajax dengan menggunakan *Framework* PRADO dan RoR.
2. Melakukan analisis terhadap penerapan Ajax pada PRADO, yang mengedepankan konsep *component-based* dan *event-driven*, dan RoR, yang mengedepankan konsep MVC.
3. Melakukan analisis perbandingan terhadap aplikasi Ajax yang dihasilkan oleh *Framework* PRADO dengan RoR.

Adapun batasan Masalah yang terdapat dalam Tugas Akhir ini adalah:

1. Aplikasi Ajax yang dibuat hanya sebatas pembuatan aplikasi blog yang memuat fitur-fitur Ajax seperti pada; *login validation*, *livesearch*, dan *dynamic link*.
2. Pengujian terhadap aplikasi yang dihasilkan akan menggunakan tools tambahan untuk menguji performansi web, misalnya dengan *browser* dan *add-ons*.

1.3 Tujuan

Tujuan dari pembuatan TA ini adalah sebagai berikut :

1. Mengetahui bagaimana perbedaan implementasi aplikasi web Ajax dengan menggunakan *Framework* PRADO yang berbasis bahasa pemrograman PHP dan menerapkan konsep *component-based* dan *event-driven*, dengan RoR yang berbasis bahasa Ruby dan menerapkan konsep MVC.
2. Menganalisis kemudahan penggunaan kedua *framework* tersebut untuk membangun dan mengembangkan aplikasi web Ajax. Diantaranya berdasarkan kelengkapan fitur untuk implementasi Ajax dan kemudahan penulisan code.
3. Mengetahui kehandalan dari masing-masing aplikasi web Ajax yang dihasilkan dengan menggunakan *tools* untuk menguji performansi web. Performansi yang akan diuji di antaranya adalah page size, load time, HTML size, dan banyaknya waktu yang digunakan untuk pencarian.

1.4 Metodologi penyelesaian masalah

Berikut ini akan diuraikan metodologi penyelesaian masalah yang akan dilakukan untuk menyelesaikan tugas akhir ini :

1. Studi Pustaka, dilakukan dalam bentuk pencarian referensi terkait dengan permasalahan yang ada pada TA ini. Dan juga melakukan pendalaman terhadap materi yang akan dibahas. Selain itu juga memahami bagaimana pembangunan software dengan konsep *Event-Driven* dan MVC (*Model-View-Control*).
2. Analisa kebutuhan untuk pembangunan aplikasi website Ajax. Dalam hal ini disesuaikan dengan kebutuhan studi kasus yang ada, yaitu aplikasi blog system.
3. Melakukan perancangan terhadap kebutuhan untuk membangun aplikasi web Ajax.
4. Melakukan implementasi terhadap aplikasi web Ajax, menggunakan *Framework* PRADO yang berbasis bahasa pemrograman PHP dan RoR yang berbasis bahasa pemrograman Ruby
5. Melakukan pengujian dan analisis terhadap aplikasi web Ajax yang dihasilkan, sekaligus melakukan perbandingan dari kedua aplikasi yang dihasilkan. Pengujian dilakukan terhadap fungsionalitas dari aplikasi, kemudahan mengimplementasikan Ajax dan kehandalan aplikasi dari sisi performansinya. Pengujian ini menggunakan *tools* tambahan yang sudah ada.
6. Pembuatan laporan penelitian TA, yang berisi dokumentasi, kesimpulan dan rekomendasi terhadap penelitian ini.