

---

## RANCANG BANGUN APLIKASI BAPPSI SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER JAKARTA STI&K

Febianto Arifien dan Devita Rizky NurSeptiani

STMIK Jakarta STI&K

Jl. BRI No. 17, Radio Dalam, Kebayoran Baru, Jakarta Selatan 12140

febianto@staff.jak-stik.ac.id, devita@staff.jak-stik.ac.id

### ABSTRAK

Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K merupakan sekolah tinggi yang memiliki program studi sistem informasi dan komputer. Bagian Administrasi Pengelolaan dan Pengembangan Sistem Informasi, merupakan bagian pengembangan sistem informasi yang berkewajiban untuk membangun aplikasi dan sistem di Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K, untuk itu diperlukan aplikasi untuk mengatur dan mengolah kegiatan-kegiatan seputar BAPPSI. Aplikasi BAPPSI menggunakan pustaka Polymer JS untuk membangun tampilan front-end serta didukung oleh model Restful API sebagai mesin back-end. Aplikasi BAPPSI dirancang dengan Unified Modelling Language (UML) yang menjadi diagram standar untuk merepresentasikan desain dari perangkat lunak. Kelebihan dari aplikasi BAPPSI adalah pada bagian best practices dan SEO, yang menunjukkan aplikasi BAPPSI sudah memenuhi dalam hal search engine optimization dan untuk kemudahan dalam mengkasas aplikasi terbilang cukup.

**Kata Kunci:** Polymer JS, Restful API, Aplikasi, BAPPSI, Lighthouse, UML

### PENDAHULUAN

Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K merupakan sekolah tinggi yang memiliki sebuah Program Studi Diploma Tiga yaitu Manajemen Informatika Program Studi Sarjana yaitu Sistem Informasi dan Sistem Komputer, serta sebuah Program Studi Magister yaitu Teknologi Informatika.

Setiap program studi memiliki aplikasi sendiri dan setiap dosen memiliki akun masing-masing untuk mengetahui mahasiswa yang mengikuti mata kuliah tersebut, serta mahasiswa juga memiliki akun masing-masing yang dapat dimonitoring dari mulai daftar ulang, mengisi KRS, sampai melihat nilai yang didapat dari setiap mata kuliah sampai dengan penulisan akhir.

Bagian Administrasi Pengelolaan dan Pengembangan Sistem Informasi, yang akan disebut kemudian dengan BAPPSI. BAPPSI merupakan bagian pengembangan sistem informasi yang berkewajiban untuk membangun aplikasi dan sistem di Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K juga membangun aplikasi untuk BAPPSI sendiri, aplikasi

BAPPSI digunakan untuk memelihara dan mengontrol semua kegiatan mahasiswa, dosen dan semua elemen yang ada di Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K yang menggunakan aplikasi sesuai kebutuhan masing-masing.

Sebelum adanya aplikasi BAPPSI, Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K sudah memiliki aplikasi untuk mengolah itu semua tetapi masih berbasis DOS yang belum terintegrasi antar bagian dan jika ingin mengirim data harus menggunakan alat bantu seperti *flashdisk* atau perangkat lain, serta beresiko dalam redundansi data.

Berdasarkan masalah yang ada, maka digagaslah aplikasi BAPPSI yang sekarang berjalan. Aplikasi BAPPSI menggunakan pustaka Polymer JS untuk membangun tampilan karena Polymer JS merupakan *framework* yang sering digunakan orang dan merupakan *library* dari JavaScript. Polymer JS mengutamakan seputar enkapsulasi data dan penggunaan kembali data.[1]

Polymer JS merupakan *library* keluaran Google yang terbuka atau *open*

source. Polymer JS berfokus pada serangkaian tugas tertentu dan bukan banyak tujuan dan tugas. Komponen web dalam Polymer JS mirip seperti HTML yang dikombinasikan dengan style CSS.[1]

Aplikasi BAPPSI menggunakan Restful API untuk membangun basis data adalah karena RESTful API tidak mengharuskan *client* dan *browser* mengetahui apapun tentang struktur API.

Aplikasi Restful API yang dibuat menggunakan tools dan teknologi Java EE (Enterprise Edition) berbasis JDK (Java Development Kit) dengan Maven untuk membuat Java Project dengan bantuan Eclipse IDE, selanjutnya di deploy ke Wildfly Server [2].

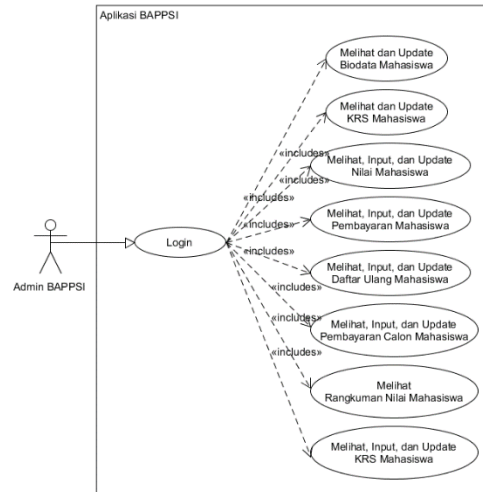
Skalabilitas RESTful API membuat RESTful API berkembang cepat dan mandiri, serta mendukung *multi-domain* dan RESTful API dengan cepat menjadi standar di berbagai layanan atau halaman web.[3] Fungsionalitas RESTful API membuat *developer* lebih mudah dalam mengalokasikan atau membatalkan alokasi sumber daya tergantung penggunaan.[4]

Setelah adanya rancangan kemudian dilanjutkan ke tahap pengkodean dan setelah itu diuji menggunakan sebuah *tools* bernama *Chrome Lighthouse*. *Chrome Lighthouse* merupakan alat yang membantu *developer* untuk menilai kualitas halaman web yang telah dibangun supaya *developer* dapat meningkatkan kualitas halaman web setelah melihat penilaian dari *tools* tersebut karena *tools* ini memberikan cara yang jelas untuk meningkatkan kualitas dengan adanya penilaian *performance*, *accessibility*, kompatibilitas *progressive web app*, *best practices*, dan penilaian SEO. *Chrome Lighthouse* bersifat *open source* jadi pengguna atau *developer* tidak perlu mengeluarkan biaya lagi untuk mendapatkan penilaian yang sudah lengkap.[5]

## METODE PENELITIAN

Aplikasi BAPPSI dirancang dengan *Unified Modelling Language (UML)* yang menjadi diagram standar untuk merepresentasikan desain dari perangkat lunak. UML memiliki beberapa jenis diagram dengan fungsi masing-masing, dalam penelitian ini hanya digunakan satu

jenis diagram yaitu *class diagram* yang digunakan untuk memahami perangkat lunak.[6] Berikut UML yang digunakan dalam penelitian ini:



Gambar 1. Use Case BAPPSI

Pada Gambar 1 menjelaskan yang dapat dilakukan oleh Admin BAPPSI pada Aplikasi BAPPSI. Setelah Login, Admin BAPPSI dapat Melihat dan Update Biodata Mahasiswa; Melihat dan Update KRS Mahasiswa; Melihat, Input, dan Update Nilai Mahasiswa; Melihat, Input, dan Update Pembayaran Mahasiswa; Melihat, Input, dan Update Daftar Ulang Mahasiswa; Melihat, Input, dan Update Pembayaran Calon Mahasiswa; Melihat Rangkuman Nilai Mahasiswa; Melihat, Input, dan Update KRS Mahasiswa.

Rancangan *use case* yang sudah ada kemudian langkah selanjutnya adalah merancang tampilan aplikasi. Berikut rancangan aplikasi BAPPSI:

Pada gambar 2. terlihat rancangan tampilan halaman login yang terdiri dari sebuah *text* di atas untuk penempatan judul aplikasi, 2 (buah) isian atau *field* yang digunakan *username* dan *password* untuk masuk aplikasi BAPPSI. *username* dan *password* yang digunakan menggunakan API dengan metode RESTful dan di bawah *field* tersebut terdapat tombol login yang apabila diklik akan melanjutkan ke halaman masuk.

Pada gambar 3. terlihat rancangan tampilan halaman setelah login, terdapat judul dan pada sebelah kanan atas terdapat

tombolbeberapa menu yang berada di sebelah kiri, menu tersebut jika dipilih maka akan tampil dalam isi halaman.

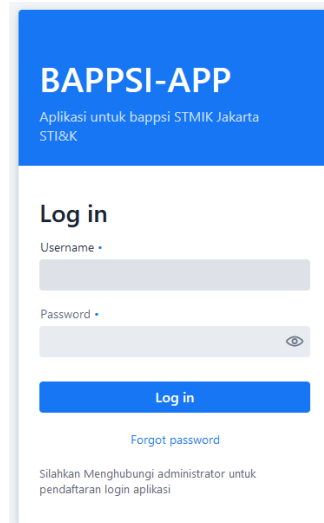
**Gambar 2.** Rancangan Tampilan Halaman Login

**Gambar 3.** Rancangan Tampilan Halaman Setelah Login

**HASIL DAN PEMBAHASAN**

Setelah rancangan selesai maka akan dilakukan proses pengkodean dengan Polymer JS dan menghasilkan aplikasi BAPPSI seperti di bawah ini:

Pada gambar 4 terlihat halaman login yang sudah melalui tahap pengkodean sesuai dengan tahap perancangan sebelumnya di mana terdapat *text* yang digunakan untuk judul aplikasi, kemudian dua buah *field* yang digunakan untuk *username* dan *password* dan sebuah tombol yang digunakan untuk masuk halaman selanjutnya.



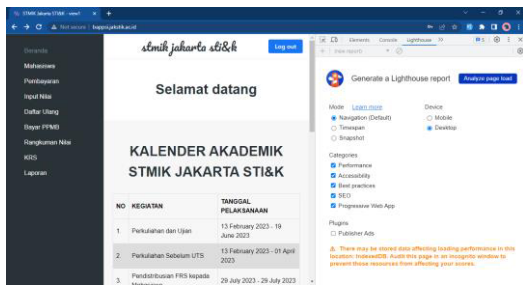
**Gambar 4.** Tampilan Halaman Login

Pada gambar 5 terlihat halaman setelah login yang sudah melalui tahap pengkodean sesuai dengan tahap perancangan sebelumnya di mana terdapat *text* yang digunakan untuk judul aplikasi, kemudian sebuah tombol di sebelah kanan atas yang digunakan untuk keluar dari aplikasi, di sebelah kiri ada menu yang dapat dipilih untuk keperluan masing-masing dan jika di pilih maka akan tampil di sebelah kanan.

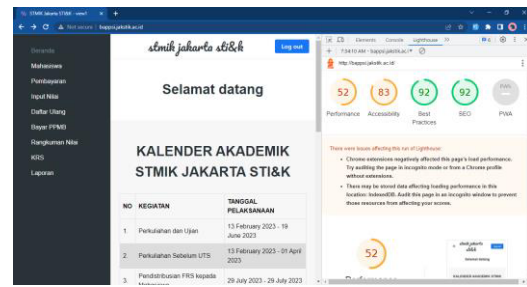


**Gambar 5.** Tampilan Halaman Setelah Login

Setelah selesai tahap pengkodean, maka langkah selanjutnya adalah pengujian aplikasi BAPPSI yang sudah dibangun dengan *tools Chrome Lighthouse* seperti pada gambar 6 sampai dengan gambar 9.

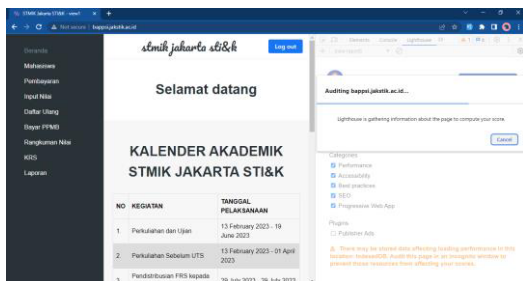


Gambar 6. Tahap Pemilihan Penilaian Chrome Lighthouse



Gambar 9. Penilaian Chrome Lighthouse Tipe Mobile

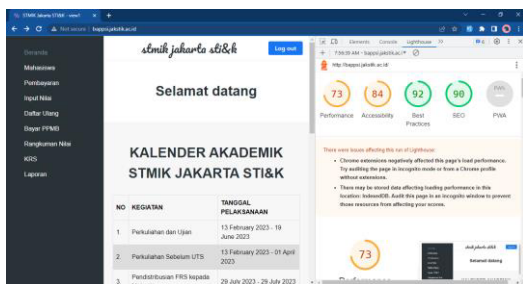
Sebelum mendapatkan penilaian, maka harus memilih penilaian berdasarkan apa yang diinginkan dan apakah dengan tipe *mobile* atau tipe *desktop* seperti pada gambar 6.



Gambar 7. Tahap Proses Penilaian Chrome Lighthouse

Setelah mengatur penilaian yang diharapkan, maka klik *analyze page load* dan kemudian akan memproses pemilihan tersebut seperti gambar 7.

Setelah proses audit atau analisa aplikasi BAPPSI selesai, maka akan terlihat nilai seperti gambar 8 yang memilih untuk tipe *desktop* dan gambar 9 yang memilih tipe *mobile*.



Gambar 8. Penilaian Chrome Lighthouse Tipe Desktop

## PENUTUP

Berdasarkan penelitian yang sudah dijalani dari tahap perancangan sampai pengujian, maka didapatkan hasil cukup memuaskan tetapi ada beberapa hal yang tetap harus ditingkatkan seperti di penilaian *performance* pada tipe *mobile* yang masih kurang dan untuk penilaian bagian *progressive web app* juga masih belum ada, jadi *developer* harus membuat aplikasi BAPPSI menjadi lebih *progressive*.

Di samping kekurangan yang harus ditingkatkan, ada kelebihan yang aplikasi BAPPSI dapatkan yaitu pada bagian *best practices* dan *SEO*, yang menunjukkan aplikasi BAPPSI sudah memenuhi dalam hal *search engine optimization* atau mudah untuk dicari menggunakan *browser* dan untuk kemudahan dalam mengakses aplikasi terbilang cukup walaupun aplikasi ini belum menggunakan *HTTPS* yang menunjukkan halaman web lebih aman. Aplikasi BAPPSI juga terbilang cukup membantu dalam pekerjaan yang ada dan sudah terintegrasi dengan bagian terkait.

## DAFTAR PUSTAKA

- [1] F. Arifien and Rozi, "Implementasi Arsitektur Microservice Pada Sistem Informasi Akademik STMIK Jakarta STI&K Menggunakan Model Enterprise Javabeans (EJB) dan Polymer JS," vol.5, 2021.
- [2] F. Arifien and M. Riastuti, "Model Interoperabilitas Web Service Feeder PDDIKTI Menggunakan Enterprise Javabeans ( EJB ) dan REST-API," vol.3, 2019.
- [3] A. Borrego, M. Bermudo, F. Sola, D. Ayala, I. Hernandez, D. Ruiz, "Silence — A web framework for an agile

- generation of RESTful APIs,” *Elsivier. SoftwareX*, vol. 20, no. 101260, December 2022.
- [4] D. Felicio, J. Simao, N. Datiam, “RapiTest: Continuous Black-Box Testing of RESTful Web APIs,” *Elsivier. Procedia Computer Science*, vol. 219, pages 537-545, 2023.
- [5] “Peluncuran kategori Audit SEO di ekstensi Chrome Lighthouse”. [Online]. Available: <https://developers.google.com/search/blog/2018/02/seo-audit-category-in-lighthouse?hl=id>. accessed Juli 2023.
- [6] G. Bergstrom, F. Hujainah, T. Ho-Quang, R. Jolak, S.A. Rukmono, A. Nurwidyantoro, M.R.V. Chaudron, “Evaluating the layout quality of UML class diagrams using machine learning,” *Elsivier. Journal of Systems and Software*, vol.192, no. 111413, October 2022.