BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan pustaka

Osinski dan Weiss (2005) dalam penelitiannya menjelaskan perancangan framework untuk *clustering* dokumen. Penelitiannya mengimplementasikan algoritma lingo dalam mengelompokan hasil pencarian search engine dengan menggunaka data teks potongan isi hasil pencarian yang disebut dengan *snippet.* Penelitiannya menghasilkan sebuah *framework opensource* bernama *carrot*.

Pada penelitian Hervilorra Eldira, Entin Martiana, dan Nur Rosyid mendapatkan kesimpulan bahwa, Pada proses pencarian dan penyimpanan dokumen secara online masih terbatas pada file html yang diunduh dari Rss berita *online* di internet, diharapkan pengembangan berikutnya tidak hanya berupa file html saja, namun bisa bermacam file seperti: pdf, ppt, doc, dll. Sehingga proses *clustering* tidak hanya terbatas pada dokumennya saja namun diharapkan bisa juga *clustering* untuk type data dokumen sekaligus.

Penelitian tentang implementasi JSON (*Javascript Object Notation*) pada Sistem Informasi Akademik Berbasis *Mobile* dengan Sistem Operasi Android yang di lakukan di Universitas Respati Yogyakarta. Sistem ini dapat menampilkan data akademik yaitu biodata, jadwal kuliah, kartu rencana studi, kartu hasil studi dan transkrip nilai dari masing-masing mahasiswa. Perancangan sistem menggunakan UML (*Unified Modeling Language*) yaitu *usecase diagram, class diagram, activity diagram,*dan *sequence diagram (*Widodo, 2013).

Penelitian serupa pernah di bahas oleh samsudin yang disusun pada tahun 2012 dengan judul " Implementasi SMS Gateway Pada Penyediaan Informasi Lowongan Pekerjaan". Pada penelitian tersebut dengan metode *SMS Gateway* hanya memberikan informasi lowongan pekerjaan kepada user (Alumni STMIK AKAKOM).

Tabel 21.Acuan Tinjauan Pustaka

Penulis	Objek	Metode	Bahasa Pemograman	Interface
Osinski dan Weiss (2005)	penelitiannya perancangan framework untuk *clustering* dokumen	*clustering*	PHP	Website
Hervilorra Eldira, Entin Martiana, Nur Rosyid (2007)	Pencarian Dokumen Bahasa Inggris Menggunakan Hill Climbing Automatic Clustering	*Hill climbing clustering*	Html	Website
Widodo (2013)	Sistem Informasi Akademik Berbasis *Mobile* dengan Sistem Operasi Android	JSON	Java	GUI
Samsudin (2012)	SMS Gateway system informasi lowongan pekerjaan	SMS Gateway		Text

2.1 Dasar Teori

Sejalan dengan tujuan yang diinginkan, yakni penerapan *Progressive Web Apps* pada aplikasi lowongan pekerjaan dengan teknologi *service worker* di *Akakom Career Center*,maka dilakukan penelitian ulang terhadap model informasi lowongan pekerjaan yang sudah ada di website ACC.akakom.ac.id , kemudian berdasarkan hasil penelitian itu akan dirumuskan satu analisis dan desain sistem untuk mengembangkan informasi lowongan pekerjaan yang sudah ada tersebut ke dalam model aplikasi lowongan pekerjaan dengan penerapan PWA. Adapun beberapa teori yang menyangkut penerapan PWA, diantaranya :

2.2.1 Teknologi *Progressive Web Apps* (PWA)

Teknologi PWA mengambil kelebihan dari teknologi baru yang mengambil bagian dari aplikasi *mobile* dan aplikasi *native*. Teknologi ini dikemukakan oleh para developer google pada saat acara I/O google di *mountain view California*. Menurut Rica Handayani selaku *Stayegic Partner Manager Google Asia Tenggara*, menyatakan bahwa PWA akan membuat sebuah situs seperti aplikasi yang akan memberikan performa lebih baik. (Neha Sharma,2015)

Progressive Web Apps(PWA) adalaah aplikasi native yang mendukung hybrid secara penuh dan apklikasi ini tidak perlu proses penginstallan terlebih dahulu namun langsung dapat digunakan secara penuh. Program PWA memiliki banyak kelebihan yang akan memudahkan penggunan dalam menyelami sebuah *website* secara penuh. Apabila dibandingkan dengan *hybrid*, PWA ini penengah antara *native* dan *hybrid* sehingga kondisinya akan lebih stabil namun tetap *up to date* sesuai kondisi *hybrid* yang sebenarnya. *Icon* dapat dipasang pada bagian

desktop atau *screenhome* pada *mobile* agar pengguna dapat melihat notifikasi dengna lebih mudah. Hanya saja untuk saat ini browser yang support dengan PWA ini hanya chrome diatas 47.

Pada dasarnya teknologi progressive web apps bekerja layaknya aplikasi website lainnya,namun yang menjadi perbedaan dengan teknologi aplikasi website lainnya pada PWA bekerja dengan konektivitas yang independen. Artinya, aplikasi PWA dapat bekerja secara offline atau pada jaringan berkualitas rendah dengan adanya *service worker*. Tentu saja selain mampu berjalan di koneksifitas rendah atau offline, PWA juga menggunakan teknologi *instant loading* yang membuat aplikasi website tersebut berjalan dengan cepat, *screenhome* dimana aplikasi website tersebut dapat dijadikan icon pada desktop atau *homescreen* dan *notifikasi* yang artinya pada PWA dapat menampilkan pemberitahuan kepada pengguna tentang adanya pembaruan informasi pada aplikasi website tersebut.

PWA akan bekerja dengan me*load* file HTML,CSS dan javascript minimum yang diperlukan untuk membentuk antarmuka pengguna PWA dan juga merupakan salah satu komponen yang memastikan website dapat berjalan sangat cepat dan langsung di simpan sementara ke perangkat lokal dalam browser untuk nantinya jika setiap kali pengguna membuka aplikasi website, file antarmuka akan dimuat dari penyimpanan sementara perangkat lokal yang membuat waktu *loading* semakin cepat. Penyimpanan sementara secara lokal tersebut menggunakan *service worker* sehingga pada pemuatan berikutnya PWA hanya perlu mengambil data yang di butuhkan, daripada memuat semuanya.

2.2.2 Service Worker

Progressive Web App harus cepat,yang berarti bahwa tetap berfungsi saat *online,offline,* dan pada koneksi yang lamban serta tidak stabil. Untuk mencapainya,PWA harus menyimpan sementara tampilan antarmuka dari aplikasi website dengan menggunakan *service worker,* sehingga selalu tersedia dengan cepat. *Service worker* adalah skrip yang dijalankan browser yang berjalan di latar belakang yang terpisah dari laman web, yang membuka pintu ke berbagai fitur dimana tidak memerlukan laman web ataupun interaksi pengguna. Saat ini service worker sudah menyertakan berbagai fitur seperti pemberitahuan, *push,* dan sinkronisasi latar belakang. *Service worker* juga berkemampuan mencegah dan menangani permintaan jaringan termasuk mengelola *cache respons* melalui program.

Yang membuat API ini menarik adalah karena memungkinkan pengguna mendukung pengalaman *offline.* Sebelum *service worker,* ada satu API lain yang memberikan pengguna pengalaman *offline* di website,yaitu *AppCache.* Namun pada *AppCache* jumlah *gotcha* yang ada serta fakta bahwa meskipun desain bekerja dengan sangat baik,untuk laman aplikasi web tunggal, namun ternyata tidak begitu baik untuk situs multi-laman. *Service Worker* telah didesain untuk menghindari titik-titik menyulitkan yang sudah umum tersebut.

Untuk memasang *service worker*, perlu mendaftakannya sebagai *file JavaScript.* Mendaftarkan *service worker* akan menyebabkan *browser* memulai

langkah pemasangan *service worker* di latar belakang. Biasanya selama langkah pemasangan, perlu menyimpan sementara beberapa asset statis. Jika semua file berhasil di simpan sementara, maka *service worker* akan terpasang. Jika ada file yang gagal di unduh dan disimpan sementara, maka langkah pemasangan akan gagal dan *service worker* tidak akan diaktifkan sehingga membuat aplikasi website tersebut tidak dapat berjalan secara *offline*. Setelah *service worker* aktif, maka secara bersamaan akan menangani manajemen penyimpanan sementara yang sebelumnya. Lalu *service worker* akan mengontrol semua laman yag berada dalam cakupannya, walaupun laman yang mendaftarkan *service worker* untuk pertama kali tidak akan di control hingga dimuat ulang.jika *service worker* sudah terkontrol,*service worker* akan berada dalam salah satu dari dua keadaan,yaitu : *service worker* akan dihentikan untuk menghemat memori, atau *service worker* akan menangani pesan untuk menjalankan permintaan saat jaringan buruk

2.2.3 HTTPS

Pada dasarnya HTTPS(Hypertext Transfer Protocol Secure) memiliki pengertian yang sama dengan HTTP, hanya saja pada HTTPS memiliki kelebihan dalam gungsi keamanannya. Https bukan protokol yang terpisah, tetapi mengacu pada kombinasi dari interaksi HTTP normal melalui Socket Layer terenkripsi SSL (Secure) atau Transport Layer Security (TLS) mekanisme transportasi.

Hal ini menjamin perlindungan yang wajar dari penyadap dan (asalkan dilaksanakan dengan benar dan otoritas sertifikasi tingkat atas melakukan pekerjaan mereka dengan baik) serangan.