

**ANALISIS SISTEM UJIAN ONLINE BERBASIS KOMPUTER PADA
JARINGAN LOW BANDWIDTH
STUDI KASUS SEKOLAH MENENGAH PERTAMA**



TUGAS AKHIR

*Disusun dalam rangka memenuhi salah satu persyaratan
Untuk menyelesaikan program Strata-1 Departemen Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Makassar*

Disusun Oleh :

AINUN MARDIAH

D421 15 004

**DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR**

2022

LEMBAR PENGESAHAN SKRIPSI
ANALISIS SISTEM UJIAN ONLINE BERBASIS KOMPUTER PADA
JARINGAN LOW BANDWIDTH STUDI KASUS SEKOLAH MENENGAH
PERTAMA

Disusun dan diajukan oleh

AINUN MARDIAH

D42115004

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas
Teknik Universitas Hasanuddin pada tanggal 03 Oktober 2022 dan dinyatakan
telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

Dr. April Ahmad Ilham, S.T., MIT.
Nip. 197310101998021001

Iqra' Aswad, S.T., M.T.
Nip. 199011282019043001



Dr. Indrabayu, S.T., M.T.
Nip. 197507162002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : AINUN MARDIAH

NIM : D421 15 004

Program Studi : TEKNIK INFORMATIKA

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**ANALISIS SISTEM UJIAN ONLINE BERBASIS KOMPUTER PADA
JARINGAN LOW BANDWIDTH STUDI KASUS SEKOLAH MENENGAH
PERTAMA**

Adalah karya tulis saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 1 September 2022

Yang Menyatakan



Ainun Mardiah

AINUN MARDIAH

ABSTRAK

Sekolah – sekolah menengah pertama maupun menengah atas dewasa ini diharapkan untuk menggunakan ujian berbasis komputer sebagai sarana evaluasi belajar siswa di berbagai sekolah. Dalam pelaksanaannya ujian berbasis komputer membutuhkan infrastruktur berupa komputer dan jaringan yang memadai. Namun dalam realitanya hal tersebut belum bisa terekseskusi sampai ke bawah terutama bagi sekolah menengah pertama dan menengah atas yang ada di daerah pelosok dan pedesaan atau daerah rural lainnya. Hal ini terutama disebabkan karena jaringan internet yang belum memadai dan kekurangan dari segi infrastruktur di sekolah menengah pertama dan sekolah menengah atas yang ada di daerah pedesaan, pelosok dan daerah rural lainnya. Penetrasi jaringan hingga ke pedesaan dan daerah rural belum merata optimalisasinya jika dibandingkan dengan pusat kota, oleh karenanya sekolah – sekolah menengah pertama dan sekolah menengah atas di daerah rural tersebut masih menggunakan sistem dengan memanfaatkan jaringan LAN yang dibangun di sekolah masing – masing. Oleh karena itu, dalam upaya membantu meringankan masalah jaringan tersebut diperlukan suatu sistem ujian online yang bisa berjalan dalam jaringan walaupun dengan bandwidth yang rendah. Pada penelitian ini diterapkan salah satu teknologi *modern javascript* yakni gatsby Js yang dapat menunjang kinerja sistem ujian online yang dibangun agar dapat bekerja secara baik dan cepat dalam kondisi jaringan dengan bandwidth rendah. Dan dipadukan dengan express Js. serta indexedDB dan *web service* REST API untuk menunjang kinerja sistem ujian yang dibangun. Pengujian dilakukan dengan beberapa metode dengan membandingkan kinerja sistem ujian online yang dibangun pada jaringan 4G, 3G, dan 2G/edge. Hasil yang diperoleh menunjukkan sistem bisa berjalan pada ketiga kriteria jaringan uji dengan baik dan mendapatkan skor *web vitals* (*LCP, FID, CLS*) rata - rata pada taraf *good*

Kata kunci : Gatsby Js, *web vitals* , headless web, *low bandwidth*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Puji dan syukur penulis panjatkan kepada Alla SWT. Tuhan yang Maha Esa karena berkat rahmat dan karunia-Nya sehingga tugas akhir yang berjudul “ANALISIS SISTEM UJIAN ONLINE BERBASIS KOMPUTER PADA JARINGAN LOW BANDWIDTH STUDI KASUS SEKOLAH MENENGAH PERTAMA” ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan laporan tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada:

1. Allah SWT. atas segala berkat, karunia, rahmat, kasih sayang dan pertolongan – Nya yang tak terhingga kepada penulis dalam setiap langkah pembuatan program hingga penulisan skripsi yang melalui banyak tantangan yang tidak mudah hingga bisa menyelesaikan tugas akhir ini.
2. Rasulullah SAW sebagai panutan, shalawat dan juga salam tidak lupa penulis panjatkan kepada Nabi Muhammad SAW beserta segenap keluarganya, para sahabat, dan seluruh umatnya yang mengikuti sunnahnya sampai hari pembalasan.
3. Kedua orang tua penulis serta kedua saudara penulis, serta keluarga yang senantiasa memberikan dukungan, doa, motivasi, bimbingan moral, materi,

kasih sayang yang tidak kepada penulis serta selalu sabar dalam mendidik penulis sejak kecil;

4. Bapak Dr.Amil Ahmad Ilham, ST., M.IT., selaku pembimbing I dan Bapak Iqra Aswad, ST., M.T selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir;
5. Bapak Dr. Indrabayu, S.T., M.T., M.Bus.Sys., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis;
6. Segenap Staf dan Dosen Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis.
7. Teman - teman HYPERVISOR '15 FT-UH atas dukungan dan semangat yang diberikan selama ini;
8. Kak Jannah, Kak Reyna, dan Kak Andriani serta Afifah yang selalu memberikan dukungan moril dan senantiasa mendoakan penulis dalam penyusunan tugas akhir ini
9. Teman – teman IMM Cabang Makassar Timur yang selalu menemani dan memberikan dukungan dan bantuan kepada penulis.
10. Kak Faturahman Y dan Mr. Gyen Abu Bakar yang telah memberikan dukungan, ilmu, dan pengetahuan yang bermanfaat dalam proses penyusunan tugas akhir ini.

11. Para sahabat dan teman-teman yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data dan diskusi progress penyusunan tugas akhir;
12. Orang-orang berpengaruh lainnya yang tanpa sadar telah menjadi inspirasi penulis serta seluruh pihak yang tidak sempat disebutkan satu persatu yang telah banyak meluangkan tenaga, waktu, dan pikiran selama penyusunan laporan tugas akhir ini.

Akhir kata, penulis berharap semoga Allah SWT. berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin.

Wassalam,

Makassar, 19 Juni 2022

Penulis

DAFTAR ISI

ABSTRAK.....	iii
KATA PENGANTAR	v
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
BAB I.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masaah Penelitian	3
1.6 Sistematika Penulisan.....	4
BAB II	5
2.1 Modern Website	5
2.2 <i>Progressive Web Apps</i> (PWA)	9
2.3 Node Js.....	12
2.4 Headless Website	14
2.5 Gatsby Js	15
2.6 JSON (JavaScript Object Notation).....	19
2.7 RESTful API.....	23
2.8 IndexedDB	27
2.9 Express Js.....	29
BAB III.....	31
3.1 Lokasi dan Waktu Penelitian	31
3.2 Instrumen Penelitian.....	31
3.3 Tahapan Penelitian	32
3.4 Gambaran Umum Sistem	34
3.5 Perancangan Sistem.....	37
3.6 Skenario Pengujian.....	50
BAB IV.....	56

4.1.	Hasil dan Implementasi Sistem	56
4.2.	Pengujian.....	63
4.2.1.	Pengujian <i>BlackBox Testing</i>	63
4.2.2.	Pengujian Proses Browser	69
4.2.3.	Pengujian Response Time	78
BAB V	86
5.1	Kesimpulan	86
5.2	Saran	87
DAFTAR PUSTAKA	89

DAFTAR GAMBAR

Gambar 3.1. Gambaran umum sistem	35
Gambar 3.2 Use Case Diagram	36
Gambar 3.3.Sequence diagram login.....	37
Gambar 3.4. Sequenece diagram laman ujian	38
Gambar 3.5. Sequenece diagram laman ujian	39
Gambar 3.6.Sequence diagram laman answer	40
Gambar 3.7. Sequence diagram laman answer	40
Gambar 3.8. Sequnce diagram keseluruhan web apps ujian online	41
Gambar 3.9.Activity diagram login exam	42
Gambar 3.10. Activity diagram fetch data soal di awal	43
Gambar 3.11.Activity diagram laman rules.....	43
Gambar 3.12.Activity diagram laman ujian	44
Gambar 3.13.Activity diagram laman answer	45
Gambar 3.14.Activity diagram laman akhir	45
Gambar 3.15. Activity diagram tampilkan soal	46
Gambar 3.16. Activity diagram input soal baru	46
Gambar 3.17. Activity diagram update soal	47
Gambar 3.18. Activity diagram hapus soal	47
Gambar 3.19. Activity diagram menampilkan daftar user.....	48
Gambar 3.20. Activity diagram input user	48
Gambar 3.21. Activity diagram update user	49
Gambar 3.22. Activity diagram hapus user	49
Gambar 4.23 Laman login user	57

Gambar 4.24 Laman rules/peraturan ujian	57
Gambar 4.25 Laman ujian.....	58
Gambar 4.26 Laman ujian 2 (contoh pengisian ujian).....	59
Gambar 4.27 Laman jawapan siswa.....	59
Gambar 4.28 laman akhir ujian	60
Gambar 4.29 Laman login admin.....	61
Gambar 4.30 Laman dashboard soal	61
Gambar 4.31 Laman dashboard user	62
Gambar 4.32 Grafik DOMContentLoaded pada jaringan 4G.....	71
Gambar 4.33 Grafik Load pada jaringan 4G	72
Gambar 4.34 Grafik DomContentLoaded pada jaringan 3G	74
Gambar 4.35 Grafik Load pada jaringan 3G	74
Gambar 4.36 Grafik DomContentLoaded pada jaringan 2G	76
Gambar 4.37 Load pada jaringan 2G	77
Gambar 4.38 Cumulative Layout Shift	83
Gambar 4.39 Largest Contentful Paint.....	83
Gambar 4.40 First Input Delay.....	83

DAFTAR TABEL

Tabel 4.1. Hasil balckbox testing laman login sistem.....	64
Tabel 4.2. Hasil blackbox testinglaman ujian	66
Tabel 4.3. Hasil blackbox testing laman jawaban siswa	66
Tabel 4.4. Hasil blackbox testing laman nilai	67
Tabel 4.5. Hasil blackbox testing laman dashboard soal dan user.....	69
Tabel 4.6. Hasil pengujian DomContentLoaded dan Load pada jaringan 4G	71
Tabel 4.7. Hasil pengujian DomContentLoaded dan Load pada jaringan 3G	73
Tabel 4.8. Hasil pengujian DomContentLoaded dan Load pada jaringan 2G	76
Tabel 4.9. Hasil pengujian response time pada page login user	79
Tabel 4.10. Hasil pengujian response time pada page admin	79
Tabel 4.11. Hasil pengujian response time pada page rules	80
Tabel 4.12. Hasil pengujian response time pada page ujian	80
Tabel 4.13. Hasil pengujian response time pada page jawaban siswa	81
Tabel 4.14. Hasil pengujian response time pada page nilai	81
Tabel 4.15. Hasil pengujian response time pada page dashboard soal	82
Tabel 4.16. Hasil pengujian response time pada page dashboard user	82

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring berjalannya waktu, sekolah – sekolah menengah pertama maupun menengah atas diharapkan untuk menggunakan ujian berbasis komputer sebagai sarana evaluasi belajar siswa di berbagai sekolah. Dalam pelaksanaannya ujian berbasis komputer membutuhkan infrastruktur berupa komputer dan jaringan yang memadai. Namun dalam realitanya hal tersebut belum bisa tereksekusi sampai ke bawah terutama bagi sekolah menengah pertama dan menengah atas yang ada di daerah pelosok dan pedesaan atau daerah rural lainnya

Hal ini terutama disebabkan karena jaringan internet yang belum memadai dan kekurangan dari segi infrastruktur di sekolah menengah pertama dan sekolah menengah atas yang ada di daerah pedesaan, pelosok dan daerah rural lainnya. Penetrasi jaringan hingga ke pedesaan dan daerah rural belum merata optimalisasinya jika dibandingkan dengan pusat kota, oleh karenanya sekolah – sekolah menengah pertama dan sekolah menengah atas di daerah rural tersebut masih menggunakan sistem dengan memanfaatkan jaringan LAN yang dibangun di sekolah masing – masing.

Sistem LAN yang dibangun di sekolah – sekolah menengah atas dan menengah pertama tersebut membutuhkan infrastruktur yang tidak sedikit. Dimulai dari penyediaan bangunan kelas untuk membangun sistem, penyediaan pengkabelan dengan semua instalasi listrik yang dibutuhkan untuk mendukung agar jaringan

tetap stabil selama proses ujian berlangsung bagi siswa – siswi di sekolah – sekolah tersebut

Oleh karena itu, pada tugas akhir ini diusulkan suatu sistem ujian online berbasis komputer yang mampu berjalan pada jaringan *low bandwidth* dengan menggunakan teknologi modern website, *headless web*, dan *progressive web apps* yang diharapkan bisa mengatasi permasalahan jaringan yang ada di sekolah – sekolah menengah pertama dan menengah atas di daerah – daerah rural tersebut.

1.2 Rumusan Masalah

1. Bagaimana cara membuat sistem ujian online yang bisa bekerja pada jaringan low bandwidth?
2. Bagaimana cara membuat sistem ujian online dengan mengintegrasikan teknologi *RestAPI* dan service worker dan Modern javascript?

1.3 Tujuan Penelitian

1. Untuk membuat sistem ujian online tingkat sekolah menengah yang bisa berjalan pada jaringan low bandwidth.
2. Untuk mengetahui kinerja sistem ujian online yang berjalan pada jaringan low bandwidth.

1.4 Manfaat Penelitian

- 1 Manfaat yang diharapkan dari penelitian ini bagi peneliti adalah dapat menambah wawasan dan pengetahuan mengenai website berbasis PWA (*Progressif Web Apps*) yang diintegrasikan dengan teknologi modern javascript seperti Gatsby Js yang digunakan dalam penelitian kali ini.
- 2 Bagi pendidikan yakni dapat dijadikan sebagai referensi website dengan teknologi *Progressif Web Apps* yang diharapkan dapat membantu siswa – siswi sekolah – sekolah menengah atas dan menengah pertama yang tinggal di daerah yang memiliki problem jaringan lemah/kurang.

1.5 Batasan Masaah Penelitian

Yang menjadi batasan masalah dalam tugas akhir ini adalah:

1. Low bandwidth yang dimaksud pada tugas akhir ini adalah jaringan 2G.
2. Basis data yang digunakan adalah *indexedDB* dan *MySQL* dengan menggunakan web service *REST API* .
3. Pembuatan website menggunakan *Modern Javascript*
4. Aplikasi web yang dibangun dimaksudkan untuk ujian akhir sekolah bagi siswa sekolah menengah.
5. Aplikasi web yang dibangun hanya mampu digunakan untuk mengerjakan ujian akhir sekolah siswa/siswi sekolah menengah, memasukkan data soal dan user baru, memperbarui data soal dan user, serta menghapus data soal dan user .

1.6 Sistematika Penulisan

BAB I PENDAHULUAN : Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan.

BAB II TINJAUAN PUSTAKA : Pada bab ini akan dijelaskan teori-teori yang menunjang percobaan yang dilakukan.

BAB III METODOLOGI PENELITIAN : Bab ini berisi analisis kebutuhan sistem, perancangan sistem, dan skenario pengujian.

BAB IV HASIL DAN PEMBAHASAN : Bab ini berisi hasil penelitian dan pembahasan penjabaran dari penelitian yang dilakukan.

BAB V PENUTUP : Bab ini berisi kesimpulan dari hasil penelitian dan saran

BAB II

TINJAUAN PUSTAKA

2.1 Modern Website

Website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web* (WWW) di Internet. Sebuah halaman *web* adalah dokumen yang ditulis dalam format *HTML* (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui *HTTP*, yaitu *protokol* yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser* (Trimarsiah, Yunita dan Muhajir Arafat, 2017).

Website adalah sebutan untuk kumpulan halaman web (*webpage*) yang umumnya merupakan bagian dari suatu nama domain atau subdomain di WWW di internet. Website dapat bersifat statis ataupun dinamis yang membentuk satu kesatuan yang saling terkait, yang mana masing-masing dihubungkan dengan *hyperlink*. Suatu website bersifat statis apabila isi informasi tetap, jarang berubah dan informasinya searah hanya dari pemilik website. Website bersifat dinamis apabila isi informasi suatu *website* selalu berubah-ubah.

Diakrenakan website memiliki fungsi untuk menyampaikan informasi , maka pada umumnya website akan menekankan pada kualitas konten yang tersemat di dalamnya. (Krisnayani, Putu., Ketut Resika Arthana., dan I Gede Mahendra Darmawiguna, 2016).

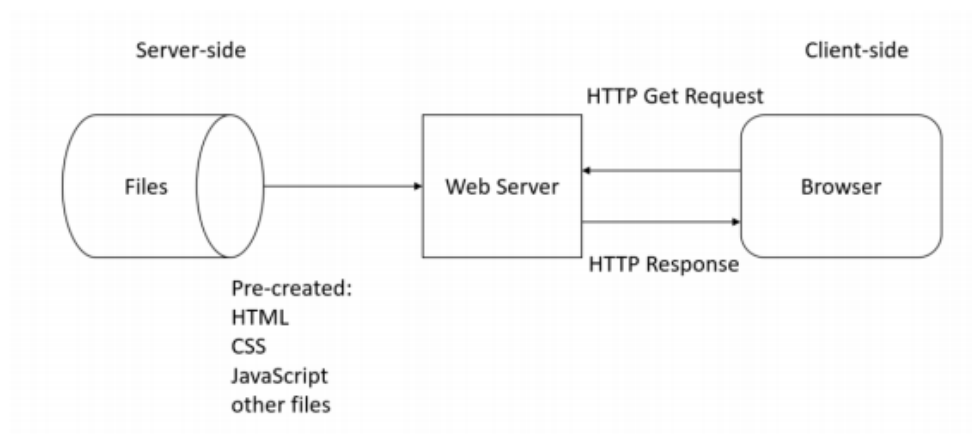
Halaman-halaman dari *website* akan bisa diakses melalui sebuah *URL* yang biasa disebut *Homepage*. *URL* ini mengatur halaman-halaman situs untuk menjadi sebuah *hirarki*, meskipun, *hyperlink-hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan. Beberapa *website* membutuhkan *subskripsi* (data masukan) agar para user bisa mengakses sebagian atau keseluruhan isi *website* tersebut (JavaCreativity,2014).

Salah satu poin penting dalam membangun aplikasi berbasis web adalah berkaitan dengan perancangan desain antar muka. Aplikasi yang akan dibangun harus memiliki tampilan antar muka atau interface yang *user-friendly*. Pembuatan suatu system informasi harus dilakukan dengan memperhatikan faktor kemudahan penggunaan (*usability*) (Krisnayani, Putu., Ketut Resika Arthana., dan I Gede Mahendra Darmawiguna, 2016).

Pada zaman sekarang ini pengembangan aplikasi web dapat dilakukan dengan berbagai macam cara. Dimulai dengan memahami arsitektur apa yang harus dipilih - statis atau dinamis merupakan hal yang harus dilakukan dalam memulai mengembangkan aplikasi web. Kemudian langkah selanjutnya yaitu penentuan lebih lanjut dari tools dan *development service*. Situs web statis terdiri dari sekumpulan halaman HTML, halaman CSS yang ditautkan bersama oleh hyperlink. Situs web dinamis berisi konten yang terletak di database dan ditampilkan atas permintaan *user*. Terkecuali untuk halaman HTML dan CSS, situs web statis dapat

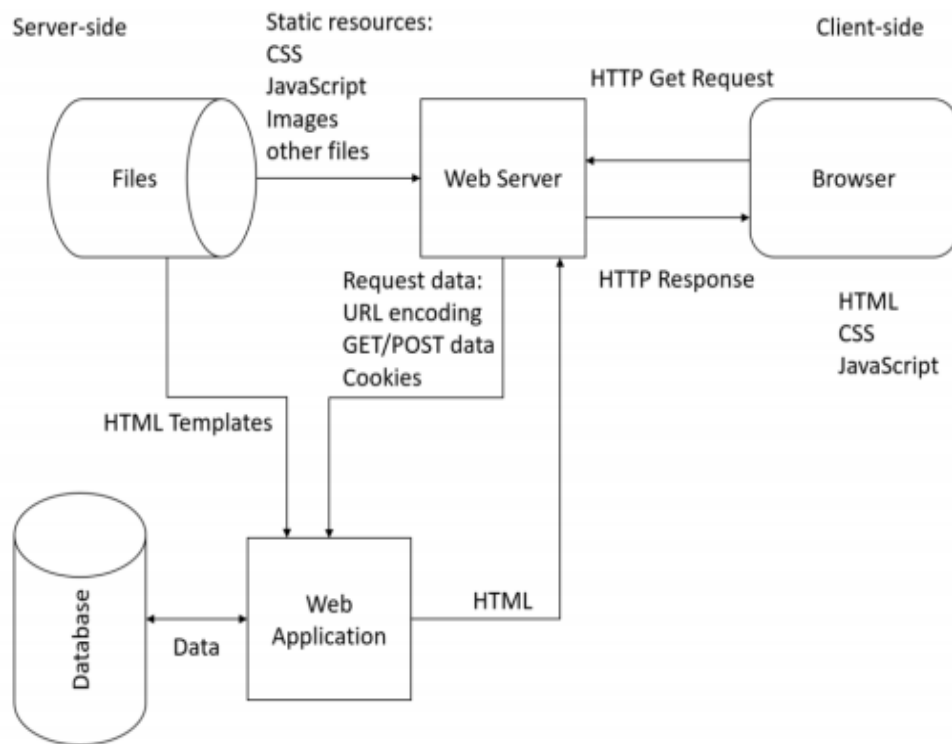
menyertakan beberapa *javascript* di front-end, sedangkan situs web dinamis berisi skrip back-end (Hannonen, 2017).

Skrip front-end adalah skrip sisi klien yang dijalankan oleh browser dan skrip back-end, yaitu skrip sisi server dieksekusi oleh server web. Oleh karena itu, skrip sisi server berfungsi di bagian belakang situs, yang tidak dilihat pengguna. Hal ini menciptakan perancah bagi situs untuk mengakses database-nya, semua mekanisme di balik layar yang mengatur dan memberdayakan situs web.



Gambar 2.1 : Diagram *Website* statis (Hannonen,2017).

Dalam situs web statis ketika *user* ingin membuka halaman, browser mengirimkan permintaan HTTP "GET" dengan menentukan URL-nya. Server mengambil dokumen yang diminta dari sistem filenya dan mengembalikan respons HTTP yang berisi dokumen dengan status success. Jika file tidak dapat diambil karena alasan tertentu, yang kembali adalah pesan atau status *error* .



Gambar 2.2 : Diagram *Website* statis (Hannonen,2017).

Menganalisis dari kedua diagram, ada beberapa keuntungan dan kerugian yang teridentifikasi.Keunggulan *website* statis yakni ada pada mudah dan cepatnya proses pengembangan (*development*). Namun kekurangannya terdapat pada kebutuhan dalam melakukn perubahan pada kode HTML dan atau CSS setiap kali pembaruan pada halaman web dibutuhkan.Psda *website* dinamis memiliki keuntungan dalam proses pembaruan yang lebih fungsional dan lebih mudah serta mengizinkan bekerja dengan data yang dinamis. Namun untuk websiste dinamis membutuhkan lebih dari satu programmer untuk memastikan fungsionalitas dari web itu sendiri dan bagian desain interface yang baik.

Saat sekarang ini, penggunaan web framework sebagai asisten dalam pengembangan aplikasi web sangat digemari. Framework aplikasi web biasanya menyediakan manajemen sesi pengguna (user session management), penyimpanan data (data storage), dan sistem template (templating system), framework ini dapat dibagi menjadi dua main group: front-end dan back-end.

Framework pada bagian server merupakan framework yang berupa perangkat lunak back-end yang membuatnya lebih mudah untuk menulis, memelihara, dan menskalakan aplikasi web (maintain and scale web applications). Menyediakan tools dan library yang menyederhanakan tugas pengembangan web umum, termasuk merutekan URL ke penanganan yang sesuai, berinteraksi dengan database, mendukung sesi dan otorisasi pengguna (*user authorization*), menyediakan format output (misalnya HTML, JSON, XML), dan meningkatkan keamanan terhadap serangan web. Contoh framework back-end adalah: ASP.NET, Django, Ruby on Rails, Express.

2.2 Progressive Web Apps (PWA)

Progressive Web Apps (PWA) didefinisikan sebagai serangkaian konsep dan kata kunci seperti progresif, responsif, konektivitas yang independen, menyerupai aplikasi *native*, segar, aman, mudah ditemukan, dapat direkayasa ulang, dapat dipasang dan ditautkan (A. Biørn-Hansen, T. A. Majchrzak and T.-M. Grønli, 2017).

Progressive Web App juga merupakan suatu teknik untuk dapat mengakses dengan cepat pengalaman website dan aplikasi menjadi satu tanpa harus

menginstall / memasang aplikasi tersebut. *Progressive Web App* memiliki fitur untuk mengatur caching, jaringan, dan notifikasi yang bernama *service worker* (Riady, James. Henry N.Palit., dan Justinus Andjarwiraawan, 2019).

Dalam literatur lain disebutkan bahwa *Progressive Web App* (PWA) adalah teknologi baru yang disarankan oleh Google sebagai cara untuk mengatasi batasan web seluler dalam memberikan pengalaman yang memuaskan bagi pengguna, terutama karena ketergantungan yang kuat pada kondisi jaringan, kurangnya dukungan untuk notifikasi push, dan sebagainya (Malavolta, Ivano dkk, 2017).

PWA berguna untuk pengguna sejak pertama membuka halaman sebuah web dengan konsep PWA, dan seiring dengan pengguna menggunakan aplikasi web lebih banyak lagi, aplikasi akan menjadi semakin powerful. Aplikasi dapat dimuat dengan cepat, bahkan dalam kondisi internet yang kurang baik sekalipun, dapat mengirim *push notifications*, mempunyai ikon aplikasi di *home screen*, dan bisa berjalan dalam mode layar penuh (L. Adi, R. J. Akbar and W. N. Khotimah, 2017).

PWA sendiri mempunyai keunggulan yang terlihat jelas jika dibandingkan dengan aplikasi web klasik, misalnya: PWA dapat diluncurkan dari ikon di layar beranda perangkat (seperti yang dilakukan aplikasi asli), PWA langsung dimuat terlepas dari ketersediaan jaringan, PWA juga mendukung push notification. Inti dari teknologi baru ini adalah konsep pekerja layanan atau lebih familiar disebut *service worker*, yakni sekumpulan API yang memungkinkan pengembang untuk secara terprogram menyimpan dan memuat aset dan data, mengelola push notification, dan lainnya. Secara teknis, *service worker* adalah modul JavaScript yang berjalan di utasnya (thread) sendiri dan menyediakan titik masuk umum untuk

pemrosesan latar belakang yang didorong fungsi *event* (misalnya, reaksi terhadap penerimaan push notification) (Malavolta, Ivano dkk, 2017).

Pada saat sekarang ini, diperlukan sebuah aplikasi berbasis web yang memiliki performa yang baik meskipun dalam kondisi internet yang buruk masih dapat mengakses serta memanfaatkan fitur-fitur dalam *smartphone* yang digunakan untuk asesment. Oleh karenanya PWA hadir sebagai salah satu aplikasi berbasis web yang dapat diakses melalui *smartphone* dan mampu berjalan dalam kondisi jaringan yang buruk sekalipun. PWA merupakan *Mobile Web* yang menggunakan teknologi-teknologi terbaru guna menghasilkan *mobile* web yang memiliki pengalaman pengguna yang jauh lebih baik dari *Mobile Web* tradisional (M. R. Ridho, A. Pinandito and R. K. Dewi, 2018).

Pada dasarnya, PWA merupakan pendekatan pengembangan untuk aplikasi web agar bisa digunakan pada multiplatform, artinya ditujukan agar bisa digunakan oleh sebagian besar platform, dan bisa diakses melalui browser yang hampir available atau tersedia di hampir seluruh platform juga. Dalam pengembangannya Progressive Web Apps bertumpu pada beberapa teknologi utama yakni service worker, Cache API, Background Sync, dan manifest file.

Service worker merupakan file dalam format javascript yang berjalan pada thread lain dan berfungsi untuk mendengar(listen) event - event request ke server yang terjadi dalam sebuah PWA sehingga service worker bisa menanganinya dengan tepat. Untuk cache API sendiri merupakan application programming interface yang bisa menyimpan dan mengembalikan konten, objek, atau data ke

PWA terutama saat offline sehingga PWA tetap bisa berjalan bahkan pada saat kondisi jaringan low.

Background Sync bekerja sama dengan service worker untuk melakukan sinkronisasi ke server pada saat jaringan memungkinkan untuk melakukan pertukaran data, sehingga refresh data bisa tetap terjadi. File manifest memberikan informasi tentang aplikasi seperti nama, pembuat/author, ikon, dan deskripsi mengenai aplikasi tersebut, serta memungkinkan untuk menginstall aplikasi melalui sebuah web install banner.

PWA juga mengimplementasikan pendekatan pada Application Shell Architecture, dengan memfokuskan pada jumlah minimum dari HTML, CSS, dan JavaScript yang dibutuhkan oleh sebuah halaman inti untuk ditampilkan. File - file shell ini nantinya akan disimpan pada cache saat melakukan load pertama kali sehingga pada load atau visit selanjutnya, PWA akan melakukan load secara instan dari cache dengan menggunakan pendekatan tersebut sehingga walaupun jaringan

2.3 Node Js

Node.JS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. Node.JS dapat juga disebut sebagai runtime environment. Aplikasi ini ditulis dalam campuran Bahasa C++ dan juga JavaScript, mempunyai model event driven (basis event) dan asynchronous I/O. Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada web browser, Node.JS dieksekusi sebagai aplikasi server. Node.JS dapat berjalan di server karena dukungan dari V8 Engine buatan Google dan beberapa modul

bawaan yang terintegrasi seperti modul http, modul filesystem, modul security dan beberapa modul penting lainnya (Fajrin, Rachmat, 2017)

Selain itu Node.js juga didesain untuk dapat mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan server web tanpa menggunakan program server web seperti Apache atau Nginx (Rafly, Muhammad Azmi, 2020)

Node.js juga merupakan *web server environment* bersifat *open source* yang paling populer saat ini yang memungkinkan kode JavaScript dapat dijalankan di luar browser web. Hal tersebut membuat halaman web dinamis dengan membantu JavaScript untuk menulis alat baris perintah dan skrip sisi server sebelum halaman dikirim ke browser web pengguna (Pengenalan Express / Node, 2020). Node.js cocok untuk fondasi pustaka web atau kerangka kerja karena memiliki HTTP warga kelas satu di Node.js, dirancang dengan streaming dan latensi rendah. Node.js menggunakan Google Mesin Chromes v8 untuk menjalankan JavaScript. Node.js, secara umum, membuat pengembangan aplikasi lintas platform, sisi server, dan jaringan jauh lebih mudah (Chettri, N., 2016.).

2.4 Headless Website

Istilah "headless" berasal dari konsep memotong "kepala" web , yaitu dalam hal ini bagian ujung depan dari "tubuh" (ujung belakang, yaitu repositori konten). Headless website tetap dengan antarmuka untuk mengelola konten dan RESTful atau GraphQL API untuk mengirimkan konten di mana pun dibutuhkan. Karena pendekatan ini, headless website tidak peduli tentang bagaimana dan di mana konten ditampilkan, namun hanya memiliki satu fokus: menyimpan dan mengirimkan konten terstruktur dan memungkinkan editor konten untuk berkolaborasi pada konten baru (Storyblok, diakses 29/05/2022)

Beberapa alasan penggunaan *headless/decoupled architecture* dalam website yakni sebagai berikut

- Performance

Dalam hal performansi, penggunaan *headless/decoupled architecture* dapat membuat web lebih cepat dan mengoptimalkan web sehingga mampu bekerja secara high-available pada jaringan edge.

- Security

Headless/Decoupled web yang memisahkan frontend dan backend, memperkecil kemungkinan serangan malicious karena bekerja terpisah, sehingga aman.

- Scalability

Tidak perlu menskalakan web server dan tidak perlu membeli sumber daya bahkan pada *spiky traffic* (penambahan jumlah visitor web yang naik dengan tajam).

- Flexibility

Sangat flexibel dalam menggunakan sumber daya , *third-party service*, atau APIs lainnya untuk membangun situs dengan user *experience* yang baik.

- Productivity

Memberikan keluasan bagi developer untuk berkreasi dengan tools dan *workflow* yang diinginkan dan dibutuhkan untuk membangun situs web (Jack Sellwood , 2022. webinar “ Scaling High-Performance Localization wit StoryBlok”).

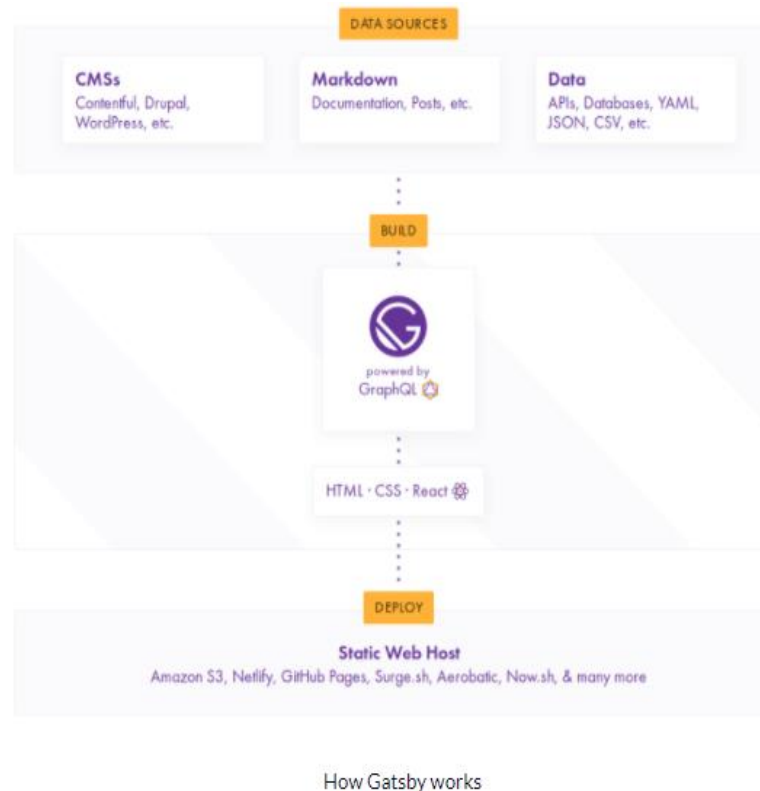
2.5 Gatsby Js

Gatsby Js adalah generator situs statis (static site generator) lainnya seperti halnya Hugo, Jekyll, dan sebagainya. Hal yang menjadikannya istimewa adalah Gatsby Js dapat digunakan untuk membangun situs statis yang merupakan Progresif Web Apps (PWA).Tapi bukan hanya itu, Gatsby Js juga mengikuti standar web terbaru, dan dioptimalkan agar memiliki performansi tinggi. Gatsby js di-*bust* dengan menggunakan teknologi terbaru dan populer termasuk ReactJS, Webpack, GraphQL, ES6 + JavaScript dan CSS modern.(freecodecamp, 2018)

Dari sisi pengguna, user mengharapkan *smooth experience* pada web, namun dari sisi pengembang atau developer , developer terbiasa dengan situs atau web yang terdiri dari masing - masing file HTML yang memiliki link internal.Pada React kita dapat menjalankan aplikasi dengan perintah minimal create-react-app (CRA), namun jika kembali menilik struktur project, kita akan mendapati struktur yang mungkin tiddak familiar bagi pemula atau developer yang berasal dari stack teknologi lain (freecodecamp, 2018).

Pola yang digunakan React sangat berbeda dengan apa yang pernah kita lihat sebelumnya. Ini karena React bertujuan membangun SPA atau Single Page Applications tanpa setup tambahan, sedangkan untuk menambahkan perutean, halaman atau pengoptimalan SEO akan membutuhkan lebih banyak tools dan konfigurasi. Hal tersebut kelihatan kurang nyaman untuk static site. Jadi Gatsby Js hadir dengan pengoptimalan pada kasus tersebut yang mana lebih intuitif bagi para developer karena halaman dibuat dari komponen dan memiliki internal link yang biasa digunakan selama ini (freecodecamp, 2018).

Bagaimana Gatsby bekerja secara internal itu tidak seperti SPA yang membuat permintaan API saat Anda menjalankan aplikasi, Gatsby melakukan semua pengambilan data, termasuk sumber data dari file lokal, selama waktu pembuatan. Semua data ini kemudian digunakan untuk menghasilkan file HTML, JavaScript, dan CSS statis. Rendering statis inilah yang membuat segalanya bekerja lebih cepat (freecodecamp, 2018).



Gambar 2.3 : Cara kerja Gatsby Js (freecodecamp, 2018)

Gatsby Js sebagai salah satu static site generator yang dibutuhkan dalam penelitian saya kali ini dikarenakan situs statis / static site sebagai hasil dan tujuan dari Gatsby Js memiliki keunggulan – keunggulan dibandingkan situs dinamis/ dynamic site , yaitu:

- tidak memerlukan penyiapan server yang rumit (complex server setup) untuk database, maintenance, dan tidak memiliki scaling issues.
- Data secara keseluruhan aman. CMS dan API memiliki private feature tetapi datanya masih ada di server yang dapat dieksploitasi. Gatsby hanya mengambil data yang diperlukan untuk ditampilkan dari sumber

dan data pribadi atau aman bahkan tidak ada dalam versi final. Mana yang paling aman yang bisa didapat.

- Daripada mengandalkan server untuk menghasilkan (generate) halaman secara dinamis, pre-render semuanya di *build* dan gunakan CDN untuk mendapatkan *blazing fast and smooth experince* untuk smeua user..
- Gatsby melakukan *static rendering*. Yang membuat konten tersedia sebagai HTML, dan pengoptimalan search engine, dan initial long time yang tidak lama.

Salah satu hal yang harus dilakukan dalam mempertahankan kecepatan Gatsby site, adalah dengan fokus pada pengontrolan empat penggerak utama page bloat (The Four Key Drivers of Page Bloat) , yakni

1) Delay Speed Load

Karena skrip pihak ketiga seringkali dapat menjadi bagian paling mahal dari produksi situs Gatsby, Gatsby menyediakan pengait (*hooks*) yang memungkinkan untuk menunda skrip ini dengan mudah hingga setelah sisa laman/page dimuat. Selain itu, Gatsby dapat menyejajarkan skrip dan menghemat biaya panggilan jaringan (*cost of a network call*).

2) Trim Bloated Bundle

Gatsby memecah aplikasi menjadi potongan-potongan Javascript yang ditarik ke setiap halaman sesuai kebutuhan. Selain itu, Gatsby memiliki *plugin* dan dokumentasi untuk memantau berat potongan halaman (*page chunk weight*), memvisualisasikan berat halaman untuk mengidentifikasi potongan JavaScript yang bermasalah (seperti

perpustakaan pihak ketiga yang besar). Selain itu, memungkinkan penentuan prioritas dengan mengelompokkan bobot halaman ke dalam potongan global, potongan khusus template, dan potongan khusus halaman.

3) Modularize Global CSS

Dengan mendokumentasikan *library* CSS khusus yang dapat Anda gunakan untuk memastikan CSS modular untuk mencegah CSS global ditarik ke setiap halaman.

4) Optimize Images and Fonts

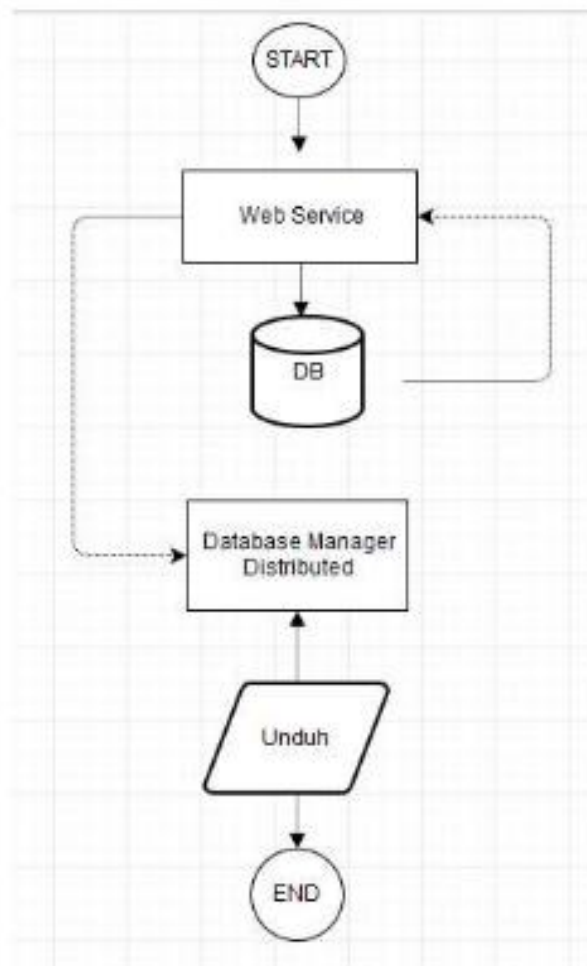
Plugin gambar Gatsby, yang secara deskriptif berjudul gatsby-plugin-image, menawarkan gambar responsif yang siap pakai. Menggunakan gatsby-plugin-image dapat meningkatkan skor Lighthouse sebesar 10-15 poin untuk halaman dengan gambar pahlawan besar.

(Sam Bhagwat , 2022)

2.6 JSON (JavaScript Object Notation)

Java Script Object Notation (JSON) adalah format pertukaran data yang ditemukan oleh Douglas Crockford pada tahun 2006 yang memiliki ukuran data yang lebih kecil serta waktu proses yang lebih cepat dibandingkan dengan XML yang sudah terlebih dulu ada. Sebelum ditemukannya JSON, web service yang ada menggunakan XML sebagai media pertukaran data, yang sudah menjadi standar dan umum digunakan oleh para programmer, namun sekarang JSON bisa juga digunakan sebagai media alternatif pertukaran data didalam web service (Warsito, Arya Budi dkk, 2017).

JSON memiliki kelebihan yakni meningkatkan keamanan database karena file – file berisi database yang telah diubah bentuknya ke dalam format JSON akan dikirim dalam bentuk object data JSON dengan ekstensi .json sehingga jika terjadi hal – hal yang tidak diinginkan seperti rusaknya data di tengah perjalanan atau sistem diretas maka data asli akan tetap aman di dalam database .



Gambar 2.4 : Flowchart Web Service (Warsito, Arya Budi dkk, 2017).

Pada flowchart di atas menjelaskan bagaimana data diambil dari *webservice* yang selanjutnya akan dikirimkan ke database manager distributed yang akan mengubah data ke dalam format atau bentuk JSON yang selanjutnya akan diletakkan dalam direktori yang siap untuk diunduh.

JSON adalah salah satu format penyimpanan data yang banyak digunakan untuk bertukar data karena memiliki kemudahan dalam hal pembacaan dan penulisannya oleh user dan mudah di-generate atau dibuat oleh komputer. JSON menggunakan gaya bahasa yang umum digunakan (C, C++, python, JavaScript, dll) yang karenanya menjadikan JSON ideal sebagai format dalam changing data atau pertukaran data. Struktur JSON yang terdiri dari key dan value, serta array yang merupakan struktur data universal membuat JSON bisa digunakan pada hampir semua bahasa pemrograman modern yang mendukung jenis struktur data tersebut.

Beberapa kelebihan JSON dibandingkan yang lain yakni :

1. JSON lebih ringkas, cepat, dan mudah
2. Tidak menggunakan tag penutup
3. Lebih cepat dalam pembacaan dan penulisannya
4. Menggunakan array
5. Tidak perlu menggunakan parse khusus untuk diubah ke objek, cukup menggunakan fungsi default di Java Script
6. Dapat mengirim banyak data ke server atau sebaliknya
7. Performances kecepatan aksesnya lebih dibandingkan XML karena menggunakan javascript dan lebih mudah diakses dengan javascript itu

sendir dibandingkan XMI yang harus meminta request server dan DOM XML.

2.7 RESTful API

REST merupakan arsitektur web service yang dikembangkan dari beberapa gaya arsitektur berbasis jaringan yang sering diterapkan dalam layanan berbasis web (R. T. Fielding, 2000). Arsitektur REST pada umumnya dijalankan melalui HTTP (Hypertext Transfer Protocol), melibatkan proses pembacaan web page tertentu yang memuat sebuah file XML atau JSON. Setiap permintaan bersifat independen, server tidak menyimpan keadaan permintaan apa pun. *Application Programming Interface (API)* yang mengikuti gaya REST disebut RESTful API. RESTful API menggunakan Uniform Resource Identifier (URI) untuk mewakili sumber daya. Setiap sumber data diidentifikasi menggunakan tautan URI. Metode yang digunakan dalam REST diantaranya: GET untuk mendapatkan sumber daya, POST digunakan untuk membuat sumber daya baru dan metode PUT digunakan untuk memperbarui sumber daya berdasarkan sumber daya. Sedangkan metode DELETE digunakan untuk menghapus sumber daya atau kumpulan sumber daya (Gunawan , Rohmat. dan Alam Rahmatulloh, 2019).

CONTOH RESTFUL API

Resource	Method			
	Get	Post	Put	Delete
/api/student	Get a list of all student	Create a new list of student.	Update a list of student	Delete all student
/api/student/1	Get a student by student's ID	Treat as a collection. Create a new student in it.	If student exists, update the student. If student does not exist. Create a new student.	Delete the student.

Tabel 2.1 : Contoh Restful API (Gunawan , Rohmat. dan Alam Rahmatulloh, 2019).

RESTful (Representational State Transfer).REST sendiri memungkinkan system request dapat mengakses dan memanipulasi teks yang direpresentasikan dari sebuah web service.Web service API yang menggunakan REST disebut dengan RESTful API. Tidak seperti jenis web service lainnya, RESTful API tidak memiliki standar yang resmi untuk notasinya dikarenakan REST merupakan sebuah arsitektur (Tanaem, P.F., Manongga, D., dan Iriani,A. 2016).

REST menentukan sekumpulan prinsip arsitektur yang mana dapat digunakan untuk merancang web service yang berfokus pada sumber daya sistem, termasuk bagaimana sumber daya yang dialamatkan dan ditransfer melalui HTTP oleh berbagai klien yang ditulis dalam Bahasa pemrograman yang berbeda (Tanaem, P.F., Manongga, D., dan Iriani,A. 2016).

Dimana tujuannya adalah untuk menjadikan sistem memiliki performa yang baik, cepat, dan mudah untuk dikembangkan (scale) terutama dalam pertukaran dan komunikasi data. Umumnya menggunakan HTTP (Hypertext Transfer Protocol) sebagai protokol untuk komunikasi data (Edy, dkk , 2019)

Pada arsitektur REST, REST server menyediakan resources (sumber daya /data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya. Seperti yang disebutkan tadi di atas bahwa setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML (Edy, dkk , 2019).

Keuntungan REST: interaksi berbasis REST menggunakan Hypertext Transfer Protocol (HTTP) internet yang merupakan hal yang sudah umum digunakan. Contoh dari pengaturan ini adalah interaksi berbasis REST semua berkomunikasi status mereka menggunakan kode status HTTP standar. Jadi, 404 berarti sumber daya yang diminta tidak ditemukan; kode 401 berarti permintaan tidak diotorisasi; kode 200 berarti semuanya OK; dan 500 berarti ada kesalahan aplikasi yang tidak dapat dipulihkan pada server. Demikian pula, detail seperti enkripsi dan integritas transport data diselesaikan bukan dengan menambahkan kerangka kerja atau teknologi baru, tetapi dengan mengandalkan enkripsi Secure Sockets Layer (SSL) dan Transport Layer Security (TLS) yang terkenal. Jadi, seluruh arsitektur REST dibangun di atas konsep yang sudah dikenal sebagian besar pengembang (Rizal, Randi. dan Alam Rahmatulloh. 2019).

Adapun cara kerja RESTful WS yaitu bermula dari *client* mengirimkan sebuah data atau *request* melalui *HTTP Request*, kemudian *server* merespons melalui *HTTP Response*. Komponen dari *HTTP Request* adalah sebagai berikut. (A. Rahmatulloh, H. Sulastri and R. Nugroho, 2018)

- *Verb. HTTP Method* yang digunakan di antaranya GET (hanya menyediakan akses baca pada *resource*), PUT (digunakan untuk menciptakan *resource* baru), DELETE (digunakan untuk menghapus *resource*), POST (digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru), OPTIONS (digunakan untuk mendapatkan operasi yang didukung pada *resource*).
- *Uniform Resource Identifier (URI)* untuk mengidentifikasi lokasi *resource* pada *server*.
- *HTTP Version*, menjelaskan versi dari HTTP yang akan digunakan, contohnya HTTP v1.1.
- *Request Header*, berisi metadata untuk *HTTP Request*. Contohnya adalah tipe *client/browser*, format yang didukung oleh *client*, format dari *body* pesan, dan *setting cache*.
- *Request Body*, yaitu konten dari data.

Sedangkan komponen dari *HTTP Response* adalah sebagai berikut (A. Rahmatulloh, H. Sulastri and R. Nugroho, 2018):

- *Status/Response Code*, menjelaskan status *server* pada *resource* yang di-*request*. Contohnya 404, artinya *resource* tidak ada dan 200 *response OK*.

- *HTTP Version*, menunjukkan versi dari HTTP yang digunakan. Contohnya HTTP v1.1.
- *Response Header*, berisi metadata untuk *HTTP Response*. Contohnya tipe *server*, panjang *content*, tipe *content*, dan waktu *response*.
- *Response Body*, yakni konten dari data yang diberikan.

Ada dua bagian pesan yang digunakan untuk membangun komunikasi dengan *server*, yaitu pesan *Header* dan pesan *Body*. *HTTP header* adalah “catatan” kecil pada setiap transaksi data HTTP yang dikirim *browser/server* baik pada proses *request* maupun *response*. Ada beberapa elemen yang terdapat pada *HTTP header*, contohnya *HTTP status*, *cachecontrol*, jenis *Web server*, dan sebagainya (A. Rahmatulloh, H. Sulastri and R. Nugroho, 2018)

2.8 IndexedDB

IndexedDB merupakan suatu sistem penyimpanan NoSQL berskala besar yang ada pada browser. Hal ini memungkinkan untuk menyimpan apa saja di browser user. Selain perintah *search*, *get*, dan *put actions*, IndexedDB juga mendukung transaksi (*transaction*). IndexedDB dapat menyimpan sejumlah besar data terstruktur (*structured-data*). Database IndexedDB terdiri atas beberapa *objectStore*. Dan setiap *objectStore* adalah keranjang (*bucket*) yang dapat menyimpan data, yang sama fungsinya dengan tabel di database relasional yang lain untuk menyimpan data, namun *bucket* ini bersifat non-relasional. Istilah lain dalam IndexedDB adalah *Index*. Indeks ini nantinya dapat digunakan untuk mengambil data record dari *objectstore*. Untuk memastikan integritas database

indexedDB, objectStore hanya dapat dibuat dan dihapus dalam fungsi callback di `idb.open`. Callback akan menerima sebuah instance dari UpgradeDB, sebuah special object di pustaka IDB Promised yang digunakan untuk membuat objectStore, Untuk membuat objectStore pada suatu database dalam indexedDB, kita perlu memanggil metode `createObjectStore` di UpgradeDB (Working with indexedDb, 2020).

IndexedDB memungkinkan kita menyimpan komponen-komponen yang ada pada aplikasi pada saat jaringan internet dalam kondisi *offline*, yang mana hal ini sangat berguna untuk memastikan setiap *request* yang dikirim dapat tersimpan didalam *database* aplikasi (Lampah, Nawan Tutu Syah. dan Eko Budi Setiawan, 2018).

Dalam indexedDB, operasi seperti *read* dan *write* dibuat di dalam bentuk transaksi (transaction). Pertama, transaksi dibuka dengan memberikan nama database (database name), versi database (version) dan callback opsional (optional callback). Namun beberapa browser ada yang tidak mendukung *IndexedDB*, jadi sebelum menggunakan indexedDB hal yang harus dilakukan adalah memeriksa apakah browser bisa menggunakan idB atau tidak.

2.9 Express Js

Express Js merupakan kerangka kerja web Node yang paling populer, dan merupakan pustaka dasar untuk sejumlah kerangka kerja web Node yang lumayan populer lainnya. Express Js menyediakan mekanisme untuk beberapa hal berikut :

- Menulis *handlers* untuk *request* dengan kata kerja HTTP yang berbeda di URL path (routes) yang berbeda.
- Mengintegrasikan dengan "view" rendering engines untuk menghasilkan respons dengan memasukkan data ke dalam template.
- Menetapkan pengaturan *web application* umum seperti port yang digunakan untuk menghubungkan, dan lokasi template yang digunakan untuk merender respons.
- Menambahkan additional request processing "middleware" pemrosesan permintaan tambahan di titik mana pun dalam request handling pipeline

Meskipun Express Js itu sendiri cukup minimalis, developer telah membuat paket middleware yang kompatibel untuk mengatasi hampir semua masalah pengembangan web. Ada perpustakaan untuk bekerja dengan cookie, sesi, login pengguna, parameter URL, data POST, header keamanan, dan banyak lagi. (Express/Node introduction, diakses pada 29/05/2022).

Express Js menyediakan metode untuk menentukan fungsi apa yang dipanggil untuk kata kerja HTTP tertentu (GET, POST, SET, dll.) dan pola URL ("Route"), dan metode untuk menentukan template ("view") engine apa yang digunakan, di mana template file berada, dan template apa yang digunakan untuk membuat respons. Dengan express js dapat menggunakan middleware express untuk menambahkan dukungan untuk cookie, sesi, dan pengguna, mendapatkan parameter POST/GET, dll. Selain itu juga dapat menggunakan mekanisme basis data apa pun yang didukung oleh Node (Express tidak mendefinisikan *behavior* terkait basis data apa pun) (Express/Node introduction diakses pada 29/05/2022).