

**SISTEM PENDUKUNG KEPUTUSAN PEMELIHARAAN MESIN UNTUK
INDUSTRI KECIL DAN MENENGAH MENGGUNAKAN *PROGRESSIVE WEB
APPLICATION* DENGAN *VUE.JS***



TUGAS AKHIR

Disusun dalam rangka memenuhi salah satu persyaratan

Untuk menyelesaikan program Strata-1 Departemen Teknik Informatika

Fakultas Teknik Universitas Hasanuddin

Makassar

Disusun Oleh :

DEWI KURNIA SAFITRI

D421 15 013

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2020

LEMBAR PENGESAHAN

“SISTEM PENDUKUNG KEPUTUSAN PEMELIHARAAN MESIN UNTUK INDUSTRI KECIL DAN MENENGAH MENGGUNAKAN PROGRESSIVE WEB APPLICATION DENGAN VUE.JS”

Disusun Oleh:

DEWI KURNIA SAFITRI

D421 15 013

Triptisi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 12 Juni 2020. Diterima dan disahkan sebagai salah satu syarat memperoleh gelar Sarjana Teknik (S.T) pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Gowa, 9 November 2020

Disetujui Oleh:

Pembimbing I,



Dr. Eng. Zulkifli Tahir, S.T., M.Sc
NIP. 19840403 201012 1 004

Pembimbing II,



Ir. Christoforus Yohannes, M.T
NIP. 19600716 198702 1 002

Diterima dan disahkan oleh:
Ketua Departemen Teknik Informatika



Dr. Amil Alimad Ilham, S.T., M.IT
NIP. 19731010 199802 1 001

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Dewi Kurnia Safitri

Nomor Induk Mahasiswa : D421 15 013

Jenjang Pendidikan : S1

Program Studi : Teknik Informatika

Menyatakan bahwa Skripsi yang berjudul "Sistem Pendukung Keputusan Pemeliharaan Mesin Untuk Industri Kecil Dan Menengah Menggunakan *Progressive Web Application* Dengan *Vue.js*" adalah BENAR merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pemikiran orang lain.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini. Skripsi ini hasil karya orang lain atau dikutip tanpa menyebut sumbernya, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 29 November 2020



(Dewi Kurnia Safitri)

ABSTRAK

Kontribusi Industri Kecil dan Menengah (IKM) berperan penting dalam mendorong pertumbuhan manufaktur dan perekonomian di Tanah Air. Kementerian Perindustrian Republik Indonesia mencatat, jumlah unit usaha IKM di dalam negeri terus mengalami peningkatan setiap tahun. Yang sering menjadi kendala adalah IKM yang rata-rata berada pada sektor pedesaan menyebabkan kurangnya ketersediaan sistem informasi yang stabil dalam koneksi internet yang minim atau kondisi *offline*. *Service worker* merupakan salah satu API (*Application Programming Interface*) *JavaScript* yang memungkinkan pengembang dalam melakukan pemrograman *cache* dan melakukan *load asset data* dan yang lainnya. Dengan kemampuan *service worker* menjadikan sebuah *website* dapat difungsikan walaupun koneksi jaringan yang lemah ataupun tidak ada koneksi internet sekalipun (*offline*). Sistem informasi dibuat dengan mengintegrasikan *VueJs*, *NodeJs* dan *database firestore*. Teknologi ini diterapkan pada sebuah web Pemeliharaan Mesin pada Industri Kecil dan Menengah, dimana pada web tersebut dapat digunakan untuk membantu pengambilan keputusan untuk pemeliharaan mesin dengan metode *Decision Making Grid* (DMG). Analisis yang dilakukan menggunakan indikator waktu rata – rata akses, proses *browser*, dan *throughput*. Hasil penelitian menunjukkan bahwa *web* yang terintegrasi *service worker* memberikan nilai waktu proses *browser* yang lebih cepat dibandingkan dengan *web* tanpa *service worker*. Untuk hasil pengujian *throughput web service worker* juga memberikan nilai hasil yang lebih besar dibandingkan dengan *web* tanpa *service worker*.

Kata Kunci : IKM, *website*, *VueJs*, *service worker*.

KATA PENGANTAR

Assalamu Alaikum Wr. Wb.

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT karena berkat Rahmat dan Karunia-Nya sehingga Tugas Akhir yang berjudul “**SISTEM PENDUKUNG KEPUTUSAN PEMELIHARAAN MESIN UNTUK INDUSTRI KECIL DAN MENENGAH MENGGUNAKAN *PROGRESSIVE WEB APPLICATION* DENGAN *VUE.JS***” ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penyusunan penelitian ini disajikan hasil suatu penelitian yang menyangkut judul yang telah diangkat dan telah melalui proses pencarian dari berbagai sumber baik jurnal penelitian, *prosiding* pada seminar-seminar nasional/internasional, buku maupun dari situs-situs di internet.

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

- 1) Tuhan Yang Maha Esa atas semua berkat, karunia serta pertolongan-Nya yang telah diberikan kepada kami disetiap langkah dalam pembuatan program hingga penulisan laporan skripsi ini;
- 2) Orang tua penulis, Bapak Luaying dan Ibu Zubaedah, serta saudara-saudara penulis, Anwar, Maylani, Sherly dan Khalid yang selalu memberikan dukungan, doa, dan semangat serta selalu sabar dalam mendidik penulis sejak kecil;

- 3) Bapak Dr. Eng. Zulkifli Tahir, ST.,M.Sc selaku pembimbing 1 dan Bapak Ir. Christoforus Yohannes, M.T. selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir;
- 4) Bapak Amil Ahmad Ilham, ST., M.IT., Ph.D selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis;
- 5) Para teman-teman dan kakak-kakak Laboratorium UBICON Unhas yang telah memberikan begitu banyak bantuan selama penelitian dan diskusi *progress* penyusunan Tugas Akhir;
- 6) Teman-teman Hypervisor FT UH atas dukungan dan semangat yang diberikan selama ini;
- 7) Segenap Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis;
- 8) Nani, Novi, Billa, Dini, Christine, Dilla, Dian, Uun, Laura, Khusnul, Charina, Arifa, dan Eka yang telah memberikan dukungan dan bantuan kepada penulis untuk menyelesaikan penulisan tugas akhir;
- 9) Keluarga besar GKM Al-Muhandis FT-UH terkhusus Departemen Kemuslimahan yang telah banyak memberikan nasehat dan dukungan selama penulisan tugas akhir ini.
- 10) Orang-orang berpengaruh lainnya yang tanpa sadar telah menjadi inspirasi penulis.

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

Gowa, 9 September 2020

Penulis

DAFTAR ISI

SISTEM PENDUKUNG KEPUTUSAN PEMELIHARAAN MESIN UNTUK INDUSTRI KECIL DAN MENENGAH MENGGUNAKAN <i>PROGRESSIVE WEB APPLICATION</i> DENGAN <i>VUE.JS</i>	Error! Bookmark not defined.
ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah Penelitian.....	3
1.6. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. Pengertian Industri kecil dan menengah	6
2.2. Pemeliharaan Mesin di Industri Kecil dan Menengah	7
2.3. Decision Making Grid.....	9
2.4. Pengertian <i>Website</i>	12
2.5. Progressive Web Apps (PWA).....	13
2.6. Apache Jmeter.....	19
2.7. Lighthouse.....	20
2.8. JavaScript	21

2.9. Basis Data.....	22
2.10. Firebase.....	24
BAB III METODOLOGI PENELITIAN.....	25
3.1. Tahapan Penelitian	25
3.2. Waktu dan Lokasi Penelitian	26
3.3. Instrumen Penelitian.....	26
3.4. Tahap Persiapan	27
3.5. Perancangan Implementasi Sistem.....	27
3.6. Skenario Pengujian.....	41
BAB IV HASIL DAN PEMBAHASAN	44
4.1. Hasil Pengujian Decision Making Grid.	44
4.2. Pengujian <i>Black box</i> Sistem.....	47
4.3. Pengujian dengan <i>Lighthouse</i>	49
4.4. Pengujian Response Time	50
4.5. Pengujian Throughput.....	51
4.6. Pengujian Proses Browser.....	52
BAB V KESIMPULAN & SARAN	58
5.1. Kesimpulan	58
5.2. Saran.....	59
DAFTAR PUSTAKA	60
LAMPIRAN.....	62

DAFTAR TABEL

Tabel 3. 1 Tabel Rancangan <i>Input</i> Data.....	36
Tabel 3. 2 Tabel Rancangan <i>Update</i> Data	49
Tabel 3. 3 Tabel Rancangan <i>Delete</i> Data	49
Tabel 3. 4 Tabel Rancangan Basis Data	49
Tabel 4. 1 Hasil Pengujian <i>black box Input</i> data.....	60
Tabel 4. 2 Hasil Pengujian <i>black box update</i> data	60
Tabel 4. 3 Hasil Pengujian <i>black box delete</i> data.....	58
Tabel 4. 4 Hasil Pengujian <i>Response Time</i>	58
Tabel 4. 5 Hasil Pengujian <i>Troughput</i>	60
Tabel 4. 6 Hasil Pengujian <i>Online</i> dengan <i>disable chace</i>	62
Tabel 4. 7 Hasil Pengujian <i>Online</i> dengan <i>chace</i>	58
Tabel 4. 8 Hasil Pengujian <i>offline</i> dan <i>online</i>	60

DAFTAR GAMBAR

Gambar 2.1 Kriteria Analisis (Labib, 1998).....	11
Gambar 2.2 Alur Kerja Aplikasi Apache JMeter	20
Gambar 3.1 Diagram Tahapan Penelitian.....	25
Gambar 3.2 Proses Manajemen Pemeliharaan Mesin IKM	28
Gambar 3.3 <i>Flowchart Decision Making Grid</i>	29
Gambar 3.4 <i>Diagram activity</i> untuk menampilkan halaman <i>Input Data</i>	32
Gambar 3.5 <i>Diagram activity</i> untuk menampilkan halaman <i>Update Data</i>	33
Gambar 3.6 <i>Diagram activity</i> untuk <i>Delete Data</i>	35
Gambar 3.7 Rancangan Halaman <i>Home</i>	36
Gambar 3.8 Rancangan Halaman <i>Maintenance Report</i>	37
Gambar 3.9 Rancangan Halaman <i>Report List</i>	38
Gambar 3.10 Rancangan Halaman <i>Maintenance List</i>	38
Gambar 3.11 Rancangan Halaman <i>DSS Result Frequency Analysis</i>	39
Gambar 3.12 Rancangan Halaman <i>DSS Result Frequency Analysis</i>	39
Gambar 4.1 <i>Output</i> Frekuensi Microsoft Excel	44
Gambar 4.2 <i>Output</i> Kriteria Frekuensi Pada Sistem DMG	44
Gambar 4.3 <i>Output Downtime</i> Microsoft Excel.....	45
Gambar 4.4 <i>Output Downtime</i> Pada Sistem DMG.....	45
Gambar 4.5 <i>Output</i> DMG Microsoft Excel	45
Gambar 4.6 <i>Output</i> DMG Pada Sistem	46
Gambar 4.7 Hasil Pengujian dengan Google <i>Lighthouse</i>	49
Gambar 4.8 Hasil Pengujian <i>Response Time</i>	50

Gambar 4.9 Grafik Hasil Pengujian <i>Throughput</i>	51
Gambar 4.10 Grafik <i>DOMContentLoaded</i> Online dengan <i>disable cache</i>	53
Gambar 4.11 Grafik <i>Load</i> Online dengan <i>disable cache</i>	53
Gambar 4.12 Grafik <i>Load</i> Online dengan <i>cache</i>	54
Gambar 4.13 Grafik <i>DOMContentLoaded</i> Online dengan <i>cache</i>	54
Gambar 4.14 Grafik Perbandingan <i>Load</i> Online dan <i>Offline</i>	56
Gambar 4.15 Grafik Perbandingan <i>Load</i> Online dan <i>Offline</i>	56

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kontribusi Industri Kecil dan Menengah (IKM) berperan penting dalam mendorong pertumbuhan manufaktur dan perekonomian di Tanah Air. Kementerian Perindustrian Republik Indonesia mencatat, jumlah unit usaha IKM di dalam negeri terus mengalami peningkatan setiap tahun. Misalnya, pada tahun 2013, sebanyak 3,43 juta IKM, naik menjadi 3,52 juta IKM pada tahun 2014. Kemudian, mampu mencapai 3,68 juta IKM di tahun 2015, dan bertambah lagi hingga 4,41 juta tahun 2016. Pada triwulan II tahun 2017, jumlah IKM berada di angka 4,59 juta unit usaha .

Pemeliharaan (*maintenance*) adalah suatu kegiatan yang dilakukan secara berulang-ulang dengan tujuan agar peralatan selalu memiliki kondisi yang sama dengan keadaan awalnya. Kurangnya modal dan ketidak-efektifan dalam melaksanakan pemeliharaan dan perawatan mesin adalah masalah utama yang ada di IKM. Sistem pengambilan keputusan berbasis web dengan konsep lama yang sudah dibuat sebelumnya menggunakan *PHP MySQL* terkendala pada konektivitas yang lambat dan *bandwidth* internet yang rendah.

Oleh karena itu, perlu dibuat sistem yang handal dalam konektivitas jaringan tersebut. Sistem yang akan dibuat juga menerapkan PWA dengan service worker dan framework *VueJs* pada konsep *Decision Making Grid* (DMG) untuk pengambilan keputusan terhadap proses pemeliharaan mesin di IKM.

Progressive Web Applications mulai diperkenalkan oleh *Google* pada tahun 2015. *Progressive Web Applications* merupakan sebuah aplikasi *web* yang dilengkapi dengan teknologi *Service Workers*, sebuah *Java Script Workers*, yang dapat melakukan *caching* pada konten untuk memberikan akses secara *offline*.

Menurut *google developer* salah satu karakteristik dari *progressive web apps* adalah *connectivity independent* dimana *service workers* membantu web untuk dapat diakses secara *offline* atau pada kualitas koneksi jaringan yang rendah.

VueJs adalah suatu librari *JavaScript* yang digunakan untuk membangun antar muka sebuah website yang interaktif. *Library* dari *VueJs* difokuskan hanya pada *view layer* dan sangat mudah diimplementasikan dan diintegrasikan dengan *library* lain ataupun juga dengan *project* yang sudah ada sebelumnya.

Sistem ini diharapkan tidak hanya mampu mendukung proses pengambilan keputusan dan tetapi juga mendorong IKM untuk menggunakan website dalam proses produksinya.

Berdasarkan uraian diatas penulis kemudian mengangkat sebuah penelitian dengan judul “**Sistem Pendukung Keputusan Pemeliharaan Mesin Untuk Industri Kecil Dan Menengah Menggunakan *Progressive Web Application* dengan *Vue.Js* ”** .

1.2 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah pada tugas akhir ini adalah:

1. Bagaimana cara membuat sistem pendukung keputusan untuk pemeliharaan mesin pada industri kecil dan menengah menggunakan PWA dengan service worker dan framework *VueJs*?
2. Bagaimana mengetahui kinerja PWA untuk proses sistem pengambilan keputusan pada IKM?

1.3 Tujuan Penelitian

Tujuan dari tugas akhir ini adalah untuk:

1. Membuat Website Pengambilan keputusan pemeliharaan mesin di IKM dengan memanfaatkan teknologi PWA dan *framework VueJs*.
2. Membuat sistem yang handal dalam konektivitas jaringan dengan memanfaatkan teknologi *service worker* dari *VueJs*.
3. Menerapkan konsep PWA yang dibuat dengan *framework VueJs* di IKM.

1.4 Manfaat Penelitian

Manfaat dari tugas akhir ini adalah :

1. Teknologi *Progressive Web Apps* dapat menjadi pilihan alternatif baru dalam pembuatan website pada IKM.
2. Penelitian ini dapat digunakan untuk menambah pengetahuan dan sebagai referensi mengenai kinerja *website* dengan konsep *Progressive Web App*.
3. Bagi institusi pendidikan, dapat digunakan sebagai referensi dalam pengembangan penelitian topik terkait untuk mempelajari pemrograman berbasis web *Progressive Web Apps*.

1.5 Batasan Masalah Penelitian

Yang menjadi batasan masalah dalam tugas akhir ini adalah:

1. *Website* yang menerapkan teknologi *Progressive Web Apps* dibuat menggunakan *framework website HTML5* yaitu *VueJs*.
2. Penelitian fokus pada Industri Kecil dan Menengah.
3. Metode pengambilan keputusan yang digunakan adalah DMG aplikasi ketika diakses oleh banyak *request* dan kecepatan pengaksesan *website* yang dibuat.
4. Parameter yang digunakan untuk pengujian kinerja dari *Progressive Web Apps* adalah *response time* dan browser proses.
5. Perangkat yang digunakan dalam membuat sistem adalah Laptop Asus X455L dengan memori RAM 6GB dan prosesor Intel core i3.

1.6. Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang diangkatnya judul penelitian sistem pendukung keputusan pemeliharaan mesin untuk industri kecil dan menengah menggunakan *progressive web application* dengan *vue.js*, disertai dengan rumusan masalah, tujuan dan manfaat penelitian, batasan masalah, metode penulisan, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan proses perancangan sistem informasi berbasis *progressive web application*.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang perancangan dan penerapan *VueJs* untuk membangun sistem informasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil pengolahan data serta pembahasan yang disertai tabel hasil penelitian.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1. Pengertian Industri kecil dan menengah

Industri Kecil dan Menengah adalah industri yang memiliki skala kecil dan menengah. Adapun pengertian Industri Kecil dan Menengah dapat didefinisikan berdasarkan beberapa sumber sebagai berikut:

Tabel 2.1 Pengertian Industri Kecil dan Menengah

No.	Pengertian	Industri Kecil	Industri Menengah
1.	Badan Pusat Statistik	suatu kegiatan ekonomi yang melakukan kegiatan mengubah barang dasar menjadi barang jadi/setengah jadi dan atau barang yang kurang nilainya menjadi barang yang lebih tinggi nilainya	suatu kegiatan ekonomi yang melakukan kegiatan mengubah barang dasar menjadi barang jadi/setengah jadi dan atau barang yang kurang nilainya menjadi barang yang lebih tinggi nilainya
2.	Departemen Perindustrian dan Perdagangan (Depperindag)	suatu kegiatan ekonomi yang mengolah bahan mentah, bahan setengah jadi dan atau barang jadi menjadi barang lebih tinggi untuk penggunaannya	suatu kegiatan ekonomi yang mengolah bahan mentah, bahan setengah jadi dan atau barang jadi menjadi barang lebih tinggi untuk penggunaannya
3.	Jumlah Pekerja	5-19 orang	20-99 orang
4.	Jumlah Investasi	Rp. 5.000.000,- (lima juta rupiah) sampai Rp.200.000.000,- tidak termasuk tanah dan bangunan usaha.	Rp. 200.000.000,- sampai 10 milyar, tidak termasuk tanah dan bangunan tempat usaha.

2.2. Pemeliharaan Mesin di Industri Kecil dan Menengah

Pemeliharaan mesin dideskripsikan sebagai semua aktivitas yang dikonsepsikan untuk menjaga dan mempertahankan kualitas mesin/peralatan agar tetap dapat berfungsi dengan baik sesuai kondisi atau standar yang dapat diterima. Secara umum, pekerjaan pemeliharaan mekanik mesin industri, digolongkan menjadi dua macam, yaitu perawatan yang direncanakan (*planned maintenance*) dan perawatan yang tidak direncanakan (*unplanned maintenance*).

2.2.1. Pemeliharaan yang Direncanakan (*Planned Maintenance*)

Dalam pemeliharaan yang direncanakan, suatu mesin atau peralatan akan mendapat giliran perawatan sesuai dengan interval waktu yang telah ditentukan. Penentuan dari interval waktu perawatan ini didasarkan pada beban kerja, dan tingkat kerumitan dari mesin/peralatan yang bersangkutan. Harapan atau tujuan dari pekerjaan perawatan yang direncanakan adalah:

1. Dapat memperpanjang usia pakai mesin/peralatan hingga mencapai 3 atau 4 kali lebih panjang.
2. Dapat mengurangi bahkan meniadakan terjadinya kerusakan yang tidak diharapkan terutama ketika mesin/peralatan digunakan dalam proses produksi.
3. Dapat menjamin ketelitian mesin/peralatan produksi yang digunakan sehingga dapat terjamin kualitas produk yang dihasilkan.
4. Dapat menjamin kualitas operasional mesin/peralatan, sehingga kelancaran dan kelangsungan produksi dapat terpelihara dengan baik.

2.2.2. Perawatan yang tidak direncanakan (*Unplanned Maintenance*)

Pemeliharaan yang tidak direncanakan adalah bentuk pekerjaan pemeliharaan yang dilakukan setelah terjadi kerusakan pada suatu mesin/peralatan. Pekerjaan pemeliharaan yang dilakukan bersifat darurat, artinya harus segera mungkin dilakukan tindakan perbaikan karena kerusakan tersebut telah mengganggu aktivitas produksi, oleh karena itu jenis pemeliharaan yang tidak direncanakan ini dikategorikan sebagai pemeliharaan darurat (*emergency maintenance*).

Beberapa nama atau istilah dalam setiap kelompok perawatan dijelaskan dalam uraian berikutnya.

1) ***Preventive Maintenance***

Preventive Maintenance adalah jenis pekerjaan perawatan/pemeliharaan yang bertujuan untuk mencegah atau menghindari terjadinya kerusakan mesin/peralatan yang mengakibatkan terhentinya atau terganggunya suatu kegiatan produksi. Dengan kata lain, *preventive maintenance* adalah kegiatan perawatan atau cara perawatan yang direncanakan untuk pencegahan (*preventif*), yaitu mencegah suatu mesin mengalami kerusakan pada saat digunakan dalam proses produksi.

2) ***Perawatan Korektif (Corrective Maintenance)***

Perawatan Korektif adalah pekerjaan pemeliharaan yang dilakukan untuk memperbaiki kinerja mesin/ peralatan dengan meningkatkan kondisi mesin/peralatan sehingga mencapai standar yang dapat diterima. Dalam perawatan korektif, perbaikan kinerja mesin dilakukan dengan cara peningkatan kualitas

bagian yang sering mengalami kerusakan, sehingga kerusakan yang sama tidak terulang kembali atau menurun frekuensinya.

3) ***Shut-down Maintenance***

Pekerjaan perawatan yang dilakukan karena terjadi kerusakan mesin/peralatan pada saat mesin/peralatan tersebut digunakan untuk kegiatan produksi. Perawatan dilakukan dengan cara memberhentikan sementara kegiatan produksi yang sedang berjalan (*shut-down*).

2.3. **Decision Making Grid**

Decision Making Grid (DMG) adalah bagan kendali dalam bentuk matriks dua dimensi. Model pemeliharaan yang lebih baik untuk manajemen kualitas dapat dibentuk dengan menangani masing-masing baris dan kolom. Matriks ini menawarkan peluang untuk memutuskan strategi perawatan apa yang diperlukan untuk pengambilan keputusan.

Labib (1998) mengusulkan model DMG sebagai penempatan performansi dari mesin yang bermasalah berdasarkan multi kriteria. DMG didefinisikan dalam diagram dua dimensi. Dimensi pertama adalah *downtime* dengan kriteria *low*, *medium* dan *high*. Dimensi kedua adalah frekuensi mesin rusak, juga dengan kriteria *low*, *medium* dan *high*.

DMG diimplementasikan sebagai berikut:

1. Analisa kriteria

Analisis kriteria dimulai dengan membuat analisis Pareto untuk setiap kriteria. Pareto adalah grafik batang yang menunjukkan masalah berdasarkan

urutan banyaknya jumlah kejadian. Urutannya mulai dari jumlah permasalahan yang paling banyak terjadi sampai yang paling sedikit terjadi.

Data kerusakan mesin dibagi kedalam tiga kriteria yaitu *high*, *low*, dan *medium* berdasarkan jumlah *downtime* dan total frekuensi. Pengelompokan kriteria tersebut dapat dilihat bawah ini. Misalan x adalah frekuensi kerusakan mesin. Kemudian $k = \frac{x_{max}-x_{min}}{3}$, maka intervalnya bisa ditentukan sebagai berikut:

$$\begin{aligned} high_{frequency} &= [x_{max}, x_{max} - k] \\ medium_{frequency} &= [x_{max} - k, x_{max} - 2k] \\ low_{frequency} &= [x_{max} - 2k, x_{min}] \end{aligned} \quad (1)$$

Hal yang sama juga berlaku pada *downtime* mesin. Misalkan y adalah *downtime*. Kemudian , $l = \frac{y_{max}-y_{min}}{3}$ maka intervalnya bisa ditentukan sebagai berikut:

$$\begin{aligned} high_{downtime} &= [y_{max}, y_{max} - l] \\ medium_{downtime} &= [y_{max} - l, y_{max} - 2l] \\ low_{downtime} &= [y_{max} - 2l, y_{min}] \end{aligned} \quad (2)$$

2. Mendefinisikan keputusan

Mendefinisikan kriteria didalam matriks. Kedua variabel data frekuensi dan *downtime* dimasukkan ke dalam matrik dua dimensi dengan dua langkah seperti berikut:

- Kedua kriteria dari setiap mesin di kategorikan menggunakan persamaan (1) dan (2) kemudian dimasukkan ke dalam matriks dua dimensi, dan
- Ketika telah dimasukkan ke dalam matriks, keputusan pemeliharaan dibuat dengan memperhatikan matriks dua dimensi dengan strategi pemeliharaan untuk setiap criteria.

		Down time		
		Low	Medium	High
Frequency	Low	OTF	TPM	CBM
	Medium	TPM	TPM	TPM
	High	SLU	TPM	DOM

OTF: Operate to Failure
 TPM: Total Productive Maintenance
 CBM: Condition Based Maintenance
 SLU: Skilled Upgrade Level
 DOM: Design Out Maintenance

Gambar 2.1 Kriteria Analisis (Labib, 1998)

3. Pengambilan keputusan

Mengidentifikasi pengerjaan yang akan diimplementasikan. Berdasarkan gambar 2.1, setiap bagian memiliki strategi pemeliharaan masing-masing. Berikut penjelasannya.

a. *Operate to Failure* (OTF)

OTF merupakan cara pemeliharaan mesin yang digunakan jika kerusakan alat atau mesin tidak mengganggu rencana produksi dan tersedia sumber daya yang cukup untuk melaksanakan perbaikan. Strategi ini digunakan sebagai perencanaan pemeliharaan pada mesin yang memiliki frekuensi kerusakan sedikit dan dengan *downtime* yang cukup singkat. Karena waktu perbaikan singkat dan frekuensi yang sedikit, mesin-mesin biasanya terus dioperasikan dan akan diperbaiki jika telah mengalami kerusakan.

b. *Fixed Time Maintenance* (FTM)

FTM adalah pemeliharaan yang dilakukan pada interval waktu yang teratur dan merupakan siklus dari pengoperasian alat. Agar FTM berjalan lancar, proses produksi yang sedang berjalan harus dihentikan untuk sementara waktu. FTM

hanya akan efektif jika kerusakan yang terjadi tergantung pada waktu penggunaan mesin.

c. *Skill Level Upagrede* (SLU)

SLU adalah pemeliharaan mesin yang dilakukan pada mesin yang memiliki frekuensi kerusakan tinggi dan periode *downtime* yang rendah. Pada strategi ini dilakukan dengan meningkatkan kemampuan operator mesin tersebut.

d. *Condition-Based Maintenance* (CBM)

CBM adalah strategi yang digunakan untuk memeriksa kondisi peralatan dengan mengukur satu atau beberapa parameter tertentu, yang dapat digunakan untuk mengetahui hal-hal yang dibutuhkan dalam melakukan pemeliharaan. Indikasi perubahan parameter dapat digunakan untuk memantau dan menilai kecenderungan kerusakan yang mungkin terjadi di masa yang akan datang. Strategi ini dapat diimplementasikan pada mesin yang jarang mengalami kerusakan namun dalam sekali kerusakan, membutuhkan *downtime* yang lama.

e. *Design Out Maintenance* (DOM)

DOM adalah strategi yang dilakukan jika biaya pemeliharaan mesin sangat tinggi. Strategi ini berbeda dari strategi lainnya. DOM bertujuan untuk mendesain ulang atau mengganti beberapa peralatan atau bagian pada mesin yang memiliki biaya perawatan paling tinggi atau yang memiliki frekuensi dan downtime kerusakan paling tinggi .

2.4. Pengertian Website

Sebuah situs web (sering pula disingkat menjadi situs saja, *website* atau *site* adalah sebutan bagi sekelompok halaman web (*web page*), yang umumnya

merupakan bagian dari suatu nama domain (*domain name*) atau subdomain di *World Wide Web* (WWW) di Internet. Sebuah web page 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* baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*) (Ali Zaki, 2009).

Penemu *website* adalah Sir Timothy John “Tim” Berners -Lee, sedangkan *website* yang tersambung dengan jaringan, pertama kali muncul pada tahun 1991. Maksud dari Tim ketika membuat *website* adalah untuk mempermudah tukar-menukar dan memperbarui informasi kepada sesama peneliti di tempat bekerja. Pada tanggal 30 April 1993, CERN (tempat dimana Tim bekerja) menginformasikan bahwa WWW dapat digunakan secara gratis oleh semua orang yang dapat diakses melalui sebuah *software* yang disebut *browser*, seperti *internet exploer*, *mozilla firefox*, *opera* dan lain-lain.(Rahmat Hidayat, 2010)

2.5. Progressive Web Apps (PWA)

Progressive Web Apps (PWAs) merupakan aplikasi berbasis web yang sama seperti halaman *web* ataupun situs *web* biasa hanya saja menawarkan fungsionalitas yang berbeda seperti dapat bekerja secara *offline*, dapat memberikan *push notification* dan akses *hardware* yang tersedia untuk aplikasi *native*. PWA ini sendiri dikembangkan oleh Google. *Progressive Web Apps*

menggunakan beberapa teknologi diantaranya *Hypertext Transfer Protocol Secure* (HTTPS), *Manifest* dan *Service Workers*.

Pada Mei 2016 Google memperkenalkan Progressive web apps. Progressive web apps dirancang oleh Frances Berriman dan Google Chrome Engineer Alex Russel. Teknologi dibalik progressive web apps adalah service worker. Menurut google developer karakteristik dari progressive web apps adalah sebagai berikut :

- **Progressive** – dapat digunakan oleh semua pengguna, terlepas browser apa yang digunakan karena aplikasi dikembangkan secara progressive.
- **Responsive** – dapat digunakan pada semua perangkat mulai dari desktop, tablet, smartphone dan lainnya.
- **Connectivity** independent – memiliki service workers untuk dapat diakses secara offline atau pada kualitas koneksi jaringan yang rendah.
- **App-like** – terasa seperti aplikasi karena model aplikasi shell memisahkan fungsi dari konten aplikasi.
- **Fresh** – selalu *up-to-date* dikarenakan update proses dari service worker.
- **Safe** – dilayani via HTTPS untuk mencegah pengintaian dan untuk memastikan konten tidak dirusak.
- **Discoverable** – yaitu diidentifikasi sebagai "aplikasi" berkat cakupan W3C dan ruang lingkup pendaftaran *service worker*, yang memungkinkan mesin pencari menemukannya.

- ***Re-engageable*** – yaitu membuat keterlibatan ulang menjadi mudah dengan adanya fitur seperti *push notification*.
- ***Installable*** – memungkinkan user untuk menambahkan aplikasi ke home screen tanpa melalui appstore.
- ***Linkable*** – dapat berbagi dengan mudah melalui URL dan tidak memerlukan instalasi.

Progressive Web Apps memiliki tiga kebutuhan yang harus dipenuhi, yaitu:

- Harus dilayani melalui *Hypertext Transfer Protocol Secure* (HTTPS).
- Memiliki *web app manifest*.
- Memiliki setidaknya satu *service worker*.

2.5.1. Single-page Application

Paradigma baru dimulai sejak sekitar tahun 2005 ketika model pemrograman *asynchronous Java script* dan XML (AJAX) menjadi daya tarik baru. AJAX memungkinkan halaman *web* untuk melakukan 12 permintaan HTTP ke *web server*, dan melakukan pembaharuan hanya pada bagian halaman *web* yang berubah. Revolusi lain dimulai pada sekitar tahun 2006 ketika dua kelompok yaitu *World Wide Web Consortium (W3C)* dan *Web Hypertext Application Technology Working Group (WHATWG)*, mulai bekerja bersama untuk sebuah standar *Hypertext Markup Language (HTML)* yang baru yaitu HTML 5. Standar HTML 5 yang baru menambahkan berbagai macam fitur kedalam sebuah

perangkat *web* yang memungkinkan pengembang untuk membangun sebuah *web* yang lebih kompleks dan lebih kaya akan fitur (Vanhala 2017).

Evolusi *web* mengantarkan kepada teknologi baru dalam membangun sebuah aplikasi *web*. *Single Page Application* (SPA) adalah aplikasi *web* yang menggunakan halaman HTML yang tidak melakukan *refresh* pada saat digunakan. Sebagai gantinya SPA menggunakan *Java script* untuk setiap interaksi pengguna dan mengandalkan AJAX ketika ingin melakukan komunikasi dengan server. SPA memberikan pengalaman user yang lebih baik, waktu respon yang lebih cepat dari *web* tradisional sejak interaksi server terjadi secara *asynchronous* pada sisi belakang aplikasi, dan memerlukan data yang sedikit dalam melakukan transaksi (Vanhala 2017).

2.5.2. Web app Manifest

Web app manifest merupakan *file JSON* yang memberitahukan *browser* tentang aplikasi *web* yang dijalankan seperti bagaimana harusnya berperilaku ketika dilakukan instalasi pada perangkat selular atau *desktop*, ini diperlukan untuk memunculkan permintaan “Tambahkan ke Layar Beranda”.

2.5.3. Service Workers

Service worker adalah salah satu jenis dari *web worker*, yaitu script yang berjalan di belakang *browser* pengguna. *Service worker* pada dasarnya adalah berkas *JavaScript* yang berjalan pada *thread* yang berbeda dengan *main thread browser*, menangani *network request*, *caching*, mengembalikan *resource* dari *cache*, dan bisa mengirimkan *push message*. Aset *web* dapat disimpan sebagai

cache lokal, sehingga dengan jaringan internet yang kurang memadai pun, pengguna tetap mendapat pengalaman yang baik. Aplikasi dapat tetap menjalankan halaman web yang sudah di-*cache* atau memberikan status koneksi tanpa *browser* menampilkan tulisan error karena ketiadaan koneksi internet.

Langkah-langkah dalam melakukan konfigurasi dasar *service worker* sebagai berikut:

- Daftarkan *service worker* melalui URL (*Uniform Resource Locator*) fungsi *serviceWorkerContainer.register()*.
- Jika berhasil, *service worker* dijalankan di *ServiceWorkerGlobalScope*.
- *Service worker* telah siap untuk memproses *event*.
- Instalasi *service worker* dicoba ketika *service worker* mengontrol halaman yang diakses setelah dan sebelumnya. *Event install* akan selalu dikirim pertama kali ke *service worker*.
- Ketika handler *oninstall* selesai, *service worker* dipasang.
- Proses aktivasi. Ketika *service worker* terpasang, selanjutnya akan menerima *event activate*. Penggunaan utama dari *onactivate* ini adalah untuk membersihkan sumber daya yang digunakan sebelumnya.

Service control sekarang dapat mengontrol halaman, tapi hanya dibuka setelah *register()* telah sukses seperti dokumen mulai aktif dengan atau tanpa *service worker* dan menjaganya selama masih digunakan. Jadi dokumen harus dimuat ulang agar benar-benar terkontrol.

2.5.4. JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, *JavaScript*, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

2.5.5. Cache

Cache interface menyediakan mekanisme penyimpanan untuk pasangan objek *Request* dan *Response* mau disimpan ke dalam *cache*, contohnya sebagai bagian dari daur hidup *service worker*. Perlu diketahui bahwa *cache interface* terbuka terhadap halaman web dan juga *workers*. *Cache* tidak harus selalu digunakan bersamaan dengan *service worker* walaupun *cache* tercantum di dalam

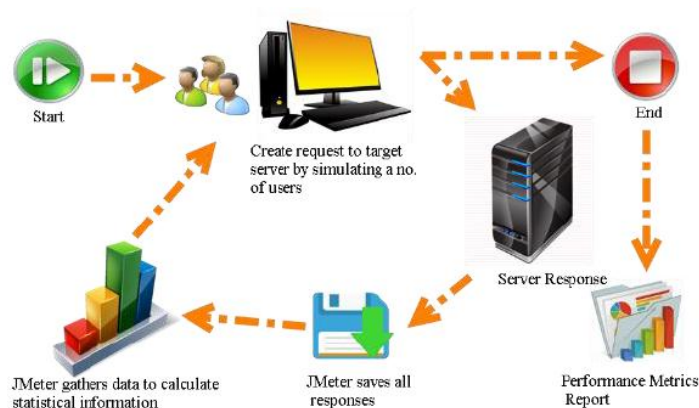
spesifikasinya. *Cache* digambarkan sebagai sebuah *array* berisi objek *Request* yang bertindak sebagai pasangan untuk responsnya yang disimpan di dalam *browser*.

2.6. Apache Jmeter

Apache JMeter merupakan aplikasi berbasis Java yang *open source* dan dapat digunakan untuk melakukan tes atau simulasi untuk memberikan sejumlah *request* untuk mengukur performansi dan maupun kinerja dari suatu sistem. Pada dasarnya JMeter mampu melakukan tes dengan mengirimkan HTTP *request* dan mampu menangkap kembali hasil yang diinginkan. Selain itu, JMeter juga dapat mensimulasikan kemampuan *load* dari suatu *server* yang diujikan.

Berikut 3 jenis test yang dapat dilakukan dengan JMeter:

1. **Uji Kinerja (*Performance Test*)** – Tes ini akan menunjukkan ekspektasi kinerja terbaik pada konfigurasi infrastruktur yang diberikan. Hal ini sangat penting pada awal proses pengujian, menunjukkan apakah diperlukan perubahan sebelum aplikasi diproduksi.
2. **Uji Beban (*Load Test*)** – Uji ini pada dasarnya digunakan untuk menguji sistem *under the top* yang telah dikonfigurasi.
3. **Uji Tekanan (*Stress Test*)** – Tes ini akan melakukan percobaan untuk merusak sistem dengan cara memberikan kinerja sumber daya secara berlebihan.



Gambar 2.2 Alur Kerja Aplikasi Apache JMeter

Pada gambar 2.2 adalah alur kerja aplikasi Apache JMeter. Cara kerja JMeter dengan *web browser* hampir sama, yaitu sama-sama melakukan HTTP *Request* ke *server*. Tetapi JMeter bukanlah *web browser*. Perbedaan utamanya adalah *web browser* dapat melakukan *request* seperti dengan menekan tombol dan lainnya tetapi JMeter hanya melakukan *request* dengan HTTP *request*. Setelah *server* memberi *respond*, *web browser* akan memparse *respond* dan mengeksekusi *Java Script*, sedangkan JMeter hanya bisa memparse *respond* tanpa mengeksekusi *java script*. Melainkan JMeter menghasilkan data yang nantinya dapat di analisis. Pengguna dapat memasukkan beberapa parameter dan elemen sehingga bisa dijadikan bahan uji menggunakan aplikasi JMeter ini. Agar aplikasi JMeter dapat digunakan sebagai alat penguji, aplikasi harus di konfigurasi sesuai kebutuhan *client* dapat menambahkan *sampler*, *listener*, *timer* dan lainnya sesuai kebutuhan. (Putra, 2019)

2.7. Lighthouse

Lighthouse adalah aplikasi *open source* yang dibangun oleh Google untuk menguji konsep PWA dengan aspek-aspeknya, *performance*, *accessibility* dan

best practices. Lighthouse dapat dijalankan dari *browser* pengguna berupa ekstensi pada Google Chrome. Pengujian menggunakan Lighthouse hanya bisa dijalankan untuk satu pengujian pada satu *browser client*. Untuk melakukan pengujian menggunakan Lighthouse pada banyak *client* digunakan layanan *third-party* Calibre.

2.8. JavaScript

JavaScript adalah bahasa scripting kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client*. *JavaScript* digunakan dalam pembuatan website agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi browser. *JavaScript* dapat merespon perintah user dengan cepat dan menjadikan halaman web menjadi responsif. *JavaScript* memiliki struktur sederhana, kodenya dapat disisipkan pada dokumen HTML atau berdiri sebagai satu kesatuan aplikasi.

2.8.1. VueJs

VueJs adalah *JavaScript framework* yang dikembangkan untuk membangun antarmuka suatu *software*. *VueJs* telah menyediakan berbagai macam fungsi *JavaScript* yang telah dimodifikasi sehingga programmer dapat lebih mudah untuk membangun *software*, tentunya dengan aturan-aturan tertentu.

VueJs pertama kali dirilis pada Februari 2014 oleh Evan You setelah bekerja di *Google* menggunakan *AngularJS* di beberapa proyek. Berbeda dengan *framework* lain yang menggunakan prinsip *MVC(Model View Controller)* *VueJs* hanya difokuskan untuk membangun tampilan atau hanya bekerja pada *view layer*.

Dalam hal lainnya *VueJs* juga dapat membantu *programmer* untuk membuat web dengan aplikasi *single page* dan dapat di kombinasikan dengan *library browser* lainnya. Selain itu *VueJs* menggunakan template sintaks berbasis *HTML* yang memungkinkan pengguna untuk mendeklarasikan data atau state kedalam *DOM*. Semua template *VueJs* adalah *HTML* yang valid yang dapat diuraikan oleh browser sesuai spesifikasi dan parser *HTML*.

DOM menyediakan struktur dokumen yang berorientasi objek dan setiap elemen yang menyusun *DOM* memiliki property dan nilai nya masing masing. Saat pertama kali *web browser* mengunjungi sebuah halaman, *web browser* akan memarsing teks tersebut menjadi *DOM*.

2.9. Basis Data

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol).

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengembalian kembali. Untuk mencapai tujuannya, syarat sebuah basis data yang baik adalah sebagai berikut: Tidak adanya redudansi dan inkonsistensi data, kesulitan pengaksesan data, dan *Multiple user*.

2.9.1. Basis Data NoSQL

Istilah *NoSQL* diciptakan oleh Carlo Strozzi pada tahun 1998 dan mengacu pada *database* non-relasional. Pada tahun 2009 Eric Evans memperkenalkan

kembali istilah *NoSQL*. *NoSQL* mengakomodasi tanda yang tidak terstruktur, kehadirannya bukan untuk menggantikan *SQL* namun kedua teknologi ini dapat saling berdampingan. Perbedaan utama kedua *database* ini adalah *SQL* memiliki skema yang kaku sementara *database NoSQL* menawarkan desain yang fleksibel yang dapat diubah tanpa downtime atau gangguan layanan.

NoSQL menjadi solusi dalam penanganan data dalam jumlah besar yang berkembang pesat saat ini. Data ini biasanya non-terstruktur, kompleks dan tidak cocok digunakan dalam model relasional. Contoh data yang bisa kita rasakan adalah data yang berasal dari smartphone yang mencatat lokasi *broadcast* setiap saat, video dan kamera bahkan halaman halaman *website* yang berisi banyak informasi serta dokumen .

Basis data *NoSQL* mempunyai karakteristik *BASE* (*Basically, Available, Soft state and Eventual Consistency*) yang merupakan kebalikan dari *ACID* pada basis data *SQL*. Setelah transaksi yang konsisten, keadaan (*state*) yang didapat adalah keadaan sementara (*soft state*) bukan keadaan tetap (*solid state*). Fokus utama dari *BASE* adalah ketersediaan permanen. Karakteristik berikutnya adalah *CAP* (*Consistency, Availability dan Partition*) yang mempunyai tiga prinsip utama:

- Data tersedia pada semua mesin harus sama di semua aspek dan *updatem* harus dilakukan terhadap semua mesin.
- Data harus tersedia secara permanen dan harus dapat diakses setiap saat.
- Pada saat terjadi kegagalan mesin atau kesalahan lainnya, basis data tetap bekerja dengan baik tanpa ada pekerjaan yang berhenti.

2.10. *Firestore*

Firestore Adalah *Backend as a Service* (BaaS) yang saat ini dimiliki oleh Google. *Firestore Database* merupakan penyimpanan basis data non-SQL yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data itu antara lain *string*, *long*, dan *boolean*. Data pada *Firestore Database* disimpan sebagai objek JSON *tree*. Tidak seperti basis data SQL, tidak ada tabel dan baris pada basis data non-SQL. Ketika ada penambahan data, data tersebut akan menjadi *node* pada struktur JSON. *Node* merupakan simpul yang berisi data dan bisa memiliki cabang-cabang berupa *node* lainnya yang berisi data pula. Proses pengisian suatu data ke *Firestore Database* dikenal dengan istilah *push*.

2.10.1. *Firestore Firestore*

Cloud Firestore adalah *database* yang fleksibel dan skalabel untuk pengembangan seluler, web, dan server di *Firestore* dan Google Cloud Platform. Seperti *Firestore Realtime Database*, *Cloud Firestore* membuat data tetap terhubung di aplikasi klien melalui *listener realtime* dan menawarkan dukungan secara *offline* untuk seluler dan web. Dengan begitu, *cloud firestore* mendukung proses pembuatan aplikasi yang responsif dan mampu bekerja tanpa harus bergantung pada latensi jaringan atau koneksi Internet. *Cloud Firestore* juga menawarkan integrasi yang lancar dengan produk *Firestore* dan Google Cloud Platform lainnya, termasuk Cloud Functions.