

Aplikasi Asesmen Nasional Berbasis Komputer Dengan
Menggunakan Progressive Web App



SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan
Pendidikan Diploma Empat (D-4) Teknik Kompur dan Jaringan

Teknik Elektro

Politeknik Negeri Ujung Pandang

La Ode Muhamad Aslan

42518037

PROGRAM STUDI D-4 TEKNIK KOMPUTER DAN JARINGAN

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI UJUNG PANDANG

MAKASSAR

2022

HALAMAN PENGESAHAN

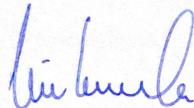
HALAMAN PENGESAHAN

Skripsi ini dengan judul “**Aplikasi Asesmen Nasional Berbasis Komputer dengan Menggunakan Progressive Web App**” oleh La Ode Muhamad Aslan NIM 42518037 telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Diploma IV (D- 4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

Makassar, 2022

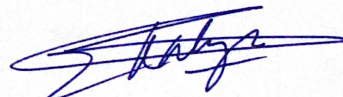
Mengesahkan,

Pembimbing I,



Lin Karmila Yusri, S.ST, M.Eng., Ph.D
NIP. 197604032002122001

Pembimbing II,



Muh. Fajri Raharjo, S.T., M.T
NIP. 197005211996011001

Mengetahui,

Koordinator Program Studi
Teknik Komputer dan Jaringan
Politeknik Negeri Ujung Pandang



Eddy Tungadi, S.T., M.T.
NIP. 197908232010121001



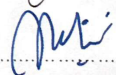
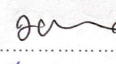


HALAMAN PENERIMAAN

HALAMAN PENERIMAAN

Pada hari ini, Jumat Tanggal 11 Agustus Tim Penguji Sidang Tugas Akhir, telah menerima dengan baik hasil skripsi oleh mahasiswa: **La Ode Muhamad Aslan** NIM 42518037 dengan judul “**Aplikasi Asesmen Nasional Berbasis Komputer dengan Menggunakan Progressive Web App**”.

Makassar, 11 Agustus 2022

Tim Penguji Ujian Skripsi :

- | | | |
|---|------------|---|
| 1. Irfan Syamsuddin, S.T., M.Com.ISM., Ph.D | Ketua | () |
| 2. Eddy Tungadi, S.T., M.T. | Sekretaris | () |
| 3. Irmawati, S.T., M.T. | Anggota | () |
| 4. Meylanie Olivya, S.T., M.T. | Anggota | () |
| 5. Iin Karmila Yusri, S.ST., M.Eng., Ph.D | Anggota | () |
| 6. Muh. Fajri Raharjo, S.T., M.T. | Anggota | () |

HALAMAN PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : La Ode Muhamad Aslan

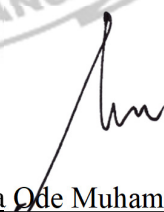
NIM : 42518037

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul “**APLIKASI ASESMEN NASIONAL BERBASIS KOMPUTER DENGAN MENGGUNAKAN PROGRESSIVE WEB APP**” merupakan gagasan dan hasil karya sendiri dengan arahan komisi pembimbing dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi manpun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan di atas tidak benar, saya siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 25 Juli 2022


La Ode Muhamad Aslan
42518037

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat Allah SWT atas berkat dan rahmatNya yang telah memberikan Kesehatan serta keselamatan sehingga saya dapat menyelesaikan skripsi ini yang berjudul “APLIKASI ASESMEN NASIONAL BERBASIS KOMPUTER DENGAN MENGGUNAKAN PROGRESSIVE WEB APP” dengan baik. Shalawat dan salam kepada baginda Rasul Muhammad Shallallahu Alaihi Wa Sallam sebagai sebaik-baik panutan bagi seluruh manusia.

Skripsi ini dibuat guna memenuhi salah satu syarat untuk menyelesaikan studi serta dalam rangka memperoleh gelar Diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang. Penulis menyadari bahwa keberhasilan penyusunan skripsi ini tidak lepas dari bantuan banyak pihak baik secara langsung maupun tidak langsung. Oleh sebab itu, melalui kesempatan ini penulis menyampaikan penghargaan dan ucapan terimakasih kepada:

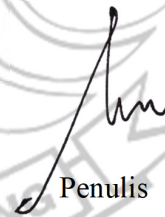
1. Kedua orang tua penulis yakni Bapak La Ode Ane dan Ibu Sarina serta seluruh keluarga penulis yang tidak hentinya memberikan dukungan, motivasi serta mendoakan penulis.
2. Bapak Prof. Ir. Muhammad Ansar, M.Si., Ph.D. selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Ahmad Rizal Sultan, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Bapak Eddy Tungadi, S.T., M.T. selaku Koordinator Program Studi Teknik Komputer dan Jaringan Politeknik Negeri Ujung Pandang.
5. Bapak/Ibu dosen dan staf jurusan Teknik Elektro khususnya Program Studi Teknik Komputer dan Jaringan yang telah memberikan ilmu pengetahuan yang sangat berharga selama mengikuti pendidikan di Politeknik Negeri Ujung Pandang
6. Ibu Iin Karmila Yusri, S.ST., M.Eng., Ph.D. selaku pembimbing I dan Bapak Muh. Fajri Raharjo, S.T., M.T. selaku pembimbing II atas segala v

ilmu, nasehat, arahan, bantuan, kesediaan waktu dan tenaga serta kesabarannya dalam membimbing saya sehingga terselesaikannya penelitian ini.

7. Teruntuk teman-teman seperjuangan di Program Studi TKJ angkatan 2018 Politeknik Negeri Ujung Pandang yang mempunyai peranan besar dalam membantu penyusunan skripsi ini dan memberikan arti kebersamaan selama menyelesaikan studi di Politeknik Negeri Ujung Pandang.
8. Semua pihak yang telah memberikan bantuan moril maupun material hingga skripsi ini dapat diselesaikan, semoga Allah SWT senantiasa memberikan imbalan pahala yang berlipat ganda untuk kita semua.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan, sehingga penulis mengharapkan kritik dan saran yang sifatnya membangun untuk perbaikan di masa mendatang. Semoga apa yang penulis lakukan dapat bermanfaat bagi pembaca supaya dapat menghasilkan sumber daya manusia yang berkualitas dan berguna bagi bangsa dan negara.

Makassar, 25 Juli 2022



Penulis

RINGKASAN

Asesmen nasional adalah program penilaian terhadap mutu setiap sekolah, madrasah, dan program kesetaraan pada jenjang dasar dan menengah. Aplikasi asesmen nasional memanfaatkan konektivitas jaringan untuk mengirim data hasil asesmen nasional dari aplikasi ke server. Tetapi, kurangnya pemerataan sarana dan infrastruktur jaringan menjadi kendala dalam penggunaan aplikasi asesmen nasional. Oleh karena itu, untuk membangun sistem aplikasi asesmen nasional yang bisa diakses tanpa memiliki ketergantungan dengan konektivitas jaringan, maka sistem akan dibangun dengan menggunakan konsep Progressive Web App (PWA). Dari hasil implementasi PWA pada sistem menunjukkan bahwa, dengan menggunakan *service worker* akan membuat *website* tidak bergantung pada konektivitas jaringan sehingga dapat diakses secara *offline*. Sedangkan, dengan menggunakan *web app manifest*, *website* dapat memunculkan *pop-up add to homescreen* yang berguna untuk melakukan instalasi sistem pada *homescreen* pengguna sehingga aplikasi bisa di akses dengan cepat melalui ikon *homescreen*. Dengan menggunakan *indexedDB*, meskipun dalam keadaan *offline* pengguna tetap bisa mengirimkan data dan data akan tersimpan ke *indexedDB* dan ketika keadaan jaringan kembali *online* data akan otomatis disinkronkan dengan server. Dari hasil pengujian kualitas PWA menggunakan *Lighthouse* menghasilkan rata-rata nilai 100 pada kriteria *progressive web app*, 96,5 pada kriteria *performance*, 92,75 pada kriteria *accessibility* dan 100 untuk kriteria *best practice* dan *seo*.

Kata kunci : Aplikasi Asesmen Nasional, *Progressive Web App* (PWA), *Service Worker* , Konektivitas Jaringan, Satuan Pendidikan.

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
HALAMAN PENERIMAAN	iii
HALAMAN PERNYATAAN	iv
KATA PENGANTAR	v
RINGKASAN	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Ruang Lingkup Penelitian.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB 2 TINJAUAN PUSTAKA	4
2.1 Asesmen Nasional	4
2.1.1 Struktur Asesmen Nasional.....	4
2.1.2 Asesmen Kompetensi Minimum.....	5
2.2 <i>Web Application</i>	6
2.3 <i>Progressive Web App</i>	6
2.3.1 <i>Web App Manifest</i>	7
2.3.2 <i>Service Worker</i>	7
2.3.3 <i>IndexedDB</i>	9
2.4 <i>Application Programming Interface (API)</i>	10
2.5 <i>Browser</i>	11
2.6 <i>Lighthouse</i>	11
BAB 3 METODE PENELITIAN	12
3.1 Tempat dan Waktu Penelitian	12
3.2 Alat dan Bahan Penelitian	12
3.3 Alur Penelitian.....	13

3.3.1	Studi Literatur	13
3.3.2	Analisis Kebutuhan	13
3.3.3	Perancangan Sistem	13
3.3.4	Implementasi	24
3.3.5	Pengujian.....	24
BAB 4	HASIL DAN PEMBAHASAN	26
4.1	Implementasi Tampilan Antar Muka	26
4.1.1	Implementasi Proses Mengelola Data Topik.....	26
4.1.2	Implementasi Proses Mengelola Soal dan Daftar Soal.....	28
4.1.3	Implementasi Proses Mengelola Data Peserta.....	31
4.1.4	Implementasi Proses Mengelola Asesmen Nasional.....	35
4.1.5	Implementasi Mengikuti Asesmen Nasional.....	35
4.2	Pengujian Sistem <i>Progressive Web App</i>	37
4.2.1	Pengujian <i>Web App Manifest</i>	37
4.2.2	Pengujian <i>Service Worker</i>	39
4.2.3	Pengujian IndexedDB	41
4.3	Pengujian Sistem Menggunakan <i>Lighthouse</i>	43
BAB 5	KESIMPULAN DAN SARAN	46
5.1	Kesimpulan.....	46
5.2	Saran.....	46
DAFTAR PUSTAKA	47
LAMPIRAN	49

DAFTAR GAMBAR

Gambar 2.1 Daur Hidup <i>Service Worker</i>	8
Gambar 3.1 Alur penelitian.....	13
Gambar 3.2 Arsitektur sistem <i>progressive web app</i>	18
Gambar 3.3 <i>Use case diagram</i> role admin dan siswa	19
Gambar 3.4 Admin mengelolah data topik dan modul	20
Gambar 3.5 Admin mengelola data soal	21
Gambar 3.6 Admin mengelolah data siswa.....	21
Gambar 3.7 Admin mengelolah data ujian	22
Gambar 3.8 Siswa mengerjakan ujian.....	22
Gambar 3.9 <i>Logical Relational Structure</i>	23
Gambar 4.1 Antar muka data topik.....	27
Gambar 4.2 Antar muka menambah topik baru	27
Gambar 4.3 Antarmuka mengubah topik	28
Gambar 4.4 Antarmuka mengelolah soal dan daftar soal	29
Gambar 4.5 Antarmuka menambahkan jawaban	30
Gambar 4.6 Antarmuka menambah jawaban	30
Gambar 4.7 Antarmuka daftar soal	31
Gambar 4.8 Antarmuka daftar group	32
Gambar 4.9 Antarmuka tambah group.....	32
Gambar 4.10 Antarmuka edit group.....	33
Gambar 4.11 Antarmuka daftar peserta	33
Gambar 4.12 Antarmuka tambah peserta.....	34
Gambar 4.13 Antarmuka edit peserta.....	34
Gambar 4.14 Antarmuka tambah tes.....	35
Gambar 4.15 Tampilan aplikasi ketika jaringan <i>online</i>	36
Gambar 4.16 <i>Page</i> dimana jaringan terputus.....	36
Gambar 4.17 <i>Page</i> soal selanjutnya.....	37
Gambar 4.18 Pengujian <i>web app manifest</i>	38
Gambar 4.19 Pengujian menginstall web app ke <i>homescreen</i>	39
Gambar 4.20 Pengujian membuka web app melalui <i>homescreen</i>	39
Gambar 4.21 Pengujian <i>Service Worker</i>	40
Gambar 4.22 <i>Request table</i> pada <i>chrome dev tools</i>	41
Gambar 4.23 Pengujian menyimpal hasil ujian secara offline.....	43
Gambar 4.24 Pengujian sinkronkan data dari indexedDB ke database server.....	43

DAFTAR TABEL

Tabel 2.1 Kriteria penilaian hasil uji <i>lighthouse</i>	11
Tabel 3.1 Analisa Sistem PWA.....	15
Tabel 4.1 Pengujian 5 <i>user</i> awal	44
Tabel 4.2 Pengujian dengan menambahkan 5 <i>user</i> pertama	44
Tabel 4.3 Pengujian dengan menambahkan 5 <i>user</i> kedua	44



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pendidikan menjadi salah satu tolak ukur dalam upaya meningkatkan taraf hidup masyarakat dan berperan sangat penting dalam membangun dan membentuk manusia yang berkualitas. salah satu strategi yang dilakukan pemerintah untuk meningkatkan kualitas Pendidikan di Indonesia yaitu dengan melakukan pendekatan teknologi (Warsihna, 2013). Perkembangan teknologi sekarang ini semakin cepat seiring dengan kebutuhan masyarakat akan layanan internet. Perkembangan teknologi yang sangat pesat bisa dimanfaatkan seperti halnya untuk membantu di dunia pendidikan. Di dunia pendidikan, salah satu peran teknologi yaitu melakukan otomatisasi dan melakukan perubahan – perubahan terhadap suatu tugas atau proses (Diskominfo, 2020). Indonesia memiliki sistem pendidikan formal yang umumnya bersifat wajib sebagai bentuk upaya dalam meningkatkan kualitas pendidikan. Untuk menyikapi kemajuan teknologi yang terus berkembang, dengan teknologi *E-Service* dapat dirancang sebuah sistem yang nantinya bisa mempermudah aktivitas sekolah. *E-Service* adalah istilah yang digunakan oleh layanan elektronik yang memanfaatkan perkembangan layanan teknologi dan komunikasi untuk mempermudah suatu pekerjaan (Pratiwi, 2013).

E-Service bisa dimanfaatkan dalam upaya meningkatkan kualitas pendidikan di Indonesia salah satunya yaitu Asesmen Nasional Berbasis Komputer (ANBK). Asesmen nasional adalah program penilaian terhadap mutu setiap sekolah, madrasah, dan program kesetaraan pada jenjang dasar dan menengah. Mutu satuan pendidikan dinilai berdasarkan hasil belajar murid yang mendasar (literasi, numerasi) serta kualitas proses belajar-mengajar dan iklim satuan pendidikan yang mendukung pembelajaran. Asesmen nasional menghasilkan informasi untuk memantau perkembangan mutu dari waktu ke waktu, dan kesenjangan antar bagian di dalam sistem pendidikan. Asesmen nasional tidak lagi dirancang untuk mengevaluasi capaian murid secara individu, akan tetapi untuk mengevaluasi dan memetakan sistem pendidikan berupa input, proses dan hasil. Inputan dari asesmen

nasional diperoleh dari tiga instrumen utama, yaitu Asesmen Kompetensi Minimum (AKM), Survei Karakter, dan Survei Lingkungan Belajar (Kemdikbud, 2021). Tetapi yang menjadi kendala adalah belum ada pemerataan sarana dan infrastruktur jaringan di Indonesia. Kemkominfo melaporkan pada tahun 2020 ada sekitar 9.113 daerah yang tidak terjangkau jaringan 4G, dan 3.435 daerah non tertinggal, terluar, terbelakang (3T) yang tidak terjangkau jaringan ini. jika ditotal, ada sekitar 12.548 daerah titik kosong di Indonesia. Salah satu faktor yang menjadi kendala dalam upaya pemerataan jaringan internet di Indonesia adalah letak geografisnya (CNN, 2020). Selain masalah jaringan, masalah lainnya yang menjadikan susah nya pelaksanaan asesmen nasional berbasis komputer adalah kurangnya fasilitas internet yang disediakan oleh sekolah (Ane, 2021). Oleh karena itu, ketersediaan aplikasi ANBK yang memenuhi kebutuhan dan stabil dalam keadaan koneksi internet yang kurang bagus sangat dibutuhkan untuk mengatasi kendala terhadap terbatasnya ketersediaan akses jaringan.

Pada penelitian kali ini, dibuat aplikasi ANBK dengan menggunakan konsep *Progressive Web App* (PWA). PWA merupakan konsep baru yang diperkenalkan oleh Alex Russel dan Frances Berriman pada tahun 2015. Konsep ini mencakup teknologi baru dari *web browser* seperti *service worker*, *web app manifest* dan *indexedDB*. Aplikasi Asesmen Nasional menggunakan PWA memungkinkan siswa tetap bisa melaksanakan asesmen nasional meskipun pada kondisi kurang stabil, hal ini dikarenakan PWA melakukan *caching* terhadap konten aplikasi asesmen nasional dan menyediakannya dalam bentuk *offline*, selain itu juga aplikasi asesmen nasional juga bisa mengirimkan data hasil asesmen ke server meskipun dalam kondisi tidak ada jaringan atau *offline* (Osmani, 2015). Dengan adanya aplikasi ini diharapkan dapat memberikan efisiensi dan mempermudah sekolah dalam melaksanakan asesmen nasional.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dibuat, maka rumusan masalah pada penelitian ini yaitu bagaimana membuat aplikasi asesmen nasional berbasis komputer dengan menggunakan *progressive web app*.

1.3 Ruang Lingkup Penelitian

1. Bentuk soal yang digunakan adalah pilihan ganda dan *essay* (isian singkat).
2. Konsep Progressive Web App diterapkan pada instrumen asesmen nasional yaitu asesmen kompetensi minimum.
3. Menggunakan Google Chrome ketika mengakses aplikasi.
4. Menggunakan Windows sebagai sistem operasi *client*.

1.4 Tujuan Penelitian

Adapun manfaat dari penelitian ini yaitu untuk membuat aplikasi asesmen nasional berbasis komputer dengan menggunakan *progressive web app*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut :

1. Membantu sekolah dalam melaksanakan asesmen nasional komputer meskipun dalam keadaan internet yang kurang stabil.
2. Dapat mempermudah sekolah dalam melaksanakan asesmen nasional sekolah.
3. Memberikan kontribusi terhadap peningkatan kualitas pendidikan di Indonesia dengan menggunakan pendekatan teknologi.

BAB 2 TINJAUAN PUSTAKA

2.1 Asesmen Nasional

Asesmen nasional adalah program penilaian terhadap mutu setiap sekolah, madrasah, dan program kesetaraan pada jenjang dasar dan menengah. Mutu satuan pendidikan dinilai berdasarkan hasil belajar murid yang mendasar (literasi, numerasi, dan karakter) serta kualitas proses belajar-mengajar dan iklim satuan pendidikan yang mendukung pembelajaran (Kemdikbud, 2021).

Asesmen Nasional perlu dilakukan untuk meningkatkan mutu pendidikan. Asesmen Nasional menghasilkan informasi untuk memantau perkembangan mutu dari waktu ke waktu, dan kesenjangan antar bagian di dalam sistem pendidikan (misalnya kesenjangan antarkelompok sosial ekonomi dalam satuan pendidikan, kesenjangan antara satuan Pendidikan negeri dan swasta di suatu wilayah, kesenjangan antardaerah, atau pun kesenjangan antarkelompok berdasarkan atribut tertentu). Asesmen Nasional bertujuan untuk menunjukkan apa yang seharusnya menjadi tujuan utama satuan pendidikan, yakni pengembangan kompetensi dan karakter murid. Asesmen Nasional juga memberi gambaran tentang karakteristik esensial sebuah satuan pendidikan yang efektif untuk mencapai tujuan utama tersebut. Hal ini diharapkan dapat mendorong satuan pendidikan dan Dinas Pendidikan untuk memfokuskan sumber daya pada perbaikan mutu pembelajaran (Pusmenjar, 2021).

2.1.1 Struktur Asesmen Nasional

Asesmen ini dirancang untuk menghasilkan informasi akurat untuk memperbaiki kualitas belajar-mengajar, sehingga meningkatkan hasil belajar murid. Informasi-informasi tersebut diperoleh dari tiga instrumen utama, yaitu Asesmen Kompetensi Minimum (AKM), Survei Karakter, dan Survei Lingkungan Belajar.

Asesmen nasional tidak lagi dirancang untuk mengevaluasi capaian murid secara individu, akan tetapi untuk mengevaluasi dan memetakan sistem pendidikan

berupa input, proses dan hasil. Asesmen nasional wajib dilaksanakan di semua sekolah, madrasah dan program pendidikan kesetaraan yang ada di Indonesia, baik di tingkat Provinsi maupun Kabupaten/Kota, baik itu negeri maupun swasta. Sampel peserta didik yang mengikuti asesmen nasional dipilih oleh Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi (Kemdikbud) secara acak. Peserta didik jenjang pendidikan dasar dipilih secara acak dengan jumlah maksimal 30 murid dan jenjang pendidikan menengah dipilih secara acak dengan jumlah maksimal 45 murid. Masing masing jenjang memiliki lima orang peserta cadangan (Kemdikbud, 2021).

2.1.2 Asesmen Kompetensi Minimum

Asesmen Kompetensi Minimum (AKM) merupakan penilaian kompetensi mendasar yang diperlukan oleh semua murid untuk mampu mengembangkan kapasitas diri dan berpartisipasi positif pada masyarakat. Terdapat dua kompetensi mendasar yang diukur AKM: literasi membaca dan literasi matematika (numerasi). Baik pada literasi membaca dan numerasi, kompetensi yang dinilai mencakup keterampilan berpikir logis-sistematis, keterampilan bernalar menggunakan konsep serta pengetahuan yang telah dipelajari, serta keterampilan memilah serta mengolah informasi. AKM menyajikan masalah-masalah dengan beragam konteks yang diharapkan mampu diselesaikan oleh murid menggunakan kompetensi literasi membaca dan numerasi yang dimilikinya. AKM dimaksudkan untuk mengukur kompetensi secara mendalam, tidak sekedar penguasaan konten. Soal literasi membaca dan literasi matematika dibuat oleh Pusat Asesmen dan Mengajar dan masing-masing topik soal terdiri dari dua bentuk soal yaitu pilihan ganda dan essay. Soal pilihan ganda terdiri dari lima pilihan jawaban dengan satu jawaban benar, soal essay terdiri dari satu atau beberapa pertanyaan yang menuntut jawaban tertentu oleh siswa secara individu berdasarkan pendapatnya sendiri. Setiap siswa memiliki kesempatan memberikan jawaban sendiri yang berbeda dengan jawaban siswa lain. Soal AKM berdasarkan indikator-indikator 6 kompetensi yang membentuk lintasan kompetensi hasil belajar yang bersifat kontinum (Pusmenjar, 2021).

2.2 *Web Application*

Wep application atau yang kita ketahui sebagai *web app*, merupakan perangkat lunak yang berbentuk aplikasi yang dapat diakses melalui web browser (Techtarget, 2019). Banyak pengembangan yang dilakukan terhadap aplikasi ini sehingga bisa digunakan di berbagai jenis perangkat dan semua jenis browser. Contoh pengembangan web app yaitu *dinamic web*, *responsif web* dan *progressive web*. *Dinamic web* merupakan website yang memiliki isi konten yang dengan otomatis berubah-ubah setiap waktu. ketika pengguna mengaksesnya maka data dan isi yang ada pada database akan diperbarui setiap saat (Dicoding, 2020). *Responsive web* adalah pengembangan aplikasi web yang di rancang agar dapat menyesuaikan diri sesuai dengan ukuran layar pengguna sehingga diharapkan bisa memberikan kecepatan akses website saat dibuka (Perdana, 2020). *Progressive Web App* adalah metode yang dibuat untuk pengembangan aplikasi web yang diakses menggunakan website tetapi memberikan pengalaman seperti layaknya aplikasi mobile (Developers, 2021).

2.3 *Progressive Web App*

Progressive Web App adalah perangkat lunak yang menggunakan teknologi web, dibangun menggunakan teknologi web umum menggunakan *Hypertext Markup Language (HTML)*, *Cascading Style Sheets (CSS)*, dan *javascript* yang dapat bekerja pada platform manapun yang menggunakan browser sesuai standar, termasuk desktop dan perangkat seluler. Konsep ini mencakup teknologi baru dari web browser seperti *service worker*, *indexedDB* dan *web app manifest* (Osmani, 2015). PWA memiliki karakteristik utama sebagai berikut :

1. *Progressive* : beroperasi bagi semua pengguna, tanpa melihat jenis browser karena dibuat dengan pengembangan progresif dari sebuah inti aplikasi.
2. *Connectivity independent* : dengan *service worker* memungkinkan aplikasi dapat beroperasi dalam keadaan *offline* atau jaringan yang lemah.
3. *App-like* : bekerja seperti layaknya aplikasi native, karena model *app shell* yang memisahkan fungsional aplikasi dari kontennya.

4. *Installable* : pengguna bisa menambahkan aplikasi ke *homescreen* tanpa harus repot mendownload aplikasi di app store atau google play.
5. *Linkable* : dapat membagikan aplikasi dengan membagikan URL, dan tidak memerlukan instalasi yang rumit.

PWA mencakup tiga jenis arsitektur yaitu *Server-Side Rendering* (SSR), *Client-Side Rendering* (CSR), dan gabungan dari keduanya. Arsitektur SSR bekerja melalui browser dengan mengambil data dari halaman melalui HTTPS dan kemudian dikembalikan dalam bentuk halaman lengkap dengan data dinamis yang sudah di-render (Developers, 2021).

2.3.1 Web App Manifest

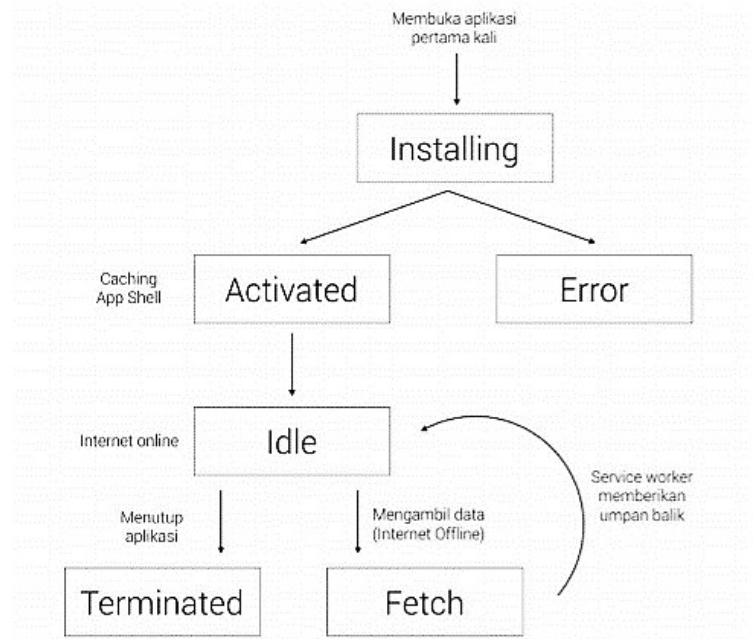
Web app manifest berisi informasi tentang aplikasi, seperti nama, pemilik, ikon dan deskripsi, dalam bentuk file teks. Dengan adanya manifest pengguna bisa menginstall aplikasi web ke *homescreen* sebuah smartphone android, sehingga pengguna mendapatkan akses yang lebih cepat dan pengalaman yang lebih baik. *Web app manifest*, sebagaimana didefinisikan dalam spesifikasi *Web app manifest*, memberikan informasi tentang aplikasi web dalam file teks JSON, yang diperlukan agar aplikasi web diunduh dan disajikan kepada pengguna mirip dengan aplikasi asli (mis., diinstal di layar beranda perangkat, memberikan pengguna akses yang lebih cepat dan pengalaman yang lebih kaya). Manifest PWA mencakup nama, penulis, ikon, versi, deskripsi, dan daftar semua sumber daya yang diperlukan (antara lain).

2.3.2 Service Worker

Service worker merupakan jenis web worker yang terdiri dari script yang bekerja di belakang browser pengguna. Sederhananya *service worker* dibuat dari berkas *JavaScript* (JS) yang bekerja pada thread browser namun lokasinya berbeda dengan thread inti pada browser, yaitu menangani *network request*, *caching*, dan mengembalikan *resource* dari *cache*, dan dapat mengirimkan *push notification*.

Aset halaman web dapat disimpan pada *cache local*, sehingga pengguna tetap mendapatkan pengalaman yang baik meskipun dalam kondisi jaringan yang kurang

memadai. Aplikasi dapat tetap menjalankan halaman web yang sudah disimpan di *cache* atau browser memberikan status koneksi tanpa menampilkan pesan eror karena ketidakadaan koneksi internet.



Gambar 2.1 Daur Hidup *Service Worker*

Service worker memiliki daur hidup yang berbeda dengan halaman web yang ditampilkan pada Gambar 2.1. Untuk bisa menggunakan *service worker* di laman, kita harus mendaftarkannya dengan menggunakan *JavaScript* yang ada 9 pada halaman web. Setelah diregistrasi, browser akan melakukan install *service worker* di latar. Setelah aktif, *service worker* akan menangani semua halaman di bawah scope dimana *service worker* di-install, kemungkinan yang bisa terjadi yaitu terminated menghemat memori, atau menangani fetch ketika ada network request dari halaman web.

Service worker bisa menyimpan berbagai jenis *request*, tetapi hanya semua request jenis GET yang bisa disimpan kedalam *cache*. Pada *service worker* kita bisa aplikasikan strategi *caching* sesuai dengan yang kita butuhkan, tidak ada strategi terbaik yang digunakan untuk *caching* konten dinamis, tergantung situasi ketika kita melakukan *caching* (Developers, 2021).

Ada lima strategi *caching* yang umum digunakan pada *service worker*, yaitu:

1. *Cache first, network fallback*: pertama halaman berkas HTML dan *JavaScript* yang ada di dalam *cache* di kelola oleh *service worker*, jika memungkinkan, melewati network. Jika *cache* tidak tersedia, maka respon dari network dikembalikan oleh *service worker* dan menyimpan *respons* ke *cache*. Strategi ini digunakan untuk menyimpan *resource* yang bersifat statis atau hampir tidak berubah, misalnya gambar.
2. *Network first, cache fallback*: pertama mengecek apakah network memberikan *respons*, dan jika berhasil, data akan dikembalikan langsung ke halaman. Jika gagal, data dari *cache* dikembalikan oleh *service worker*. Strategi ini digunakan jika kita membutuhkan data yang selalu baru misalnya *respons* API, tapi butuh menampilkan sesuatu jika network tidak tersedia.
3. *Cache/network race*: *request* yang sama dilakukan oleh pengguna dijalankan bersamaan ke *network* dan ke *cache*. Pada umumnya, data *cache* dimuat terlebih dahulu dan langsung ditampilkan ke halaman. Sementara *respons* dari network digunakan untuk memperbarui data di *cache*. Proses ini berjalan di latar sehingga tidak mengganggu rendering konten dari *cache*. Strategi ini 10 digunakan ketika konten sering diperbarui secara berkala, seperti artikel, media sosial, dan sebagainya.
4. *Network only*: strategi ini bekerja dengan mengecek network, tidak ada data halaman yang disimpan ke dalam *cache*. Ketika dalam kondisi tidak ada jaringan, maka *request* akan gagal. Strategi ini digunakan dimana hanya data terbaru yang ditampilkan di halaman.
5. *Cache only*: hanya data yang ditambahkan ke *cache* yang akan ditampilkan saat instalasi *service worker* berlangsung. Biasa di gunakan jika kita hanya ingin menampilkan data statis di halaman (Developers, 2021).

2.3.3 IndexedDB

IndexedDB adalah *Application Programming Interface (API) low-level* untuk penyimpanan sisi klien dari sejumlah besar data terstruktur, termasuk

file/gumpalan. API ini menggunakan indeks untuk mengaktifkan pencarian performa tinggi dari data ini. Meskipun Penyimpanan Web berguna untuk menyimpan data dalam jumlah yang lebih kecil, namun kurang berguna untuk menyimpan data terstruktur dalam jumlah yang lebih besar. IndexedDB memberikan solusi.

Dengan IndexedDB ini kita dapat menyimpan data pada browser secara permanen sehingga aplikasi PWA dapat berjalan tanpa harus terhubung dengan server. Dengan cara ini aplikasi dapat berjalan secara cepat karena tidak perlu meminta informasi ke jaringan internet.

Pada IndexedDB ini kita dapat menyimpan data dengan format terstruktur dalam ukuran yang cukup besar. IndexedDb ini merupakan transactinal database system mirip dengan *Relational Database Management System* (RDBMS). Perbedaan IndexedDB dengan RDBMS adalah IndexedDB berbasis Javascript Object sedangkan RDBMS berbasis table.

Sesuai namanya yaitu IndexedDB kita dapat memahami adanya sistem indexing dalam IndexedDB ini. Kita dapat menggunakan IndexedDB untuk 11 menyimpan dan mengambil data berbentuk object dengan menggunakan suatu key. Operasional yang dijalankan oleh IndexedDB adalah asinkron (MDN contributors, 2022)

2.4 *Application Programming Interface (API)*

API adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Pada konteks website, API merupakan pemanggil fungsi melalui *Hyper Text Transfer Protocol* (HTTP) dan mendapatkan respon berupa *Extensible Markup Language* (XML) atau *Javascript Object Notation* (JSON).

Tujuan penggunaan dari API adalah untuk saling berbagi data antar aplikasi yang berbeda. Tujuan penggunaan API lainnya yaitu untuk mempercepat proses

pengembangan aplikasi dengan cara menyediakan sebuah *function* yang terpisah sehingga developer tidak perlu lagi merancang fitur serupa. API yang bekerja pada tingkat sistem operasi membantu aplikasi berkomunikasi dengan layer dasar dan satu sama lain mengikuti serangkaian protokol dan spesifikasi yang telah disesuaikan (Hasanuddin et al., 2022) .

2.5 *Browser*

Peramban web atau *browser* adalah bagian terpenting dari perangkat lunak yang harus dikerjakan oleh perancang web. Tujuan utama dari *browser* adalah untuk terhubung ke *server web*, meminta dokumen, dan kemudian memformat dan menampilkan dokumen-dokumen tersebut dengan benar. Peramban web juga dapat menampilkan file di komputer lokal, mengunduh file, mengirim/menerima email, dapat memainkan animasi flash, membuka file pdf dan sebagainya. *Browser* yang berbeda mungkin memformat dan menampilkan file yang sama dengan cara yang berbeda, tergantung kemampuan sistem dan bagaimana *browser* dikonfirmasi (AZIS, 2021).

2.6 *Lighthouse*

Lighthouse adalah perangkat lunak yang bersifat *open source* untuk meningkatkan kualitas web. *Lighthouse* bisa dijalankan sebagai *ekstensi chrome* atau baris perintah. *Lighthouse* akan melakukan serangkaian pengujian terhadap laman sebuah website, kemudian menghasilkan sebuah laporan mengenai seberapa bagus laman tersebut. *Lighthouse* lebih berfokus pada fitur *progressive web app* seperti *add to homescreen* dan dukungan *offline*. *Lighthouse* mengembalikan skor PWA antara 0 dan 100 (Adi et al., 2018). Penilaian terhadap kualitas PWA bisa dilihat pada tabel 2.1.

Tabel 2.1 Kriteria penilaian hasil uji *lighthouse*

Kode	Kriteria	Skor
SB	Sangat Baik	90-100
B	Baik	50-89
K	Kurang	0-49

BAB 3 METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di Laboratorium Sistem Informasi program studi Teknik Komputer dan Jaringan, Kampus I Politeknik Negeri Ujung Pandang pada bulan Mei sampai dengan bulan Juli tahun 2022. Penelitian ini di implementasikan di wilayah kota Makassar, Sulawesi Selatan.

3.2 Alat dan Bahan Penelitian

Perangkat yang digunakan dalam penelitian ini dapat dikategorikan menjadi perangkat keras (hardware) dan perangkat lunak (software). Adapun perangkat yang digunakan dalam melaksanakan penelitian ini adalah :

1. Alat

a. Perangkat Keras

Laptop memiliki spesifikasi sebagai berikut :

- 1) Memiliki Processor core i5
- 2) Memiliki RAM 8 GB
- 3) Memiliki SSD 1 TB

b. Perangkat lunak

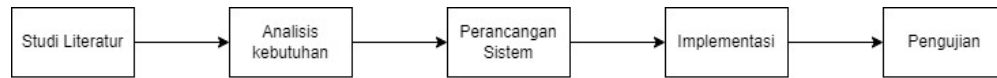
- 1) Sistem operasi windows 10
- 2) MERN stack digunakan untuk menyimpan data, mengelola routing pada website, membuat tampilan pada website, dan menjalankan kode pada sisi server
- 3) Visual Studio Code digunakan untuk menuliskan kode selama proses pembuatan website
- 4) Postman digunakan untuk melakukan pengujian API yang telah dibuat

2. Bahan

- a. Informasi data siswa untuk dijadikan sampel asesmen nasional
- b. Data bank soal asesmen kompetensi minimum (Literasi dan Numerasi) untuk dikerjakan peserta asesmen nasional

3.3 Alur Penelitian

Berikut ini merupakan gambaran alur penelitian yang digambarkan melalui diagram pada Gambar 3.1.



Gambar 3.1 Alur penelitian

3.3.1 Studi Literatur

Studi literatur dilakukan untuk mengumpulkan data dan memperoleh data yang tepat dalam penelitian dengan cara mempelajari, membaca dan mencatat literatur baik berupa jurnal ilmiah atau artikel ilmiah yang tersedia di internet yang berkaitan dengan masalah penelitian.

3.3.2 Analisis Kebutuhan

Kebutuhan sistem yang dibutuhkan untuk penelitian ini terdiri dari perangkat keras dan perangkat lunak sebagai berikut :

a. Kebutuhan Fungsional

Data yang diperlukan sebagai berikut:

1. Data Siswa yang digunakan sebagai informasi yang terdiri dari nama, Nomor Induk Keluarga (NIK), dan lain sebagainya.
2. Data bank soal yang digunakan untuk soal asesmen yang nantinya akan dikerjakan oleh siswa

b. Kebutuhan non-fungsional perangkat lunak

1. Sistem operasi perangkat lunak Windows 10 64-bit
2. MongoDB, Express, ReactJS, dan NodeJs (MERN) stack
3. Web Browser Google Chrome (Version 67.0.3396^)
4. VS Code

3.3.3 Perancangan Sistem

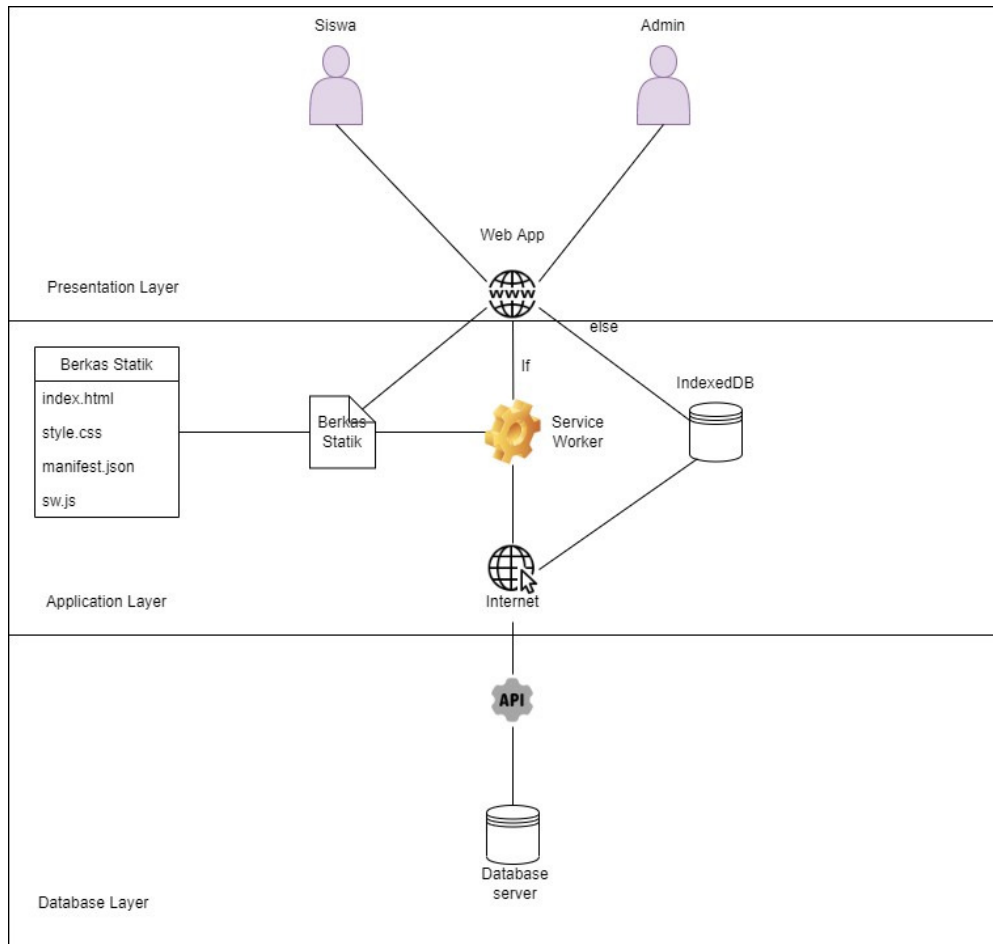
Dalam perancangan sistem akan diuraikan menjadi enam bagian, yaitu analisa sistem asesmen nasional, perancangan arsitektur *progressive web app*,

analisis kasus pengguna, *Use case diagram*, *Activity diagram*, dan *Logical Relational Structure* (LRS).

3.3.3.1 Analisa Sistem Asesmen Nasional

Pada bagian analisa sistem akan diberikan gambaran yang nantinya menghasilkan rancangan arsitektur yang akan dibangun dan kebutuhan fungsional yang akan digunakan oleh sistem. Teknik analisa kali ini dibuat dengan membagi sistem menjadi tiga layer, yaitu *presentation layer*, *application layer*, dan *database layer*. Pada *presentation layer*, pengguna dapat mengakses aplikasi sesuai dengan role yang telah ditentukan dengan menggunakan web browser.

Pada bagian *application layer* terdapat *service worker* yang mengatur event dan menyimpan file statik yang ada pada web browser *client* seperti file .html, .css, .png, json dan .js ke *cache storage*. Data yang dikirimkn dari web app akan di simpan langsung ke *database server* dengan menggunakan *Representational State Transfer Application Programming Interface* (REST API) jika konektivitas jaringan online. REST API digunakan untuk menghubungkan *application layer* dan *database layer*. Aplikasi akan menyimpan data pada *local storage* (indexedDB) jika konektivitas jaringan offline. Data yang ada pada indexedDB akan diteruskan secara *asynchronous* ke server dengan menggunakan REST API jika konektivitas jaringan kembali *online*. Ketika melakukan request untuk tambah, hapus dan ubah data pada database, harus mengirimkan request *Hypertext Transfer Protocol* (HTTP) disertai dengan alamat *Uniform Resource Location* (URL) *enpoint* REST API yang telah dibuat.



Tabel 3.1 Analisa Sistem PWA

3.3.3.2 Perancangan Arsitektur *Progressive Web App*

Arsitektur *Progressive Web App* pada sistem akan dibagi menjadi empat container utama yaitu:

- a. *Container layout* atau antarmuka aplikasi

Pada container layout ditunjukkan bagian-bagian utama pada Aplikasi asesmen nasional yang terdiri dari header, sidebar, dan footer. Layout inilah yang nantinya akan di *caching* oleh *container progressive web app*.

- b. *Container progressive web app*

Container *progressive web app* terdiri dari beberapa subbagian yaitu manifest.json yang berisikan informasi berupa nama, penulis, ikon, versi, deskripsi, dan daftar semua sumber daya yang diperlukan (antara lain), *app*

shell yang bertujuan menyediakan kembali file *static* yang terdiri dari *index.html*, *style.css*, *main.js*, dan *server.js* sehingga *container layout* antarmuka akan bersifat tetap dan hanya akan mereload bagian content dari *container layout*, *service worker* terdiri dari file *sw.js* yang berisi perintah untuk melakukan register pada browser dan menentukan metode *caching* yang akan digunakan.

c. *Representational State Transfer Application Programm Interface (REST API)*

REST API terdiri dari serangkaian perintah *javascript* yang digunakan *container browser* untuk mengakses berkas *static* yang ada pada *container progressive web app*.

d. *Container browser*

Container browser terdiri dari manifest yang merupakan *output* dari *manifest.json* sehingga membuat website bisa di install ke halaman desktop, *cache API* merupakan daftar dari berkas *static* website yang sudah di *cache* dan siap untuk digunakan kembali, *indexedDB* terdiri dari nama database dan tabel yang akan digunakan untuk menyimpan state atau data berupa *object* yang ada pada perangkat user.

e. *Container data* atau *state* pada *indexedDB*

Container ini terdiri dari state atau data berupa *object* sebagai tempat penyimpanan sementara pada arsitektur *progressive web app* sehingga website tetap bisa mengirim data meskipun dalam keadaan *offline*.

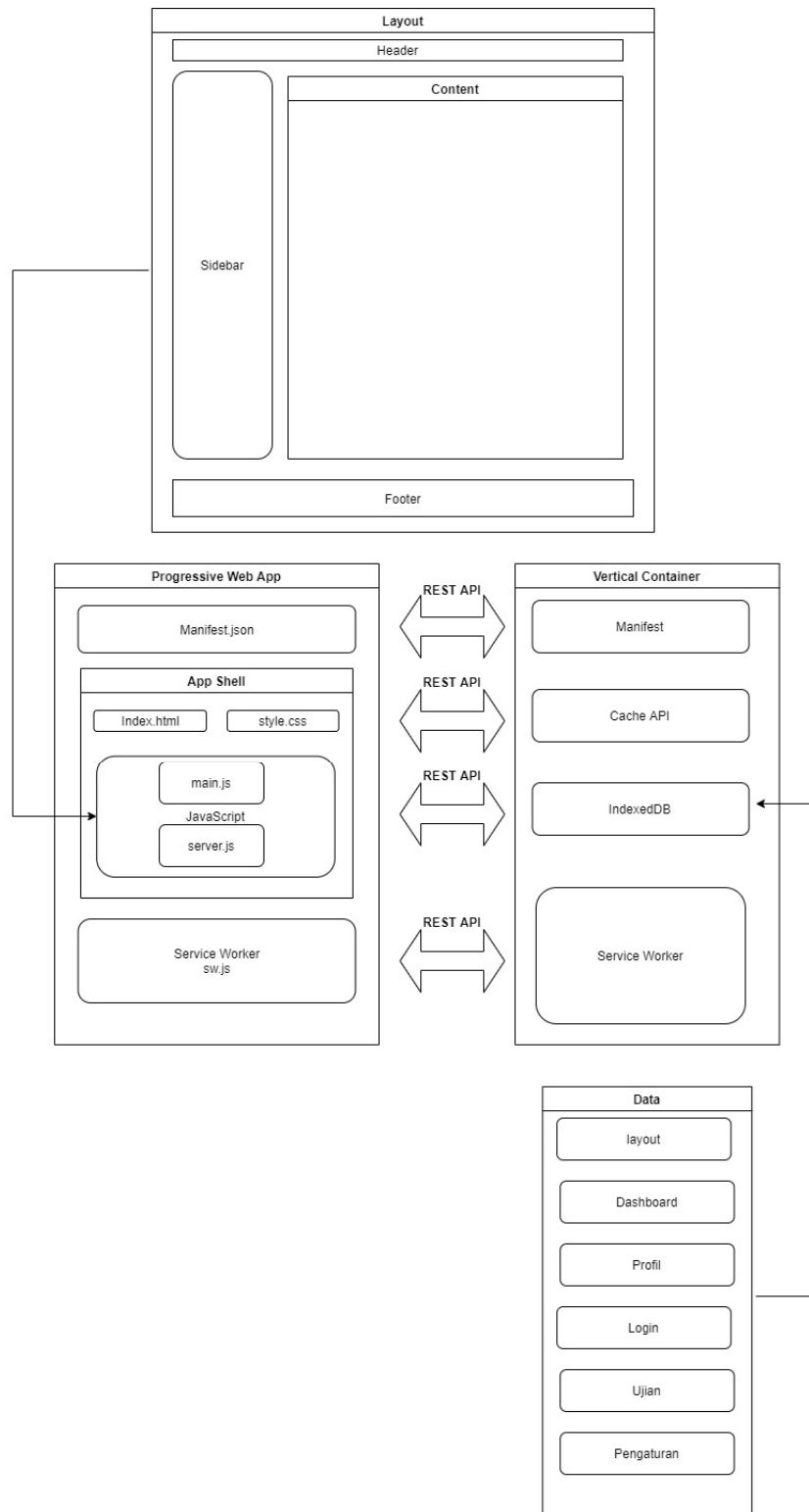
Pada penelitian ini, di implementasikan *strategi cache first, network fallback*, sesuai yang telah dijelaskan pada subbab 2.5 tentang strategi *caching service worker*.

Service worker bertugas meneruskan setiap *request* dari halaman web ke server ketika mendapatkan jaringan, lalu menggandakan respons server dan disimpan ke dalam *cache* di browser menggunakan REST API, lalu response server diteruskan kembali ke halaman web. Setiap halaman yang pernah dikunjungi oleh pengguna secara otomatis akan disediakan dalam versi *offlinenya* sehingga aplikasi

hanya mereload bagian content pada website. Cara ini bermanfaat sehingga pengguna tetap bisa mengakses web app sambil menunggu jaringan kembali *online*, hanya saja pengguna tidak dapat mengirim data ke server. Melainkan untuk sementara disimpan ke indexedDB kemudian di sinkronkan ketika terhubung ke jaringan.

Ketika berada dalam kondisi *offline* atau tidak ada respons dari server, halaman dari *cache* akan dikembalikan jika halaman tersebut sudah disimpan di dalam *cache* sebelumnya. Ini lebih baik dari pada menampilkan laporan kesalahan jaringan.





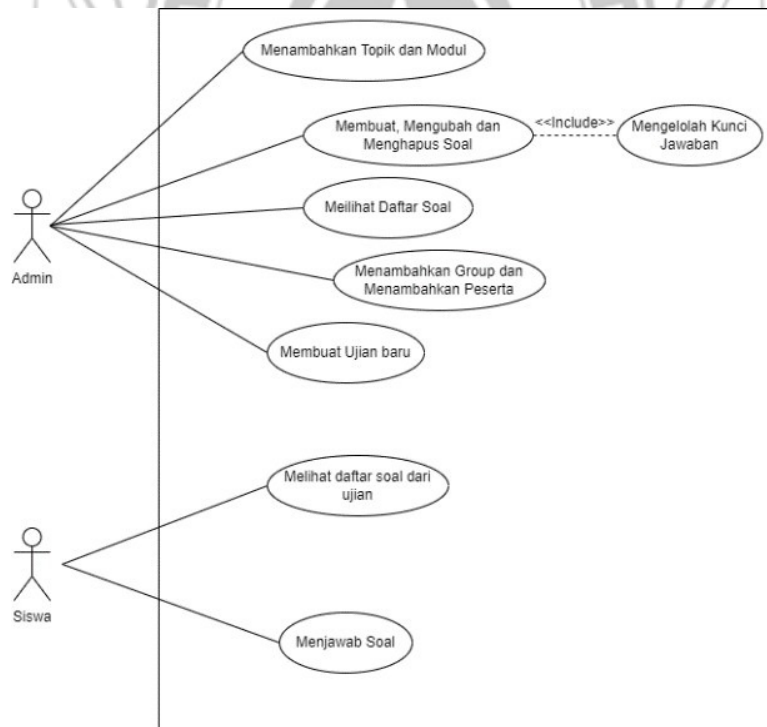
Gambar 3.2 Arsitektur sistem *progressive web app*

3.3.3.3 Analisis Kasus Pengguna

Kasus penggunaan dirancang untuk menunjukkan kebutuhan fungsional sistem. Perancangan kasus pengguna ini dibagi berdasarkan tugas-tugas yang telah dikelompokkan dari pengguna yang ada. Pengguna dibagi menjadi 2 role, yaitu administrator dan siswa. Administrator bertugas untuk melakukan manajemen-konten, administrator dapat melakukan semua manajemen yang ada pada sistem dan input soal, membuat soal-soal sesuai dengan kebutuhan dan dapat melihat hasil ujian. Siswa hanya diberikan hak akses hanya menjawab soal asesmen.

3.3.3.4 Use Case Diagram

Use case diagram atau diagram kasus pengguna ini melibatkan admin dan siswa sebagai pusat aktifitas pembelajaran. Diagram kasus pengguna bisa dilihat pada gambar 3.3.



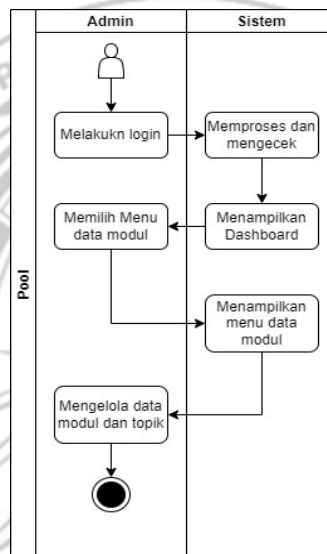
Gambar 3.3 *Use case diagram* role admin dan siswa

Pengguna yang membutuhkan kemampuan *offline* adalah pengguna dengan role siswa yaitu melihat daftar soal dari ujian dan menjawab soal. Pengguna dengan role admin tidak didukung kemampuan *offline*. Role admin dapat mengelola data modul (menambahkan topik dan modul), mengelola data soal (membuat, melihat, mengubah dan menghapus soal), dan mengelola data ujian.

3.3.3.5 Activity Diagram

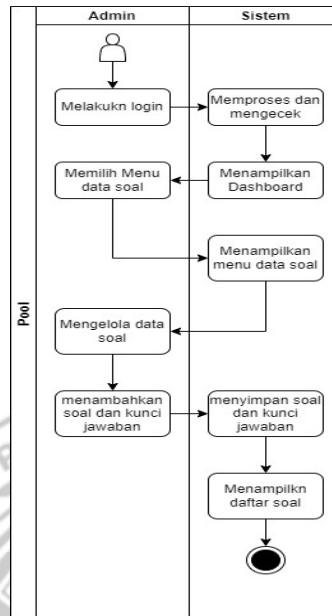
1. Activity Diagram Role Admin

a. Admin mengelola data topik dan modul



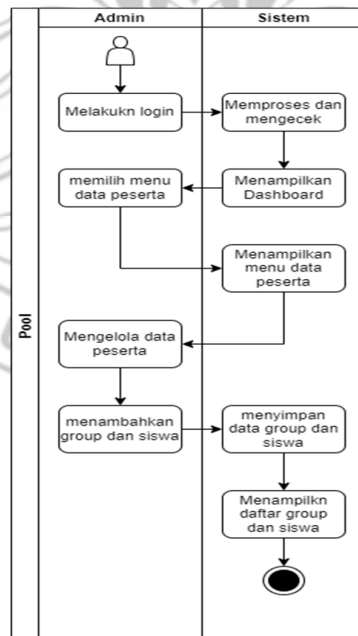
Gambar 3.4 Admin mengelolah data topik dan modul

b. Admin mengelola data soal



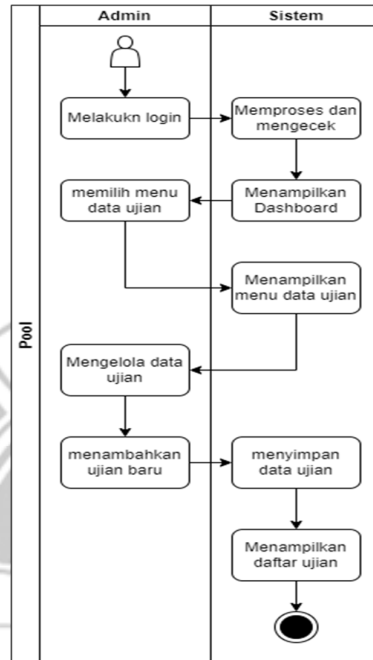
Gambar 3.5 Admin mengelola data soal

c. Admin mengelola data siswa



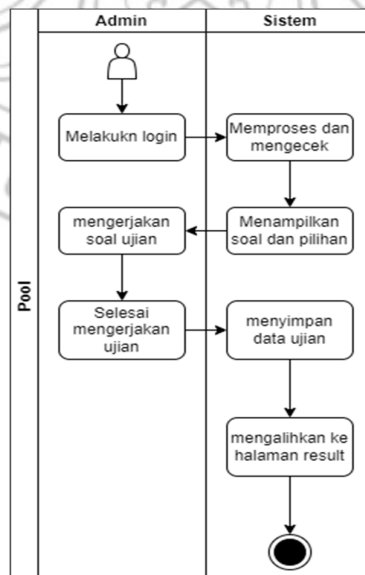
Gambar 3.6 Admin mengelolah data siswa

d. Admin mengelola data ujian



Gambar 3.7 Admin mengelolah data ujian

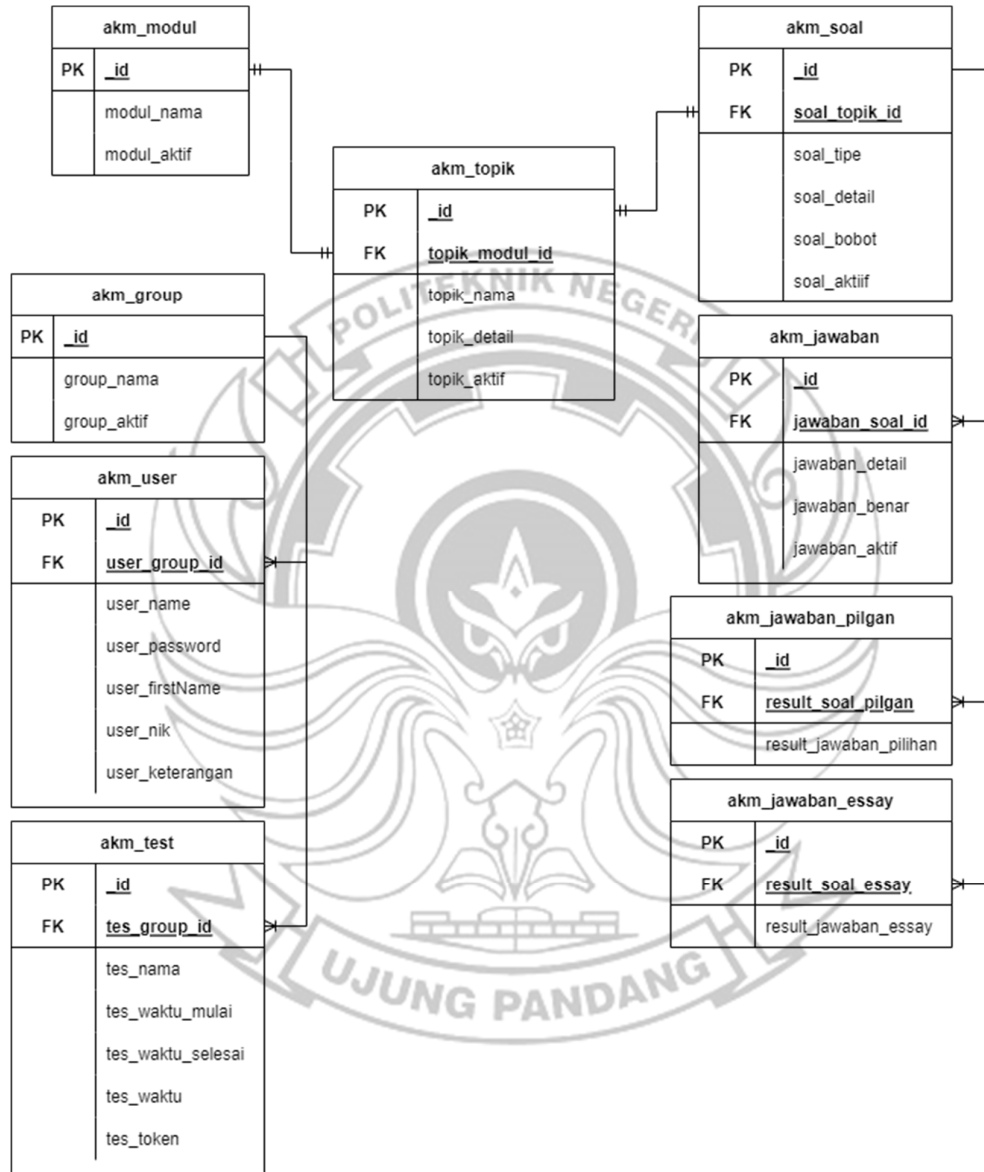
2. Activity Diagram Role Siswa



Gambar 3.8 Siswa mengerjakan ujian

3.3.3.6 Logical Relasional Structure (LRS)

Perancangan basis data dilakukan menggunakan LRS dengan tujuan dapat mengetahui kebutuhan data yang akan diolah pada sistem.



Gambar 3.9 Logical Relasional Structure

3.3.4 Implementasi

Sistem yang telah dirancang sebelumnya akan di implementasikan ke bahasa pemrograman pada tahap ini. sehingga, dihasilkan aplikasi asesmen nasional berbasis PWA. Pada tahap ini juga akan dilakukan implementasi penerapan PWA yang terdiri dari penerapan sistem, *Web App Manifest*, *service worker* dan *indexedDB*.

3.3.5 Pengujian

Pengujian sistem yang telah dibangun akan dilakukan pada tahap ini. Proses pengujian kualitas PWA ini dilakukan secara manual dan menggunakan bantuan *lighthouse*.

a. Pengujian Manual

Pada pengujian manual, akan di uji kemampuan arsitektur utama pada sistem PWA, yaitu

1) Pengujian kemampuan *web app manifest*

Pada pengujian *web app manifest*, akan dilakukan pengujian install atau menambahkan aplikasi website ke *homescreen* secara manual dan memastikan aplikasi bisa memunculkan *pop-up add to homecreen* dan berjalan layaknya aplikasi *native*.

2) Pengujian kemampuan *service worker*

Pada pengujian *service worker* akan di lakukan *registrasi service worker* pada aplikasi website dan memastikan *service worker* dapat melakukan *caching* terhadap file statik website sehingga aplikasi website tetap bisa menampilkan halaman aplikasi website meskipun dalam keadaan jaringan *offline*.

3) Pengujian kemampuan *indexedDB*

Pada pengujian *indexedDB* akan dilakukan pengujian dengan melakukan pengiriman data melalui antarmuka client (siswa) dalam keadaan jaringan *offline* dan memastikan data akan tersimpan sementara di *local storage* (*indexedDB*) dan akan disinkronkan dengan server ketika kondisi jaringan kembali *online*.

b. Pengujian Menggunakan *Lighthouse*

Pada pengujian menggunakan *lighthouse*, dilakukan serangkaian pengujian terhadap laman aplikasi ANBK. Pengujian menggunakan *lighthouse* akan dilakukan dengan menambahkan jumlah user (siswa) secara bertahap, kemudian dilakukan pengujian menggunakan *lighthouse* setiap kali melakukan penambahan jumlah user (siswa).



BAB 4 HASIL DAN PEMBAHASAN

4.1 Implementasi Tampilan Antar Muka

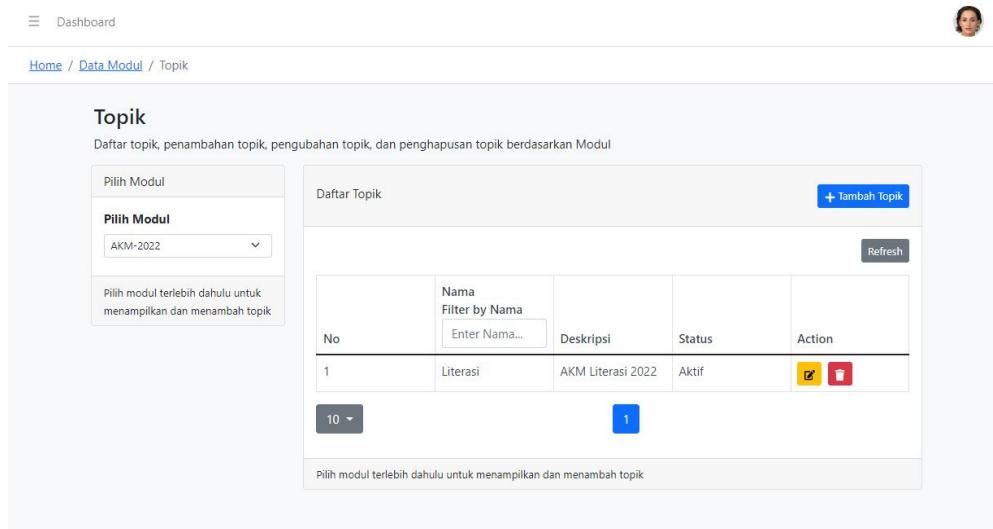
Pada tahap ini akan dilakukan implementasi terhadap sistem yang dirancang ke dalam bahasa pemrograman. Sehingga dihasilkan aplikasi asesmen nasional berbasis komputer dengan menggunakan *progressive web app*. Web app dibangun dengan menggunakan MERN Stack yaitu Nodejs untuk membangun REST API pada sistem, *framework* ReactJs untuk membangun *client-side* pada sistem, database Mongoddb untuk menyimpan data web app.

Pada implementasi tampilan antar muka akan dilakukan dengan dua skenario sebagai perbandingan yaitu ketika kondisi jaringan *online* dan *offline*.

4.1.1 Implementasi Proses Mengelola Data Topik

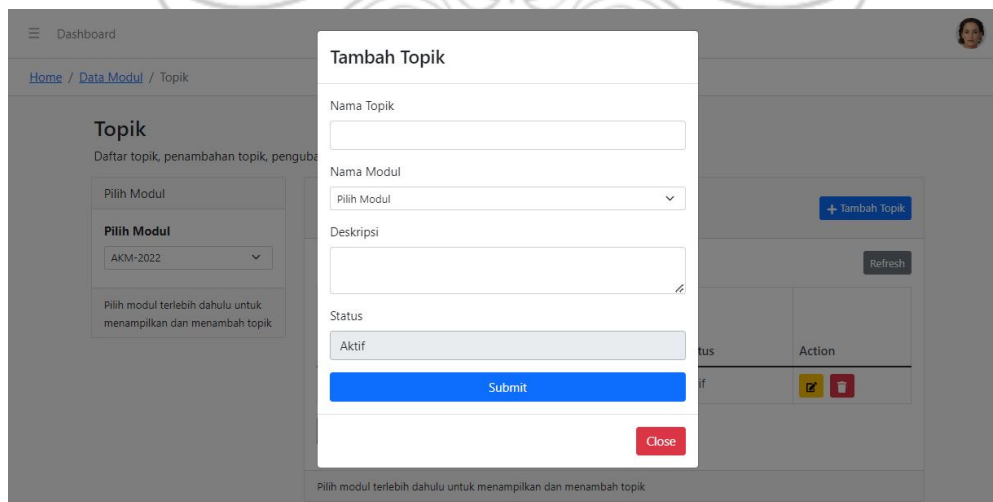
Di dalam proses mengelola data topik, admin dapat melihat daftar topik, daftar modul, membuat, mengubah dan menghapus topik.

Admin memulai proses ini dari halaman data modul seperti ditunjukkan pada gambar 4.1. Pada halaman data topik, ditampilkan daftar topik yang ada, dengan nama, deskripsi dan status topik. Di atas daftar topik terdapat tombol “Tambah Topik” untuk menambahkan topik baru berdasarkan modul, yang akan ditampilkan pada pop-up modal.



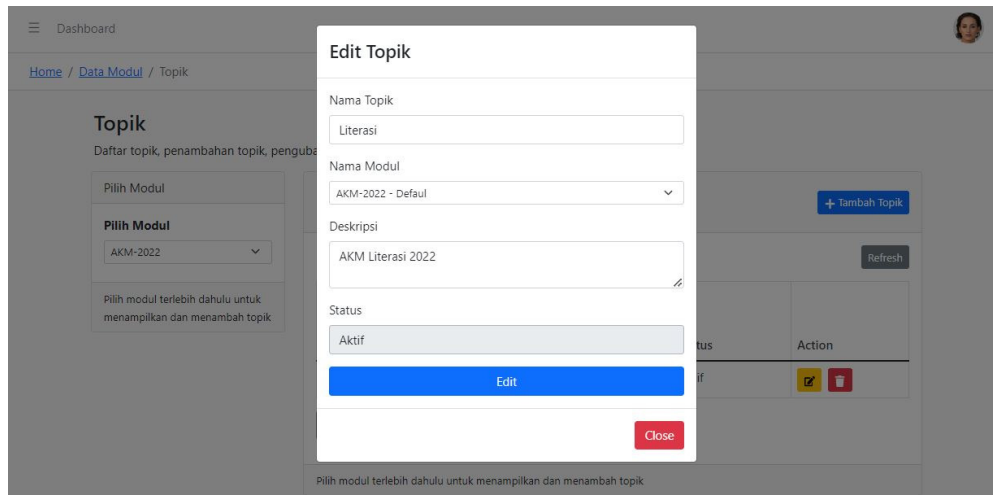
Gambar 4.1 Antar muka data topik

Di dalam proses membuat topik baru, admin dapat membuat topik baru berdasarkan modul seperti yang ditunjukkan gambar 4.2. satu buah modul bisa memiliki banyak topik.



Gambar 4.2 Antar muka menambah topik baru

Admin dapat mengubah topik dengan menekan tombol “Edit” (berwarna kuning) pada kolom “action” sehingga akan muncul pop-up modal seperti yang ditunjukkan pada gambar 4.3.

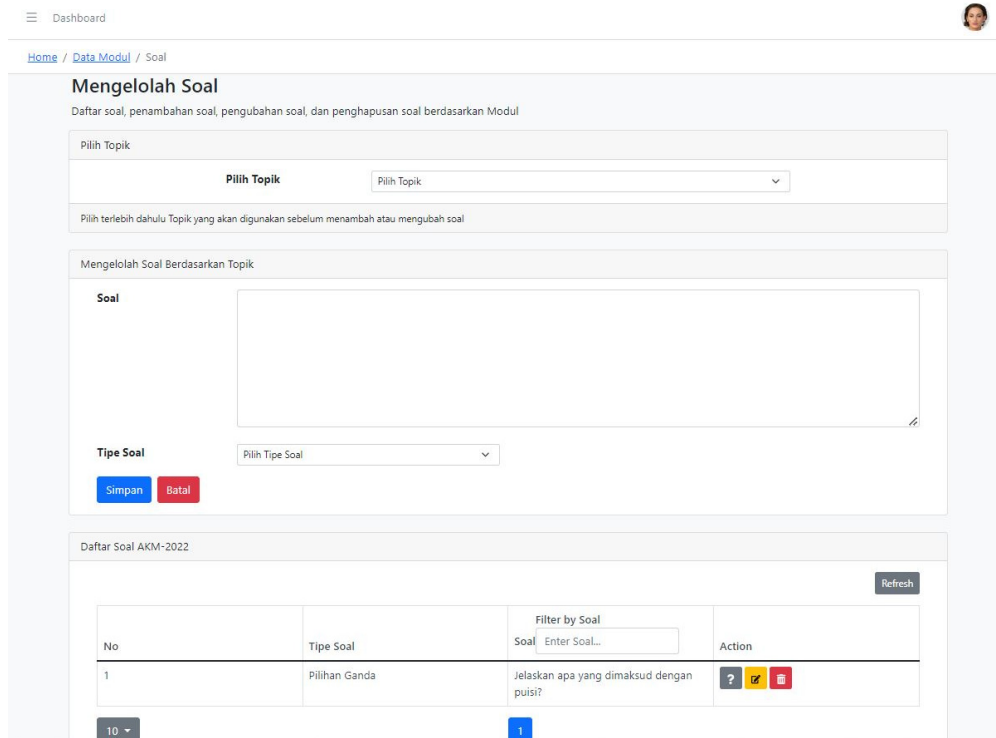


Gambar 4.3 Antarmuka mengubah topik

Admin dapat menghapus topik dengan menekan tombol “Hapus” (berwarna merah) pada kolom “action”, sehingga secara asinkron topik akan terhapus.

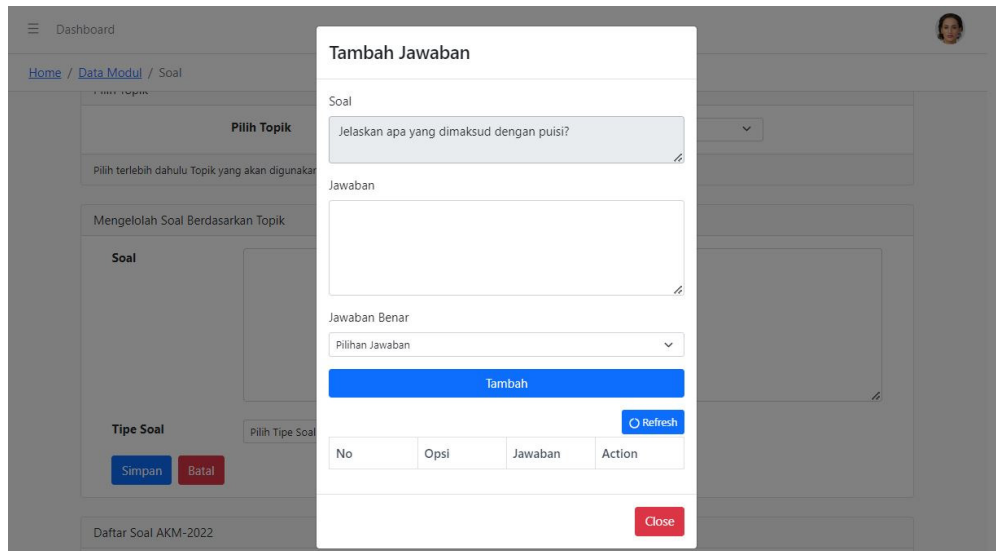
4.1.2 Implementasi Proses Mengelola Soal dan Daftar Soal

Admin memulai proses ini dari halaman data soal seperti yang ditunjukkan oleh gambar 4.4. admin bisa membuat soal baru berdasarkan topik yang ada. Ada dua tipe soal yang bisa dipilih, yaitu pilihan ganda dan essay. Ketika admin memilih tipe soal pilihan ganda, maka akan muncul kolom baru untuk mengisi bobot soal. Setelah admin membuat soal, maka soal yang dibuat akan muncul di tabel daftar soal. Tabel daftar soal dibuat berdekatan dengan card tambah soal agar mempermudah admin untuk melihat daftar soal dan menambahkan jawaban pada soal yang bertipe pilihan ganda. Pada daftar soal, soal yang bertipe pilihan ganda akan mempunyai menu tambah jawaban pada kolom action.



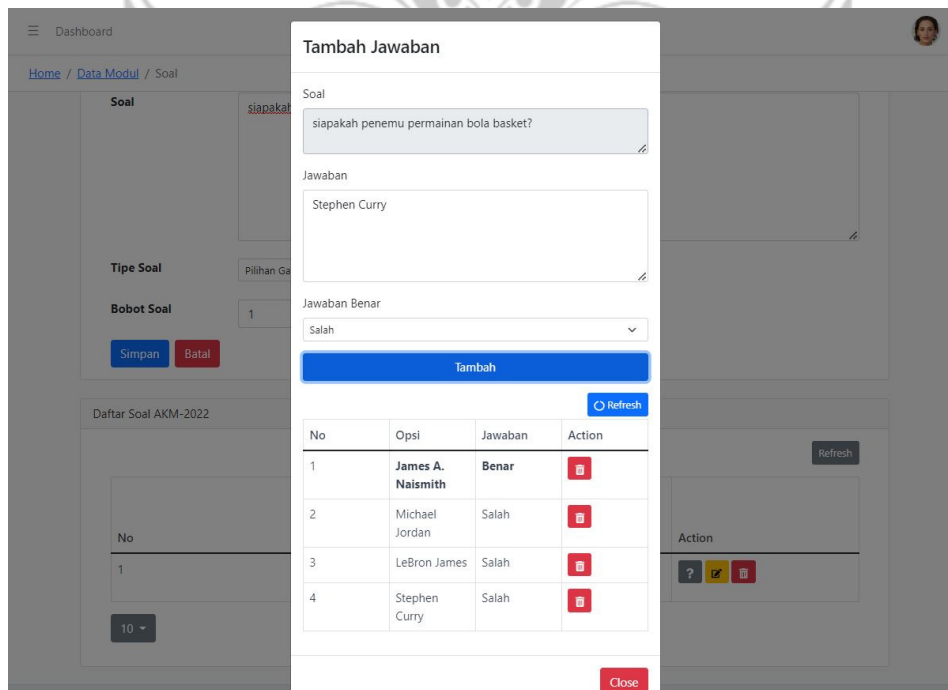
Gambar 4.4 Antarmuka mengelolah soal dan daftar soal

Admin bisa menambahkan jawaban pada soal dengan tipe soal pilihan ganda dengan menekan tombol “tambah jawaban” (berwarna abu-abu). Ketika menekan tombol tambah soal, maka akan muncul pop-up modal tambah jawaban yang bisa dilihat pada gambar 4.5. pada modal tambah jawaban, admin bisa mengisi jawaban dan kunci jawaban, sedangkan soal akan otomatis terbaca berdasarkan soal yang akan ditambahkan jawabannya. Pada modal tambah jawaban juga, terdapat tabel daftar jawaban untuk mempermudah admin melihat daftar dan kunci jawaban.



Gambar 4.5 Antarmuka menambahkan jawaban

Pada tabel daftar jawaban, opsi soal benar akan ditulis secara bold untuk memudahkan admin melihat kunci jawaban yang bisa dilihat pada gambar 4.6



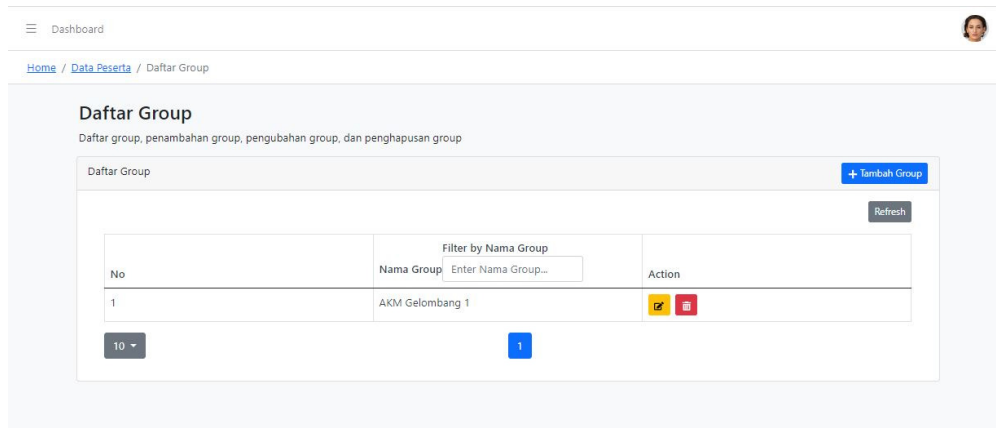
Gambar 4.6 Antarmuka menambahkan jawaban

Untuk melihat daftar soal beserta dengan kunci jawaban, admin bisa ke halamn daftar soal seperti yang ditunjukkan pada gambar 4.7. untuk melihat daftar soal, admin harus memilih topik terlebih dahulu, karena soal dibuat berdasarkan topik. pada halaman daftar soal, admin bisa megedit dan menghapus soal dan jawaban.

Gambar 4.7 Antarmuka daftar soal

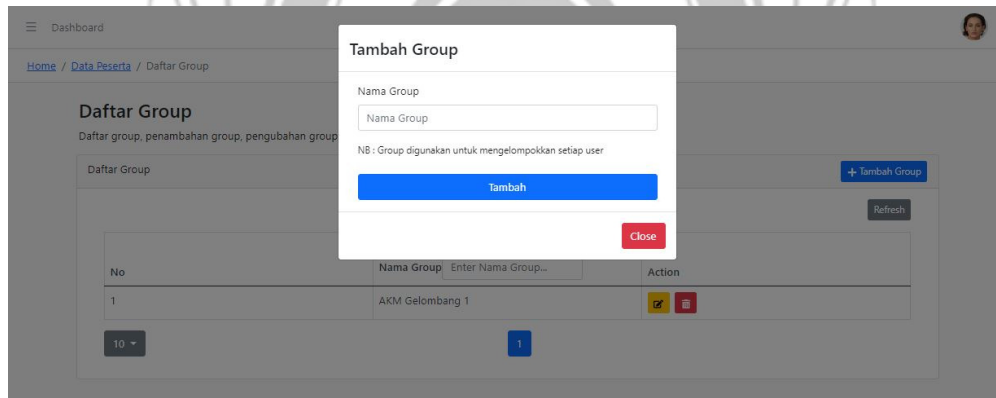
4.1.3 Implementasi Proses Mengelola Data Peserta

Data peserta terdiri dari daftar group dan daftar peserta. Untuk menambahkan peserta, admin terlebih dahulu harus menambahkan group. Untuk menambahkan data group, admin terlebih dahulu masuk ke halaman daftar group seperti yang ditunjukkan pada gambar 4.8. Pada halaman daftar group, ditampilkan daftar group yang ada. Di atas daftar topik terdapat tombol “Tambah Group” untuk menambahkan group baru, yang akan ditampilkan pada pop-up modal.



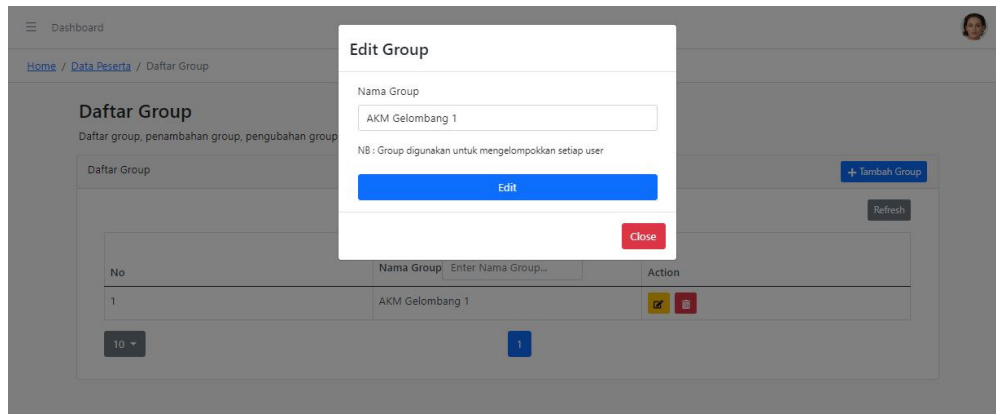
Gambar 4.8 Antarmuka daftar group

Di dalam proses membuat group baru, admin dapat membuat group baru dengan menekan tombol “Tambah Group” sehingga akan muncul modal seperti gambar 4.9.



Gambar 4.9 Antarmuka tambah group

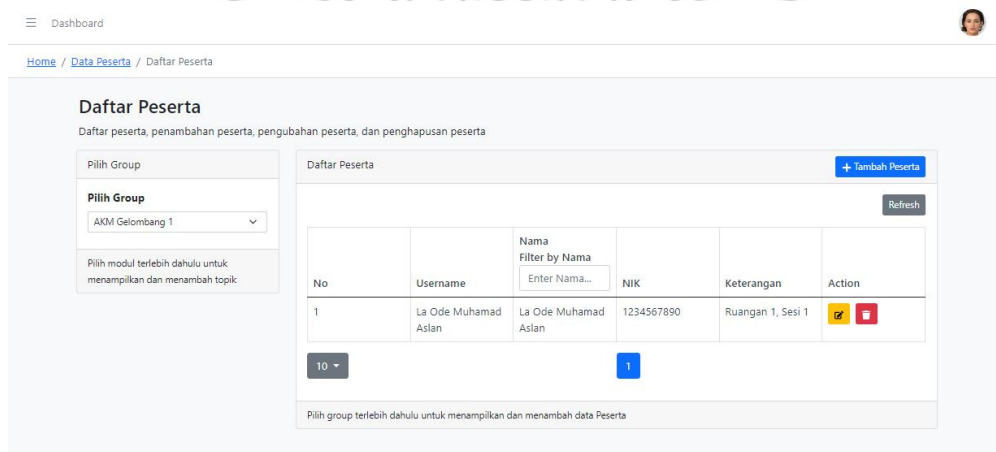
Admin dapat mengubah group dengan menekan tombol “Edit” (berwarna kuning) pada kolom action sehingga akan muncul pop-up modal seperti yang ditunjukkan pada gambar 4.10.



Gambar 4.10 Antarmuka edit group

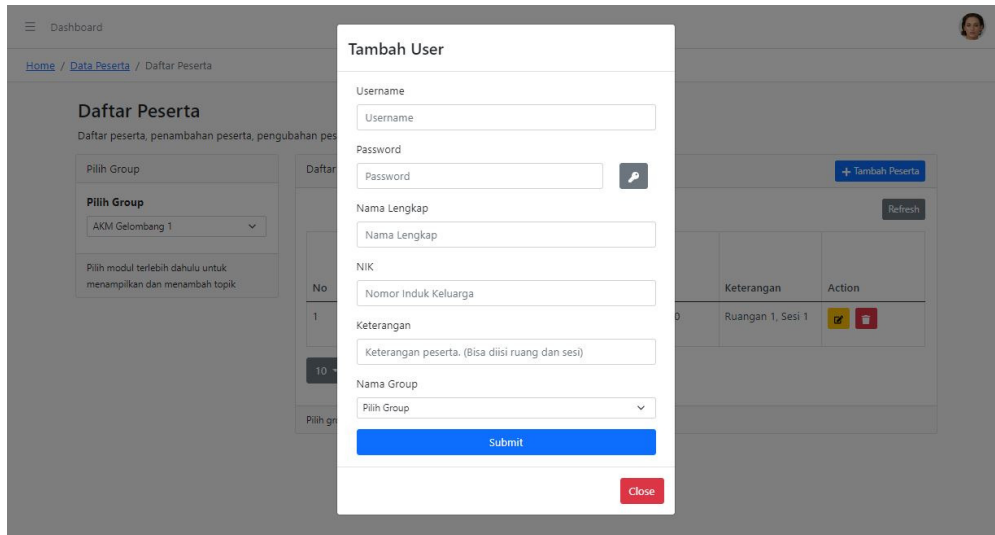
Admin dapat menghapus group dengan menekan tombol “Hapus” (berwarna merah) pada kolom action, sehingga secara asinkron group akan terhapus.

Untuk menambahkan peserta, admin terlebih dahulu masuk ke halaman daftar peserta seperti yang ditunjukkan pada gambar 4.11. Pada halaman daftar peserta, ditampilkan daftar peserta yang ada. Di atas tabel daftar peserta terdapat tombol “Tambah Peserta” untuk menambahkan peserta baru, yang akan ditampilkan pada pop-up modal.



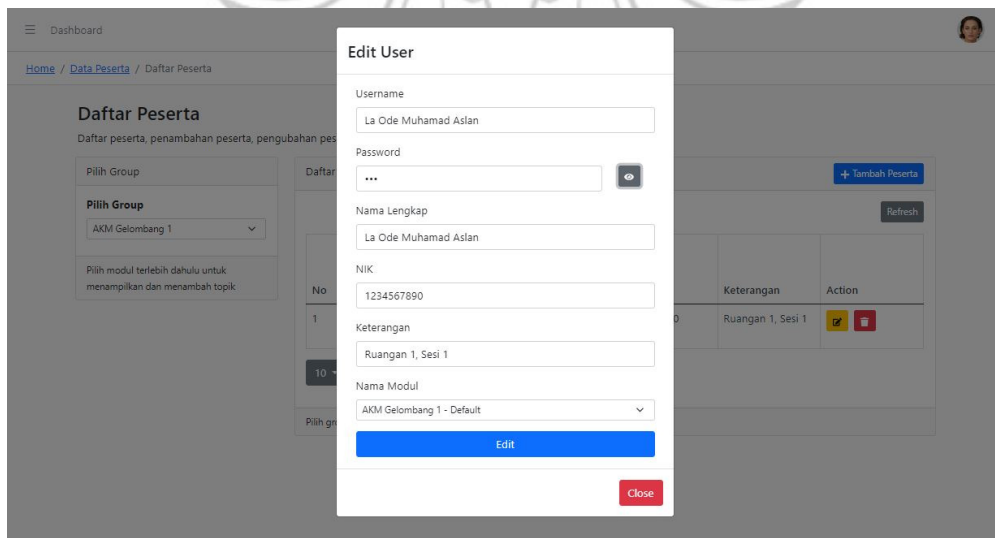
Gambar 4.11 Antarmuka daftar peserta

Di dalam proses membuat peserta baru, admin dapat membuat peserta baru dengan menekan tombol “Tambah Peserta” sehingga akan muncul modal seperti gambar 4.12.



Gambar 4.12 Antarmuka tambah peserta

Admin dapat mengubah data pengguna dengan menekan tombol “Edit” (berwarna kuning) pada kolom action sehingga akan muncul pop-up modal seperti yang ditunjukkan pada gambar 4.13.



Gambar 4.13 Antarmuka edit peserta

Admin dapat menghapus peserta dengan menekan tombol “Hapus” (berwarna merah) pada kolom action, sehingga secara asinkron data peserta akan terhapus.

4.1.4 Implementasi Proses Mengelola Asesmen Nasional

Admin memulai proses ini dari halaman tambah tes seperti ditunjukkan pada gambar 4.14. Pada halaman tambah tes, ditampilkan daftar tes yang ada. Di atas daftar topik terdapat card sebagai form menambahkan tes baru.

Tambah Tes
Melihat daftar tes, mengubah dan menghapus tes

Tambah Tes

Group: Waktu Selesai:

Nama: Waktu:

Waktu Mulai: Token:

Daftar Tes

No	Group	Nama Filter by Nama <input type="text" value="Enter Nama..."/>	Waktu	Token	Action
1	AKM Gelombang 1	TES Uji Coba 1	2022-06-27T00:02:00.000Z (10 menit)	791ZR	<input type="button" value="Hapus"/>

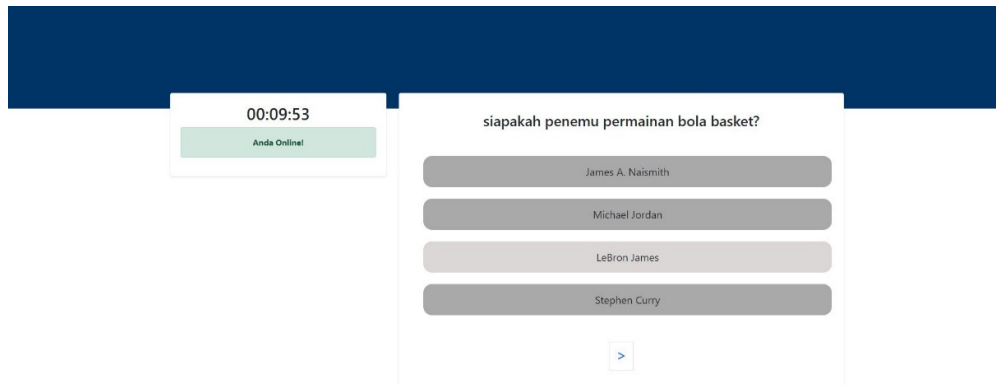
Gambar 4.14 Antarmuka tambah tes

4.1.5 Implementasi Mengikuti Asesmen Nasional

Pada implementasi mengikuti asesmen nasional akan dibuat dua skenario, yaitu ketika siswa mengerjakan soal dengan kondisi jaringan online dan ketika siswa mengerjakan soal dengan kondisi jaringan offline.

4.1.5.1 Impelementasi Ketika Jaringan *Online*

Diasumsikan siswa sudah melakukan login dan sudah mengerjakan soal bisa dilihat pada gambar 4.15. pada skenario pertama akan diperlihatkan aplikasi berjalan dengan kondisi jaringan *online*.

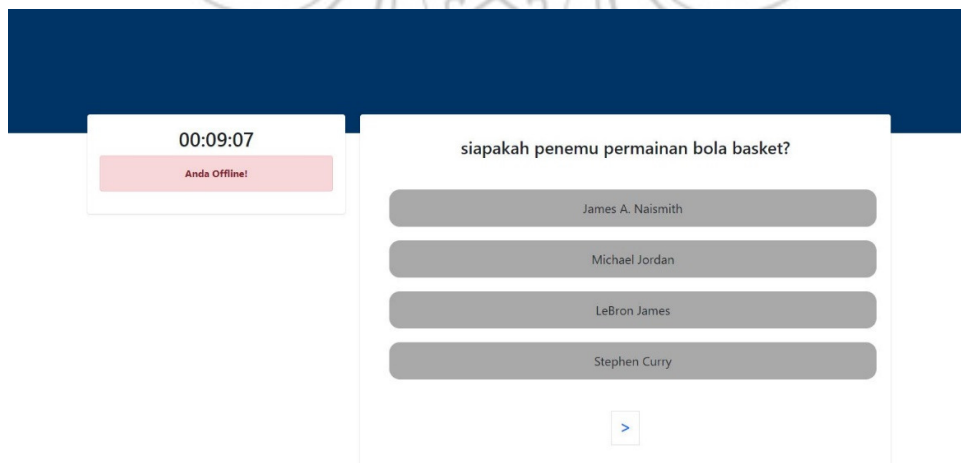


Gambar 4.15 Tampilan aplikasi ketika jaringan *online*

Ketika jaringan *online*, indikator online akan muncul, dan siswa masi bisa melihat soal dan pilihan di website. pada kondisi jaringan *online*, data dari *web app* akan dikirim langsung ke server dengan menggunakan REST API.

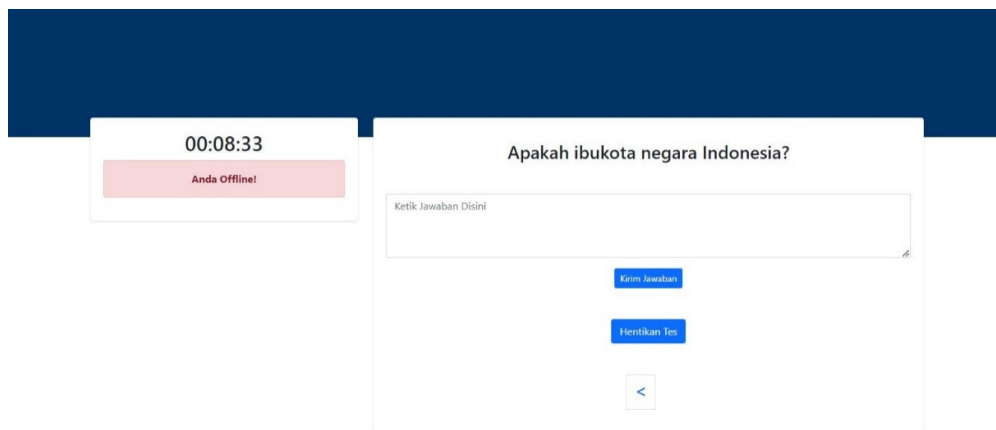
4.1.5.2 Implementasi Ketika Jaringan *Offline*

Diasumsikan siswa sudah melakukan login dan sudah mengerjakan soal bisa dilihat pada gambar 4.16. pada skenario kedua akan diperlihatkan jika jaringan tiba-tibat terputus atau *offline* ketika siswa sedang mengerjakan ujian.



Gambar 4.16 *Page* dimana jaringan terputus

Ketika jaringan tiba-tiba terputus, indikator *offline* akan muncul, dan siswa masi bisa melihat soal di *page* terakhir ketika jaringan tiba-tiba terputus. Ketika kita melihat soal selanjutnya, maka siswa masi bisa melihat dan mengerjakan ujian bisa dilihat pada gambar 4.17. pada kondisi jaringan *offline*, data dari *web app* akan disimpan di indexedDB setelah itu diteruskan ke server dengan menggunakan REST API ketika kondisi jaringan kembali *online*.



Gambar 4.17 *Page* soal selanjutnya

4.2 Pengujian Sistem *Progressive Web App*

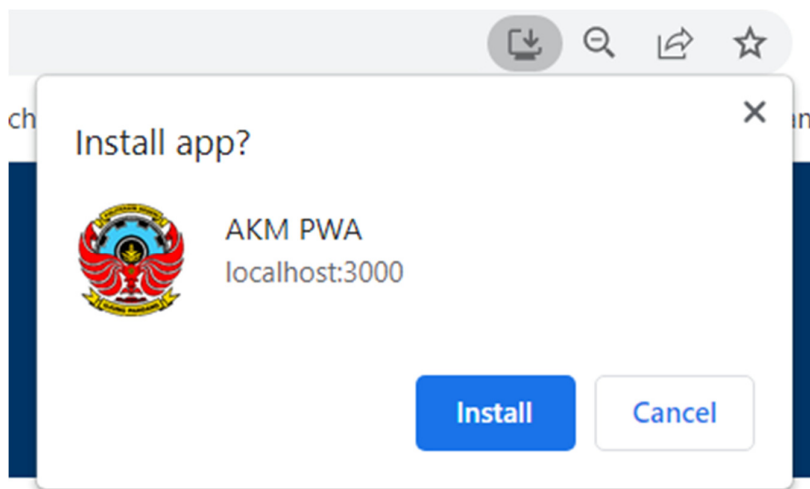
Pada bagian ini akan dijelaskan tentang kemampuan aplikasi yaitu bisa di install seperti *native app* dengan menggunakan *web app manifest*, berjalan secara *offline* dengan *service worker* dan menyimpan data di indexedDB ketika jaringan *offline* dan meneruskan data ke server secara *asinkron* ketika jaringan kembali *online*.

4.2.1 Pengujian *Web App Manifest*

web app Manifest merupakan salah satu teknologi utama dalam penerapan *progressive web app* yaitu situs web yang dapat dipasang ke layar beranda tanpa harus di download di *google play store* atau *app store*. Dengan *web app manifest* membuat website yang dibuat tidak sama seperti aplikasi biasa dengan tautan atau

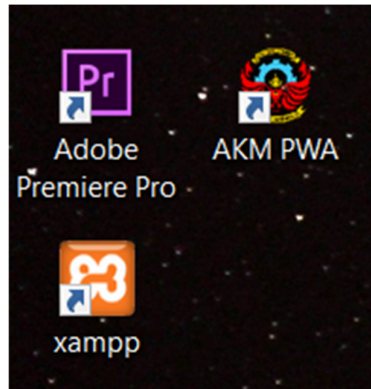
bookmark layar beranda sederhana. Dengan menggunakan PWA kita bisa menginstall terlebih dahulu, serta menggunakan API Web biasa.

Hasil pengujian *web app manifest* pada aplikasi asesmen nasional berbasis komputer dengan menggunakan PWA dapat dilihat pada gambar 4.18 di bawah ini, dimana pada gambar tersebut ditunjukkan bahwa sistem yang dihasilkan bersifat *installable* dan dapat menunjukkan *pop-up add to homescreen* ketika kita menekan tombol ikon download pada aplikasi google chrome.

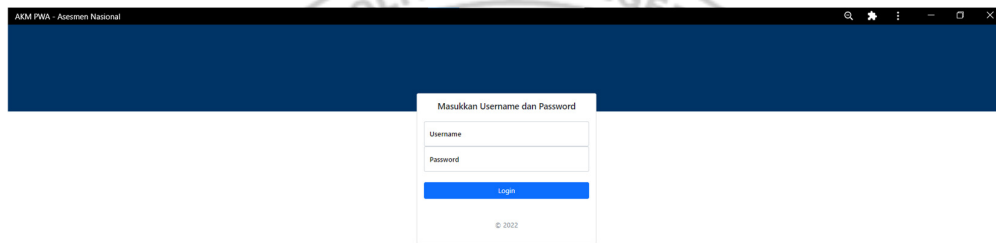


Gambar 4.18 Pengujian *web app manifest*

Implementasi *web app manifest* pada sistem selain membuat aplikasi bersifat *installable*, juga memberikan kemampuan lain pada sistem yaitu sistem dapat di akses melalui ikon pada *homescreen* perangkat yang digunakan seperti ditunjukkan pada gambar 4.19 dan 4.20.



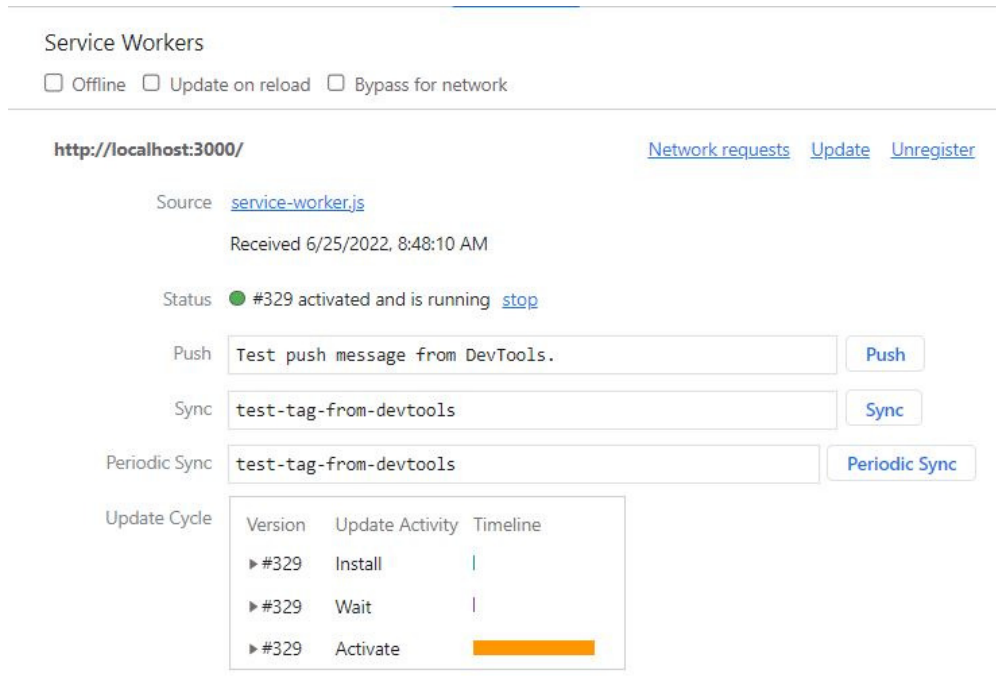
Gambar 4.19 Pengujian menginstall web app ke *homescreen*



Gambar 4.20 Pengujian membuka web app melalui *homescreen*

4.2.2 Pengujian *Service Worker*

Service worker merupakan teknologi utama dalam membangun aplikasi berbasis PWA. Dengan *service worker* membuat proses loading pada aplikasi yang dibuat menjadi lebih cepat (terlepas dari jaringan), bisa diakses secara *offline*, dan sebagainya. Implementasi *service worker* terhadap sistem dapat dilihat pada gambar 4.21 di bawah ini.



Gambar 4.21 Pengujian *Service Worker*

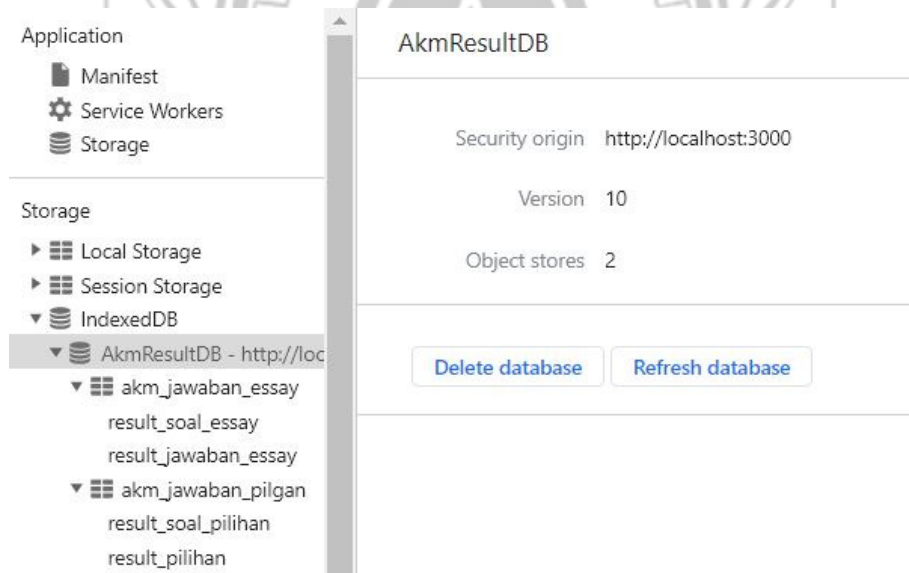
Pada gambar diatas, *service worker* telah berhasil didaftarkan secara otomatis di browser pengguna. strategi *service worker* yang di implementasikan yaitu *cache first, network fallback*. Sebelum menampilkan tampilan sesuai dengan *request* yang dikirimkan oleh siswa, *request* terlebih dahulu akan melewati *service worker*. Seperti kita mengambil kontrol penuh atas *request*, semua *response get* yang datang dari jaringan akan melalui *service worker* terlebih dahulu sebelum ke browser. *Service worker* juga akan menyimpan file statik yang dibutuhkan oleh sistem. Ketika melakukan *request* ke server, maka server akan mengembalikan *response*, setelah itu *request* dan *response* tadi akan di *caching* terlebih dahulu oleh *service worker*. Ketika kita mengakses halaman kembali, *service worker* akan mengecek terlebih dahulu apakah *request* dan *response* telah di *caching* atau belum. Jika sudah di *caching*, maka *service worker* akan mengembalikan *response* yang sudah di *caching* sehingga aplikasi tidak perlu lagi melakukan *request* ke server untuk menampilkan halaman. Data hasil *cache* akan disimpan di *cache storage*. Desain seperti ini meningkatkan kecepatan dalam memuat *resource* ke halaman web yang ditunjukkan pada gambar 4.22.

Name	Status	Type	Initiator	Size	Time
localhost	200	document	Other	(ServiceW...	17 ms
main.54974f9f.js	200	script	(index)	(ServiceW...	4 ms
main.b8a81d84.css	200	stylesheet	(index)	(ServiceW...	3 ms
manifest.json	304	manifest	(index)	113 B	4 ms
akmuser	200	xhr	xhr.js:220	211 B	14 ms
react_devtools_backend.js	200	script	injectGlobalHook.j...	525 kB	12 ms
pnup192.png	200	png	(index)	(ServiceW...	6 ms
pnup192.png	304	fetch	StrategyHandler.js:...	113 B	3 ms
service-worker.js	304	fetch	serviceWorkerRegi...	113 B	15 ms
pnup64.png	200	png	Other	(ServiceW...	4 ms
pnup64.png	304	fetch	StrategyHandler.js:...	113 B	12 ms

Gambar 4.22 Request table pada chrome dev tools

4.2.3 Pengujian IndexedDB

IndexedDB merupakan salah satu teknologi utama yang digunakan dalam membangun web app berbasis PWA. Implementasi indexedDB terhadap sistem dapat dilihat pada gambar 4.23 dibawah ini



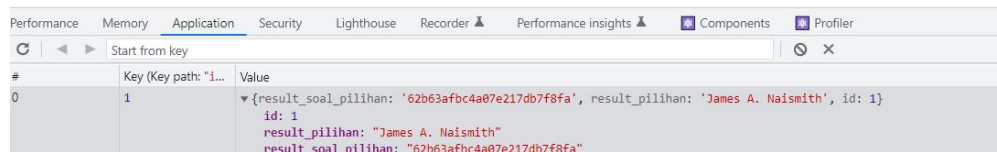
Pengujian indexedDB

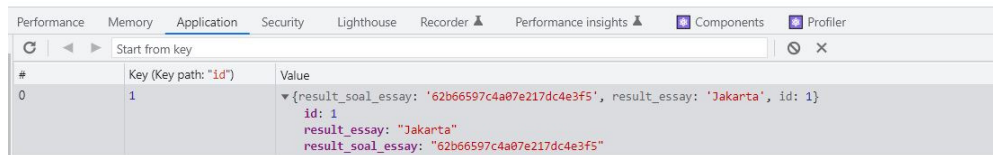
IndexedDB adalah penyimpanan data NoSQL untuk data terstruktur dan *blob* (data biner). IndexedDB bekerja secara *asinkron* sebagai tempat penyimpanan sementara untuk PWA. Dengan adanya indexedDB, membuat aplikasi bersifat

PWA bisa digunakan secara stabil meski dalam kondisi jaringan yang kurang memadai atau tidak ada.

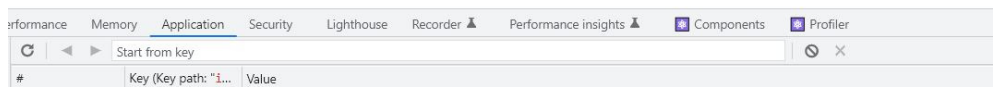
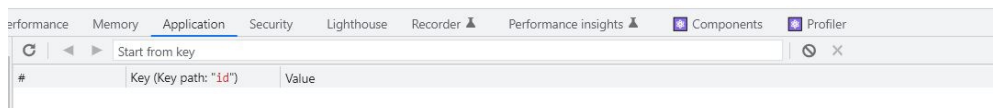
Pada gambar 4.23, dibuat indexedDB dengan nama database AkmResultDB yang terdiri dari dua *object store* atau tabel yaitu tabel `akm_jawaban_essay` dan `akm_jawaban_pilgan`. tabel `akm_jawaban_essay` mempunyai dua kolom yaitu `result_soal_essay` dan `result_jawaban_essay`. `Akm_jawaban_pilgan` juga memiliki dua kolom yaitu `result_soal_pilgan` dan `result_jawaban_pilgan`.

Ketika kita mengirimkan request post pada server, dalam kondisi jaringan yang stabil maka request akan diteruskan oleh jaringan menuju server dan server akan mengirimkan *response* berdasarkan *request* yang dilakukan oleh user. Sebaliknya ketika pada saat request post akan dikirimkan dan kondisi jaringan terputus, maka data dari request post tadi akan tersimpan pada penyimpanan indexedDB pada *local storage* browser atau terjadi *request pending*. Data akan tersimpan secara permanen sehingga aplikasi PWA dapat berjalan tanpa harus terhubung dengan server yang bisa dilihat pada gambar 4.24. Dengan cara ini aplikasi dapat berjalan secara cepat karena tidak perlu meminta informasi ke jaringan internet. Jika web kembali mendapatkan jaringan, maka data yang ada pada indexedDB akan disinkronkan dengan server secara otomatis, sehingga data yang ada pada indexedDB akan diteruskan sebagai *request* post dan masuk ke database server secara *asinkron* yang bisa dilihat pada gambar 4.25.





Gambar 4.23 Pengujian menyimpan hasil ujian secara offline



Gambar 4.24 Pengujian sinkronkan data dari indexedDB ke database server

4.3 Pengujian Sistem Menggunakan *Lighthouse*

Pada subbab ini akan dibahas hasil pengujian menggunakan *lighthouse*, mengujian dilakukan sebanyak jumlah page pada antarmuka *client*. Total jumlah *user* untuk pengujian berjumlah 15 *user*, akan dilakukan pengujian setiap kali terjadi penambahan 5 *user*.

Tabel 4.1 Pengujian 5 *user* awal

Ket	Performance	Accessibility	Best Practice	SEO	PWA
Page Home	99	100	100	100	100
Page Datates	89	81	100	100	100
Page Ujian	99	95	100	100	100
Page Result	99	95	100	100	100

Tabel 4.2 Pengujian dengan menambahkan 5 *user* pertama

Ket	Performance	Accessibility	Best Practice	SEO	PWA
Page Home	99	100	100	90	100
Page Datates	74	91	100	90	100
Page Ujian	78	97	100	90	100
Page Result	88	97	100	90	100

Tabel 4.3 Pengujian dengan menambahkan 5 *user* kedua

Ket	Performance	Accessibility	Best Practice	SEO	PWA
Page Home	75	100	100	90	100
Page Datates	64	91	100	90	100
Page Ujian	80	97	100	90	100
Page Result	85	97	100	90	100

Berdasarkan hasil pengujian yang dilakukan menggunakan *lighthouse*, Pada pengujian 5 *user* awal aplikasi asesmen nasional sudah memenuhi kriteria sebagai sebuah aplikasi berbasis *progressive web app* dengan skor 100 dari 100. Aplikasi juga sudah mendapatkan nilai yang sangat baik dari keempat kriteria lainnya yaitu kriteria *performance* dengan rata-rata 96,5 dari 100, kriteria *accessibility* dengan rata-rata 92,75 dari 100, kriteria *best practice*, *seo* memiliki rata-rata 100 dari 100. Selanjutnya pada skenario kedua (menambahkan lima *user* baru) terjadi penurunan *performance* dari aplikasi asesmen nasional sebesar 14,25 dengan skor 82,25. Meskipun begitu *performance* dari aplikasi asesmen nasional masih terbilang baik. Selanjutnya pada skenario ketiga (menambahkan lima *user* baru) aplikasi asesmen nasional menunjukkan adanya penurunan *performance* sebesar 6,25 dengan skor 76. Meskipun mengalami penurunan, aplikasi asesmen nasional masih terbilang baik. Pada kriteria *performance* nilai yang didapat memiliki perubahan yang cukup

besar. Hasil ini didapatkan karena kriteria *performance* dipengaruhi oleh jumlah *user* yang mengakses web app.



BAB 5 KESIMPULAN DAN SARAN

Pada bab ini diberikan kesimpulan yang diambil selama pengerjaan tugas akhir beserta saran-saran tentang pengembangan yang dapat dilakukan pada tugas akhir akhir ini dimasa mendatang.

5.1 Kesimpulan

Selama proses perancangan, implementasi dan pengujian dapat diambil kesimpulan bahwa aplikasi asesmen nasional berbasis komputer bisa menerapkan teknologi *progressive web app* sehingga aplikasi mampu menampilkan halaman website dan bisa mengirim data meskipun dalam kondisi jaringan yang tidak memadai atau tidak ada jaringan. Selain itu, hasil pengujian menggunakan *lighthouse* menunjukkan bahwa kondisi jaringan dan jumlah *user* dapat mempengaruhi *performance* dari aplikasi asesmen nasional berbasis komputer dengan menggunakan *progressive web app*, *service worker* bisa melakukan *caching* terhadap data *static* halaman website, sehingga halaman website tetap bisa di akses meskipun dalam keadaan *offline*.

5.2 Saran

Penelitian ini masih dapat dikembangkan lebih lanjut pada penelitian selanjutnya. Beberapa saran pengembangan pada penelitian ini diantaranya dengan melakukan pengamanan atau *enkripsi* terhadap data yang di simpan di indexedDB browser pengguna.

DAFTAR PUSTAKA

- Adi, L., Akbar, R. J., & Khotimah, W. N. (2018). Platform e-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps. *Jurnal Teknik ITS*. <https://doi.org/10.12962/j23373539.v6i2.24291>
- Ane, L. (2021). *Diwawancarai oleh La Ode Muhamad Aslan*.
- AZIS, W. A. (2021). *Studi Dan Analisis Kinerja Indexed Pada Client Web Browser*.
- CNN. (2020). *Kominfo Ungkap Masalah Internet di Indonesia*. <https://www.cnnindonesia.com/teknologi/20201215131630-213-582359/kominfo-ungkap-masalah-internet-di-indonesia>
- Developers, G. (2021). *Introduction to Progressive Web App Architectures | Google Developers*. <https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures>
- Dicoding. (2020, January 15). *Ngoding Web Dinamis atau Statis, Apa Perbedaannya? - Dicoding Blog*. <https://www.dicoding.com/blog/ngoding-web-dinamis-atau-statis/>
- Diskominfo. (2020). *Peranan Teknologi Informasi Dalam Dunia Pendidikan*. <https://diskominfo.kedirikab.go.id/baca/peranan-teknologi-informasi-dalam-dunia-pendidikan>
- Hasanuddin, Asgar, H., & Hartono, B. (2022). Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan. *Jurnal Informatika Teknologi Dan Sains*, 4(1), 8–14. <https://doi.org/10.51401/jinteks.v4i1.1474>
- Kemdikbud. (2021, August 19). *Asesmen Nasional - Tahun 2021*. <https://anbk.kemdikbud.go.id/>
- MDN contributors. (2022, June 1). *IndexedDB API - Web APIs | MDN*.

https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API

Osmani, A. (2015, December). *Getting Started with Progressive Web Apps* | *Google Developers*. <https://developers.google.com/web/updates/2015/12/getting-started-pwa>

Perdana, A. (2020, December 2). *Responsive Web Design: Definisi, Prinsip Penting dan Keuntungannya*. [Https://Glints.Com](https://Glints.Com).
<https://glints.com/id/lowongan/responsive-web-design-adalah/#.YQ15AIgzZPY>

Pratiwi, Y. (2013). Implementasi E-Service Pada Organisasi Publik Di Bidang Pelayanan Pendidikan. *Jurnal Administrasi Publik Mahasiswa Universitas Brawijaya*.

Pusmenjar. (2021, November 24). *Asesmen Nasional*.
<https://pusmenjar.kemdikbud.go.id/an/>

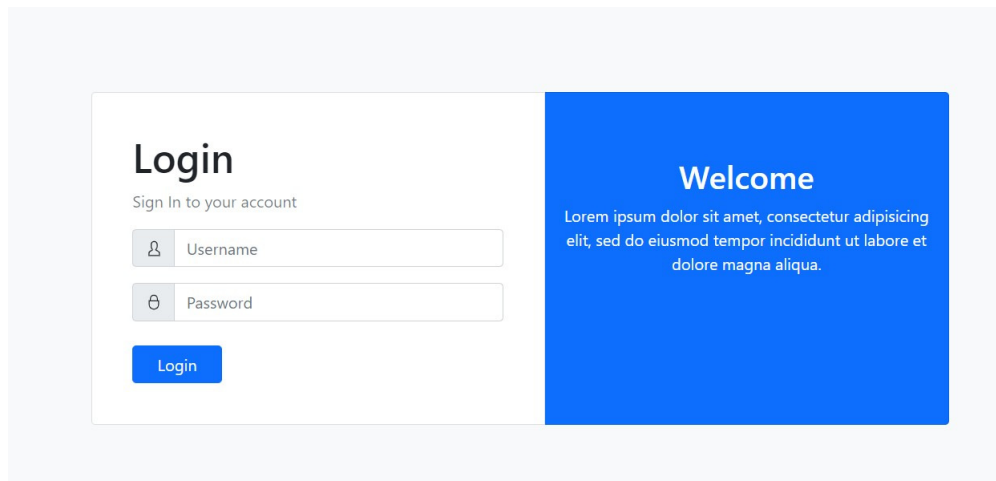
Techtarget. (2019, August 13). *What is Web Application (Web Apps) and its Benefits*. [Https://Searchsoftwarequality.Techtarget.Com](https://Searchsoftwarequality.Techtarget.Com).
<https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>

Warsihna, J. (2013). Pemanfaatan Teknologi Informasi dan Komunikasi (TIK) untuk Pendidikan di Daerah Terpencil, Tertinggal, dan Terdepan (3T). *Jurnal Teknodik*, 17(2), 238–245.

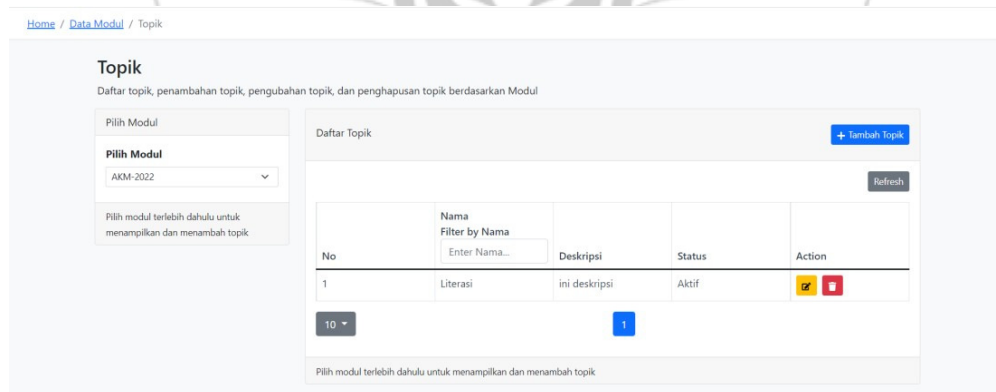
LAMPIRAN

Lampiran 1. Tampilan Antar Muka Role Admin

a. Halaman login admin



b. Halaman Data Modul : Topik



Tambah Topik

Nama Topik

Nama Modul

Pilih Modul ▼

Deskripsi

Status

Aktif

Submit

Close

Edit Topik

Nama Topik

Nama Modul

AKM-2022 - Defaul ▼

Deskripsi

Status

Aktif

Edit

Close

c. Halaman Data Modul : Soal

Mengelolah Soal

Daftar soal, penambahan soal, perubahan soal, dan penghapusan soal berdasarkan Modul

Pilih Topik

Pilih Topik

Pilih terlebih dahulu Topik yang akan digunakan sebelum menambah atau mengubah soal

Mengelolah Soal Berdasarkan Topik

Soal

Tipe Soal

Tambah Jawaban

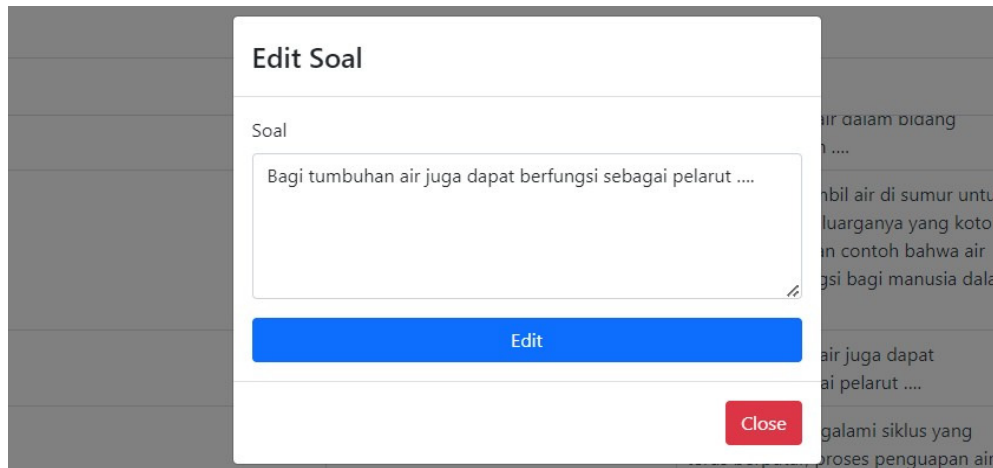
Soal

Bagi tumbuhan air juga dapat berfungsi sebagai pelarut

Jawaban











Jawaban Benar

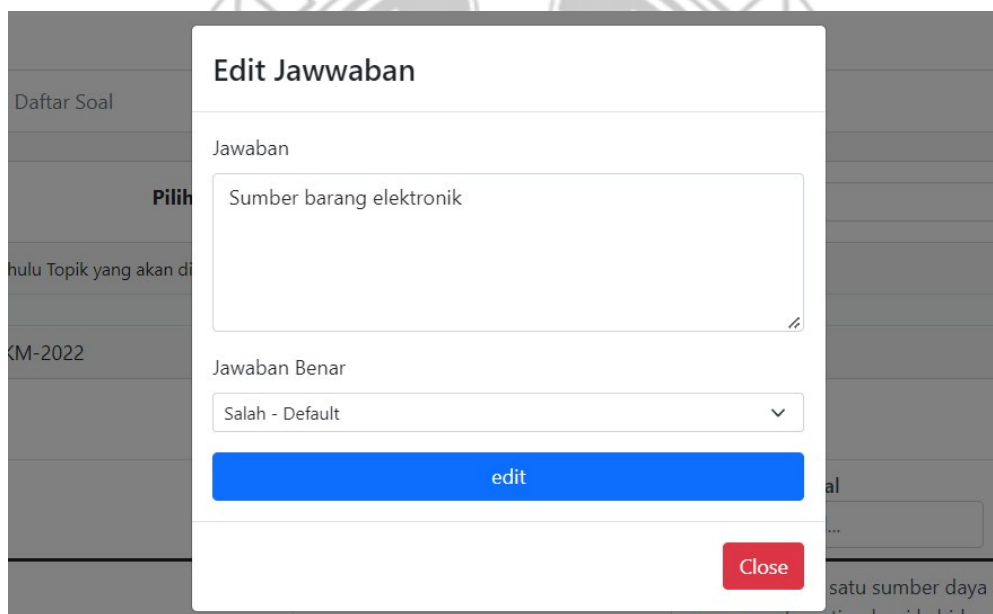
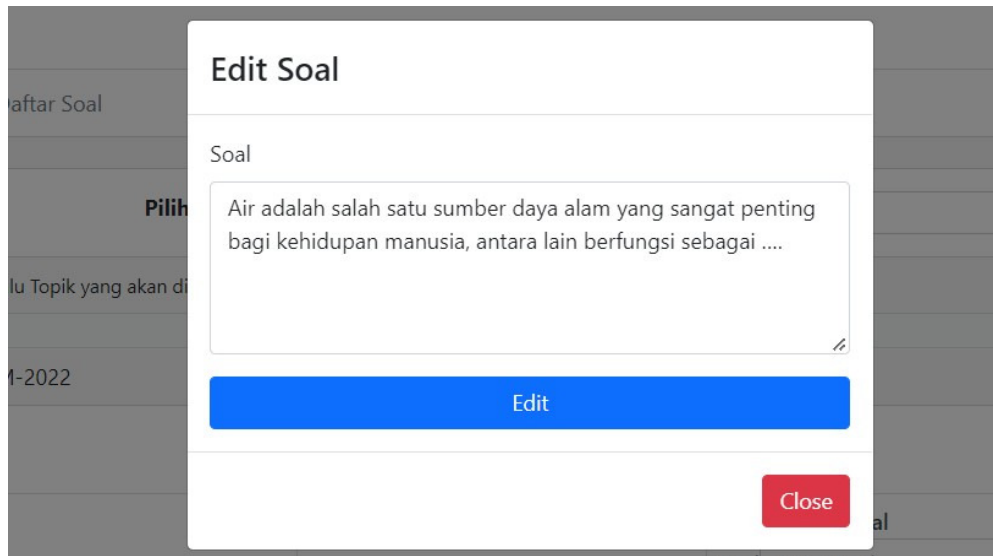
No	Opsi	Jawaban	Action
1	Zat Hara	Benar	<input type="button" value="Hapus"/>
2	Oksigen	Salah	<input type="button" value="Hapus"/>
3	Cahaya	Salah	<input type="button" value="Hapus"/>
4	Hama dan Gulma	Salah	<input type="button" value="Hapus"/>



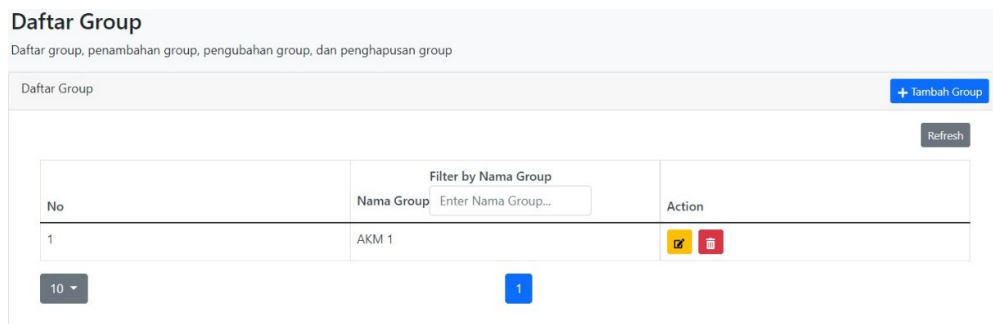
d. Halaman Data Modul : Daftar Soal

Daftar Soal AKM-2022 Refresh

No	Tipe Soal	Filter by Soal		Action
		Soal	Enter Soal...	
1	Pilihan Ganda	Air adalah salah satu sumber daya alam yang sangat penting bagi kehidupan manusia, antara lain berfungsi sebagai		 
	Salah	Sumber barang elektronik	 	
	Salah	Alat untuk membuat tanaman	 	
	Benar	Sumber minuman	 	
	Salah	Alat untuk bahan bakar	 	



e. Halaman Data Peserta : Daftar Group



Tambah Group

Nama Group

NB : Group digunakan untuk mengelompokkan setiap user

Tambah

Close

Edit Group

Nama Group

NB : Group digunakan untuk mengelompokkan setiap user

Edit

Close

f. Halaman Data Peserta : Daftar Peserta

Tambah Tes

Melihat daftar tes, mengubah dan menghapus tes

Tambah Tes

Group: Waktu Selesai:

Nama: Waktu:

Waktu Mulai: Token:

Simpan **Batal**

Daftar Tes

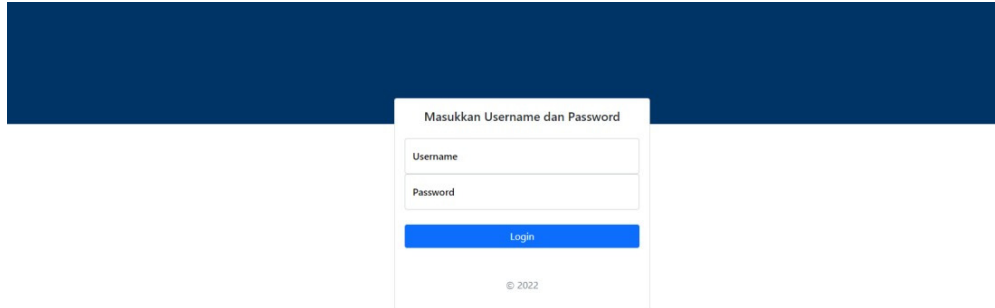
Refresh

No	Group	Nama Filter by Nama <input type="text" value="Enter Nama..."/>	Waktu	Token	Action
1	AKM 1	AKM TES 1	2022-08-08T12:22:00.000Z (60 menit)	GK2RP	

10 **1**

Lampiran 2 . Tampilan Antar Muka Role Siswa

a. Halaman Login Siswa



Masukkan Username dan Password

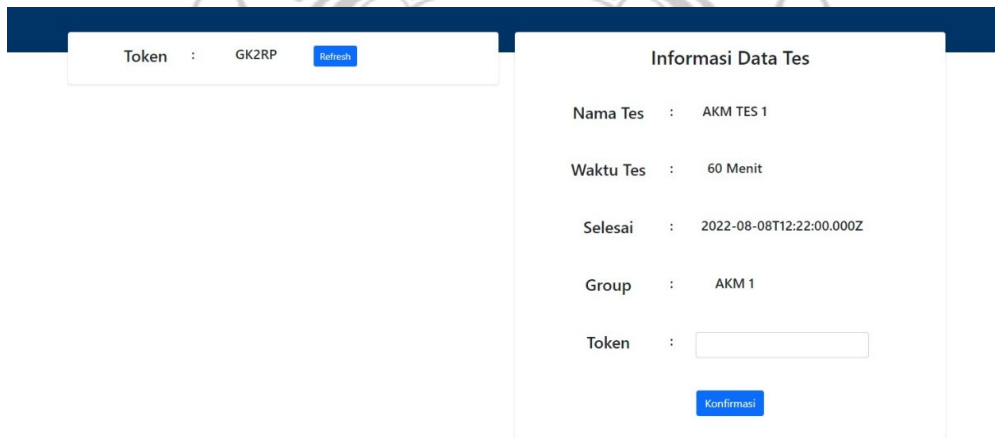
Username

Password

Login

© 2022

b. Halaman Konfirmasi Data Tes



Token : GK2RP Refresh

Informasi Data Tes

Nama Tes : AKM TES 1

Waktu Tes : 60 Menit

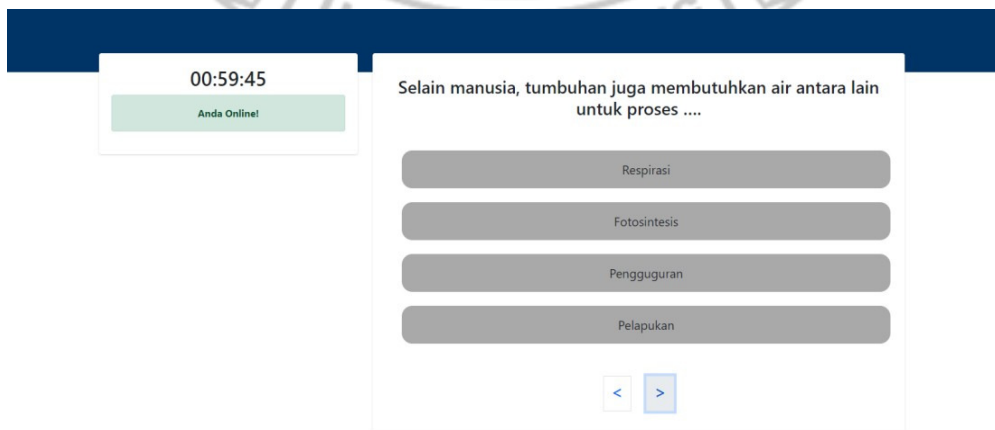
Selesai : 2022-08-08T12:22:00.000Z

Group : AKM 1

Token :

Konfirmasi

c. Halaman Ujian



00:59:45

Anda Online!

Selain manusia, tumbuhan juga membutuhkan air antara lain untuk proses

Respirasi

Fotosintesis

Pengguguran

Pelapukan

< >

d. Halaman Result



Lampiran 3. Hasil Pengujian Menggunakan *Lighthouse*

- Skenario pertama (lima user awal)

a. Halaman Login



b. Halaman Konfirmasi Data



c. Halaman Ujian



d. Halaman Result



- Skenario kedua (Menambahkan 5 user pertama)

a. Halaman Login



b. Halaman Konfirmasi Data



c. Halaman Ujian



d. Halaman Result



- Skenario kedua (Menambahkan 5 user kedua)

a. Halaman Login



b. Halaman Konfirmasi Data



c. Halaman Ujian



d. Halaman Result

