

BAB V

Progressive Web App (PWA) #1

5.1. Tujuan

1. Praktikan dapat memahami konsep dasar teknologi *Progressive Web App*
2. Praktikan dapat mengunggah suatu project PWA ke *hosting firebase*
3. Praktikan dapat memahami konsep dasar tentang *framework* Vue.js
4. Praktikan dapat mengetahui penggunaan Node.js
5. Praktikan dapat memahami konsep dasar penggunaan NPM
6. Praktikan mampu *mendeploy* ke *smartphone* hasil dari praktikum tentang PWA

5.2. Alat dan Bahan

5.2.1. Laptop



Gambar 5.1 Laptop

Laptop digunakan sebagai media dalam pengerjaan praktikum kali ini. Kita meng*install* dan melakukan kegiatan *programming* pada laptop.

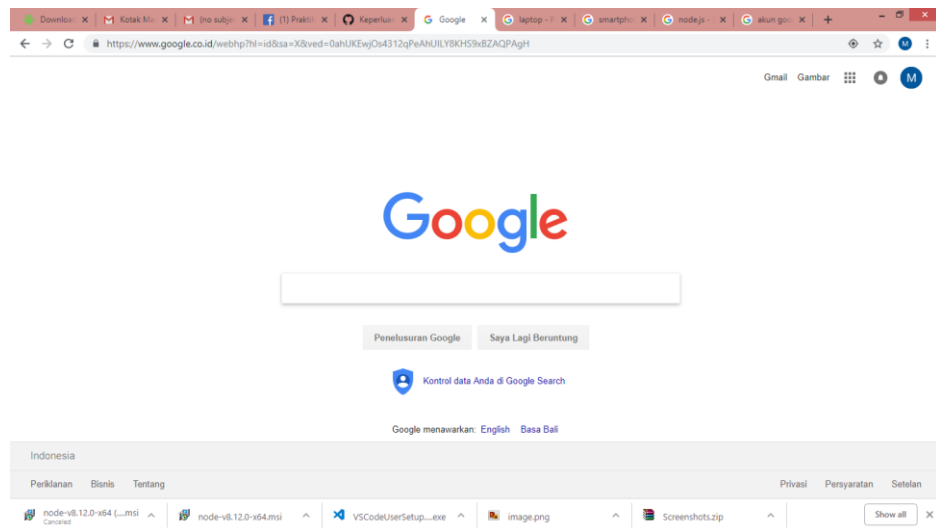
5.2.2. Smartphone



Gambar 5.2 Smartphone

Smartphone digunakan sebagai media untuk melihat hasil akhir tampilan dari program yang telah kita buat. Program yang dibuat nantinya akan di*deploy* ke *smartphone*.

5.2.3. Aplikasi Google Chrome



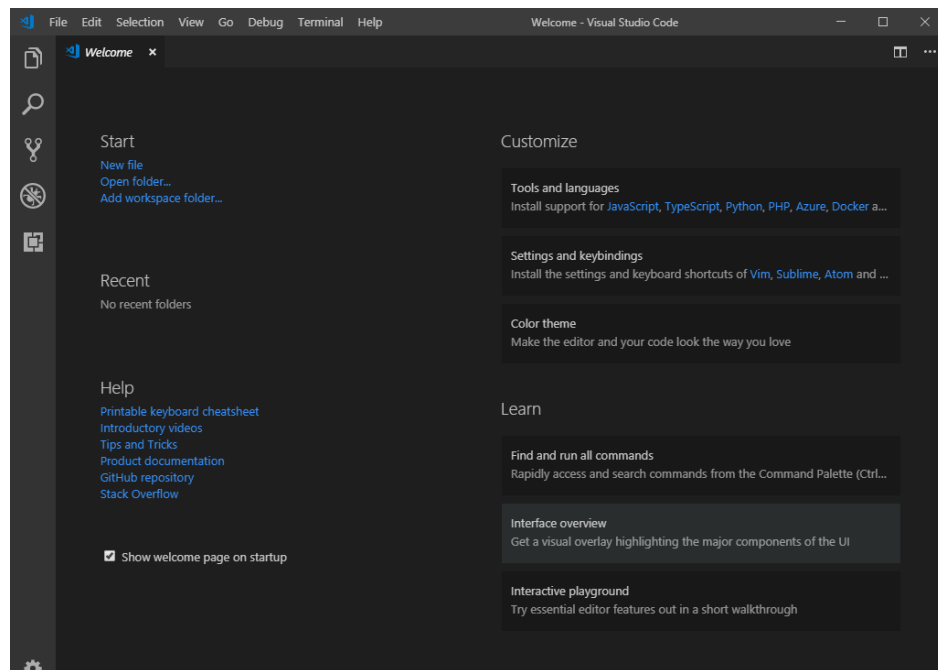
Gambar 5.3 Google Chrome Pada Laptop



Gambar 5.4 Google Chrome Pada *Handphone*

Google Chrome digunakan untuk kita membuka alamat *hosting* yang telah dibuat sebelumnya. Kita dapat melihat tampilan pada alamat *hosting* yang telah dibuat.

5.2.4. Text Editor



Gambar 5.5 Visual Studio Code

Text Editor yang akan digunakan pada praktikum kali ini yaitu Visual Studio Code. Di sinilah kita akan melakukan kegiatan *coding program* untuk praktikum.

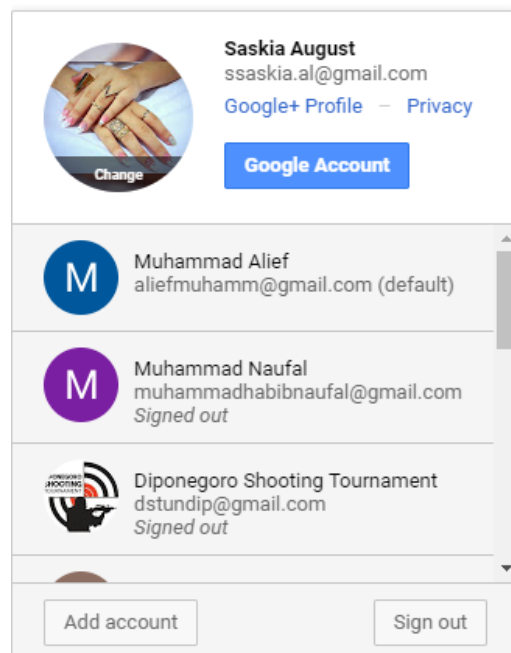
5.2.5. Node.js



Gambar 5.6 Node.js

Kita menggunakan Node.js pada praktikum kali ini sebagai perangkat lunak untuk mengembangkan aplikasi berbasis *web* dan ditulis dalam sintaks bahasa pemrograman JavaScript.

5.2.6. Akun Google aktif



Gambar 5.7 Akun google aktif

Dibutuhkan akun google aktif untuk mendaftar pada firebase dikarenakan pada praktikum ini digunakan firebase untuk hosting konten statis dan konfigurasi ringan untuk PWA yang rumit.

5.3. Dasar Teori

5.3.1. Progressive Web App (PWA)

Progressive Web App adalah pengalaman yang menggabungkan yang terbaik dari *web* dan yang terbaik dari aplikasi. Pengalaman ini bermanfaat untuk pengguna dari kunjungan pertamanya di *tab browser*, tanpa harus melakukan pemasangan. Pengguna secara progresif akan membangun hubungan dengan aplikasi, yang semakin lama semakin kuat. Aplikasi dimuat cepat, bahkan pada jaringan yang tidak stabil, mengirimkan pemberitahuan *push* yang relevan, memiliki ikon pada layar beranda, dan dimuat dengan pengalaman tingkat atas selayar penuh.

PWA memiliki sifat-sifat sebagai berikut :

- **Progresif** - Bekerja untuk setiap pengguna, apa pun pilihan *browser* mereka karena dibangun dengan peningkatan progresif sebagai konsep intinya.
- **Responsif** - Cocok dengan setiap faktor bentuk: perangkat *desktop*, seluler, tablet, atau apa saja yang muncul berikutnya.
- **Konektivitas independen** - Disempurnakan dengan *service worker* agar bisa bekerja *offline* atau pada jaringan berkualitas-rendah.
- **Seperti-Aplikasi** - Terasa seperti sebuah aplikasi untuk pengguna dengan interaksi dan navigasi bergaya-aplikasi karena mereka dibangun di atas model *shell* aplikasi.
- **Segar** - Selalu terkini berkat proses pembaruan *service worker*.
- **Aman** - Disediakan melalui HTTPS untuk mencegah *snooping* dan memastikan materi belum dirusak.
- **Dapat ditemukan** - Dapat diidentifikasi sebagai "aplikasi" berkat manifes W3C dan cakupan registrasi *service worker*, yang memungkinkan mesin telusur untuk menemukannya.
- **Bisa dilibatkan-kembali** - Kemudahan untuk dilibatkan-kembali dengan fitur seperti pemberitahuan *push*.
- **Dapat dipasang** - Memungkinkan pengguna untuk "menyimpan" aplikasi yang mereka anggap paling berguna di layar beranda tanpa kerumitan toko aplikasi.

- **Bisa ditautkan** - Dapat dengan mudah dibagikan melalui URL, tidak memerlukan pemasangan yang rumit.

Sumber: <https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=id>

Diakses pada 26 Oktober 2018

5.3.2. Web Server

Server atau *Web server* adalah sebuah *software* yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada klien yang dikenal dan biasanya kita kenal dengan nama *web browser* (Mozilla Firefox, Google Chrome) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman *web* dan pada umumnya akan berbentuk dokumen HTML.

Fungsi utama *Server* atau *Web server* adalah untuk melakukan atau akan mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. Halaman *web* yang diminta terdiri dari berkas teks, video, gambar, *file* dan banyak lagi. Pemanfaatan *web server* berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman *web* termasuk yang di dalam berupa teks, video, gambar dan banyak lagi.

Salah satu contoh dari *Web Server* adalah Apache. Apache (*Apache Web Server – The HTTP Web Server*) merupakan *web server* yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya.

Sumber: <https://idcloudhost.com/pengertian-web-server-dan-fungsinya/>

Diakses pada 26 Oktober 2018

5.3.3. Service Worker

Service worker adalah skrip yang dijalankan *browser* Anda di latar belakang, terpisah dari laman *web*, yang membuka pintu ke berbagai fitur yang tidak memerlukan laman *web* atau interaksi pengguna. Saat ini, *service worker* sudah menyertakan berbagai fitur seperti pemberitahuan *push* dan sinkronisasi latar belakang. Di masa mendatang, *service worker* akan mendukung hal-hal lainnya seperti sinkronisasi berkala atau *geofencing*. Fitur inti yang didiskusikan

dalam tutorial adalah kemampuan mencegat dan menangani permintaan jaringan, termasuk mengelola *cache* respons lewat program.

Yang membuat API ini menarik adalah karena memungkinkan Anda mendukung pengalaman *offline*, yang memberikan *developer* kontrol penuh atas pengalaman.

Sebelum *service worker*, ada satu API lain yang memberi pengguna pengalaman *offline* di web, yang disebut AppCache. Masalah utama pada AppCache adalah jumlah *gotcha* yang ada serta fakta bahwa meskipun desain bekerja dengan sangat baik untuk laman aplikasi *web* tunggal, namun ternyata tidak begitu baik untuk situs multi-laman. *Service worker* telah didesain untuk menghindari titik-titik menyulitkan yang sudah umum ini.

Sumber: <https://developers.google.com/web/fundamentals/primers/service-workers/?hl=id>

Diakses pada 26 Oktober 2018

5.3.4. Web App Manifest

Manifes aplikasi web adalah *file* JSON sederhana yang memberikan Anda, *developer*, kemampuan untuk mengontrol bagaimana aplikasi terlihat oleh pengguna di daerah yang mereka harap akan melihat aplikasi (misalnya, layar beranda perangkat seluler), mengarahkan apa yang bisa diluncurkan pengguna, dan menentukan tampilannya pada saat peluncuran.

Manifes aplikasi *web* menyediakan kemampuan untuk menyimpan *bookmark* situs ke layar beranda perangkat. Ketika sebuah situs diluncurkan dengan cara ini:

- Situs akan memiliki ikon dan nama yang unik sehingga pengguna bisa membedakannya dari situs yang lain.
- Situs akan menampilkan sesuatu kepada pengguna selagi sumber daya diunduh atau dipulihkan dari *cache*.
- Situs akan menyediakan karakteristik tampilan *default* ke *browser* untuk menghindari transisi yang terlalu mendadak bila sumber daya situs tersedia.

Sumber: <https://developers.google.com/web/fundamentals/web-app-manifest/?hl=id>

Diakses pada 26 Oktober 2018

5.3.5. Node.js

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis *web* dan ditulis dalam sintaks bahasa pemrograman JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi *client / browser* saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi *server*, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan *server web* tanpa menggunakan program *server web* seperti *Apache* atau *Nginx*.

Untuk mengeksekusi Javascript sebagai bahasa *server* diperlukan *engine* yang cepat dan mempunyai performansi yang bagus. *Engine* Javascript dari Google bernama V8-lah yang dipakai oleh Node.js yang juga merupakan *engine* yang dipakai oleh *browser* Google Chrome.

Berbeda dengan bahasa pemrograman sisi *server* pada umumnya yang bersifat *blocking*, Node.js bersifat *non-blocking*, sebagaimana halnya JavaScript bekerja. Node.js berjalan dengan basis event (*event-driven*). Maksud dari *Blocking* secara sederhana adalah, bahwa suatu kode program akan dijalankan hingga selesai, baru kemudian beralih ke kode program selanjutnya.

Sumber: <https://www.codepolitan.com/mengenal-nodejs-5880234fe9ae3>

Diakses pada 26 Oktober 2018

5.3.6. Node Package Management (npm)

NPM adalah sebuah *manager* paket untuk bahasa pemrograman JavaScript yang menjadi *manager default* untuk lingkungan *runtime* Node.js. Npm terdiri dari *command line client* yang disebut **npm**, dan sebuah *database online* yang berisi paket publik dan paket pribadi berbayar, yang disebut *npm registry*. *Registry* dapat diakses melalui klien, dan ketersediaan paket dapat dicari melalui *website* npm. *Manager* paket dan *registry* dikelola oleh npm, Inc.

NPM dapat mengelola ketergantungan paket dari suatu proyek, serta pemasangan program JavaScript secara global. Ketika digunakan sebagai *manager* paket untuk proyek lokal, npm dapat menginstal semua paket ketergantungan dengan satu perintah melalui *file* package.json. Pada *file* package.json, setiap paket dapat ditentukan versi yang akan dipasang dengan menggunakan skema *semantic versioning*. Hal ini memungkinkan pengembang untuk melakukan pemberharuan paket otomatis dan juga menghindari benturan perubahan yang tidak diinginkan. NPM juga menyediakan alat *version-bumping* untuk memilih paket dengan versi tertentu. Selain package.json, npm juga menyediakan *file* package-lock.json yang bertugas mengunci versi paket yang terpasang pada proyek.

Sumber: [https://id.wikipedia.org/wiki/Npm_\(perangkat_lunak\)](https://id.wikipedia.org/wiki/Npm_(perangkat_lunak))

Diakses pada 26 Oktober 2018

5.3.7. Vue.js

Vue js merupakan salah satu *framework* yang banyak digunakan. Jika anda telah terbiasa dengan menggunakan angular js, maka untuk melakukan migrasi ke vue js tidak lah terlalu susah karena mereka mempunyai kemiripan di bagian sintak, hanya saja angular js memiliki *development stack* yang lebih rumit dibandingkan dengan vue js. Bagi anda yang penasaran dengan angular js, silahkan lihat di artikel belajar angular js. Vue js juga menggunakan fitur *data binding* sama seperti angular js. Jika di angular js kita menggunakan fungsi *controller*, maka di vue js kita akan menggunakan pendekatan dom yaitu kita akan menggunakan id, *name* atau *class* dari suatu *component* html.

Sumber: <https://rizkimufrizal.github.io/belajar-vue-js/>

Diakses pada 26 Oktober 2018

5.3.8. Firebase

Firebase Hosting adalah layanan *hosting* konten *web* yang berkelas produksi untuk *developer*. Hanya dengan satu perintah, Anda dapat menerapkan aplikasi *web* serta menyajikan konten dinamis dan statis ke jaringan penayangan konten (CDN) global dengan cepat dan mudah.

Firebase Hosting dibuat untuk *developer web modern*. Dengan hadirnya *framework* JavaScript *front-end* seperti Angular dan fitur generator statis seperti

Jekyll, situs dan aplikasi kini jadi lebih tangguh daripada sebelumnya. Jika Anda menerapkan halaman landing aplikasi yang sederhana atau Progressive Web App (PWA) yang kompleks, Hosting menyediakan infrastruktur, fitur, dan rangkaian sarana yang disesuaikan untuk menerapkan serta mengelola situs dan aplikasi.

Hosting menambahkan *subdomain* ke *project* Anda di *domain* firebaseapp.com. Dengan *Firebase CLI*, Anda dapat menerapkan *file* dari direktori lokal di komputer Anda ke *server Hosting*. Selain menyajikan konten statis, Anda dapat menggunakan *Cloud Functions for Firebase* untuk menyajikan konten dinamis di situs Anda. Semua konten disajikan melalui koneksi SSL dari *server edge* terdekat di CDN global kami.

Selain *hosting* konten, *Firebase Hosting* menawarkan opsi konfigurasi *hosting* yang ringan kepada Anda untuk membuat *Progressive Web App* yang canggih. Anda dapat menulis ulang URL untuk perutean sisi klien atau menyiapkan header kustom dengan mudah.

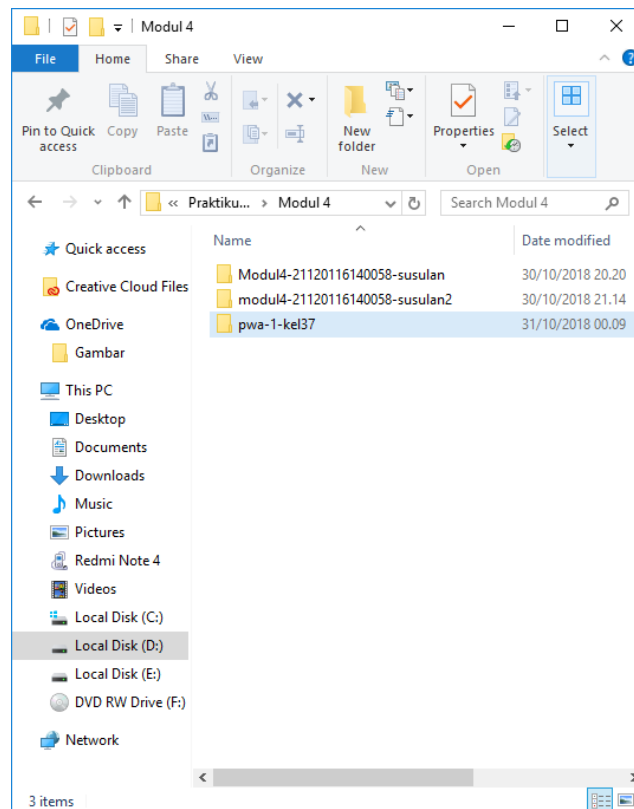
Sumber: <https://firebase.google.com/docs/hosting/?hl=id>

Diakses pada 26 Oktober 2018

5.4. Langkah Percobaan

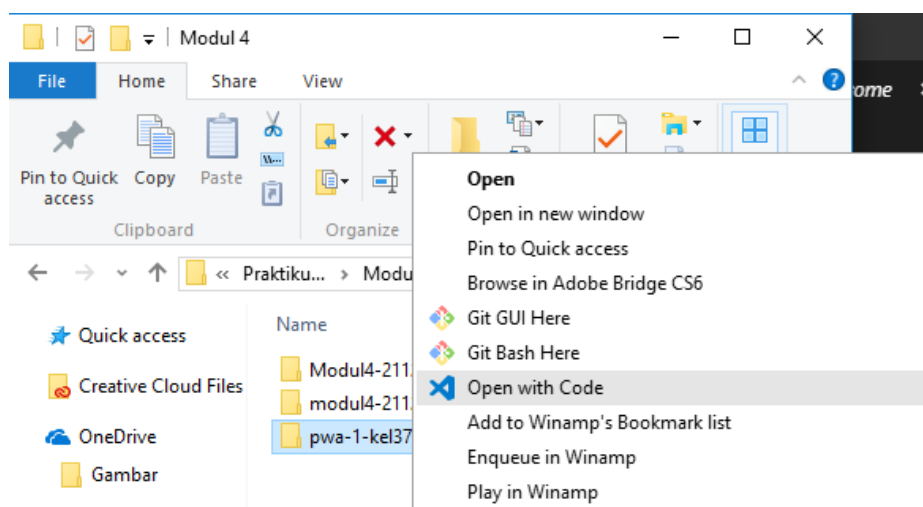
5.4.1. Pembuatan Halaman Website

1. Buat *folder* beri nama “**pwa-1-kel37**”, untuk penempatan *foldernya* bebas.



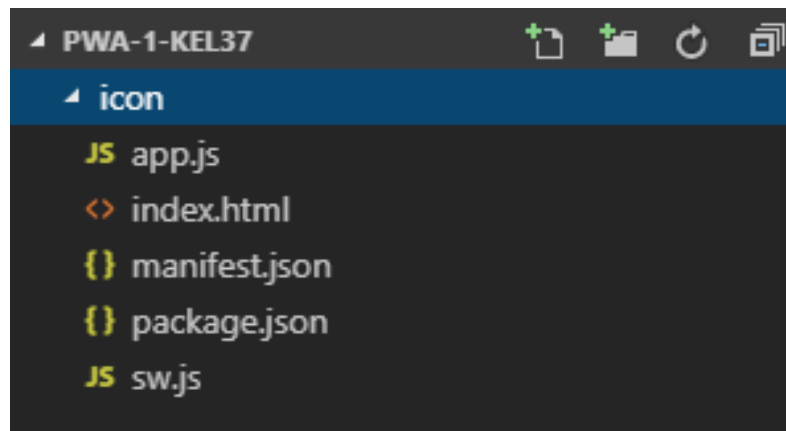
Gambar 5.8 Folder pwa-1-kel37

2. Setelah itu klik kanan *folder* tersebut dan pilih *Open with Code*, untuk membuka *VS Code* beserta *foldernya*.



Gambar 5.9 Membuka *folder* dengan *VS Code*

3. Setelah itu buatlah 5 *file* dan 1 *folder* seperti gambar di bawah.



Gambar 5.10 Folder dan File yang dibuat pada Pwa-1-kel37

4. Pada file **package.json** diberi kode seperti ini.

```
{
  "name": "pwa-1-kel37",
  "version": "1.0.0",
  "description": "Praktikum MDP PWA Jilid 1",
  "main": "index.js",
  "scripts": {
    "start": "http-server -c-1 -p 8888"
  },
  "keywords": [
    "pwa"
  ],
  "author": "Asisten MDP PWA 1",
  "license": "UNLICENSED",
  "devDependencies": {
    "http-server": "^0.10.0"
  }
}
```

5. Setelah itu, isikan file **index.html** dengan kode berikut:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Modul PWA</title>
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/boo
tstrap.min.css">
  <link rel="shortcut icon" href="icon/192.ico"
type="image/x-icon" />
  <link rel="icon"
href="https://upload.wikimedia.org/wikipedia/commons/f/f2/Lam
bang_Kota_Semarang.png" sizes="32x32" />
```

```

<!-- buat manifest json -->
</head>
<body>

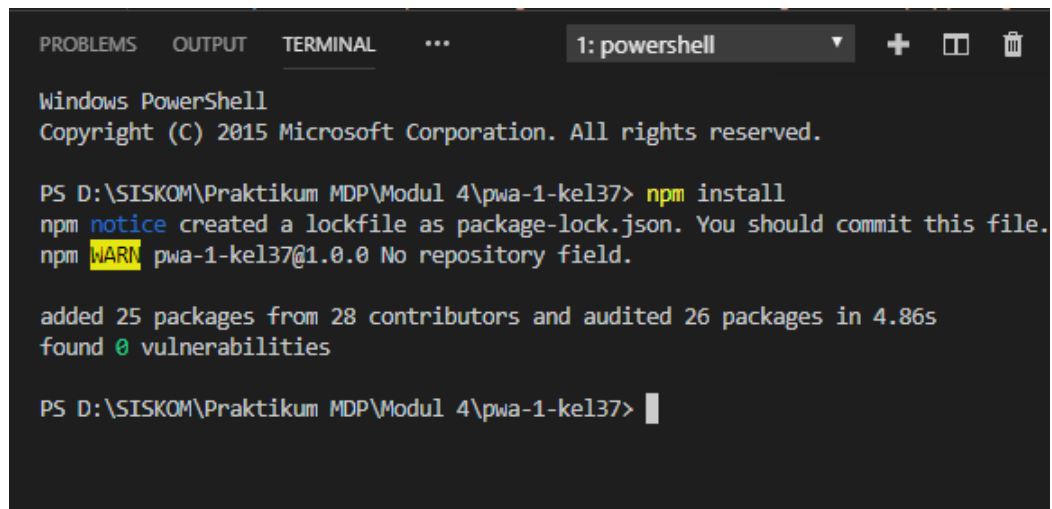
<!-- untuk program utama -->

</body>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script
src="https://unpkg.com/axios/dist/axios.min.js"></script>
  <script src="app.js"></script>
  <!-- untuk service worker -->

</html>

```

6. Setelah itu kita buka *terminal* pada *VS Code*, yaitu dengan menekan **Ctrl+`**, setelah itu kita ketikkan perintah **npm install**. Pastikan sudah menginstall *package* npm di *visual studio code*.



```

PROBLEMS OUTPUT TERMINAL ... 1: powershell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37> npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN pwa-1-kel37@1.0.0 No repository field.

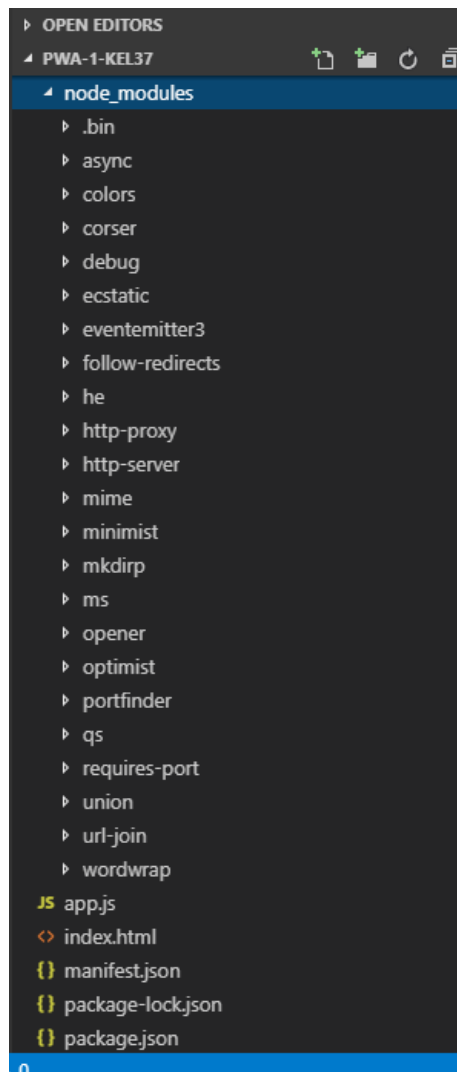
added 25 packages from 28 contributors and audited 26 packages in 4.86s
found 0 vulnerabilities

PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37>

```

Gambar 5.11 *Install* NPM menggunakan *terminal* *VSCode*

7. Setelah itu akan muncul *folder node_module* dan **package-lock.json**



Gambar 5.12 node_modules dan package-lock.json

8. Setelah itu buka *file app.js* dan isikan dengan kode berikut

```
new Vue({
  el: '#app',
  data: {
    kodepos      :      [], search      :      '',
    awal:0,akhir:10,halaman:1,tampil:true //deklarasi variabel
  }, //1 halaman menampilkan 10 data
  created(){ //method mengambil data
    this.getData(); //untuk mengambil data
    console.log(this.awal) //dari variabel awal
  },
  methods:{
    next(){ //ketika menekan tombol next
      this.awal=this.awal+=10 //menampilkan 10 data
    selanjutnya
      this.akhir=this.akhir+=10
    }
  }
})
```

```

        this.halaman=this.halaman+1 //halaman +1
    },
    prev(){ //ketika menekan tombol prev
        this.awal=this.awal-=10 //menampilkan 10 data
        sebelumnya
        this.akhir=this.akhir-=10
        this.halaman=this.halaman-=1 //halaman -1
    },
    getData(){ //method untuk mengambil data
        var kd=this;
        axios.get('https://kodepos-
2d475.firebaseio.com/kota_kab/k110.json?print=pretty').then(f
unction(response){ //link untuk kode pos semarang
        kd.kodepos=response.data;
        })
    }
},
computed:{
    cari(){ //method untuk mencari
        return this.kodepos.filter(pos =>{
            return
pos.kecamatan.toLowerCase().match(this.search.toLowerCase());
//mengembalikan nilai kode pos
        });
    }
}
});

```

9. Setelah kita mengisi app.js, selanjutnya kita melengkapi kode pada *file index.html* dengan kode berikut diletakkan dibawah tulisan **<!-- untuk program utama -->**

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Modul PWA</title>
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/boot
strap.min.css">
    <link rel="shortcut icon" href="icon/192.ico"
type="image/x-icon" />
    <link rel="manifest" href="manifest.json">
    <link rel="icon"
href="https://upload.wikimedia.org/wikipedia/commons/f/f2/Lamb
ang_Kota_Semarang.png" sizes="32x32" />
<!-- buat manifest json -->
</head>
<body>

<div class="container-fluid">

```



```

<div id="app">
  <center>
    <div class="card bg-warning mb-3">
      <div class="card-body">
        
        <br><br>
        <h5 class="card-title">Daftar Kode POS
Semarang</h5>
        <p class="card-text">Berikut adalah semua kode
pos yang ada di Kab. Semarang dan Kota Semarang</p>
        <p>Kurang lebih datanya ada
{{kodepos.length}}</p>
        <a href="#" v-on:click="tampil = !tampil" v-
show="tampil" class="btn btn-primary">Tampilan</a>
      </div>
    </div>
    <div v-show="!tampil">
      <div class="row">
        <div class="col-sm-4 col-md-4 col-lg-4 col-4">
          <button class="btn btn-primary btn-sm" v-
show="awal>0" v-on:click="prev()">Balek Wae</button>
        </div>
        <div class="col-sm-4 col-md col-lg-4 col-4">
          <span>Halaman : {{halaman}}</span>
        </div>
        <div class="col-sm-4 col-md-4 col-lg-4 col-4">
          <button class="btn btn-primary btn-sm" v-
show="akhir" v-on:click="next()">Lanjoooot</button>
        </div>
      </div>
      <div class="row" style="margin:20px">
        <div class="col-sm-3"></div>
        <div class="col-sm-6">
          <input type="text" v-model="search"
class="form-control" placeholder="Pencarian berdasarkan
Kecamatan">
        </div>
      </div>
    </div>
    <div v-show="!tampil" class="table-responsive container">
      <table id="example" class="table table-hover table-
bordered">
        <tr>
          <th>Kecamatan</th>
          <th>Kelurahan</th>
          <th>Kode POS</th>
        </tr>
        <tr v-for="pos in cari.slice(awal,akhir)">
          <td style="width:40%">{{pos.kecamatan}}</td>
          <td style="width:40%">{{pos.kelurahan}}</td>
          <td style="width:20%">{{pos.kodepos}}</td>
        </tr>
      </table>
    </div>
  </center>
</div>

```

```

        </tr>
      </table>
    </div>
  </div>

</body>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/u
md/popper.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootst
rap.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script
src="https://unpkg.com/axios/dist/axios.min.js"></script>
  <script src="app.js"></script>

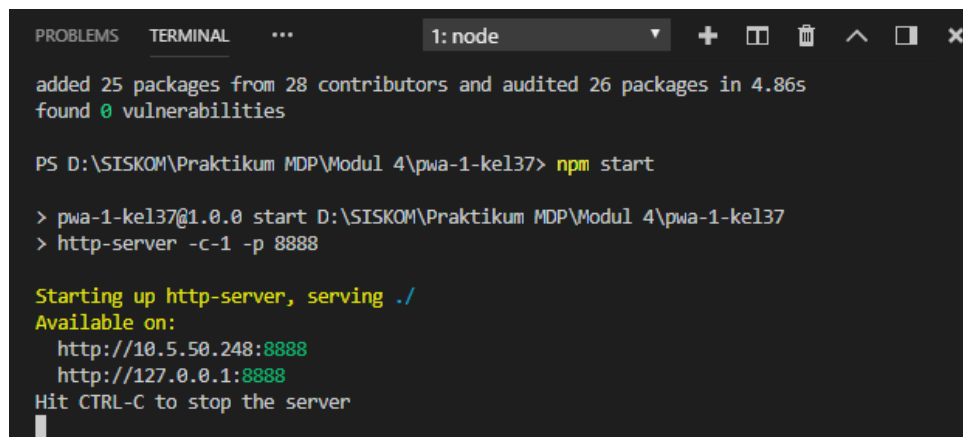
  <script src="sw.js"></script>

  <!-- untuk service worker -->

</html>

```

10. Buka *terminal* kembali, dan ketikkan perintah ***npm start***



```

PROBLEMS  TERMINAL  ...  1: node
added 25 packages from 28 contributors and audited 26 packages in 4.86s
found 0 vulnerabilities

PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37> npm start

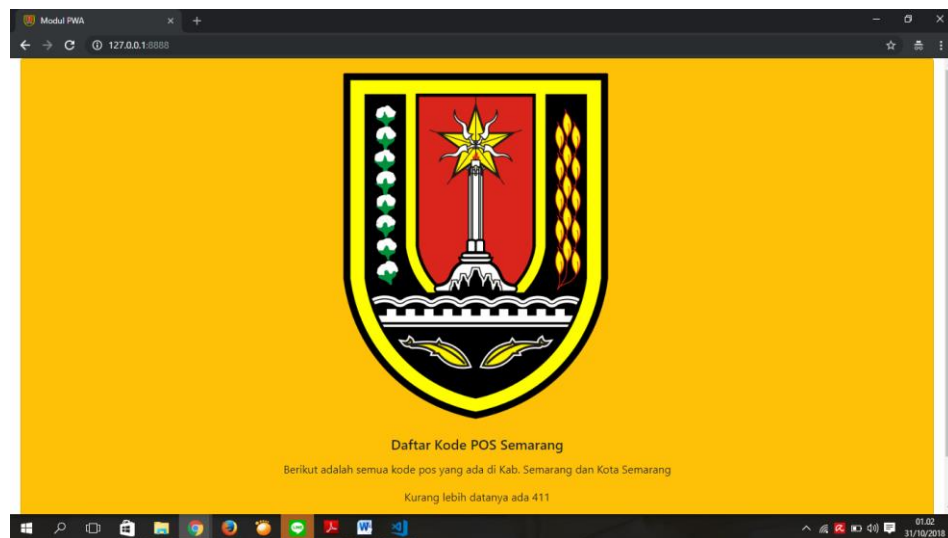
> pwa-1-kel37@1.0.0 start D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37
> http-server -c-1 -p 8888

Starting up http-server, serving ./
Available on:
  http://10.5.50.248:8888
  http://127.0.0.1:8888
Hit CTRL-C to stop the server

```

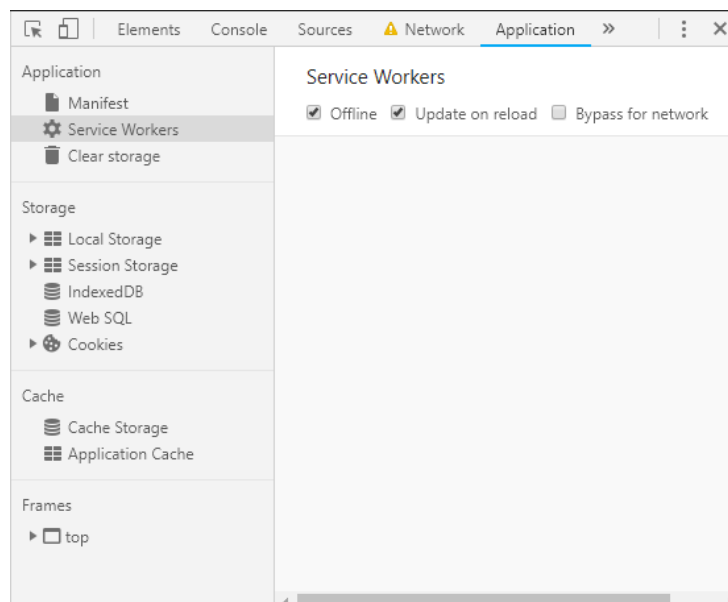
Gambar 5.13 Mengetikkan perintah *npm start*

Akan ada 2 *link*, kita pilih yang 127.0.0.1:8888, setelah itu kita masukan di *web browser* google chrome. Hasilnya seperti ini



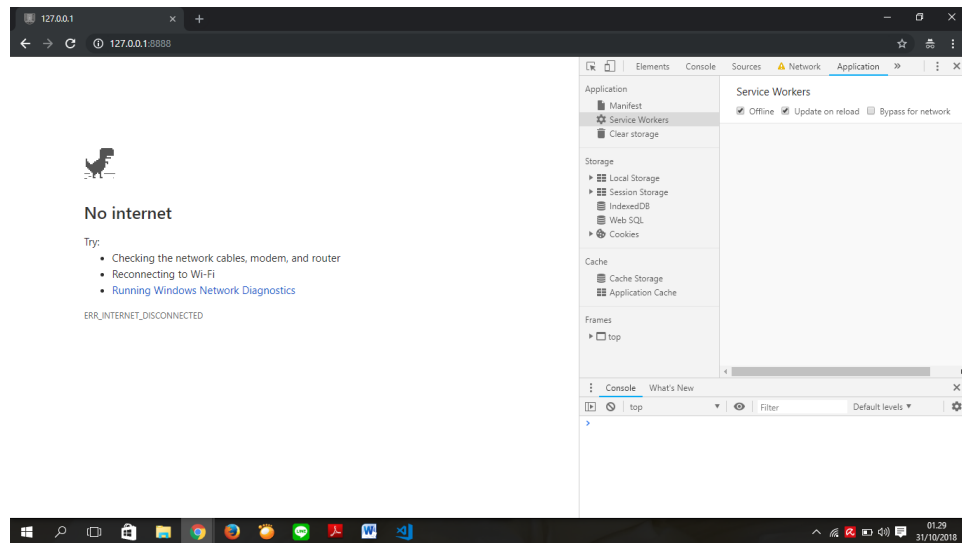
Gambar 5.14 Tampilan yang muncul di *browser*

11. Setelah berhasil menampilkan halaman seperti gambar diatas, selanjutnya kita buka *inspect element*, dengan mengetikan `ctrl+shift+I` atau klik kanan pilih *inspect element*
12. Setelah sampai di *inspect element*, kita pilih tab application dan pilih **service worker**. Kita beri centang pada *offline*.



Gambar 5.15 Halaman *inspect elements*

Setelah kita beri centang pada *offline*, reload halaman webnya maka akan muncul seperti ini.

Gambar 5.16 Halaman *web* saat *offline*

13. Setelah itu hilangkan centang pada *offline* dan *reload* kembali.
14. Buka file **sw.js** dan isikan kode berikut.

```
self.addEventListener('install', function (event) {
  console.log('SW Installed');
  event.waitUntil(
    caches.open('static')
      .then(function (cache) {
        cache.addAll([
          '/',
          '/index.html',
          '/app.js',
          '/manifest.json',
          'https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css',
          'https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js',
          'https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js',
          'https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js',
          'https://cdn.jsdelivr.net/npm/vue/dist/vue.js',
          'https://unpkg.com/axios/dist/axios.min.js',
          'https://kodepos-2d475.firebaseio.com/kota_kab/k110.json?print=pretty',
          'https://upload.wikimedia.org/wikipedia/commons/f/f2/Lambang_Kota_Semarang.png'
        ]);
      })
  );
});
```

```

    });

self.addEventListener('activate', function () {
    console.log('SW activated');
});

self.addEventListener('fetch', function (e) {
    console.log('[ServiceWorker] Fetch', e.request.url);
    e.respondWith(
        caches.match(e.request).then(function (response) {
            return response || fetch(e.request);
        })
    );
});

```

Dan buka **manifest.json** dan masukkan kode berikut

```

{
  "name": "Pos Semarang",
  "short_name": "Kode Pos Semarang",
  "theme_color": "#ffc107",
  "background_color": "#ffc107",
  "display": "standalone",
  "orientation": "portrait",
  "scope": ".",
  "start_url": ".",
  "icons": [
    {
      "src": "/icon/32.ico",
      "sizes": "32x32",
      "type": "image/ico"
    },
    {
      "src": "/icon/96.ico",
      "sizes": "96x96",
      "type": "image/ico"
    },
    {
      "src": "/icon/128.ico",
      "sizes": "128x128",
      "type": "image/ico"
    },
    {
      "src": "/icon/192.ico",
      "sizes": "192x192",
      "type": "image/ico"
    },
    {
      "src": "/icon/256.ico",
      "sizes": "256x256",
      "type": "image/ico"
    }
  ]
}

```

Setelah menambahkan kode tersebut, jangan lupa kita tambahkan kode berikut pada **index.html**

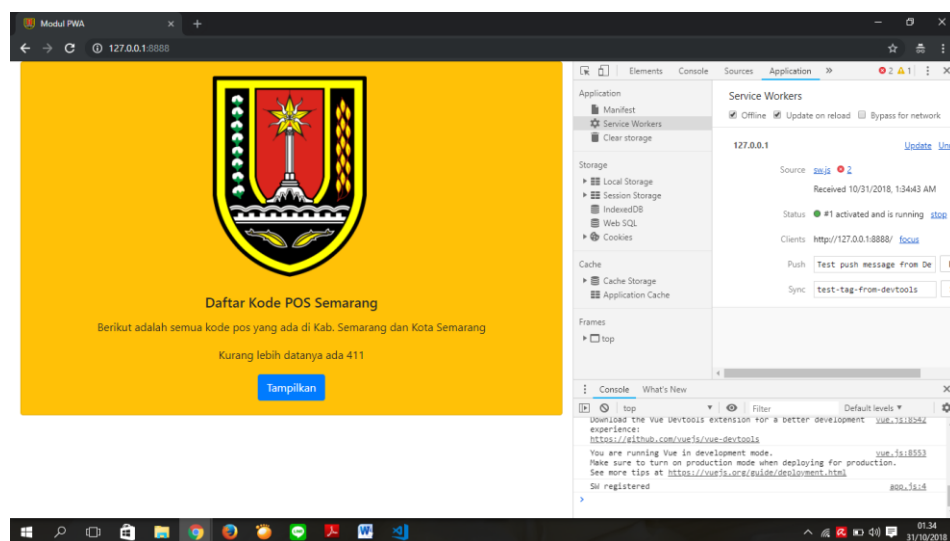
```
<link rel="manifest" href="manifest.json">
<script src="sw.js"></script>
```

Diletakan sebagaimana terlihat pada **index.html**

15. Tambahkan kode berikut pada **app.js**

```
if ('serviceWorker' in navigator){
  navigator.serviceWorker.register('/sw.js')
    .then(function(){
      console.log('SW registered');
    });
}
```

16. Setelah itu di-*reload* terlebih dahulu dan buka inspect elemen, dan buka tab *application* dan lakukan seperti langkah 12 dan lihat perbedaan.

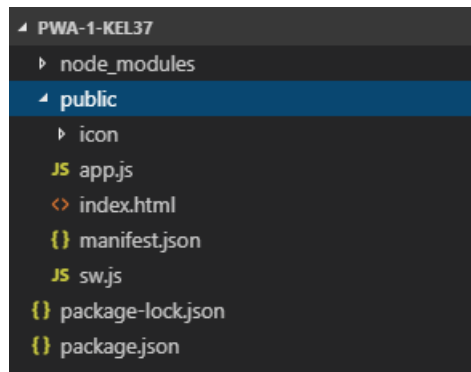


Gambar 5.17 Halaman *web* setelah ditambahkan *service worker*

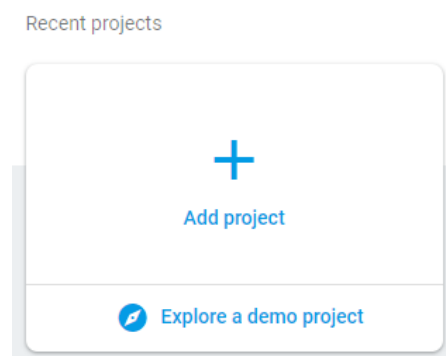
17. Setelah kita menambahkan *service worker* dan *manifest*, maka halaman *web* tetap bejalan.

5.4.2. Deploy Halaman Website ke Firebase

1. Setelah berhasil di langkah sebelumnya, pada praktikum ini kita *deploy* dengan menggunakan *firebase*.
2. Ubah struktur *folder* proyek kalian menjadi seperti gambar dibawah

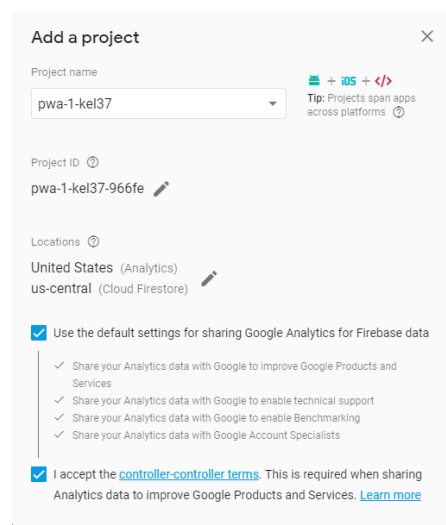
Gambar 5.18 Struktur *folder* yang baru

3. Buka link ini <https://firebase.google.com/> dan jangan lupa *login* terlebih dahulu.
4. Tambahkan proyek baru



Gambar 5.19 Menambahkan proyek baru

5. Nama proyeknya **pwa-1-kel37**



Gambar 5.20 Membuat proyek baru bernama pwa-1-kel37

6. Buka terminal dan masukkan perintah berikut untuk menginstall *firebase*

```
npm install -g firebase-tools
```

```
PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37> npm install -g firebase-tools
C:\Users\ASUS\AppData\Roaming\npm\firebase -> C:\Users\ASUS\AppData\Roaming\npm\node_modules\firebase-tools\lib\bin\firebase.js
+ firebase-tools@6.0.0
updated 1 package in 39.86s
PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37> |
```

Gambar 5.21 Menginstall *firebase*

7. Setelah itu *login* ke *firebase* dengan mengetikan perintah berikut

```
firebase login
```

```
PROBLEMS TERMINAL ... 1: powershell
PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37> firebase login
? Allow Firebase to collect anonymous CLI usage and error reporting information? Yes
S

Visit this URL on any device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgm47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response_type=code&state=1029124400&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...

+ Success! Logged in as ssaskia.al@gmail.com
```

Gambar 5.22 Berhasil *login* ke *firebase* dengan akun Gmail

8. Langkah selanjutnya kita mengetikan perintah **firebase init** , untuk menginisialisasi proyek yang akan *dideploy* ke *firebase*

```
PROBLEMS TERMINAL ... 1: node
+ Success! Logged in as ssaskia.al@gmail.com
PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37> firebase init

#####
## ## ## ## ## ## ## ## ## ##
#####
## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ##

You're about to initialize a Firebase project in this directory:

D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37

? Are you ready to proceed? (Y/n) |
```

Gambar 5.23 Menginisialisasi proyek yang akan *dideploy* ke *firebase*

9. Jika tampilan dari *terminal* seperti gambar diatas, pada saat pertanyaan “**Are you ready to proceed ?**” kita ketikan huruf “Y”.

```
You're about to initialize a Firebase project in this directory:

D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37

? Are you ready to proceed? Yes
```

Gambar 5.24 Menyetujui proses inisialisasi

10. Setelah kita menekan huruf “Y”, akan muncul tampilan untuk memilih fitur yang akan digunakan, kita pilih “**Hosting: Configure and Deploy Firebase Hosting sites**” dengan cara menekan “**Space**” setelah itu tekan “**Enter**”

```
? Which Firebase CLI features do you want to setup for this folder? Press Space to
select features, then Enter to confirm your choices.
( ) Database: Deploy Firebase Realtime Database Rules
( ) Firestore: Deploy rules and create indexes for Firestore
( ) Functions: Configure and deploy Cloud Functions
>(*) Hosting: Configure and deploy Firebase Hosting sites
( ) Storage: Deploy Cloud Storage security rules
```

Gambar 5.25 Menentukan fitur yang akan digunakan

11. Setelah itu akan muncul semua proyek yang kita buat pada firebase, kita pilih proyek “**pwa-1-kel37**” terus tekan enter.

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Select a default Firebase project for this directory:
[don't setup a default project]
modul4-21120116140058-susulan (Modul4-21120116140058-susulan)
modul4-21120116140058-susulan2 (modul4-21120116140058-susulan2)
> pwa-1-kel37-966fe (pwa-1-kel37)
[create a new project]
```

Gambar 5.26 Menentukan proyek *firebase* yang akan digunakan

12. Akan ada 3 pertanyaan yaitu:
- *What do you want to use as your public directory ?* kita tekan **enter** saja
 - *Configure as a single-page app (rewrite all urls to /index.html)?* kita tekan **N**
 - *File public/index.html already exists. Overwrite ?* kita tekan **N**

Setelah itu kita tunggu saja.

```
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? No
+ Wrote public/404.html
? File public/index.html already exists. Overwrite? No
i Skipping write of public/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
i Writing gitignore file to .gitignore...

+ Firebase initialization complete!
PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37>
```

Gambar 5.27 Menjawab pertanyaan yang muncul

13. Setelah selesai, kita tuliskan perintah “firebase deploy” pada terminal dan tunggu sampai tampilan terminal seperti pada gambar dibawah.

```
PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37> firebase deploy

=== Deploying to 'pwa-1-kel37-966fe'...

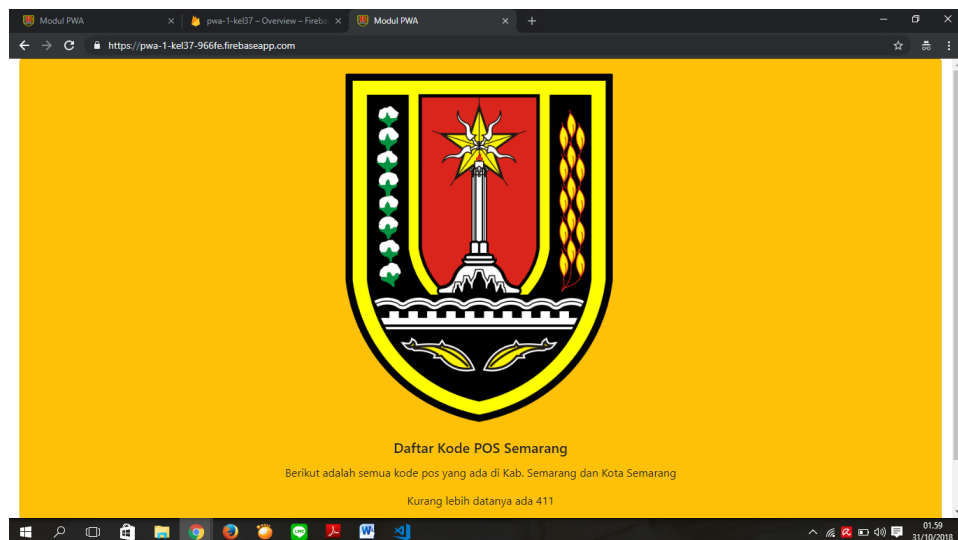
i deploying hosting
i hosting[pwa-1-kel37-966fe]: beginning deploy...
i hosting[pwa-1-kel37-966fe]: found 5 files in public
+ hosting[pwa-1-kel37-966fe]: file upload complete
i hosting[pwa-1-kel37-966fe]: finalizing version...
+ hosting[pwa-1-kel37-966fe]: version finalized
i hosting[pwa-1-kel37-966fe]: releasing new version...
+ hosting[pwa-1-kel37-966fe]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/pwa-1-kel37-966fe/overview
Hosting URL: https://pwa-1-kel37-966fe.firebaseio.com
PS D:\SISKOM\Praktikum MDP\Modul 4\pwa-1-kel37>
```

Gambar 5.28 Deploy proyek ke *firebase*

14. Setelah berhasil, akan muncul link, di *Hosting URL*, kita langsung menuju ke halaman tersebut, maka kita sudah berhasil mendeploy halaman *web* yang kita buat.



Gambar 5.29 Tampilan halaman *web* yang telah berhasil dideploy ke *firebase*

5.4.3. Website ke Mobile

Untuk menjadikan halaman *website* ini menjadi aplikasi *mobile* caranya yaitu:

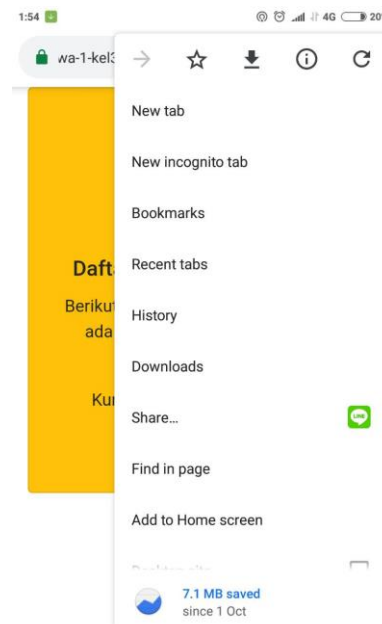
1. Buka *Smartphone* kalian yang sudah diinstal aplikasi *google chrome*

2. Buka alamat *hosting* yang sudah dibuat di *google chrome*



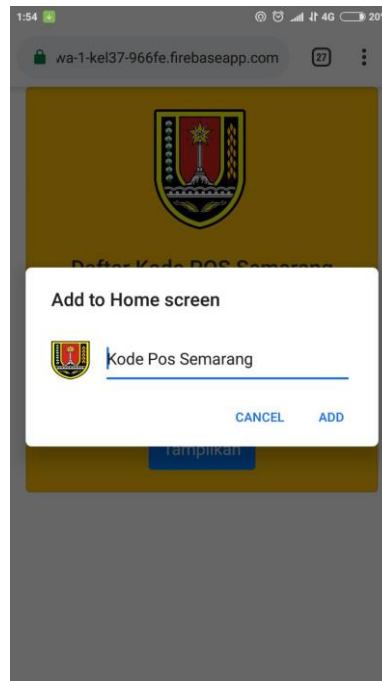
Gambar 5.30 Tampilan halaman *web hosting* pada *smartphone*

3. Setelah itu klik tombol kanan pojok pada *google chrome*, terus pilih “Tambahkan ke Layar Utama”



Gambar 5.31 Add Webpage to Home Screen

4. Akan muncul *pop up*, kita pilih tambahkan, maka aplikasi ini akan muncul pada *homescreen* di *smartphone* kalian, silahkan dicoba

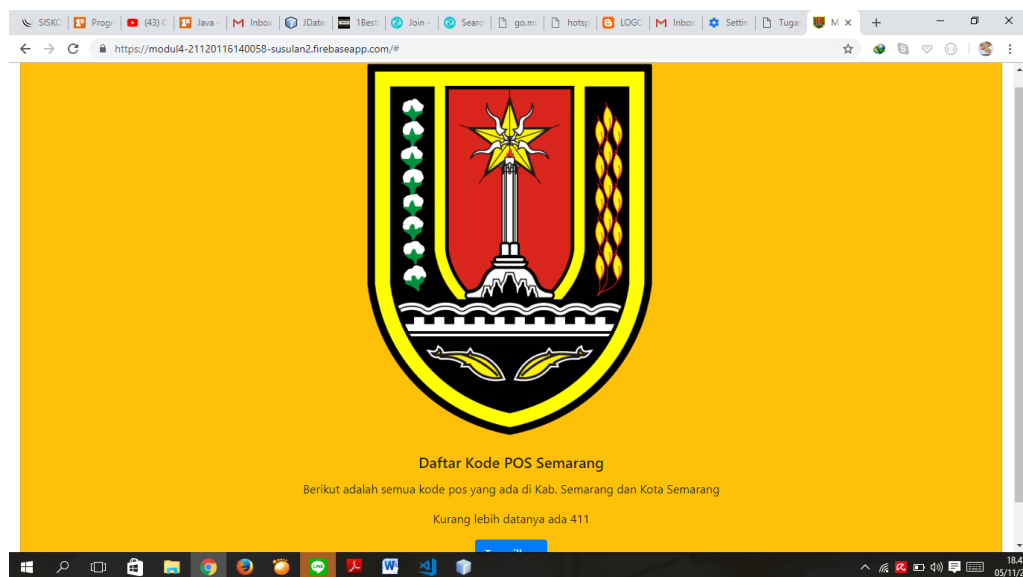


Gambar 5.32 *Pop Up* yang muncul untuk menambahkan ke layar utama

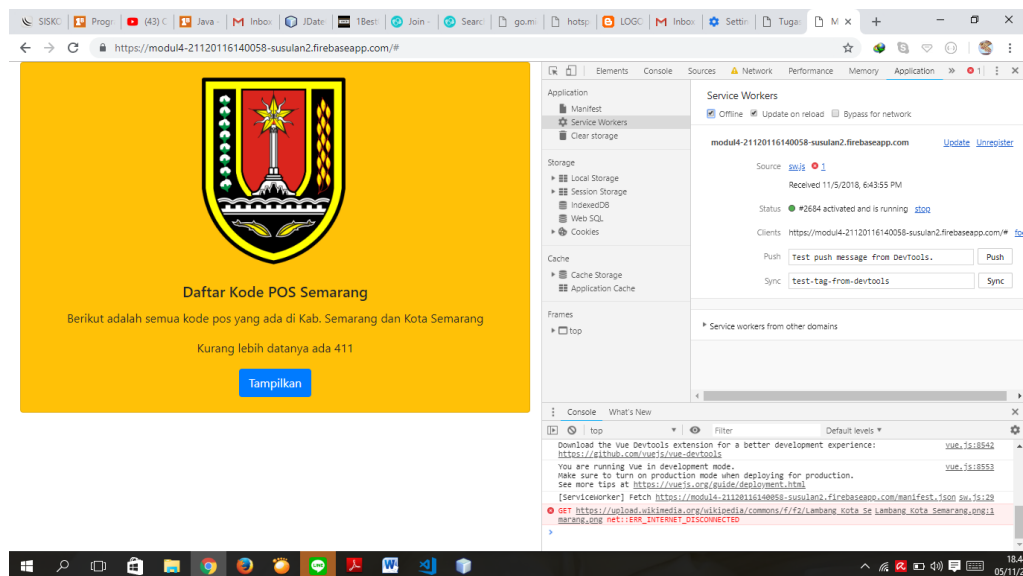


Gambar 5.33 Tampilan *shortcut webpage* pada layar utama

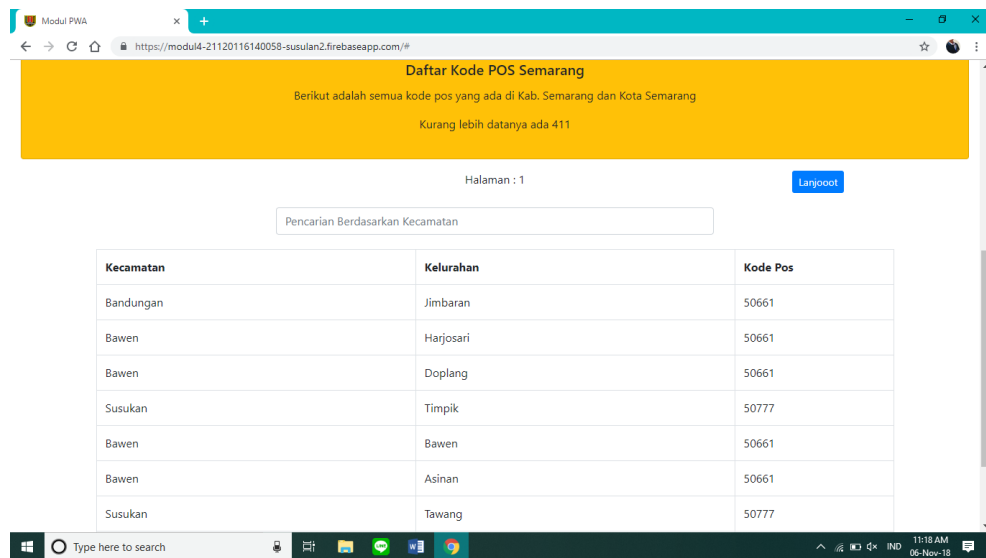
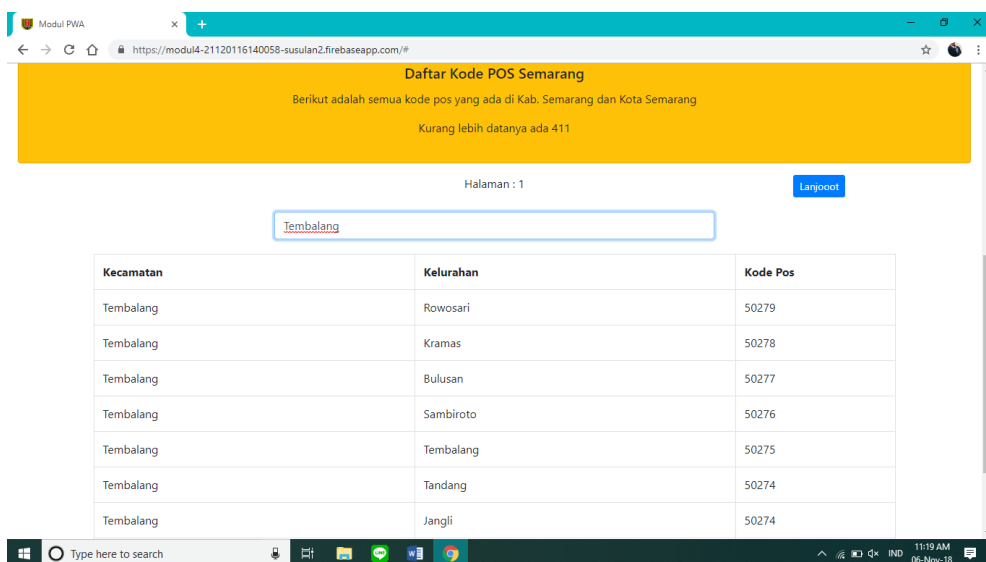
5.5 Analisa Hasil Percobaan



Gambar 5.34 Tampilan *project* pada web

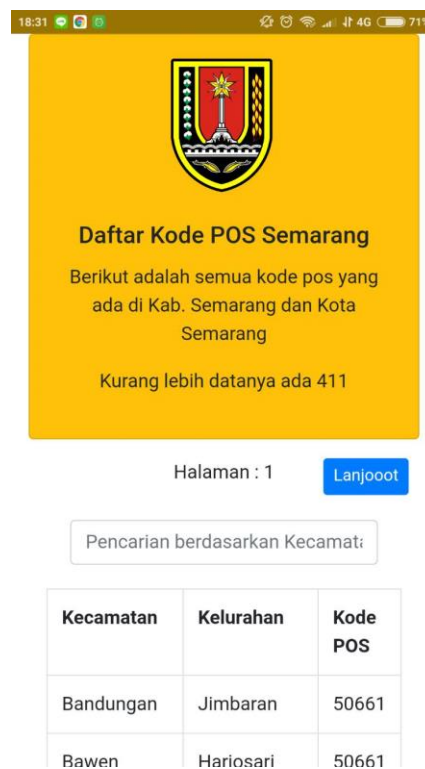


Gambar 5.35 Tampilan *project* pada web saat kondisi *Offline*

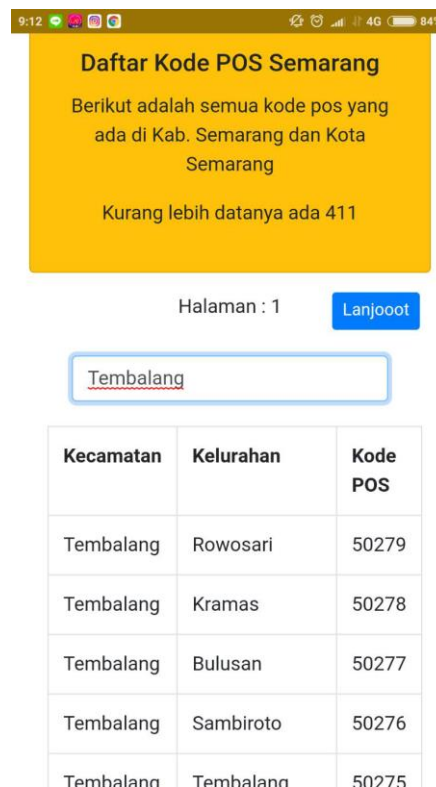
Gambar 5.36 Tampilan *project* setelah tombol Tampilkan ditekanGambar 5.37 Tampilan *project* saat melakukan pencarian data



Gambar 5.38 Tampilan *splash screen project* pada *mobile*



Gambar 5.39 Tampilan *project* pada *mobile*



Gambar 5.40 Tampilan *project* pada saat melakukan pencarian pada *mobile*

```
{
  "name": "Pos Semarang",
  "short_name": "Kode Pos Semarang",
  "theme_color": "#ffc107",
  "background_color": "#ffc107",
  "display": "standalone",
  "orientation": "portrait",
  "scope": ".",
  "start_url": ".",
  "icons" : [
    {
      "src": "/icon/32.ico",
      "sizes": "32x32",
      "type": "image/ico"
    },
    {
      "src": "/icon/96.ico",
      "sizes": "96x96",
      "type": "image/ico"
    },
    {
      "src": "/icon/128.ico",
      "sizes": "128x128",
      "type": "image/ico"
    },
    {
      "src": "/icon/192.ico",
      "sizes": "192x192",

```



```

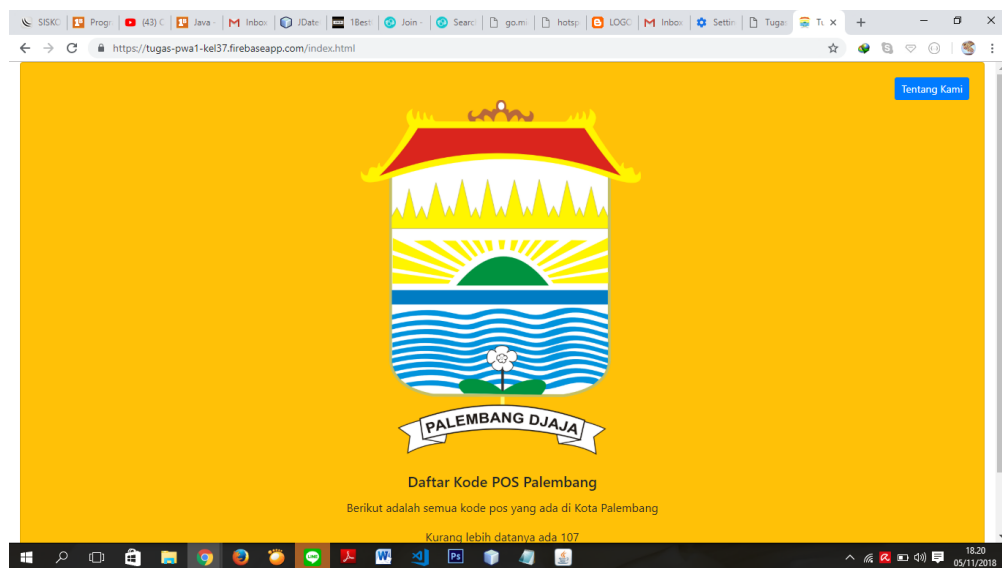
        "type": "image/ico"
      },
      {
        "src": "/icon/256.ico",
        "sizes": "256x256",
        "type": "image/ico"
      }
    ]
  }
}

```

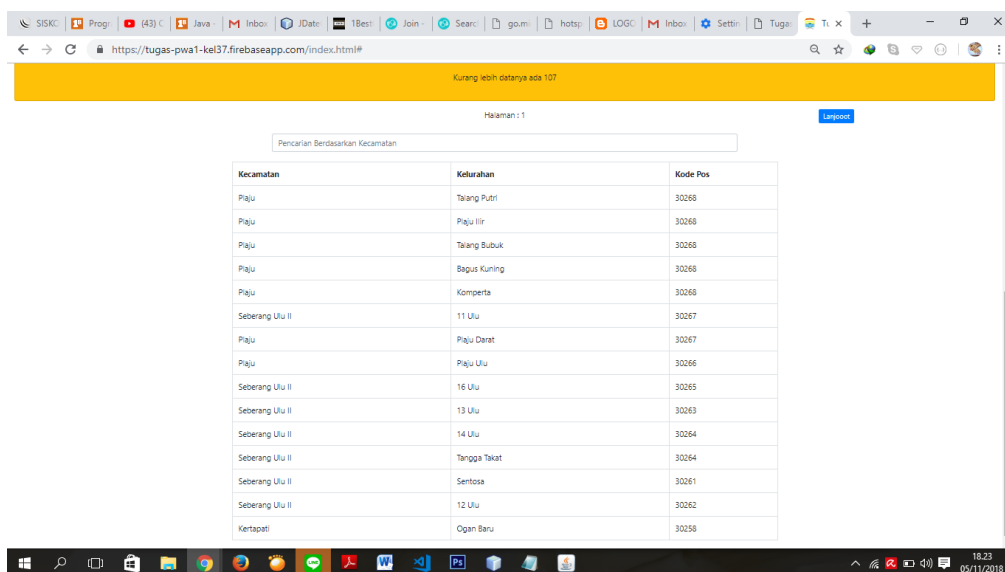
Manifest.json digunakan untuk menjadikan link yang telah kita buat, menjadi aplikasi pada handphone. Jika kita membuat *progressive web apps*, kita membutuhkan manifest.json, manifest.json merupakan file json dengan semua kebutuhan informasi yang dibutuhkan untuk menyimpan web app ke home screen handphone pengguna. Pada file manifest.json berisi mengenai *user experience* seperti splash screen, theme color, icon, dan lainnya. Pada splash screen diberi nama Pos Semarang, judulnya Kode Pos Semarang, theme color dan background color adalah #ffc107 (warna kuning), orientasi yang digunakan adalah portrait, lalu memanggil icon-icon yang akan digunakan, disini menggunakan 5 icon.

<https://modul4-21120116140058-susulan2.firebaseio.com/#>

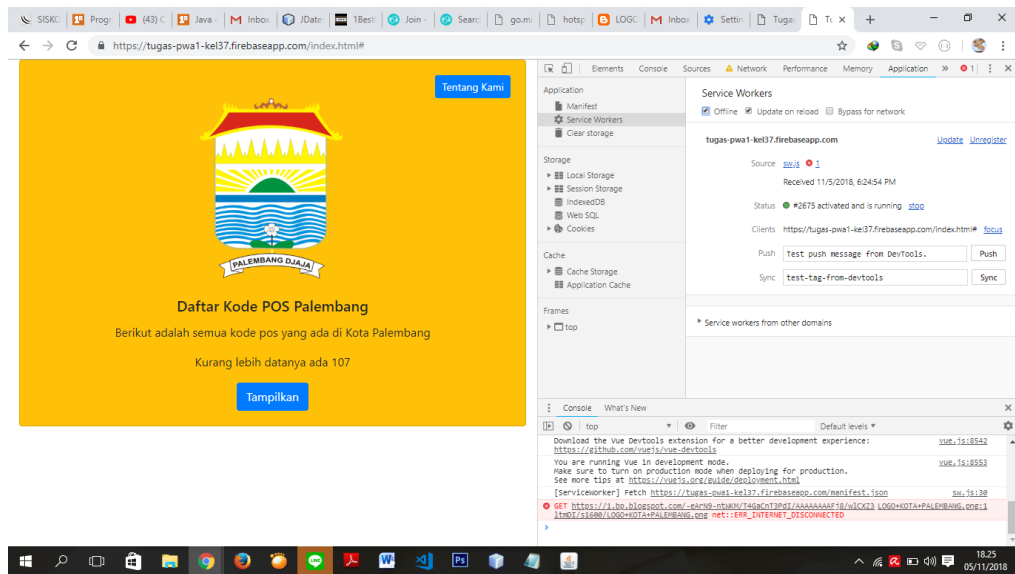
5.6 Tugas



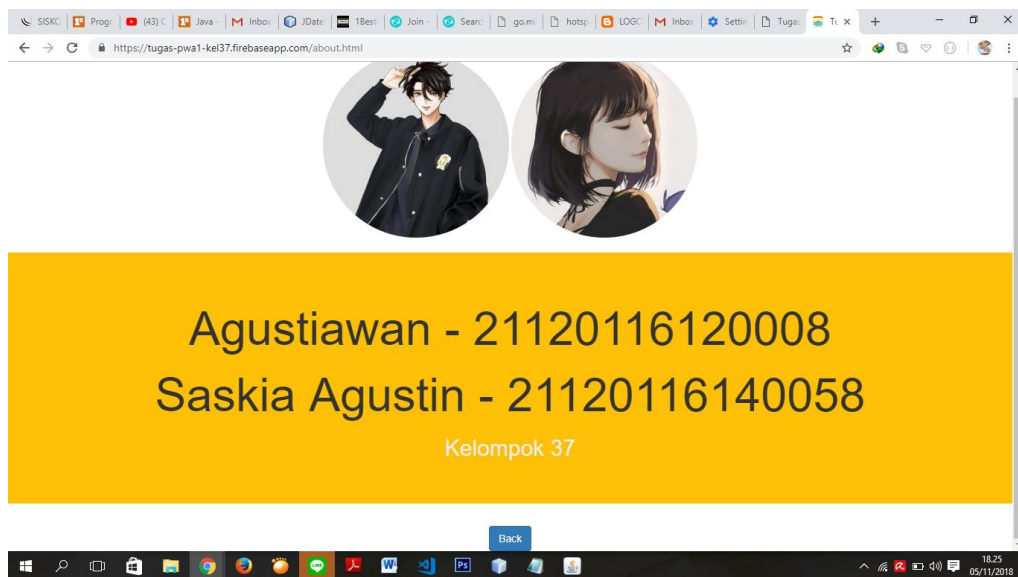
Gambar 5.41 Tampilan *project* tugas pada *web*



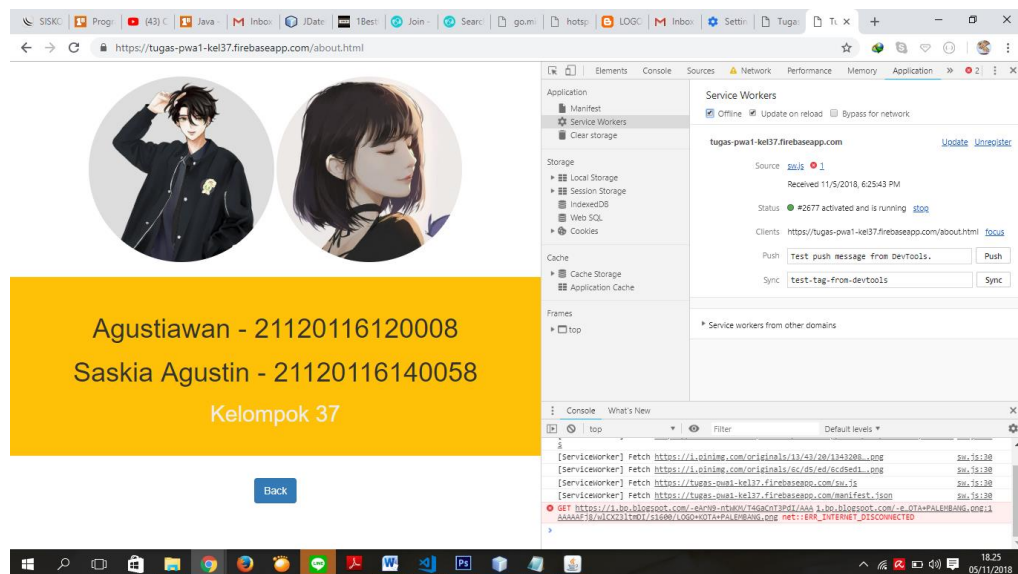
Gambar 5.42 Tampilan *project* tugas pada *web* daftar kode pos berjumlah 15



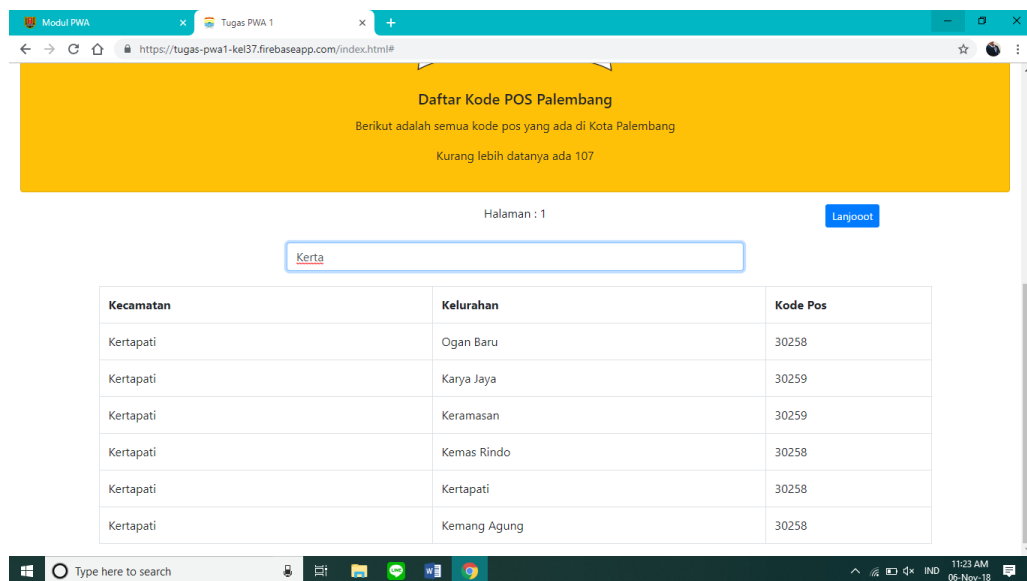
Gambar 5.43 Tampilan *project* tugas pada web saat *Offline*



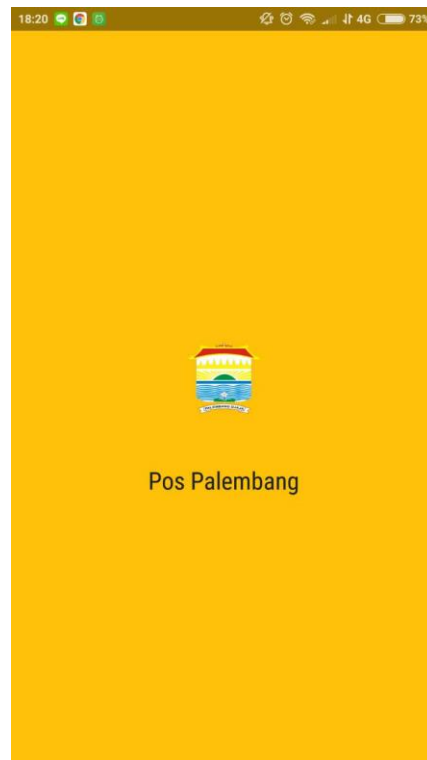
Gambar 5.44 Tampilan *project* tugas halaman *About* pada web



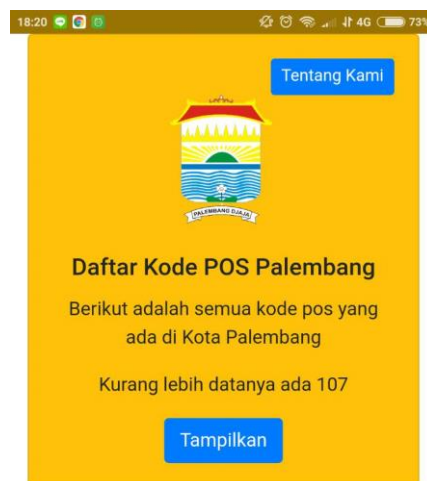
Gambar 5.45 Tampilan *project* tugas halaman *About* pada *web* saat *Offline*



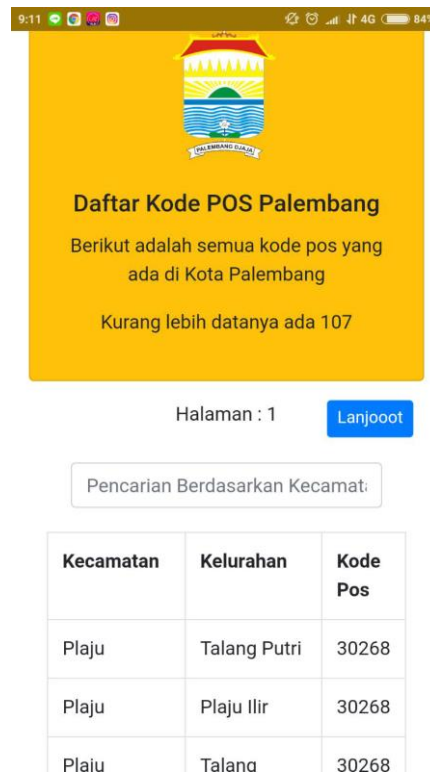
Gambar 5.46 Tampilan *project* saat melakukan pencarian pada *web*



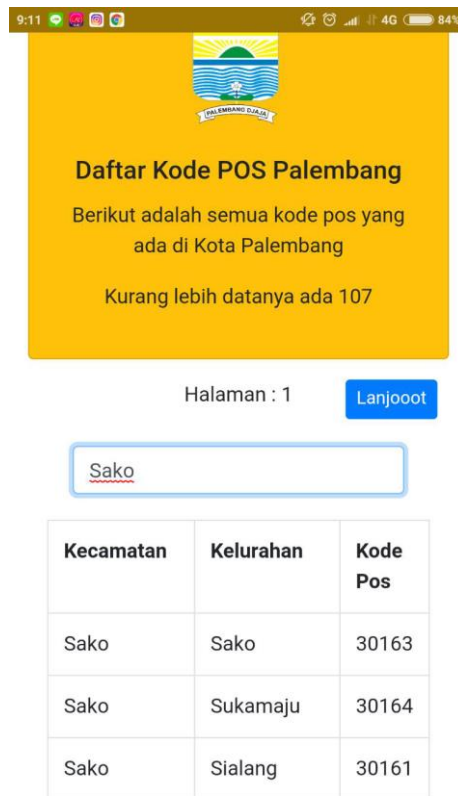
Gambar 5.47 Tampilan *splash screen project tugas* pada *mobile*



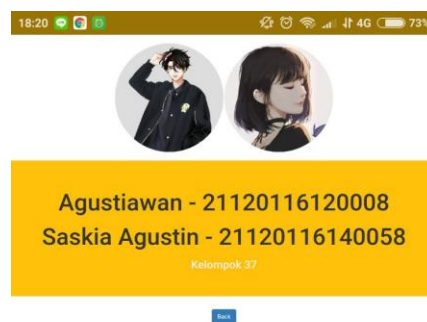
Gambar 5.48 Tampilan *project tugas* pada *mobile*



Gambar 5.49 Tampilan *project* tugas pada *mobile*



Gambar 5.50 Tampilan *project* tugas saat melakukan pencarian pada *mobile*



Gambar 5.51 Tampilan *project* tugas halaman *About* pada *mobile*

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Tugas PWA 1</title>
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <link rel="manifest" href="manifest.json">
  <link
                                rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstr
ap.min.css">
  <link rel="shortcut icon" href="icon/192.ico" type="image/x-
icon" />
  <link rel="icon" href="https://1.bp.blogspot.com/-eArN9-
ntWKM/T4GaCnT3PdI/AAAAAAAAAFj8/wlCXZ3ltmDI/s1600/LOGO+KOTA+PALEMBA
NG.png" sizes="32x32" />
  <!-- buat manifest json -->
</head>
<body>

<!-- untuk program utama -->
<div class="container-fluid">
  <div id="app" >
    <center>
      <div class="card bg-warning mb-3">
        <div class="card-body">
          <div align="right">
            <a href="about.html" class="btn btn-primary
btn-sm">Tentang Kami</a>
          </div>
          
          <br><br>
          <h5 class="card-title">Daftar Kode POS
Palembang</h5>
          <p class="card-text">Berikut adalah semua kode
pos yang ada di Kota Palembang</p>
          <p>Kurang lebih datanya ada
{{kodepos.length}}</p>
          <a href="#" v-on:click="tampil = !tampil" v-
show="tampil" class="btn btn-primary">Tampilkan</a>
        </div>
      </div>
      <div v-show="!tampil">
        <div class="row">
          <div class="col-sm-4 col-md-4 col-lg-4 col-4">
            <button class="btn btn-primary btn-sm" v-
show="awal>0" v-on:click="prev()">Balek Bae</button>
          </div>
          <div class="col-sm-4 col-md-4 col-lg-4 col-4">
            <span>Halaman : {{halaman}}</span>
          </div>
        </div>
      </div>
    </center>
  </div>
</div>

```



```

<div class="col-sm-4 col-md-4 col-lg-4 col-4">
  <button class="btn btn-primary btn-sm" v-
show="akhir" <= kodepos.length" v-
on:click="next()">Lanjoooot</button>
</div>
</div>
<div class="row" style="margin:20px">
  <div class="col-sm-3"></div>
  <div class="col-sm-6">
    <input type="text" v-model="search"
class="form-control" placeholder="Pencarian Berdasarkan
Kecamatan">
  </div>
</div>
</div>
</center>
<div v-show="!tampil" class="table-responsive container">
  <table id="example" class="table table-hover table-
bordered">
    <tr>
      <th>Kecamatan</th>
      <th>Kelurahan</th>
      <th>Kode Pos</th>
    </tr>
    <tr v-for="pos in cari.slice(awal,akhir)">
      <td style="width:40%">{{pos.kecamatan}}</td>
      <td style="width:40%">{{pos.kelurahan}}</td>
      <td style="width:20%">{{pos.kodepos}}</td>
    </tr>
  </table>
</div>
</div>
</body>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.mi
n.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/
popper.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap
.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script
src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script src="app.js"></script>
<script src="sw.js"></script>
<!-- untuk service worker -->
</html>

```

Index.html berisi sintaks-sintaks yang mengatur tentang segalanya yang akan ditampilkan saat *link* diakses. Pada *file* index.html ini di bagian *Head* ditentukan

title saat *link project* diakses yaitu Tugas PWA 1. Kemudian diatur *icon* yang akan muncul yaitu gambar yang sesuai dengan *link* yang dipanggil, dimasukkan juga *manifest.json* agar dapat berjalan pada saat halaman awal ini diakses. Pada bagian *body* diatur susunan serta tampilan yang diinginkan. Dibuat *button* “About” yang apabila diklik akan mengakses halaman *about.html* terletak di sebelah kanan. Pada tampilan awal ini dimasukkan gambar lambang dari Kota Palembang sesuai dengan *link* yang dimasukkan dengan pengaturan besar 30%, terletak di bagian atas. Berikutnya ada tulisan ‘Daftar Kode Pos Palembang’ di bawah gambar, lalu di bawahnya ada tulisan lagi yaitu ‘Berikut adalah semua kode pos yang ada di Kota Palembang’, di bawahnya lagi tertulis ‘Kurang lebih datanya ada ‘ yang di akhirnya terdapat jumlah data kodepos yang ada. Di bawah tulisan tersebut ada *button* tampilkan yang apabila diklik akan memunculkan tabel daftar kode pos yang ada. Setelah tabel dimunculkan akan ada keterangan halaman berapa yang tengah ditampilkan, *search box* untuk melakukan pencarian berdasarkan kecamatan, terdapat *button* Lanjoooot untuk pindah ke halaman berikutnya, dan akan muncul *button* Balek Bae di halaman setelah halaman 1 untuk kembali ke halaman sebelumnya. Tabel diatur memiliki 3 kolom yaitu, Kecamatan, Kelurahan, dan Kode Pos. Pada bagian akhir dilakukan pemanggilan beberapa *link* dan file *app.js* serta *sw.js* untuk mengkoneksikan halaman *index.html* dengan *link* dan isi dari file *app.js* serta *sw.js* untuk mengaktifkan *service worker* agar isi *link* tetap tampil saat koneksi dimatikan.

```
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('/sw.js')
    .then(function () {
      console.log('SW registered');
    });
}
new Vue({
  el : '#app',
  data:{
    kodepos : [],search : '', awal : 0, akhir : 15, halaman
: 1,tampil:true
  },
  created(){
    this.getData();
    console.log(this.awal)
  },
  methods:{
    next(){
```

```

        this.awal = this.awal+=15
        this.akhir = this.akhir+=15
        this.halaman = this.halaman+=1
    },
    prev() {
        this.awal = this.awal -= 15
        this.akhir = this.akhir -= 15
        this.halaman = this.halaman -= 1
    },
    getData(){
        var kd = this;
        axios.get('https://kodepos-
2d475.firebaseio.com/kota_kab/k442.json?print=pretty').then(function(response){
            kd.kodepos = response.data;
        })
    }
},
computed:{
    cari(){
        return this.kodepos.filter(pos =>{
            return
pos.kecamatan.toLowerCase().match(this.search.toLowerCase());
        });
    }
}
});

```

App.js mengatur pemanggilan-pemanggilan data yang akan dimasukkan dalam *project* ini. Diaktifkan *service worker* dengan pemanggilan file *sw.js*. Data yang akan ditampilkan pada tabel yang diatur pada *index.html* diatur akan menampilkan sebanyak 15 data kode pos saja pada setiap halaman, kecuali di bagian akhir nantinya. Data kode pos yang ditampilkan dari data pada https://kodepos-2d475.firebaseio.com/kota_kab/k442.json?print=pretty. Kemudian untuk fungsi cari ini merupakan fungsi pada *search box* pada halaman *index.html* diatur agar mencari kodepos berdasarkan kecamatannya.

```

{
  "name": "tugas-pwa1-kel37",
  "version": "1.0.0",
  "description": "Praktikum MDP PWA Jilid 1",
  "main": "index.js",
  "scripts": {
    "start": "http-server -c-1 -p 8888"
  },
  "keywords": [
    "pwa"
  ],
  "author": "Asisten MDP PWA 1",
  "license": "UNLICENSED",
  "devDependencies": {
    "http-server": "^0.10.0"
  }
}

```

```
}
}
```

Package.json merupakan deskripsi mengenai *project* yang akan kita buat. Berisi nama *project*, versi, deskripsi *project*, dan halaman utama yang diakses yaitu *index.js*, serta tertera *author* dari *project* yaitu, Asisten MDP PWA 1.

```
{
  "name": "Pos Palembang",
  "short_name": "Kode Pos Palembang",
  "theme_color": "#ffc107",
  "background_color": "#ffc107",
  "display": "standalone",
  "orientation": "portrait",
  "scope": ".",
  "start_url": ".",
  "icons" : [
    {
      "src": "/icon/32.ico",
      "sizes": "32x32",
      "type": "image/ico"
    },
    {
      "src": "/icon/96.ico",
      "sizes": "96x96",
      "type": "image/ico"
    },
    {
      "src": "/icon/128.ico",
      "sizes": "128x128",
      "type": "image/ico"
    },
    {
      "src": "/icon/192.ico",
      "sizes": "192x192",
      "type": "image/ico"
    },
    {
      "src": "/icon/256.ico",
      "sizes": "256x256",
      "type": "image/ico"
    }
  ]
}
```

Manifest.json digunakan untuk menjadikan link yang telah kita buat, menjadi aplikasi pada handphone. Jika kita membuat *progressive web apps*, kita membutuhkan manifest.json, manifest.json merupakan file json dengan semua kebutuhan informasi yang dibutuhkan untuk menyimpan web app ke home screen handphone pengguna. Pada file manifest.json berisi mengenai *user experience* seperti splash screen, theme color, icon, dan lainnya. Pada splash screen diberi

nama Pos Palembang, judulnya Kode Pos Palembang, theme color dan background color adalah #ffc107 (warna kuning), orientasi yang digunakan adalah portrait, lalu memanggil icon-icon yang akan digunakan, disini menggunakan 5 icon.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Tugas PWA Modul 1</title>
    <link rel="manifest" href="manifest.json">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="icon" href="https://1.bp.blogspot.com/-eArN9-ntWKM/T4GaCnT3PdI/AAAAAAAAAFj8/wlCXZ3ltmDI/s1600/LOGO+KOTA+PALEMBANG.png" sizes="32x32" />
    <style type="text/css">
      img {
        height: 250px;
        width: 250px;
        margin-top: 20px;
        margin-bottom: 20px;
      }
      .jumbotron {
        background-color: #ffc107;
      }
      .jumbotron * {
        color: #eee;
      }
    </style>
  </head>
  <body>

    <div class="text-center">
      
      
    </div>

    <div class="jumbotron text-center">
      <h1>Agustiawan - 21120116120008</h1>
      <h1>Saskia Agustin - 21120116140058</h1>
      <h2>Kelompok 37</h2>
    </div>

    <div class="text-center">
      <a href="index.html" class="btn btn-primary">Back</a>
    </div>

  </body>
</script src="sw.js"></script>
</html>
```

About.html merupakan halaman yang memiliki fungsi yang sama dengan index.html. Pada *file* ini diatur halaman yang akan menampilkan tentang praktikan dari kelompok 37 praktikum MDP 2018. Memiliki *title head* yaitu, Tugas PWA Modul 1 dengan *icon* berupa lambang kota Palembang yang dipanggil berdasarkan *link* yang telah dimasukkan dengan ukuran *height*, *width*, *margin-top*, dan *margin-bottom* yang telah diatur serta pemilihan warna *background* juga diatur di dalamnya. Pada bagian *body* diatur pada posisi tengah yaitu, gambar yang melambangkan kedua praktikan dari kelompok 37. Berikutnya di bawah gambar diatur tulisan besar pada bagian tengah yaitu, nama dan NIM dari praktikan. Di bawahnya terdapat *button* 'back' yang berfungsi untuk kembali ke halaman index.html. Pada bagian akhir dimasukkan sintaks `<script src="sw.js"></script>` untuk mengaktifkan *service worker* agar halaman tetap dapat tampil walau koneksi mati.

<https://tugas-pwa1-kel37.firebaseio.com/index.html>

5.7 Kesimpulan

1. *Service Worker* berfungsi untuk menyimpan *cache* sehingga *project* tetap dapat ditampilkan ketika koneksi mati.
2. *Vue.js* digunakan untuk membangun antarmuka.
3. *Manifest.json* berfungsi untuk menyimpan *web app* ke *home screen* *handphone* pengguna.
4. *Firebase* digunakan untuk memberi *hosting* pada *project* yang telah dibuat.
5. *Node.js* digunakan untuk menjalankan *web server*.
6. *PWA* berguna untuk membuat aplikasi pada *web* dan dapat *dideploy* pada *Android* dengan penyajian antarmuka yang bagus.
7. Untuk beberapa kasus *project* harus diakses dengan *Google Chrome* dalam mode penyamaran.
8. *Link* yang dimasukkan sebaiknya dalam *https*, karena terkadang terjadi *error* saat menggunakan *http*.