**BAB V**

**Progressive Web App (PWA) #1**

* 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 membuat aplikasi PWA sederhana.
   1. **Alat dan Bahan**
      1. Laptop
      2. Smartphone
      3. Aplikasi Google Chrome (Laptop dan Smartphone)
      4. Text Editor (saran menggunakan Visual Studio Code)
      5. Node.js
      6. Akun Google aktif, untuk daftar firebase.
   2. **Dasar Teori**
      1. **Progressive Web App (PWA)**

PWA adalah sebuah aplikasi web yang menggunakan beberapa teknologi modern yang membuat user experience seperti menggunakan Native App (Android/iOS).

Dengan konsep offline first dan Web APIs baru, aplikasi yang berjalan pada browser tidak akan memunculkan pesan network error atau white screen seperti yang selama ini seringkali kita alami ketika berada pada jaringan yang sangat buruk. Pada PWA, ketika kita meninggalkan browser, dan kembali lagi membuka aplikasi tersebut pada jaringan buruk atau tanpa jaringan sekalipun kita tidak akan kehilangan UI dan data-data terakhir**.**

* + 1. **Web Server**

Web server adalah sebuah software yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada klien yang dikenal dan biasanya dikenal 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 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.

* + 1. **Service Worker**

Service Workers adalah teknologi dibalik PWA yang memberikan fungsionalitas offline, notifikasi, update konten, pergantian konektifitas dan lainnya. Service workers adalah semacam proxy yang berada ditengah antara aplikasi web, browser, dan jaringan, atau lebih sederhana lagi bisa dianalogikan aplikasi kita akan diatur oleh service workers agar bisa berlajan offline terlebih dahulu, mengembalikan data dari cache jika request ke jaringan gagal.

* + 1. **Web App Manifest**

Manifes aplikasi web adalah file JSON sederhana yang memberikan mendefinisikan bagaimana aplikasi terlihat oleh pengguna (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.

* Situs akan memiliki ikon dan nama yang unik sehingga pengguna bisa membedakannya dari situs yang lain seperti halnya native app.
* 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.

Semua ini dilakukan melalui mekanisme sederhana dari metadata dalam file teks. Itulah manifes aplikasi web.

* + 1. **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.

* + 1. **Node Package Management**

NPM merupakan salah satu jenis application-level package manager untuk JavaScript yang popular digunakan. Seperti dalam PHP kita mengenal Composer, di Ruby ada RubyGems, di Python kita mengenal pip, dan seterusnya. Seperti halnya package manager lain, npm menyediakan fiture auto consume dengan mendownload library dari repository dan menaruhnya ke dalam folder node\_modules.

Dengan menggunakan npm, developer dapat mencari, mendapatkan, menggunakan dan berbagi package-package/modul-modul JavaScript. Modul-modul yang ada dapat dengan mudah digunakan yakni dengan menyertakan modul yang dibutuhkan sebagai dependency dari project yang kita buat di dalam suatu file yang dituliskan dengan format JSON misalnya package.json.

* + 1. **Vue.js**

Vue.js adalah suatu librari Javascript yang digunakan untuk membangun antar muka sebuah website yang interaktif. Supaya lebih singkat, untuk selanjutnya Vue.js saya tulis Vue. Library dari Vue difokuskan hanya pada view layer dan sangat mudah diimplementasikan dan diintegrasikan dengan library lain ataupun juga dengan project yang sudah ada sebelumnya. Untuk mempelajari Vue ini setidaknya kita sudah mengerti dasar-dasar HTML, CSS dan juga Javascript.

* + 1. **Firebase**

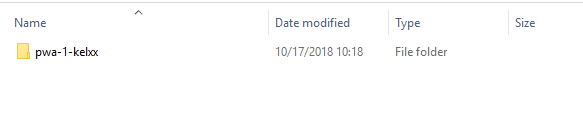
Firebase Hosting dibuat untuk developer web modern. Situs statis kini lebih hebat daripada sebelumnya, dengan hadirnya framework JavaScript front-end seperti Angular dan fitur generator statis seperti Jekyll. Hosting menyediakan infrastruktur, fitur, dan rangkaian alat yang disesuaikan untuk menerapkan dan mengelola situs web statis seperti halaman landing aplikasi yang sederhana atau Progressive App Web yang kompleks.

Hosting menambahkan subdomain ke project di domain firebaseapp.com. Dengan Firebase CLI, file dari direktori lokal di komputer dapat diunggah ke server Hosting. File disajikan melalui koneksi SSL dari server edge terdekat di CDN global yang ada.

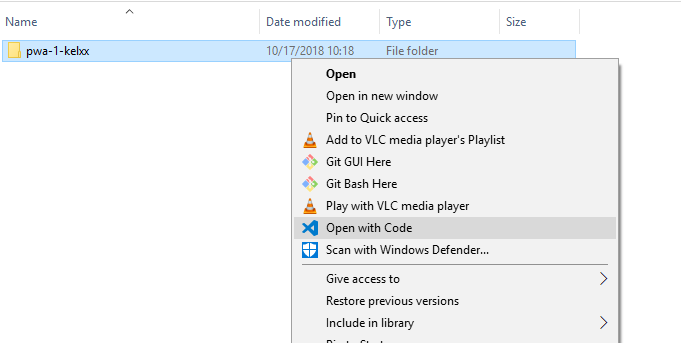
Selain hosting konten statis, Firebase Hosting menyediakan opsi konfigurasi ringan untuk Progressive Web App yang rumit. Dengan dukungan https (SSL) maka web app yang diupload ke firebase dipastikan aman

* 1. **Langkah Percobaan**
     1. **Pembuatan Halaman Website**

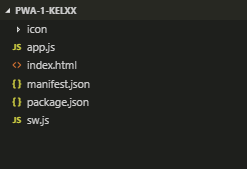
1. Buat folder beri nama “**pwa-1-kelxx**”, untuk tempat foldernya bebas.



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



1. Setelah itu buatlah 5 *file* dan 1 *folder* seperti gambar dibawah



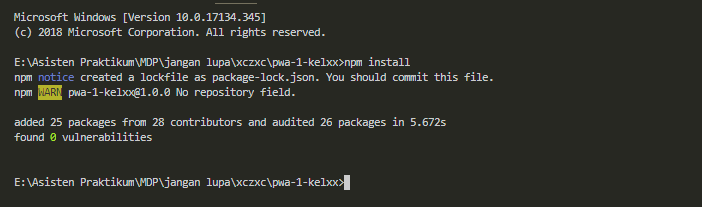
1. Pada file **package.json** diberi kode seperti ini, untuk name diganti dengan kelompok masing-masing.

|  |
| --- |
| {  "name": "pwa-1-kelxx",  "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"  }  } |

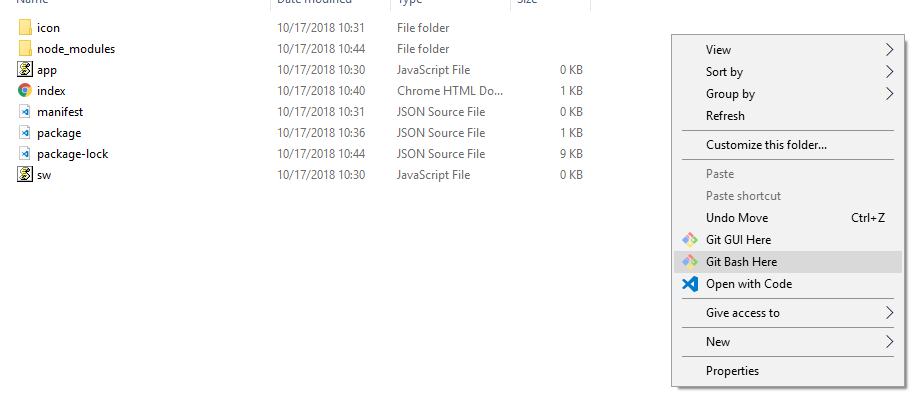
1. Setelah itu, isikan file **index.html** dengan kode berikut

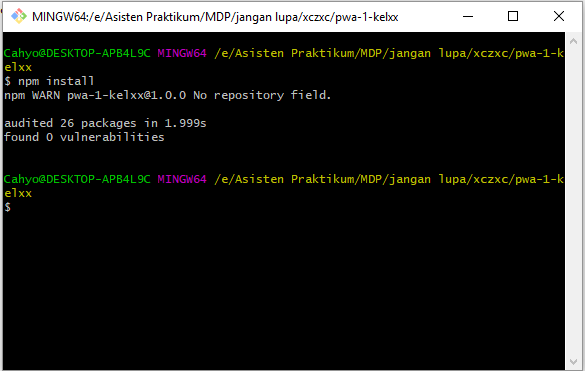
|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <title>Modul PWA</title>  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">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

1. Setelah itu kita buka terminal pada visual studio code, yaitu dengan menekan alt+`, setelah itu kita ketikan perintah **npm install.** Pastikan sudah menginstall package npm di visual studio code. Jika tidak menggunakan visual studio code bisa menggunakan git bash. Pada langkah ini membutukan koneksi internet untuk mengunduh library dari **node.js**.



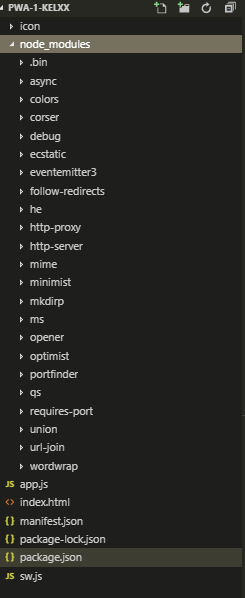
Menggunakan terminal vs code



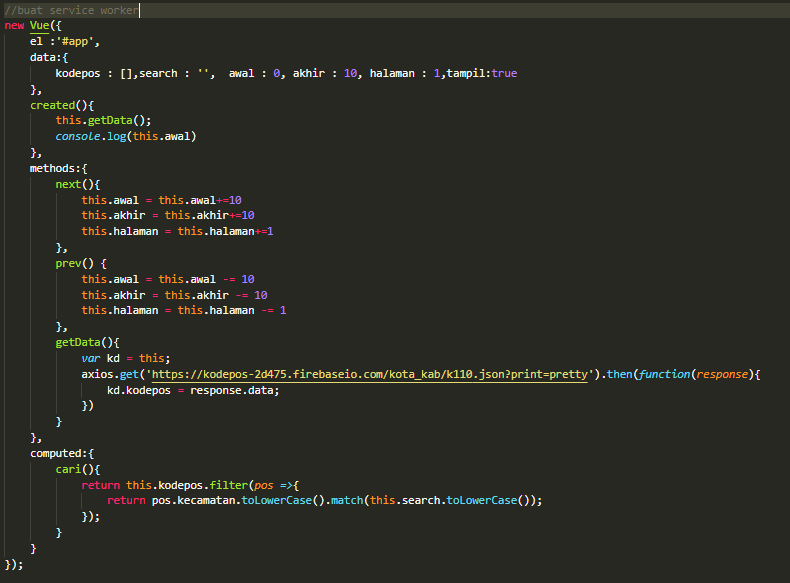


Menggunakn git bash

1. Setelah itu akan muncul folder **node\_module** dan **package-lock.json**



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



Untuk link seperti ini

|  |
| --- |
| https://kodepos-2d475.firebaseio.com/kota\_kab/k110.json?print=pretty |

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

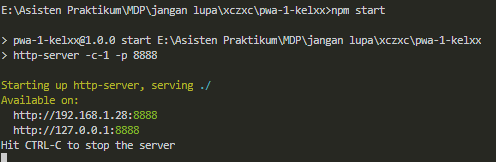




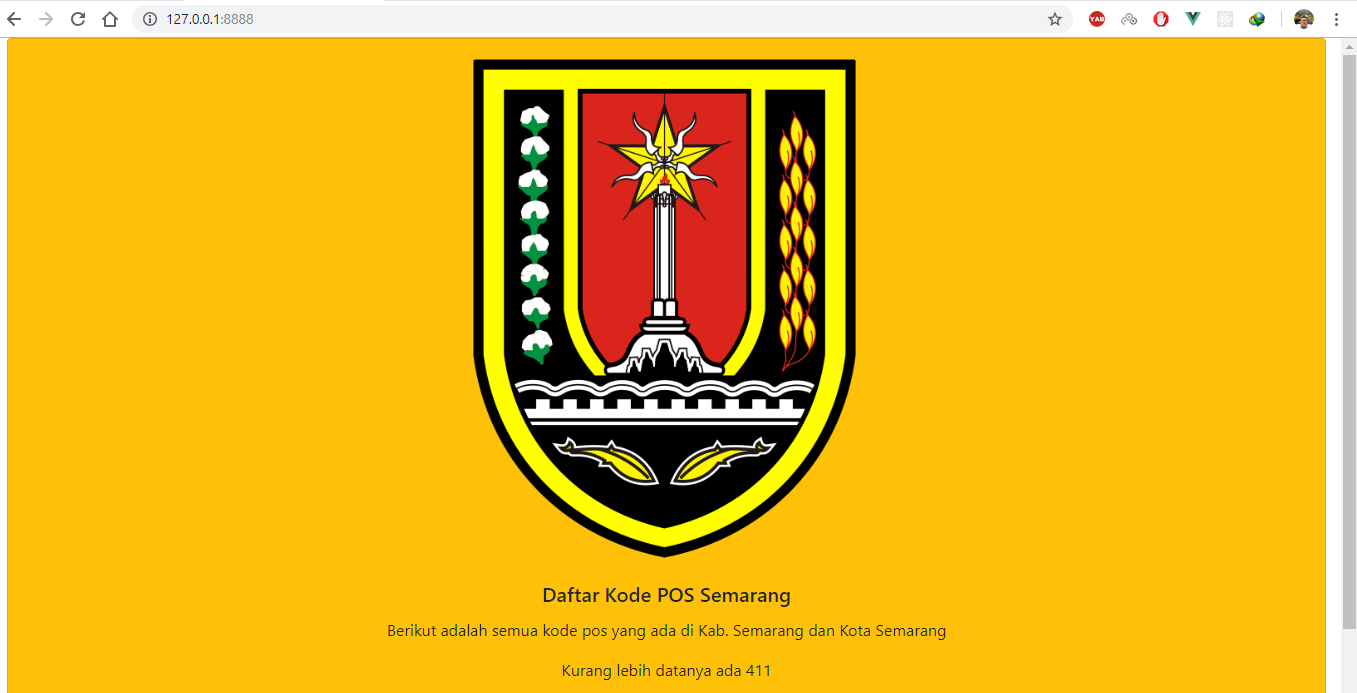
Untuk link seperti ini

|  |
| --- |
| https://upload.wikimedia.org/wikipedia/commons/f/f2/Lambang\_Kota\_Semarang.png |

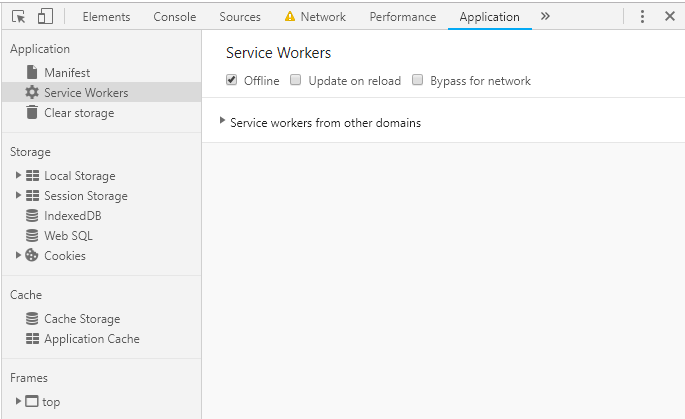
1. Buka terminal kembali, dan ketikkan perintah **npm start**



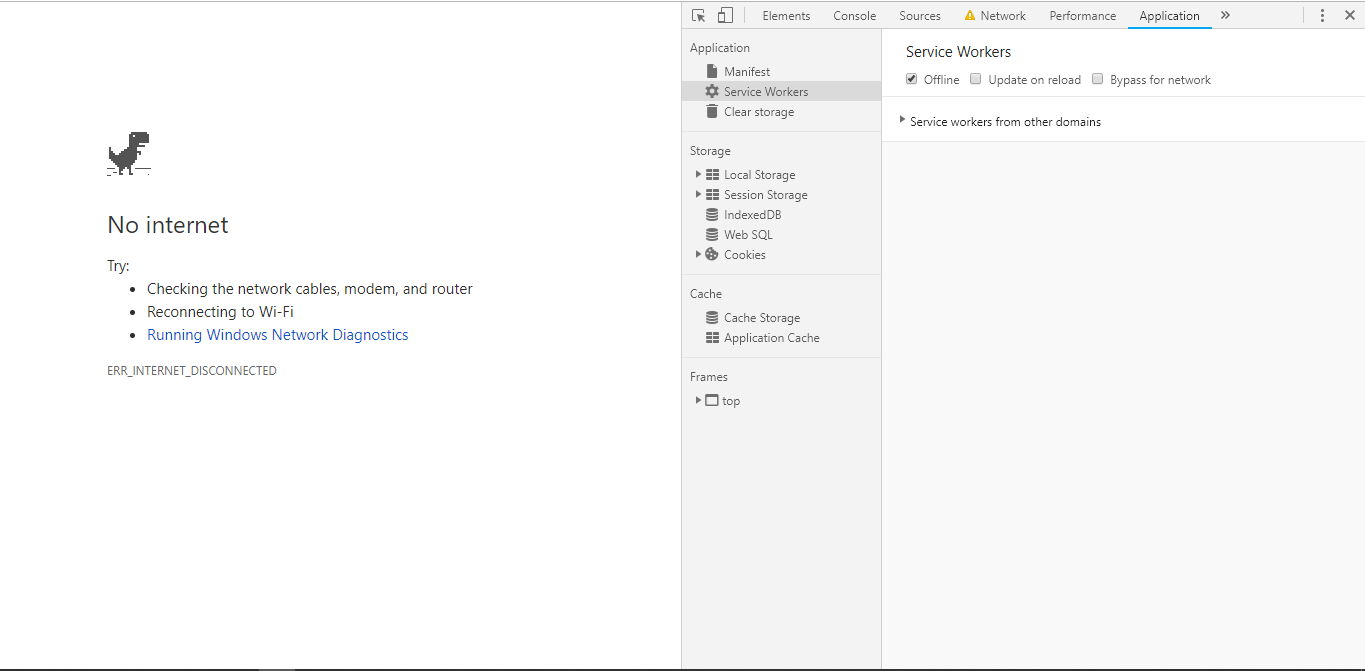
Akan ada 2 link, kita pilih yang 127.0.0.1:8888, setelah itu kita masukan di web browser google chrome. Hasilnya seperti ini



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



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



1. Setelah itu hilangkan centang pada *offline* dan *reload* kembali.
2. 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**

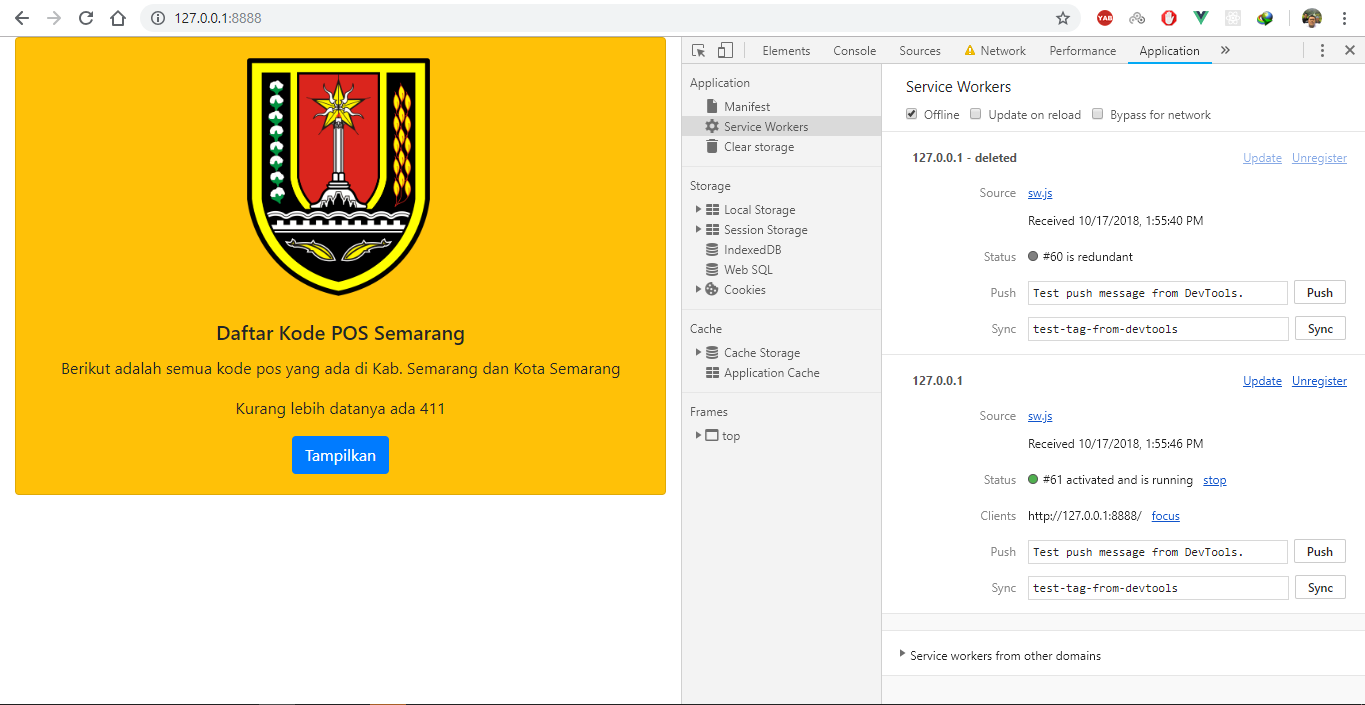
|  |
| --- |
| <link rel="manifest" href="manifest.json"> |
| <script src="sw.js"></script> |

Diletakan sebagaimana terlihat pada **index.html**

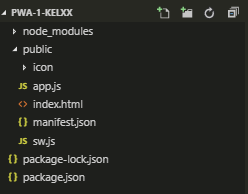
1. Tambahkan kode berikut pada **app.js**



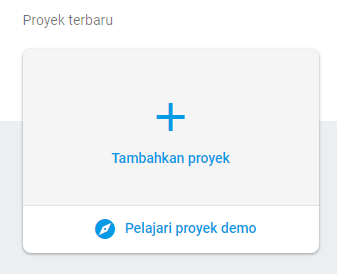
1. Setelah itu di-*reload* terlebih dan buka inspect elemen, dan buka tab *application* dan lakukan seperti langkah 12 dan lihat perbedaan.



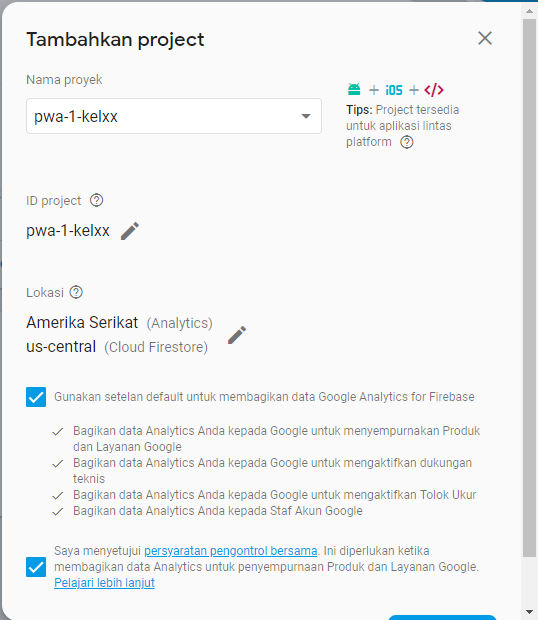
1. Setelah kita menambahkan *service worker* dan *manifest*, maka halaman *web* tetap bejalan.
   * 1. **Deploy Halaman Website ke Firebase**
2. Setelah berhasil di langkah sebelumnya, pada praktikum ini kita *deploy* dengan menggunakan *firebase*.
3. Ubah struktur *folder* proyek kalian menjadi seperti gambar dibawah



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

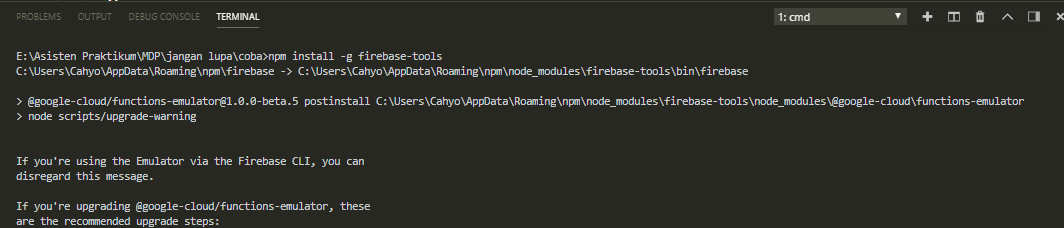


1. Nama proyeknya **pwa-1-kelxx**



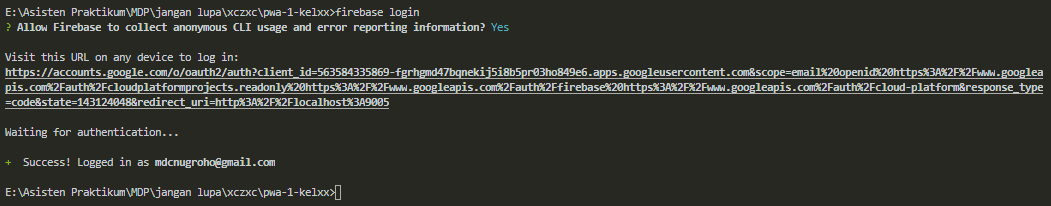
1. Buka terminal dan masukkan perintah berikut untuk menginstall *firebase*

|  |
| --- |
| npm install -g firebase-tools |

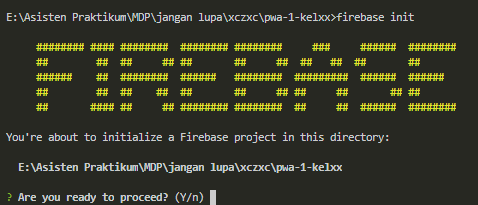


1. Setelah itu *login* ke *firebase* dengan mengetikan perintah berikut

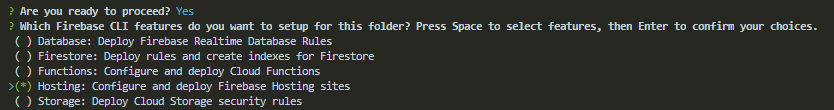
|  |
| --- |
| firebase login |



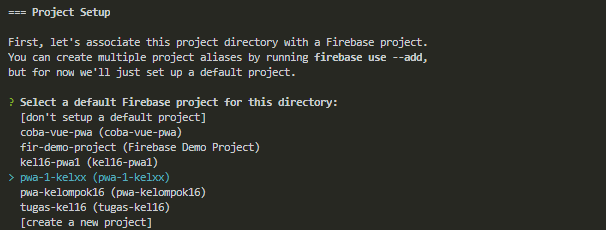
1. Langkah selanjutnya kita mengetikan perintah **firebase init** , untuk menginisialisasi proyek yang akan dideploy ke *firebase*



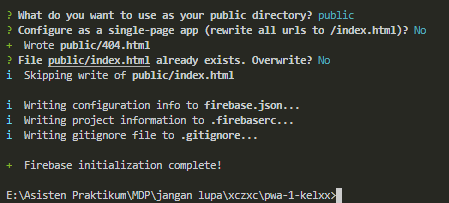
1. Jika tampilan dari terminal seperti gambar diatas, pada saat pertanyaan “**Are you ready to proceed ?”** kita ketikan huruf “Y“.



1. 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**”



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

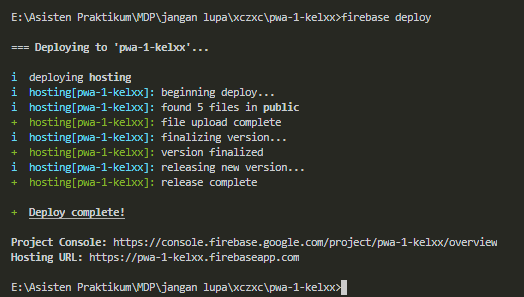


1. Akan ada 3 pertanyaan yaitu:

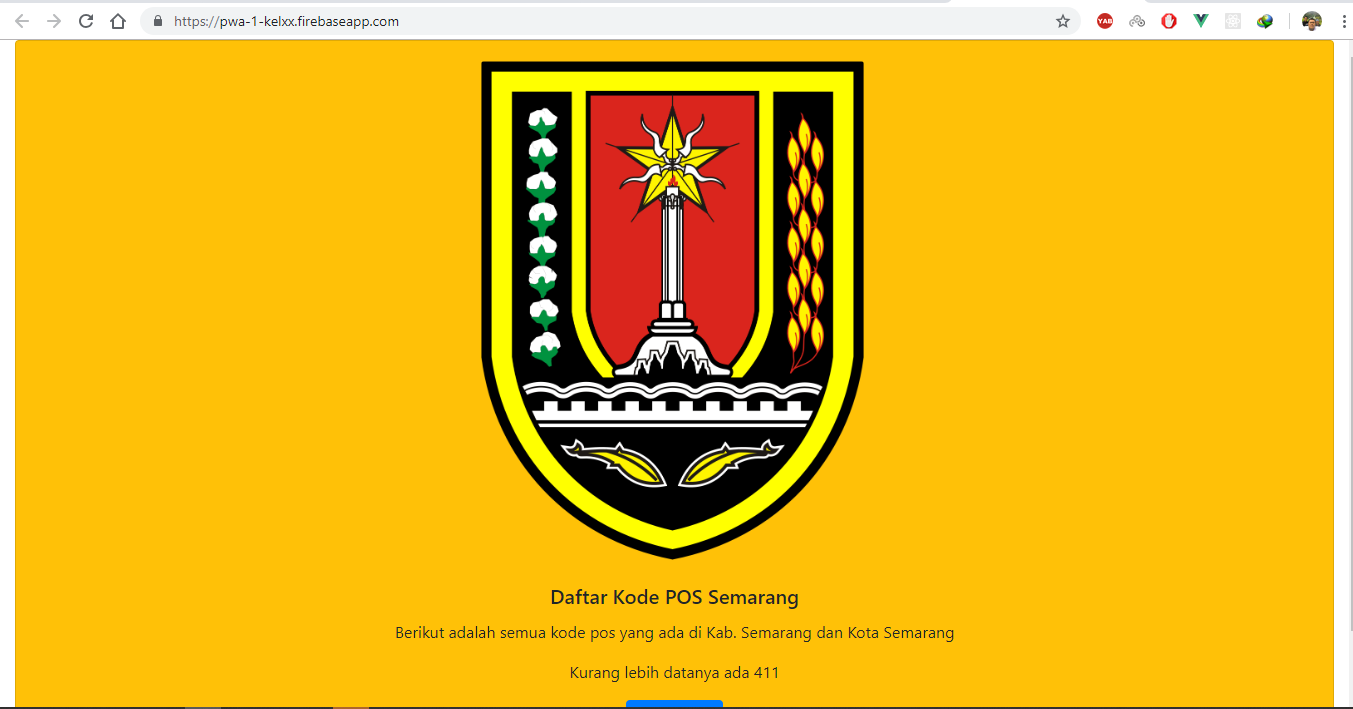
* *What do you want to use as your public directory* ? kita tekan **enter** saja
* *Configure as a single-page app (rewrite all urls to /index.html)?* kita tekan **N**
* *File public/index.html already exists. Overwrite ?* kita tekan **N**

Setelah itu kita tunggu saja.

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



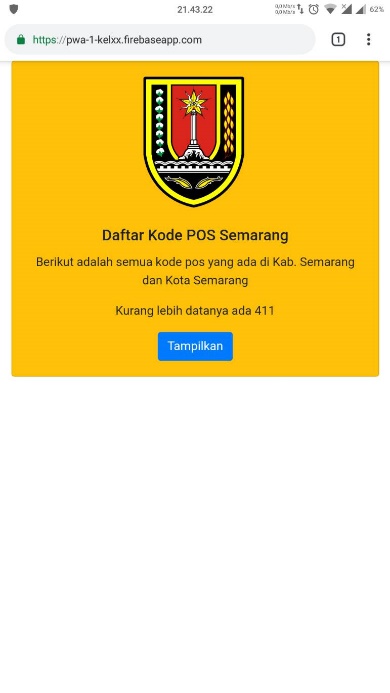
1. Setelah berhasil, akan muncul link, di *Hosting* URL, kita langsung menuju ke halaman tersebut, maka kita sudah berhasil mendeploy halaman *web* yang kita buat.



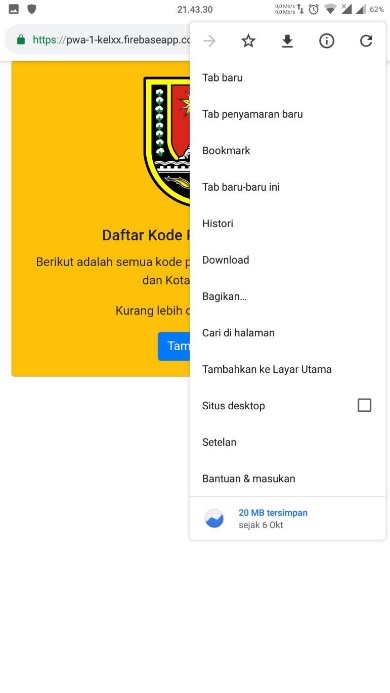
* + 1. **Website ke Mobile**

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*



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



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

