

MINGGU 12

Notifications

DESKRIPSI TEMA

Menerima notifikasi dari Firebase Cloud Messaging (FCM) menggunakan @capacitor/push-notification

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu **menghubungkan** aplikasi Ionic dengan Firebase (C3)

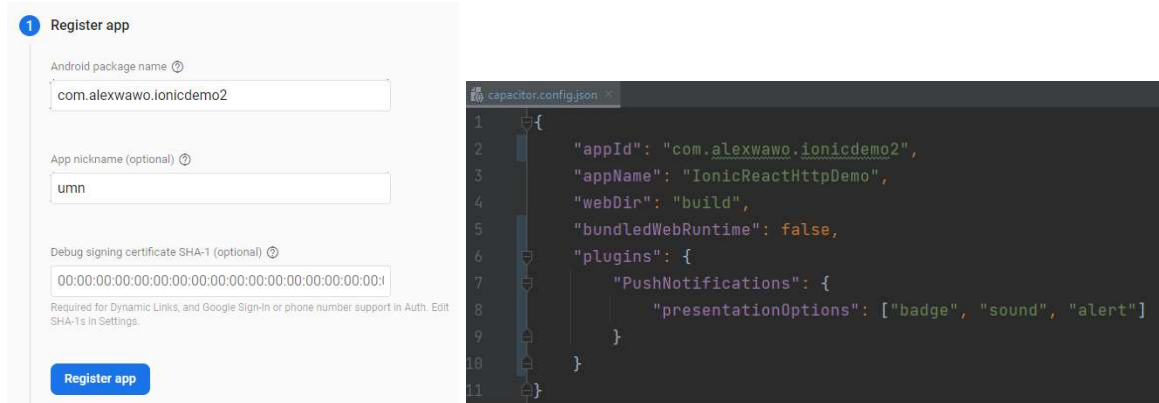
PERALATAN YANG DIGUNAKAN

WebStorm atau VS Code

LANGKAH-LANGKAH PRAKTIKUM

Firebase Setup

1. Gunakan Project Firebase yang sebelumnya pernah dibuat, atau buat project baru jika diperlukan.
2. Tambahkan app baru pada bagian project configuration dan pilih Android
3. Pastikan Android package name sama dengan appId pada capacitor.config.json pada project Ionic



4. Kemudian download file google-services.json untuk nantinya ditambahkan pada project android

SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.



See SDK instructions



google-services.json

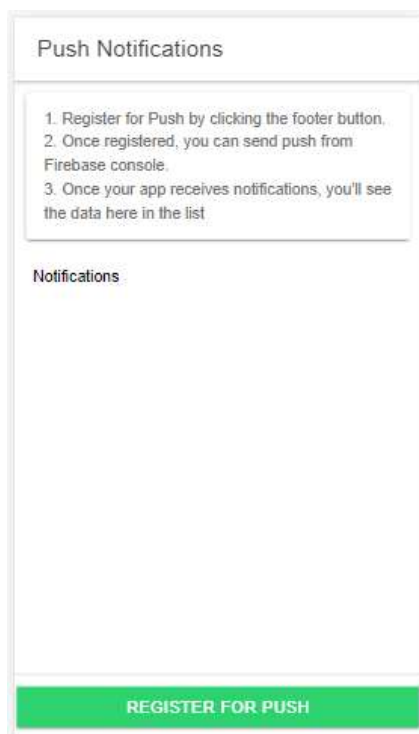
Ionic-React Project

1. Buat sebuah project Ionic-React baru, ubah page home menjadi seperti berikut

```

<IonContent fullscreen>
  <IonCard...>
  <IonListHeader...>
  <IonList>
    {notifications.map(notif=>{
      <IonItem key={notif.id}>
        <IonLabel>
          <IonText>
            <h3>{notif.title}</h3>
          </IonText>
          <p>{notif.body}</p>
          {notif.type==='foreground' && <p>This data was received in foreground</p>}
          {notif.type==='action' && <p>This data was received on tap</p>}
        </IonLabel>
      </IonItem>
    })}
  </IonList>
</IonContent>
<IonFooter>
  <IonToolbar>
    <IonButton color="success" expand="full" onClick={registerPush}>Register for Push</IonButton>
    <IonButton color="success" expand="full" routerLink={'/login'}>Go To Login</IonButton>
  </IonToolbar>
</IonFooter>

```



2. Jalankan command berikut:
 npm install @capacitor/push-notifications
 npm install @capacitor/toast

3. Tambahkan code berikut pada Home.tsx

```
const nullEntry: any[] = [];
const [notifications, setNotifications] = useState(nullEntry);

const showToast = async(msg: string) => {
  await Toast.show( options: {
    text: msg
  })
}
```

```
useEffect( effect: ()=>{
  PushNotifications.checkPermissions().then((res : PermissionStatus )=>{
    if(res.receive !== 'granted'){
      PushNotifications.requestPermissions().then((res : PermissionStatus )=>{
        if(res.receive === 'denied'){
          showToast( msg: 'Push Notifications permission denied');
        }else{
          showToast( msg: 'Push Notifications permission granted');
          registerPush();
        }
      })
    }else{
      registerPush();
    }
  })
}, deps: []);
```

```
const registerPush = () => {
  console.log('Initializing HomePage');

  // Register with Apple/Google to receive push via APNS/FCM
  PushNotifications.register();

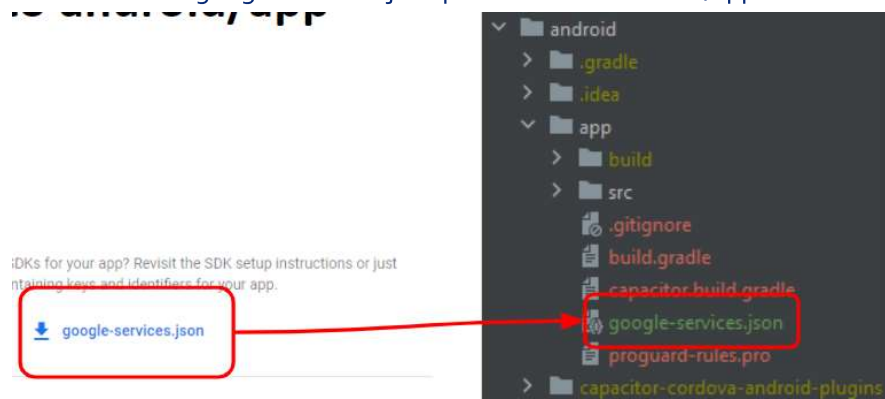
  // On success, we should be able to receive notifications
  PushNotifications.addListener( eventName: 'registration',
    listenerFunc: (token: Token) => {
      showToast( msg: 'Push registration success');
      console.log("Push Registration Success");
    });

  // Some issue with our setup and push will not work
  PushNotifications.addListener( eventName: 'registrationError',
    listenerFunc: (error:any) => {
      alert('Error on registration: ' + JSON.stringify(error));
      console.log("Push Registration Error: ", JSON.stringify(error));
    });
};
```

```
// Show us the notification payload if the app is open on our device
PushNotifications.addListener( eventName: 'pushNotificationReceived',
  listenerFunc: (notification: PushNotificationSchema) => {
    setNotifications( value: notifications => [...notifications, {
      id: notification.id, title: notification.title, body: notification.body, type: 'foreground'
    }]);
    console.log('notification:', notification);
    console.log('notifications:', notifications);
  });

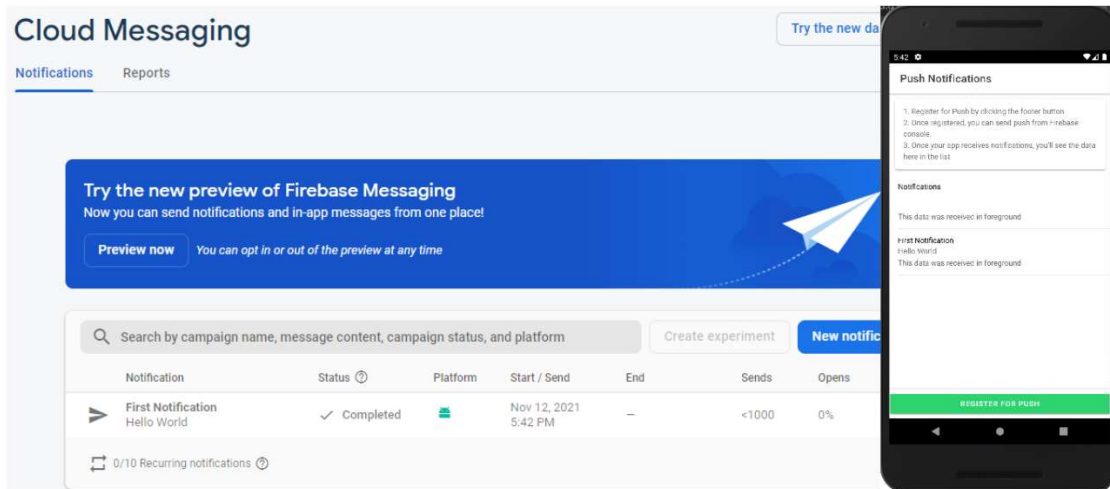
// Method called when tapping on a notification
PushNotifications.addListener( eventName: 'pushNotificationActionPerformed',
  listenerFunc: (notification: ActionPerformed) => {
    setNotifications( value: notifications => [...notifications, {
      id: notification.notification.data.id, title: notification.notification.data.title,
      body: notification.notification.data.body, type: 'action'
    }])
    console.log('notification:', notification);
    console.log('notifications:', notifications);
  });
}
```

4. Jalankan command berikut
 ionic capacitor add android
 ionic capacitor copy android
5. Tambahkan file google-services.json pada direktori android/app

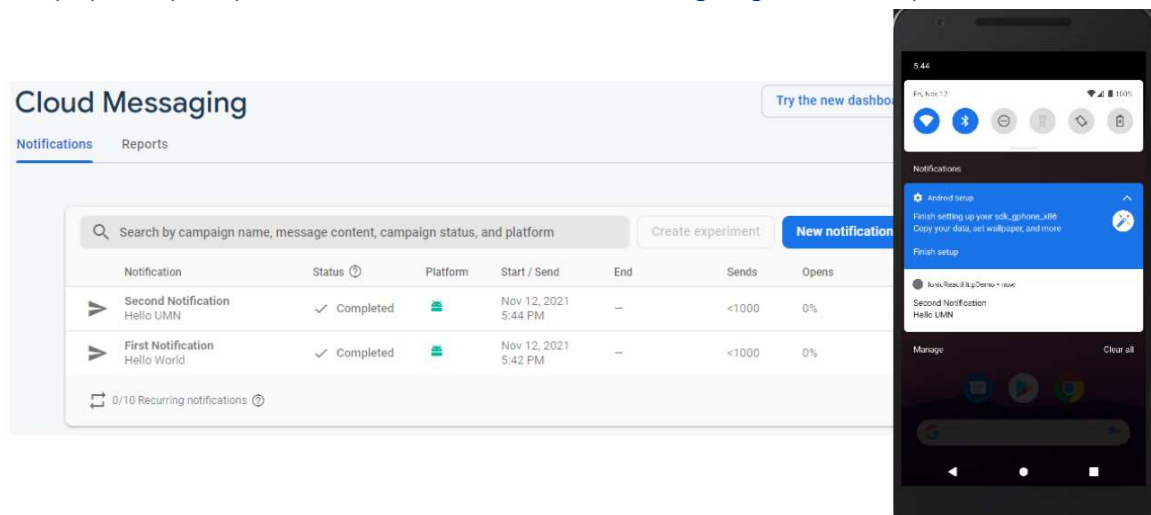


Testing

1. Buka project Android menggunakan Android Studio dan jalankan
2. Buka halaman Firebase Cloud Messaging pada project firebase Anda, dan coba kirimkan pesan baru ketika Aplikasi sedang aktif/tampil di layar ponsel



3. Tutup aplikasi pada ponsel, kemudian coba kirimkan message lagi dari FCM, apakah notifikasi masuk?



TUGAS

1. Pelajari cara penggunaan Firebase Cloud Function
2. Coba buat function untuk mendeteksi Firestore pada project Memories (project minggu lalu), jika ada memories baru yang ditambahkan, maka akan mentrigger FCM untuk mengirimkan message
3. Implementasi sehingga aplikasi memories akan memunculkan notifikasi jika ada memory baru yang ditambahkan.
4. Coba lakukan testing dengan menambahkan manual data memory di Firestore, atau menggunakan 2 device, yang pertama untuk menambah memory baru dan yang kedua untuk menerima notifikasi.