

# MINGGU 7 (TEMA)

## DESKRIPSI TEMA

- Toolbar Buttons & Floating Action Button
- Alert
- Toast
- Modal
- Context

## CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu menggunakan berbagai jenis Ionic Components untuk menghasilkan tampilan aplikasi yang interaktif.

## PERALATAN YANG DIGUNAKAN

Web Storm atau Visual Studio Code

## LANGKAH-LANGKAH PRAKTIKUM

Gunakan project IonMail dari pertemuan sebelumnya.

### Toolabr Button & Floating Action Button

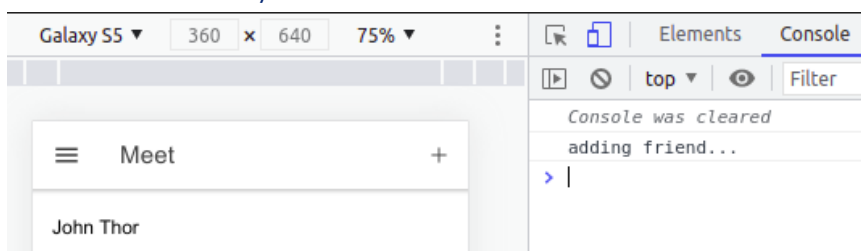
1. Tambahkan method berikut pada file Meet.tsx

```
const startAddFriendHandler = () => {  
  console.log("adding friend...");  
};
```

2. Tambahkan tombol add pada toolbar halaman Meet

```
<IonButtons slot="end">  
  <IonButton onClick={startAddFriendHandler}>  
    <IonIcon icon={addOutline} />  
  </IonButton>  
</IonButtons>  
<IonTitle>Meet</IonTitle>
```

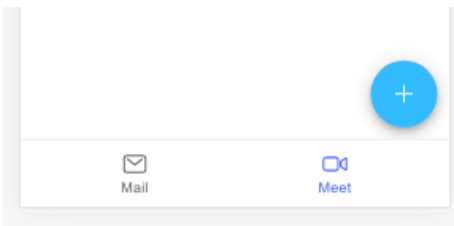
3. Lihat dan coba hasilnya di browser



4. Tambahkan FAB pada halaman Meet

```
<IonFab horizontal="end" vertical="bottom" slot="fixed">
  <IonFabButton color="secondary" onClick={startAddFriendHandler}>
    <IonIcon icon={addOutline} />
  </IonFabButton>
</IonFab>
</IonContent>
```

5. Coba hasilnya di browser

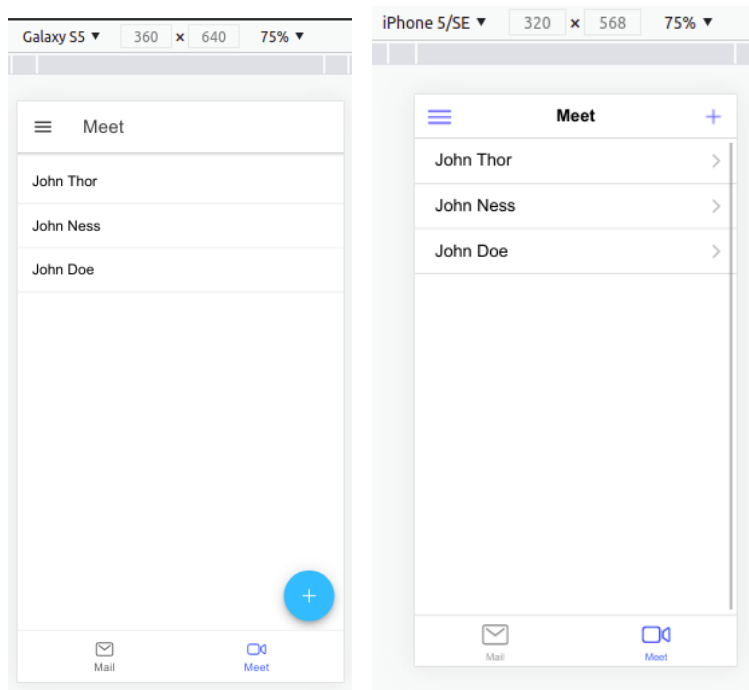


6. Modifikasi kode menjadi seperti berikut, sehingga tombol add pada toolbar hanya tampil pada platform selain android dan tombol FAB hanya tampil pada platform android.

```
{!isPlatform( platform: 'android') && (
  <IonButtons slot="end">
    <IonButton onClick={startAddFriendHandler}>
      <IonIcon icon={addOutline} />
    </IonButton>
  </IonButtons>
)}
```

```
{isPlatform( platform: 'android') && (
  <IonFab horizontal="end" vertical="bottom" slot="fixed">
    <IonFabButton color="secondary" onClick={startAddFriendHandler}>
      <IonIcon icon={addOutline} />
    </IonFabButton>
  </IonFab>
)}
```

## 7. Coba hasilnya



## Alert

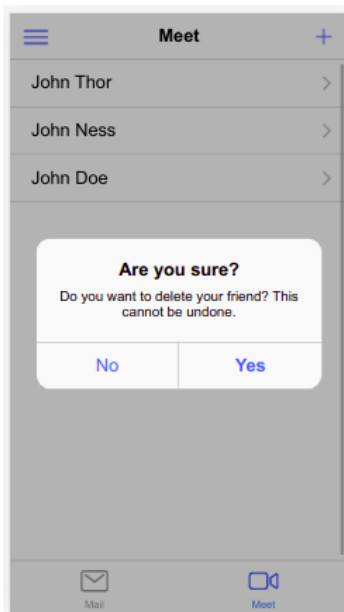
1. Modifikasi kode pada halaman Meet.tsx

```
const [startDeleting, setStartDeleting] = useState( initialState: false);
const startDeleteFriendHandler = () => {
  setStartDeleting( value: true);
  slidingOptionsRef.current?.closeOpened();
}
const deleteFriendHandler = () => {
  setStartDeleting( value: false);
  console.log("Deleting...");
};
```

2. Tambahkan elemen IonAlert pada halaman Meet.tsx

```
<React.Fragment>
  <IonAlert isOpen={startDeleting}
    header="Are you sure?"
    message="Do you want to delete your friend? This cannot be undone."
    buttons={[
      {text: 'No', role: 'cancel', handler: () => {setStartDeleting( value: false)}},
      {text: 'Yes', handler: deleteFriendHandler}
    ]} />
</IonPage>
```

3. Coba klik option delete, apakah alert muncul?



4. Kerjakan Tugas 1

## Toast

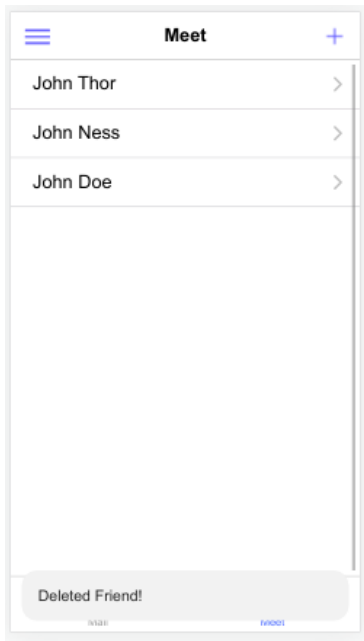
1. Tambahkan useState toastMessage

```
const [toastMessage, setToastMessage] = useState( initialState: '' );
const deleteFriendHandler = () => {
  setStartDeleting( value: false );
  setToastMessage( value: 'Deleted Friend!' );
};
```

2. Tambahkan element IonToast

```
<IonToast isOpen={!toastMessage}
  message={toastMessage}
  duration={2000}
  onDidDismiss={() => {setToastMessage( value: '' )}} />
<IonPage>
```

3. Coba hasilnya, apakah toast muncul ketika menghapus friend?



#### 4. Kerjakan Tugas 2

### Modal

1. Tambahkan kode berikut pada halaman Meet.tsx

```
const [isEditing, setIsEditing] = useState( initialState: false);
const startEditFriendHandler = () => {
  slidingOptionsRef.current?.closeOpened();
  console.log("Editing...");
  setIsEditing( value: true);
};
const cancelEditFriendHandler = () => {
  setIsEditing( value: false);
};
```

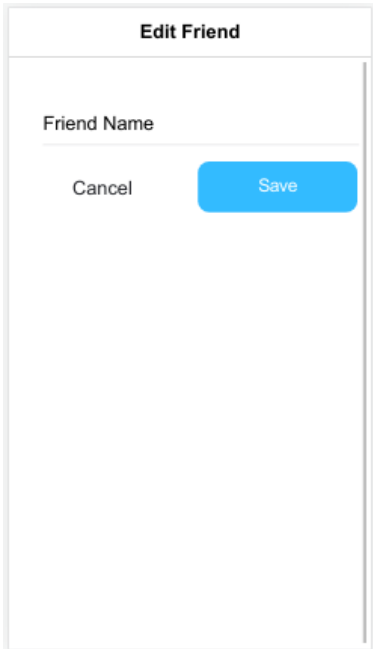
2. Tambahkan elemen IonModal, dengan IonInput untuk mengisi nama Friend

```
<IonModal isOpen={isEditing}>
  <IonHeader...>
  <IonContent>
    <IonGrid>
      <IonRow...>
        <IonRow className="ion-text-center">
          <IonCol>
            <IonButton fill="clear" color="dark" onClick={cancelEditFriendHandler}>Cancel</IonButton>
          </IonCol>
          <IonCol>
            <IonButton color="secondary" expand="block">Save</IonButton>
          </IonCol>
        </IonRow>
      </IonGrid>
    </IonContent>
  </IonModal>
```

3. Modifikasi option edit friend untuk memanggil fungsi startEditFriendHandler

```
<IonItemOptions side="end">
  <IonItemOption color="success" onClick={startEditFriendHandler}>
    <IonIcon slot="icon-only" icon={create} />
  </IonItemOption>
</IonItemOptions>
```

4. Coba jalankan, apakah modal edit Friend tampil ketika option edit dipilih?



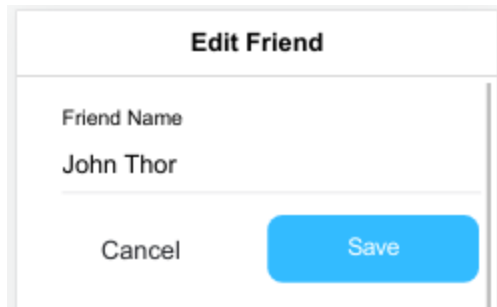
5. Modifikasi kode menjadi seperti berikut

```
const [selectedFriend, setSelectedFriend] = useState<{ id: string, name: string, photo: string } | null>();
const startEditFriendHandler = (friendId: string) => {
  slidingOptionsRef.current?.closeOpened();
  console.log("Editing...");
  const friend = FRIENDS_DATA.find(f => f.id === friendId);
  setSelectedFriend(friend);
  setIsEditing( value: true);
};

<IonItemOption color="success" onClick={startEditFriendHandler.bind( thisArg: null, friend.id)}>
  <IonIcon slot="icon-only" icon={create} />
</IonItemOption>
```

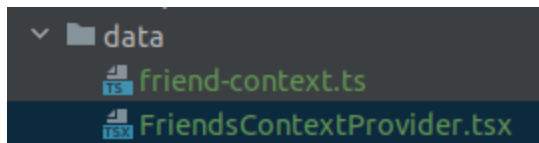
```
<IonModal isOpen={isEditing}>
  <IonHeader...>
    <IonContent>
      <IonGrid>
        <IonRow>
          <IonCol>
            <IonItem>
              <IonLabel position="floating">Friend Name</IonLabel>
              <IonInput type="text" value={selectedFriend?.name} />
            </IonItem>
          </IonCol>
        </IonRow>
      </IonGrid>
    </IonContent>
  </IonHeader>
</IonModal>
```

6. Coba jalankan, apakah nama Friend muncul pada jendela modal?



## Context

1. Buat folder data, dengan file friend-context.ts dan FriendsContextProvider.tsx.



2. Tambahkan kode berikut pada file friend-context.ts

```
import React from 'react';
export interface Friend {
  id: string,
  name: string,
  photo: string
}

interface Context {
  friends: Friend[];
  addFriend: (friendName: string, friendPhoto: string) => void,
  updateFriend: () => void,
  deleteFriend: () => void
}

const FriendsContext = React.createContext<Context>({ defaultValue: {
  friends: [],
  addFriend: () => {},
  updateFriend: () => {},
  deleteFriend: () => {}
}});
```

3. Tambahkan kode berikut pada file FriendsContextProvider.tsx

```
import React, {useState} from 'react';
import FriendsContext, {Friend} from "./friend-context";

const FriendsContextProvider: React.FC = props => {
  const [friends, setFriends] = useState<Friend[]>([...]);

  const addFriend = (name: string, photo: string) => {...};
  const updateFriend = () => {};
  const deleteFriend = () => {};

  return(
    <FriendsContext.Provider value={{friends: friends...}}...>
  );
};

export default FriendsContextProvider;
```

```
const [friends, setFriends] = useState<Friend[]>({ initialState: [
  {
    id: 'f1',
    name: 'John Thor',
    photo: ''
  }
]
});
```

```
const addFriend = (name: string, photo: string) => {
  const newFriend: Friend = {
    id: Math.random().toString(),
    name: name,
    photo: photo
  };

  setFriends( value: (currFriends :Friend[] ) => {
    return currFriends.concat(newFriend);
  });
};
```



```
return(
  <FriendsContext.Provider value={{
    friends,
    addFriend,
    updateFriend,
    deleteFriend
  }}>
    {props.children}
  </FriendsContext.Provider>
);
```

4. Tambahkan kode berikut pada file Meet.tsx

```
const startAddFriendHandler = () => {
  console.log("adding friend...");
  setIsEditing( value: true);
  setSelectedFriend( value: null);
};
```

```
const friendsCtx = useContext(FriendsContext);
```

```
const saveFriendHandler = () => {
  const enteredName = nameRef.current!.value;
  if(!enteredName) return;
  if(selectedFriend === null) {
    friendsCtx.addFriend(enteredName.toString(), friendPhoto: '');
  }
  setIsEditing( value: false);
};
```

5. Hubungkan method diatas dengan tombol Save pada Modal

```
<IonButton color="secondary" expand="block" onClick={saveFriendHandler}>Save</IonButton>
```

6. Modifikasi List pada halaman Meet.tsx untuk menggunakan data dari Context

```
<IonList>
  { /*{FRIENDS_DATA.map(friend => ({
  {friendsCtx.friends.map(friend => (
    <IonItemSliding key={friend.id} ref={slidingOptionsRef}>
```

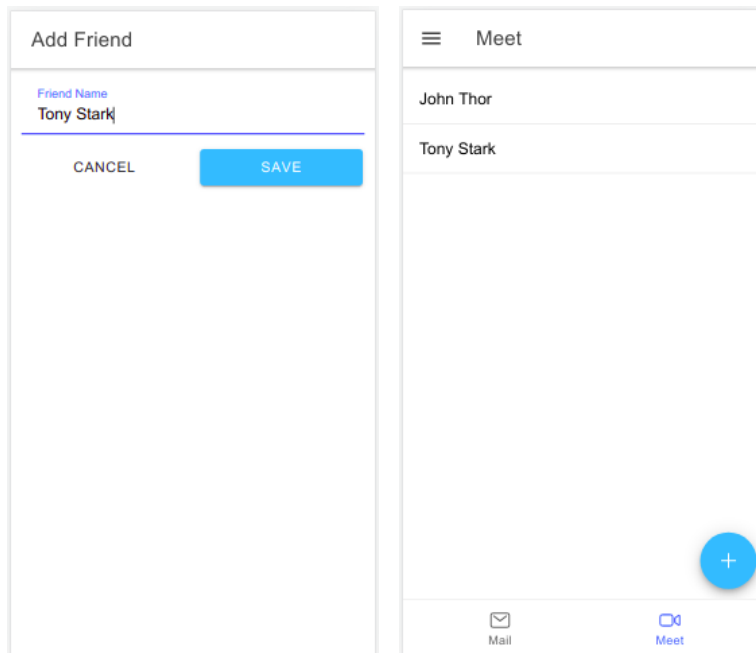
7. Bungkus IonRouterOutlet pada App.tsx dengan FriendsContextProvider

```

<FriendsContextProvider>
  <IonRouterOutlet id="main">
    <Route path="/mail/:mailId" component={MailDetail} />
    <Redirect exact path="/" to="/tabs" />
    <Route path="/tabs" component={MailTabs} />
    {/*<Route path="/mail/:mailId" component={MailDetail} />*/}
    <Route path="/settings" component={Settings} />
    {/*<Redirect exact from="/" to="/tabs" />*/}
  </IonRouterOutlet>
</FriendsContextProvider>

```

8. Coba hasilnya, apakah data Friend sudah dinamis? (Bisa menambahkan friend baru)



## TUGAS

1. Buat Alert ketika user meng-klik option block friend.
2. Tambahkan Toast ketika user mengkonfirmasi block friend.
3. Challenge: Tambahkan fitur Edit Friend dan Delete Friend.