

# MINGGU 6

## Navigation

### DESKRIPSI TEMA

- Membuat Navigasi menggunakan Tabs dan Side Menu

### CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu menerapkan berbagai jenis navigasi pada aplikasi Ionic

### PERALATAN YANG DIGUNAKAN

WebStorm atau Visual Studio Code

### LANGKAH-LANGKAH PRAKTIKUM

#### Dynamic Route

1. Buatlah sebuah project Ionic baru menggunakan React, dengan template Blank.
2. Buka project yang sudah dibuat pada IDE Editor, kemudian hapus semua file pada folder components dan pages, serta ganti code pada file App.tsx menjadi seperti berikut:

```
const App: React.FC = () => (  
  <IonApp>  
    <h2>Hello World</h2>  
  </IonApp>  
);
```

3. Tambahkan routing pada App.tsx seperti berikut:

```
<IonReactRouter>  
  <IonRouterOutlet id="main">  
    <Route exact path="/mail" component={Mail} />  
    <Route path="/mail/:mailId" component={MailDetail} />  
    <Redirect exact from="/" to="/mail" />  
  </IonRouterOutlet>  
</IonReactRouter>
```

4. Buat file ./pages/Mail.tsx dengan kode seperti berikut:

```
import * as React from 'react';

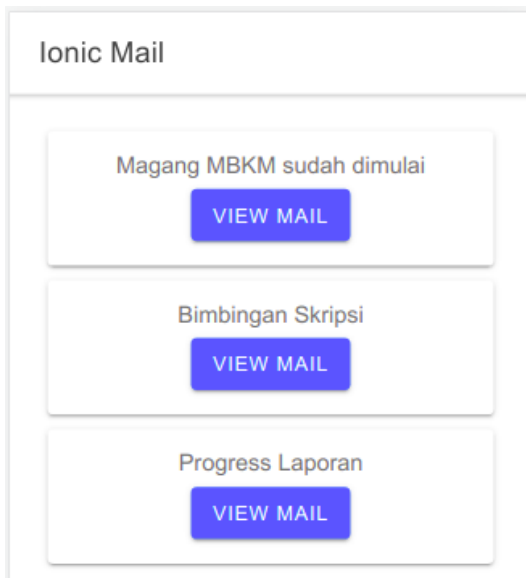
export const MAIL_DATA = [
  {id: 'm1', subject: 'Magang MBKM sudah dimulai'},
  {id: 'm2', subject: 'Bimbingan Skripsi'},
  {id: 'm3', subject: 'Progress Laporan'}
];

const Mail: React.FC = () => {...};

export default Mail;
```

```
<IonPage>
  <IonHeader>
    <IonToolbar>...</IonToolbar>
  </IonHeader>
  <IonContent className="ion-padding">
    {MAIL_DATA.map(mail => (
      <IonCard key={mail.id}>
        <IonCardContent className="ion-text-center">
          <h2>{mail.subject}</h2>
          <IonButton routerLink={` /mail/${mail.id}`}>
            View Mail
          </IonButton>
        </IonCardContent>
      </IonCard>
    ))}
  </IonContent>
</IonPage>
```

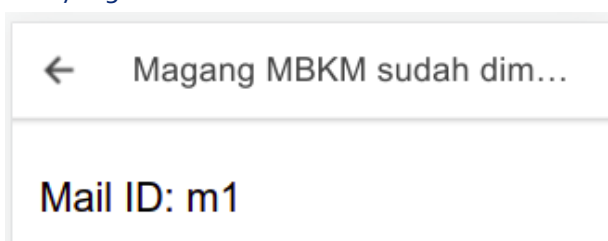
5. Coba jalankan di browser dan lihat hasilnya.



6. Buat file ./pages/MailDetail.tsx, dengan kode seperti berikut

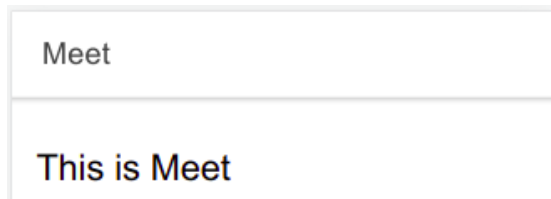
```
const MailDetail: React.FC = () => {
  const mId = useParams<{mailId: string}>().mailId;
  const selectedMail = MAIL_DATA.find(m => m.id === mId);
  return(
    <IonPage>
      <IonHeader>
        <IonToolbar>
          <IonButtons slot="start">...</IonButtons>
          <IonTitle>
            {selectedMail ? selectedMail?.subject : 'No mail found'}
          </IonTitle>
        </IonToolbar>
      </IonHeader>
      <IonContent className="ion-padding">
        <h2>Mail ID: {mId}</h2>
      </IonContent>
    </IonPage>
  );
};
```

7. Coba jalankan di browser, klik pada tombol "View Email", apakah halaman MailDetail tampil dengan data yang sesuai?



## Tabs Part 1

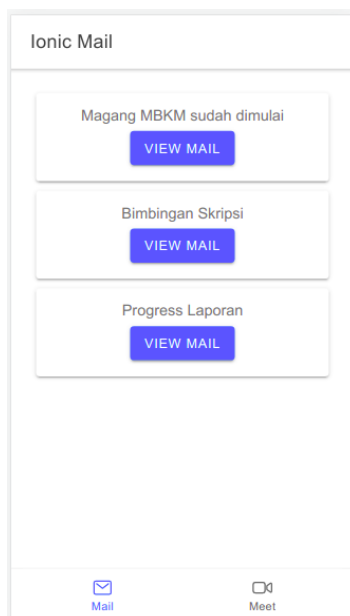
1. Buat file ./pages/Meet.tsx untuk menghasilkan tampilan seperti berikut (tambahkan juga route-nya)



2. Tambahkan kode berikut pada App.tsx

```
<IonReactRouter>
  <IonTabs>
    <IonRouterOutlet...>
    <IonTabBar slot="bottom">
      <IonTabButton tab="mail" href="/mail">
        <IonIcon icon={mailOutline} />
        <IonLabel>Mail</IonLabel>
      </IonTabButton>
      <IonTabButton tab="meet" href="/meet">
        <IonIcon icon={videocamOutline} />
        <IonLabel>Meet</IonLabel>
      </IonTabButton>
    </IonTabBar>
  </IonTabs>
</IonReactRouter>
```

3. Lihat hasilnya di browser, apakah tabs muncul?



4. Perhatikan bahwa tabs selalu muncul, ketika tombol View Mail di-klik dan menampilkan halaman Mail Detail. Bagaimana jika kita tidak ingin tabs muncul pada halaman Mail Detail?

## Tabs part 2

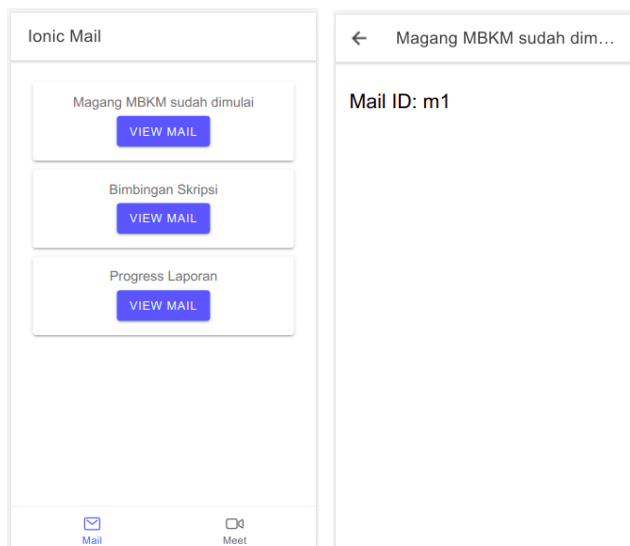
1. Buat file .pages/MailTabs.tsx dengan kode seperti berikut

```
const MailTabs: React.FC = () => {
  return(
    <IonTabs>
      <IonRouterOutlet>
        <Redirect exact path="/tabs" to="/tabs/mail" />
        <Route exact path="/tabs/mail" component={Mail} />
        <Route exact path="/tabs/meet" component={Meet} />
      </IonRouterOutlet>
      <IonTabBar slot="bottom" ...>
    </IonTabBar>
  </IonTabs>
  );
};
```

2. Ubah kode pada App.tsx menjadi seperti berikut

```
<IonApp>
  <IonReactRouter>
    <IonRouterOutlet>
      <Route path="/tabs" component={MailTabs} />
      <Route path="/mail/:mailId" component={MailDetail} />
      <Redirect exact from="/" to="/tabs" />
    </IonRouterOutlet>
  </IonReactRouter>
</IonApp>
```

3. Lihat hasilnya di browser, coba buka halaman MailDetail, apakah tabs masih muncul?



## Side Menu

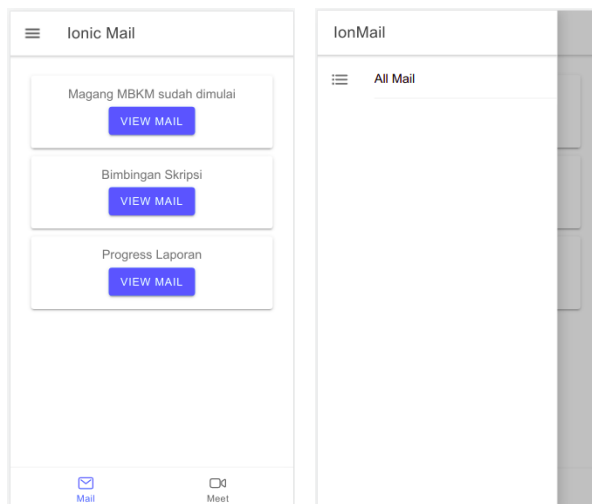
1. Tambahkan kode berikut pada App.tsx

```
<IonReactRouter>
  <IonMenu contentId="main">
    <IonHeader...>
    <IonContent>
      <IonList>
        <IonMenuToggle>
          <IonItem button routerLink="/tabs/mail">
            <IonIcon slot="start" icon={list} />
            <IonLabel>All Mail</IonLabel>
          </IonItem>
        </IonMenuToggle>
      </IonList>
    </IonContent>
  </IonMenu>
  <IonRouterOutlet id="main"...>
</IonReactRouter>
```

2. Tambahkan menu button pada halaman Mail dan Meet

```
<IonToolbar>
  <IonButtons slot="start">
    <IonMenuButton />
  </IonButtons>
  <IonTitle>Ionic Mail</IonTitle>
</IonToolbar>
```

3. Coba hasilnya di browser, apakah side menu berhasil tampil?



4. Kerjakan Tugas 1

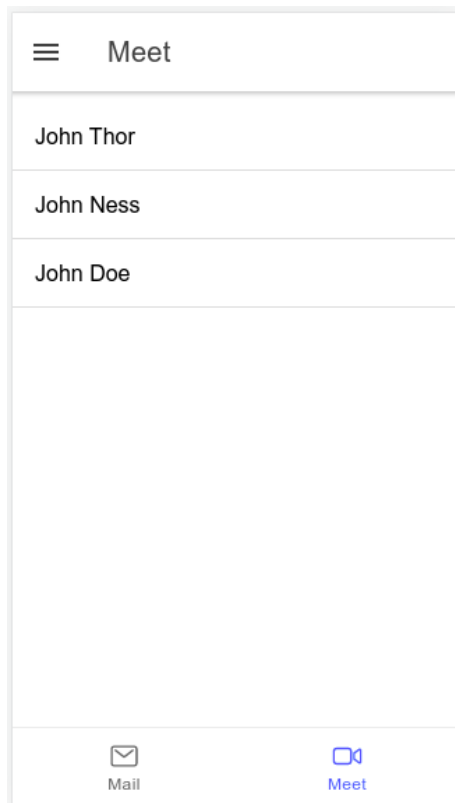
## IonList & IonItem

1. Pada `.pages/Meet.tsx`, tambahkan kode berikut

```
export const FRIENDS_DATA = [
  {id: 'f1', name: 'John Thor'},
  {id: 'f2', name: 'John Ness'},
  {id: 'f3', name: 'John Doe'}
];

<IonContent>
  <IonList>
    {FRIENDS_DATA.map(friend => (
      <IonItem key={friend.id} lines="full" button>
        <IonLabel>{friend.name}</IonLabel>
      </IonItem>
    ))}
  </IonList>
</IonContent>
```

2. Coba lihat hasilnya di browser:

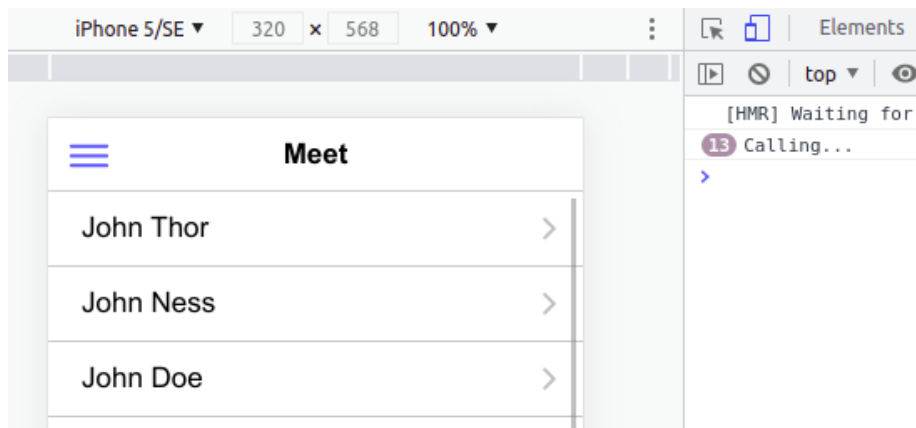


3. Tambahkan kode berikut pada `Meet.tsx` untuk menangani ketika item di-klik:

```
const callFriendHandler = () => {
  console.log("Calling...");
};
```

```
<IonItem key={friend.id}
  lines="full"
  button
  onClick={callFriendHandler}>
  <IonLabel>{friend.name}</IonLabel>
</IonItem>
```

4. Coba hasilnya di browser, dan perhatikan Console log.



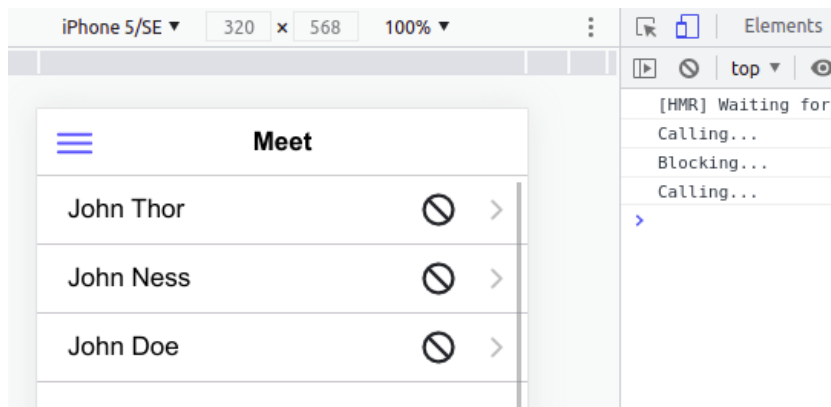
5. Tambahkan kode berikut pada Meet.tsx

```
const blockFriendHandler = () => {
  console.log("Blocking...");
};
```

```
<IonItem key={friend.id}
  lines="full"
  button
  onClick={callFriendHandler}>
  <IonLabel>{friend.name}</IonLabel>
  <IonButton fill="clear"
    color="dark"
    slot="end"
    onClick={blockFriendHandler}>
    <IonIcon icon={banSharp} slot="icon-only" />
  </IonButton>
</IonItem>
```



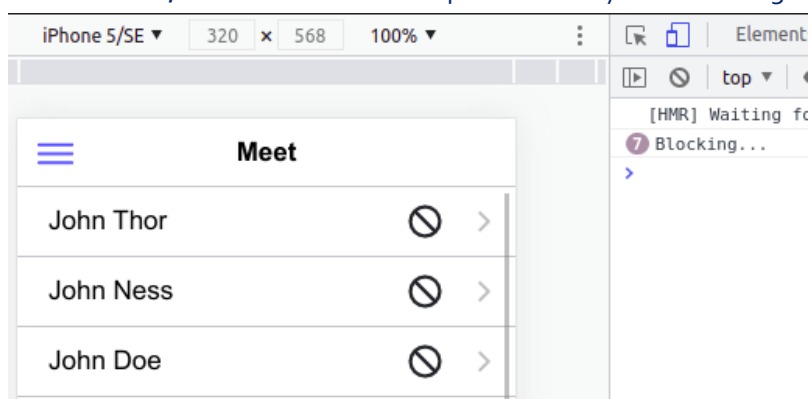
6. Coba jalankan di browser, klik pada label nama dan pada tombol block, perhatikan console log.



7. Modifikasi fungsi blockFriendHandler menjadi seperti berikut:

```
const blockFriendHandler = (event: React.MouseEvent) => {
  event.stopPropagation();
  console.log("Blocking...");
};
```

8. Coba kembali, klik tombol block dan perhatikan layar console log

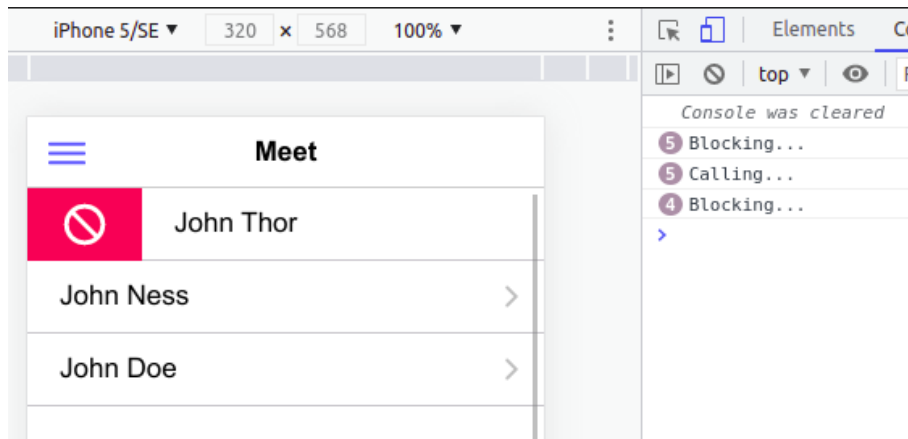


## Swipable Item

1. Modifikasi kode pada Meet.tsx menjadi seperti berikut

```
<IonItemSliding key={friend.id}>
  <IonItemOptions side="start">
    <IonItemOption color="danger" onClick={blockFriendHandler}>
      <IonIcon slot="icon-only" icon={ban} />
    </IonItemOption>
  </IonItemOptions>
  <IonItem lines="full">
    button
    onClick={callFriendHandler}>
    <IonLabel>{friend.name}</IonLabel>
  </IonItem>
</IonItemSliding>
```

2. Coba jalankan di browser, slide item dan coba di-klik.



3. Perhatikan bahwa ketika Option block di-klik, slide item tidak kembali ke posisi awal. Tambahkan kode berikut:

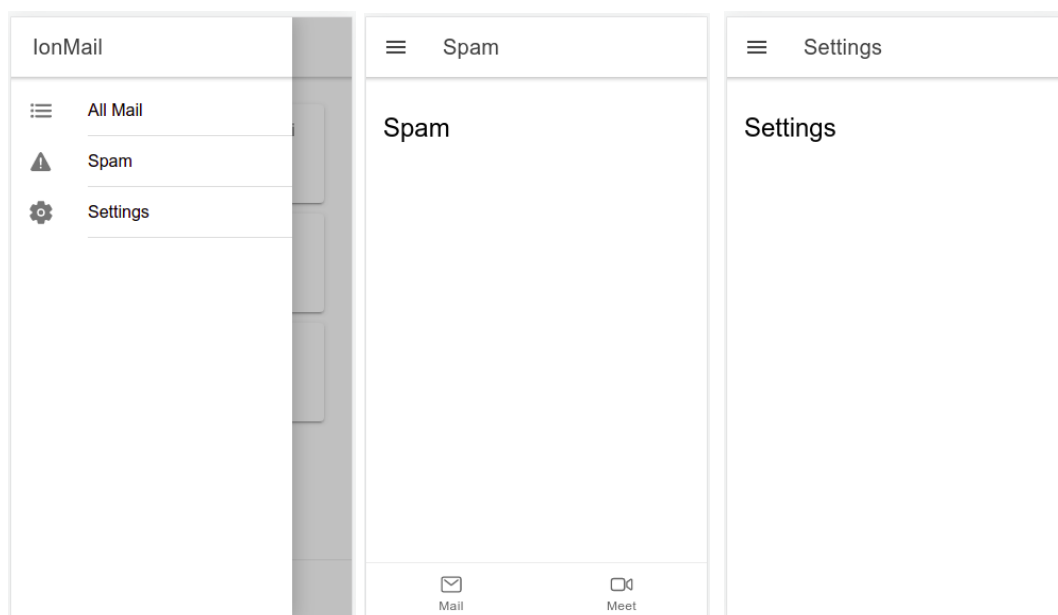
```
const slidingOptionsRef = useRef<HTMLIonItemSlidingElement>({ initialValue: null });
const blockFriendHandler = () => {
  slidingOptionsRef.current?.closeOpened();
  console.log("Blocking...");
};
```

```
<IonItemSliding key={friend.id} ref={slidingOptionsRef}>
```

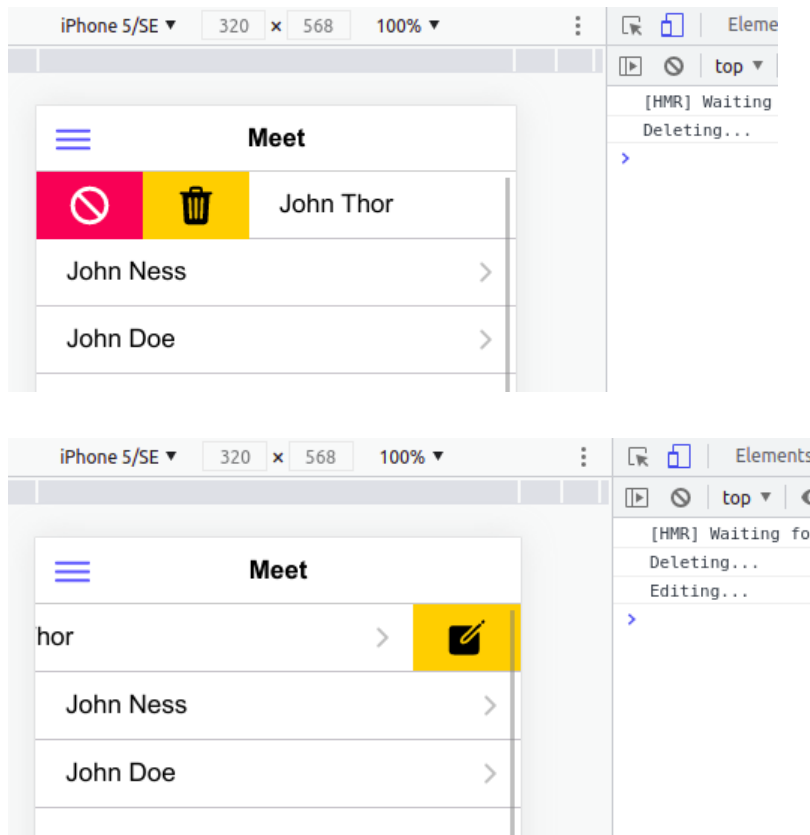
4. Coba kembali, apakah ketika option pada IonItem di-klik, posisi slidennya kembali seperti awal?
5. Kerjakan Tugas 2 dan 3

## TUGAS

1. Tambahkan menu Spam dan Settings pada Side menu, dimana halaman Spam menampilkan tabs dan halaman Settings tidak menampilkan tabs.



2. Tambahkan Option Delete dan Edit pada IonItem di halaman Meet.tsx sehingga tampil seperti berikut:



3. Tambahkan avatar pada daftar friend sehingga tampil seperti gambar berikut:

