



KASUS 9: AUTH GUARD IONIC

Catatan :
Pastikan kalian telah mengerjakan Latihan pada Kasus 8 sebelumnya. Pada Latihan 9 ini akan dibahas mengenai bagaimana melakukan memberikan penjaga (guard) di setiap halaman ionic. Kegunaan guard adl untuk mengijinkan sebuah halaman dibuka atau tidak. Misalkan halaman landing, login dan register akan kita sifati sebagai halaman public yang dapat dibuka jika user belum melakukan login (validasi user), sedangkan halaman dashboard (tabs) hanya boleh dibuka setelah user melakukan login. Proses sebaliknya tidak akan diijinkan.

Halaman yg boleh dibuka sebelum Login	Halaman yang boleh dibuka sesudah Login
Landing, Login, Register, Forgot Password	Dashboard (Tabs) dan semua halaman children di bawahnya tabs
Halaman yg tidak boleh dibuka sebelum Login	Halaman yang tidak boleh dibuka sesudah Login
Dashboard (Tabs) dan semua halaman children di bawahnya tabs	Landing, Login, Register, Forgot Password

SOAL:

1. Buka file router **index.ts** kemudian lakukan tambahkan script seperti berikut ini:

```
import { createRouter, createWebHistory } from '@ionic/vue-router';
import { RouteRecordRaw } from 'vue-router';
import TabsPage from '../views/TabsPage.vue';
import LandingPage from '@views/LandingPage.vue';
import LoginPageVue from '@views/auth/LoginPage.vue';
import RegisterPageVue from '@views/auth/RegisterPage.vue';

const routes: Array<RouteRecordRaw> = [
  {
    path: '/',
    redirect: '/landing'
  },
  {
    path: '/landing',
    component: LandingPage,
    meta: {
      public: true,
      whenLogout: true
    }
  },
  {
    path: '/login',
    component: LoginPageVue,
    meta: {
      public: true,
      whenLogout: true
    }
  },
  {
    path: '/register',
    component: RegisterPageVue,
    meta: {
      public: true,
      whenLogout: true
    }
  },
  {
    path: '/tabs/',
    component: TabsPage,
    children: [
      {
        path: '',
        redirect: '/tabs/tab1'
      },
      {
        path: 'tab1',
        component: () => import('@views/Tab1Page.vue')
      },
      {
        path: 'tab2',
        component: () => import('@views/Tab2Page.vue')
      },
      {
        path: 'tab3',
        component: () => import('@views/Tab3Page.vue')
      }
    ]
  }
]
```



```
const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes,
})

router.beforeEach((to, from, next) => {
  const isPublic = to.matched.some(record => record.meta.public);
  const onlyWhenLogout = to.matched.some(record => record.meta.whenLogout);
  const isLoggedIn = localStorage.getItem('loggedIn');

  if(!isPublic && !isLoggedIn) {
    return next('/landing')
  }
  if(isLoggedIn && onlyWhenLogout) {
    return next('/tabs/tab1')
  }
  next();
})
export default router
```

2. Perintah **router.beforeEach** akan mencegah route halaman sebelum dijalankan sesuai dengan syarat yang telah ditentukan.

Catatan:

next() – akan menjalankan route di alamat yang sekarang sedang terlihat.

```
if(!isPublic && !isLoggedIn) {
  return next('/landing')
}
```

Misal user **blm login** kemudian user mencoba membuka halaman **tabs**, maka nilai **!isPublic = true** dan **!isLoggedIn = true** sehingga route akan menjalankan **/landing** dan halaman landing akan dipanggil.

```
if(isLoggedIn && onlyWhenLogout) {
  return next('/tabs/tab1')
}
```

Misal user **sudah login** kemudian user mencoba membuka halaman login, maka nilai **isLoggedIn = true** dan **onlyWhenLogout = true** sehingga route akan menjalankan **/tabs/tab1** dan halaman tab1 akan dipanggil.

----- SELAMAT MENCOBA -----