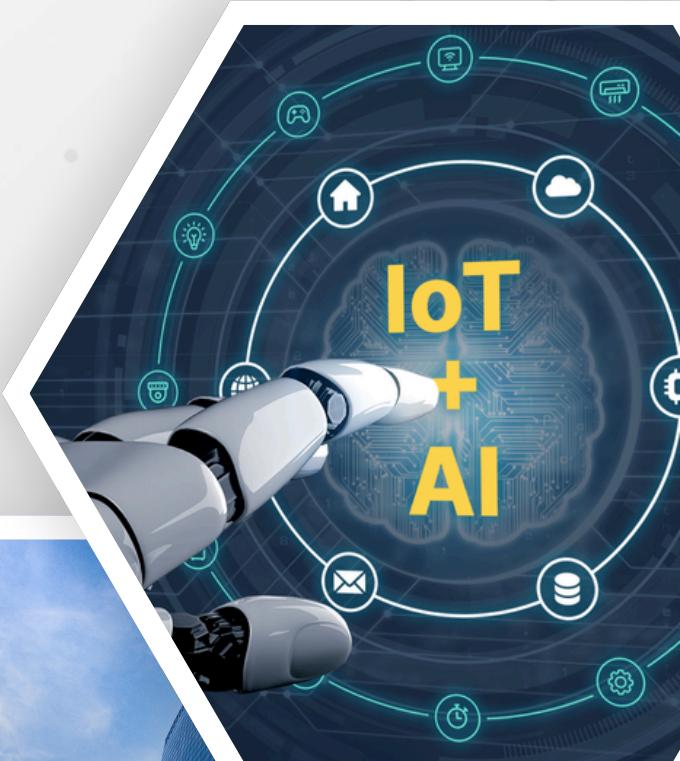
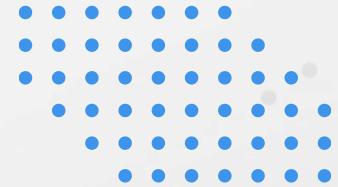




BUKU PANDUAN

*Buku Panduan Membuat Aplikasi
Controller Drone Dji Tello buatan
sendiri*

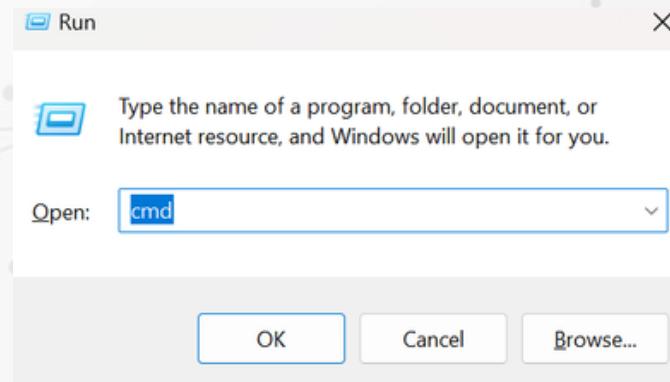
Disusun Oleh :
Tim Mentor



Menginstall Proyek Baru

Pada langkah awal, anda akan belajar membuat tampilan antarmuka pengguna (UI) untuk aplikasi Android menggunakan Ionic. Berikut adalah langkah-langkahnya:

Membuat project baru ionic dengan membuka command prompt:



Gunakan perintah ini untuk membuat proyek Ionic baru dengan template blank:

- **ionic start droneController blank**

Note: *droneController* adalah nama project kita nantinya dan *blank* untuk membuat tampilan default projek ionic kosongan

Setelah proyek berhasil dibuat, masuk ke direktori proyek:

- **cd droneController**

Jangan lupa mengubah projek capacitor menjadi cordova :

- **npm install -g cordova**
- **ionic integrations disable capacitor**
- **ionic integrations enable cordova**

Menjalankan Aplikasi di Browser/Chrome:

- **ionic serve**

Menginstall Proyek Baru

Kemudian tambahkan dukungan untuk platform Android pada proyek Anda. Ini memungkinkan Anda membangun aplikasi untuk perangkat Android/ios :

- **ionic cordova platform add android**
- **ionic corova platform add ios**

Instal plugin yang memungkinkan Anda mengakses informasi tentang perangkat, seperti jenis perangkat dan versi sistem operasi :

- **ionic cordova plugin add cordova-plugin-device**

Buat sebuah service baru bernama "tello" di dalam folder "services." Service ini akan berfungsi untuk mengatur logika dalam aplikasi Anda:

- **ionic g service services/tello**

Instal plugin untuk mengatur orientasi layar (misalnya, potret atau landscap) pada aplikasi Anda :

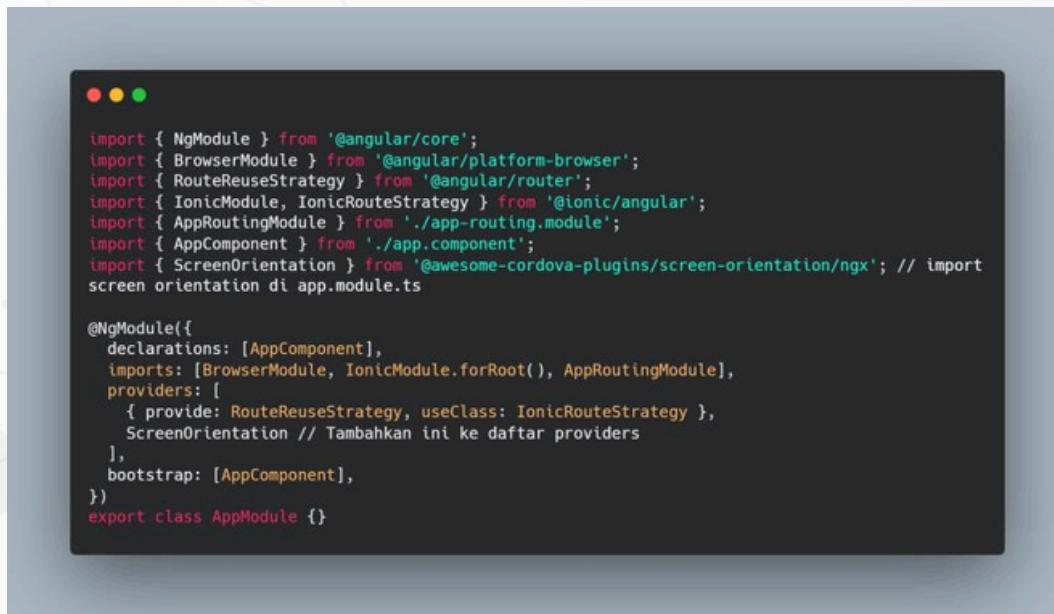
- **npm install @awesome-cordova-plugins/screen-orientation**

<preference name="Orientation" value="landscape" />

Tambahkan di file config.xml berguna untuk mengatur orientasi aplikasi menjadi landscape.

Menginstall Proyek Baru

app.module.ts



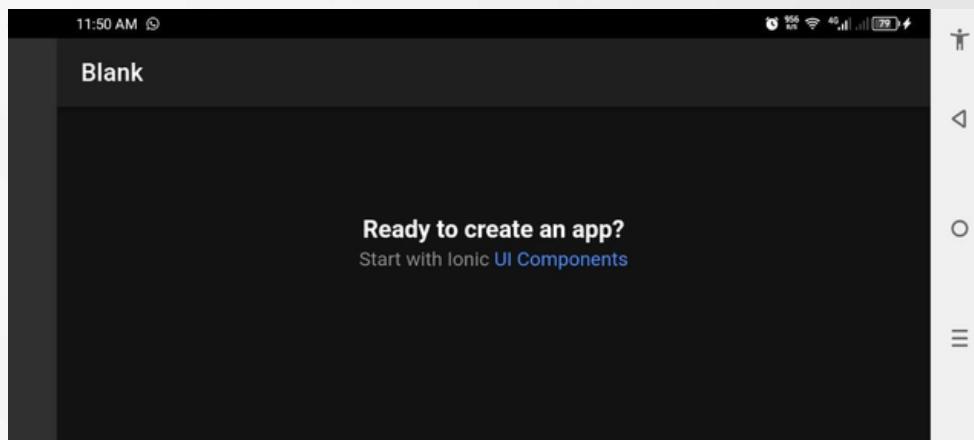
```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ScreenOrientation } from '@awesome-cordova-plugins/screen-orientation/ngx'; // import screen orientation di app.module.ts

@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule],
  providers: [
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy },
    ScreenOrientation // Tambahkan ini ke daftar providers
  ],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

Instal paket ini untuk plugin-plugin Awesome Cordova. Ini diperlukan agar plugin lain bisa berfungsi dengan baik :

- **npm install @awesome-cordova-plugins/core**

- Tampilan saat di run di android :

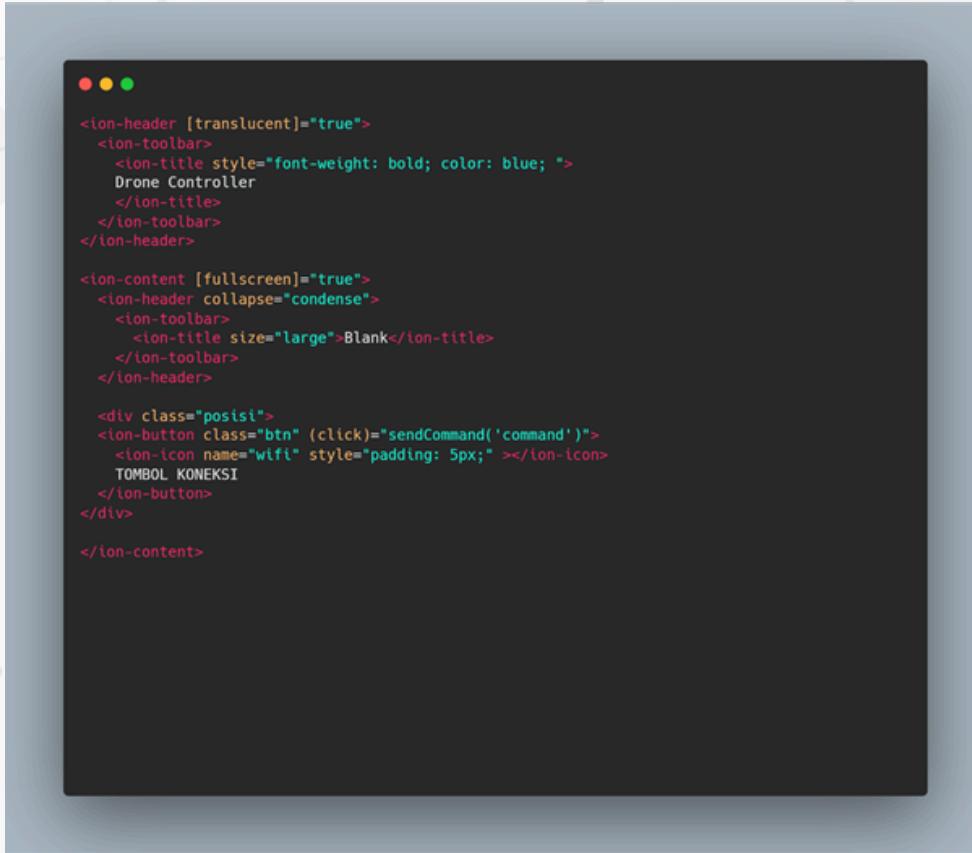


Mengkoneksi APP Ionic dengan Drone

Tambahkan Plugin tambahan yaitu untuk perantara koneksi android dengan jaringan drone :

- **ionic cordova plugin add cordova-plugin-chrome-apps-sockets-udp**

home.page.html



```
<ion-header [translucent] = "true">
  <ion-toolbar>
    <ion-title style="font-weight: bold; color: blue; ">
      Drone Controller
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen] = "true">
  <ion-header collapse="condense">
    <ion-toolbar>
      <ion-title size="large">Blank</ion-title>
    </ion-toolbar>
  </ion-header>

  <div class="posisi">
    <ion-button class="btn" (click) = "sendCommand('command')">
      <ion-icon name="wifi" style="padding: 5px;"></ion-icon>
      TOMBOL KONEKSI
    </ion-button>
  </div>
</ion-content>
```

Mengkoneksi APP Ionic dengan Drone

home.page.scss

```
● ● ●

#container {
  text-align: center;

  position: absolute;
  left: 0;
  right: 0;
  top: 50%;
  transform: translateY(-50%);
}

#container strong {
  font-size: 20px;
  line-height: 26px;
}

#container p {
  font-size: 16px;
  line-height: 22px;

  color: #8c8c8c;
}

#container a {
  text-decoration: none;
}

.posisi{
  display: flex;
  width: auto;
  height: auto;
  margin-top: 20%;
  justify-content: center;
}
ion-button{
  border-radius: 20%;
}

ion-button .btn{
  position: absolute;
  justify-content: center;
  align-items: center;
}
```

Mengkoneksi APP Ionic dengan Drone

Fungsi untuk tombol koneksi

home.page.ts

```
import { Component } from '@angular/core';
import { TelloService } from '../services/tello.service';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  canSendCommand = true;

  constructor(private telloService: TelloService) {}

  async sendCommand(command: string) {
    if (this.canSendCommand) {
      console.log(`Mengirim perintah: ${command}`);
      await this.telloService.sendCommand(command);
    }
  }
}
```

Mengkoneksi APP Ionic dengan Drone

Sebuah server atau service untuk dapat berkomunikasi dengan drone dji tello

tello.service.ts

```
import { Injectable } from '@angular/core';

declare var chrome: any;
@Injectable({
  providedIn: 'root',
})
export class TelloService {
  private telloAddress = '192.168.10.1';
  private telloPort = 8889;
  private socketId: number | null = null;
  private batteryStatusCallback: (status: number) => void = () => {};
  private accelerationCallback: (accel: {x: number, y: number, z: number}) => void = () => {};
  private lastResponseTime: number = 0;
  battery: number = 0;

  constructor() {
    document.addEventListener('deviceready', () => {
      this.createSocket();
    }, false);
  }

  createSocket() {
    if (chrome && chrome.sockets && chrome.sockets.udp) {
      chrome.sockets.udp.create({}, (socketInfo: any) => {
        this.socketId = socketInfo.socketId;
        console.log('Socket dibuat dengan ID:', this.socketId);
        this.bindSocket();
      });
    } else {
      console.error('Socket UDP tidak tersedia. Pastikan plugin terinstal.');
    }
  }

  bindSocket() {
    if (this.socketId !== null) {
      chrome.sockets.udp.bind(this.socketId, '0.0.0.0', 0, (result: any) => {
        if (result < 0) {
          console.error('Gagal bind socket:', chrome.runtime.lastError);
        } else {
          console.log('Socket berhasil di-bind ke port yang tersedia');
          this.startReceiving();
        }
      });
    }
  }
}

startReceiving() {
  if (this.socketId !== null) {
    chrome.sockets.udp.onReceive.addListener((info: any) => {
      if (info.socketId === this.socketId) {
        try {
          const dataArray = new Uint8Array(info.data);
          const message = new TextDecoder().decode(dataArray);
          console.log('Pesan diterima:', message);

          // Cek apakah responsnya adalah akselerasi
          if (message.startsWith('acceleration')) {
            const accelerationData = message.trim();
            console.log('Data akselerasi:', accelerationData);
          } else if (typeof message === 'string') {
            // Penanganan respons lain, seperti status baterai
            const batteryLevel = parseInt(message.trim(), 10);
            if (!isNaN(batteryLevel)) {
              this.battery = batteryLevel;
              if (this.batteryStatusCallback) {
                this.batteryStatusCallback(this.battery);
              }
              this.lastResponseTime = Date.now();
            }
          } else {
            console.error('Data yang diterima bukan string:', message);
          }
        } catch (error) {
          console.error('Kesalahan saat memproses data:', error);
        }
      }
    });
  }
}
```

Membuat Joystick Drone Controller

Menginstal pustaka NippleJS (terminal) :

- **npm install nipplejs**

NippleJS adalah pustaka JavaScript untuk membuat joystick virtual pada aplikasi web atau mobile, sering digunakan dalam kontrol arah seperti pengendali drone

Mengimport nipplejs ke typescript ,masukkan ke **(home.page.ts)** :



```
import { Component, AfterViewInit, OnInit, NgZone } from '@angular/core';
import { TelloService } from '../services/udp.service';
import * as nipplejs from 'nipplejs';
```

Langkah selanjutnya mulai membuat tampilan controller nya dengan joystick. Buka home.page.html, lalu ketikan kode dibawah ini



```
<div class="joystick-group">
  <div class="joystick-container">
    <div class="joystick-label top-label">Maju</div>
    <div id="movement-joystick" class="joystick"></div>
    <div class="joystick-label bottom-label">Mundur</div>
    <div class="joystick-label left-label">Kiri</div>
    <div class="joystick-label right-label">Kanan</div>
  </div>

  <div class="joystick-container">
    <div class="joystick-label top-label">Naik</div>
    <div id="directional-joystick" class="joystick"></div>
    <div class="joystick-label bottom-label">Turun</div>
    <div class="joystick-label left-label">Belok <br> Kiri</div>
    <div class="joystick-label right-label">Belok <br> Kanan</div>
  </div>
</div>
```

Membuat Joystick Drone Controller

masukkan kode ini ke **(home.page.ts)** :

```
● ● ●

import { Component, AfterViewInit, OnInit, NgZone } from '@angular/core';
import { TelloService } from '../services/udp.service';
import * as nipplejs from 'nipplejs';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage implements AfterViewInit, OnInit {
  movementJoystick: any;
  directionalJoystick: any;
  private canSendCommand = true;
  private RC: number[] = [0, 0, 0, 0]; // RC[0] = Roll, RC[1] = Pitch, RC[2] = Throttle, RC[3] = Yaw
  battery: number | null = null;
  connection: string = 'Disconnected';
  cameraOn = true;

  constructor(private telloService: TelloService, private zone: NgZone) {}

  ngAfterViewInit() {
    this.initJoysticks();
  }
}
```

Tambahkan method init Joystick untuk menginisialisasi 2 joystick menggunakan pustaka nipplejs di home.page.ts

```
● ● ●

initJoysticks() {
  const movementZone = document.getElementById('movement-joystick');
  const directionalZone = document.getElementById('directional-joystick');

  if (movementZone && directionalZone) {
    this.movementJoystick = nipplejs.create({
      zone: movementZone,
      mode: 'static',
      position: { left: '50%', bottom: '62px' },
      size: 120,
      color: 'blue',
      restOpacity: 1,
      dynamicPage: true,
    });

    this.directionalJoystick = nipplejs.create({
      zone: directionalZone,
      mode: 'static',
      position: { left: '50%', bottom: '62px' },
      size: 120,
      color: 'blue',
      restOpacity: 1,
      dynamicPage: true,
    });
  }
}
```

Membuat Joystick Drone Controller

lanjutan kode diatas:

```
this.movementJoystick.on('move', (_evt: any, data: any) => {
  this.handleMovementJoystick(data);
});

this.directionalJoystick.on('move', (_evt: any, data: any) => {
  this.handleDirectionalJoystick(data);
}

this.movementJoystick.on('end', () => this.resetRC());
this.directionalJoystick.on('end', () => this.resetRC());
} else {
  console.error('Joystick elements not found');
}
}
```

Tambahkan fungsi untuk menghandle joystick kanan dan joystick kiri dan joystick kanan

```
● ● ●

handleDirectionalJoystick(data: any) {
  if (this.canSendCommand) {
    const angle = data.angle.degree;
    const strength = Math.round(data.distance);

    if (angle > 45 && angle <= 135) {
      this.RC[2] = strength; // Naik
    } else if (angle > 226 && angle <= 315) {
      this.RC[2] = -strength; // Turun
    } else if (angle > 135 && angle <= 225) {
      this.RC[3] = -strength; // Putar Kiri
    } else {
      this.RC[3] = strength; // Putar Kanan
    }

    this.sendCommand(`rc ${this.RC[0]} ${this.RC[1]} ${this.RC[2]} ${this.RC[3]}`);
  }
}

handleMovementJoystick(data: any) {
  if (this.canSendCommand) {
    const angle = data.angle.degree;
    const strength = Math.round(data.distance);

    if (angle > 45 && angle <= 135) {
      this.RC[1] = strength; // Kiri
    } else if (angle > 226 && angle <= 315) {
      this.RC[1] = -strength; // Kanan
    } else if (angle > 135 && angle <= 225) {
      this.RC[0] = -strength; // Mundur
    } else {
      this.RC[0] = strength; // Maju
    }

    this.sendCommand(`rc ${this.RC[0]} ${this.RC[1]} ${this.RC[2]} ${this.RC[3]}`);
  }
}
```

Membuat Joystick Drone Controller

async sendCommand(command: string)

- Mengirim perintah ke drone jika canSendCommand bernilai true.
- Mencetak perintah ke konsol sebelum mengirimkannya ke TelloService.

resetRC()

- Mereset remote control (RC) ke [0, 0, 0, 0] (tidak ada input).
- Mengirim perintah 'rc 0 0 0 0' untuk menghentikan pergerakan drone.

async takeOff()

- Mencetak log "Perintah Take Off dikirim".
- Mengirim perintah 'takeoff' untuk menerbangkan drone.

async land()

- Mencetak log "Perintah Landing dikirim".
- Mengirim perintah 'land' untuk mendaratkan drone.



```
async sendCommand(command: string) {
    if (this.canSendCommand) {
        console.log(`Mengirim perintah: ${command}`);
        await this.telloService.sendCommand(command);
    }
}

resetRC() {
    this.RC = [0, 0, 0, 0];
    this.sendCommand('rc 0 0 0 0');
}

async takeOff() {
    console.log('Perintah Take Off dikirim');
    await this.sendCommand('takeoff');
}

async land() {
    console.log('Perintah Landing dikirim');
    await this.sendCommand('land');
}
```

Mini Notes Android Development With Ionic

Build Application:

1. ionic cordova build android --prod --release
2. pindah ke folder release
3. kalo belum pernah bikin file tanda tangan, bikin dulu pake perintah ini
keytool -genkey -v -keystore my-release-key.jks -keyalg RSA -keysize 2048 -validity 10000 -alias my-alias
4. jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-releasekey.jks app-release-unsigned.apk my-alias

Running Application Via Browser:

- ionic serve

Running Application Via Android:

- ionic cordova run android –livereload

Generate New Icon and Splash:

- ionic cordova resources android

Generate New Page(Halaman Baru):

- ionic generate page

Mini Notes Android Development With Ionic

Prepare New Application:

ionic start

Install Ionic on Your Computer for Android Development:

1. Install node js
2. Install android studio
3. Install JDK
4. Install JRE
5. Extract gradle on Drive C/Local Disk C
6. Set environment variable for

ANDROID_HOME

JAVA_HOME

Gradle

7. Install cordova, ionic, angular cli, cordova-res, native-run via CMD with these command:

- npm install -g @ionic/cli
- npm install -g @angular/cli
- npm install -g cordova
- npm install -g cordova-res
- npm install -g native-run