

Nama : Ramdan Hermawan
NIM : 22060021
Kelas : Informatika Pagi
UTS : -Aplikasi Berbasis Platform

Latihan 1

Fitur	Framwork7	Quasar	Ionic Framwork
Bahasa Pemrograman	JavaScript, HTML, CSS	JavaScript, Vue, dan juga mendukung TypeScript	JavaScript, TypeScript
Komponen UI	Beragam komponen UI khusus untuk aplikasi mobile	Komponen UI lengkap untuk aplikasi mobile dan desktop	Beragam komponen UI yang mudah di-customisasi
Platform	iOS, Android, Web (terutama berfokus pada mobile)	iOS, Android, Web, Windows, Mac, Linux	iOS, Android, Web
Desain	Meniru tampilan native iOS dan Material Design	Menyediakan dukungan untuk Material Design, iOS, dan bahkan Dark Mode	Meniru tampilan native iOS dan Material Design
CLI (Command Line Interface)	Tidak ada CLI khusus, menggunakan CLI eksternal	Memiliki CLI sendiri untuk konfigurasi proyek	CLI yang mudah digunakan untuk pengaturan proyek
Framework Pendukung	Native JavaScript, Vue, React	Vue	Angular, React, Vue, Svelte
PWA dan Aplikasi Web	Dukungan terbatas untuk PWA	Dukungan penuh untuk PWA, SSR (Server-Side Rendering)	Dukungan penuh untuk PWA
Integrasi Native	Terbatas, memerlukan plugin tambahan	Menggunakan Capacitor atau Cordova	Memiliki integrasi langsung dengan Capacitor
Dokumentasi	Lengkap, fokus pada mobile	Sangat lengkap, juga fokus pada desktop dan mobile	Lengkap, fokus pada mobile
Komunitas dan Ekosistem	Cukup besar, terutama pengguna mobile	Komunitas berkembang pesat, ekosistem Vue sangat besar	Komunitas besar dengan dukungan enterprise

Latihan 2

uts pa pajar > main.ts ---
// Kode TypeScript menggunakan Deno untuk class Person dan Pegawai

```
1 class Person { // <1>
2   perNik: string; // <2>
3   protected perNama: string;
4   perAlamat: string;
5
6   constructor(nik: string, nama: string) { // <3>
7     this.perNik = nik;
8     this.perNama = nama;
9   }
10 }
11
12 class Pegawai extends Person { // <4>
13   pegNpp: string; // <5>
14   private _pegJmlIanggunan: number;
15   readonly dept: string;
16   gaji: number;
17   static potongPajak = 10; // <6>
18
19   constructor(nik: string, npp: string, // <7>
20     nama: string, dept: string) {
21     super(nik, nama);
22     this.pegNpp = npp;
23     this.dept = dept;
24   }
25
26   getGaji(): number { // <8>
27     return this.gaji;
28   }
29 }
```

```
30
31   setGaji(gajiBaru: number): void { // <9>
32     this.gaji = gajiBaru;
33   }
34
35   getPotonganPajak(): number { // <10>
36     return this.gaji * (Pegawai.potongPajak / 100);
37   }
38
39   presensi(): void { // <11>
40     let dateTime = new Date();
41     console.log("Presensi pada " +
42       dateTime.toLocaleTimeString() +
43       ' - ' + dateTime.toDateString());
44   }
45 }
46
47 let pakBangang = new Pegawai('nik122', 'npp123', // <12>
48   'Bangang Purnomosidi', 'IT');
49 pakBangang.setGaji(15750500); // <13>
50 console.log(pakBangang.getGaji());
51 pakBangang.presensi();
52 console.log(pakBangang.getPotonganPajak());
53
54 // Error:
55 // Property '_pegJmlIanggunan' is private and only
56 // accessible within class 'Pegawai'
57 // console.log(pakBangang._pegJmlIanggunan);
58 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\User\Desktop\uts pa pajar> deno run --allow-read main.ts
15750500
Presensi pada 12:26:12 - Wed Nov 13 2024
1575050
PS C:\Users\User\Desktop\uts pa pajar> █
```

Latihan 3

```
UNTITLED...  ts > src > contoh.ts > ...
  ts
  dist
  contoh.js
  node_modules
  src
    contoh.ts
    contoh.txt
    package-lock.json
    package.json
    tsconfig.json
1  import { promises as fs } from 'fs';
2
3  async function readFileAsync(path: string): Promise <void>
4  {
5      try {
6          const data = await fs.readFile(path, 'utf8');
7          console.log(data);
8      } catch (error) {
9          console.error('Error Reading File', error);
10     }
11 }
12
13
14 const filePath = 'src/contoh.txt';
15 readFileAsync(filePath);
```

```
PS C:\Users\User\Desktop\ts> node dist/contoh.js
Ramdan Hermawan
PS C:\Users\User\Desktop\ts> 
```

```
> outdir
1  import express from "express";
2  import { Request, Response } from "express";
3  import * as fs from "fs";
4
5  const app = express();
6  const port = 3000;
7
8  const jsonData = JSON.parse(fs.readFileSync("./src/resources/maps.json", "utf-8"));
9
10 app.get("/data", (req: Request, res: Response) => {
11     res.json(jsonData);
12 });
13
14 app.listen(port, () => {
15     console.log(`Server berjalan di ${port}`);
16 });
17
```

```
{
  "markers": [
    {
      "name": "Rixos The Palm Dubai",
      "position": [25.1212, 55.1535]
    },
    {
      "name": "Shangri-La Hotel",
      "location": [25.2084, 55.2719]
    },
    {
      "name": "Grand Hyatt",
      "location": [25.2285, 55.3273]
    }
  ]
}
```