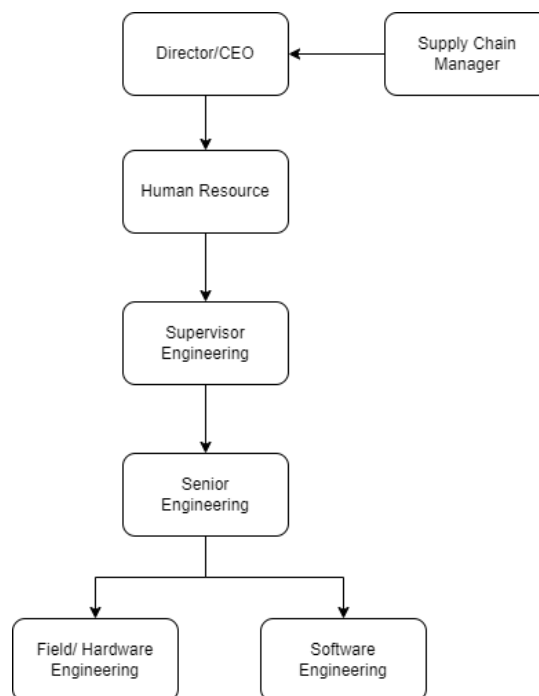


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama proses magang di PT Infra Datatech Indonesia, posisi penulis adalah sebagai Software Engineering. Saat magang, penulis berkoordinasi dengan Bapak Zulfikili sebagai Director/CEO sekaligus menjadi supervisor di tempat magang penulis. Selama magang penulis berada di posisi Software Engineering dibawah posisi Engineering.



Gambar 3.1 Struktur Perusahaan

Untuk posisi Software Engineering terdapat dua orang, yaitu:

Nama: Fina Valentina

Asal Kampus: Universitas Multimedia Nusantara

Jurusan: Teknik Komputer

Angkatan: 2021

Dan juga satu orang anak magang juga dengan data pribadi dibawah ini:

Nama: Fauzi Salman

Asal Kampus: Universitas Pendidikan Indonesia

Jurusan: Mekatronika

Angkatan: 2021

3.2 Tugas dan Uraian Kerja Magang

3.2.1 Tugas Kerja Magang

Berikut deskripsi tugas yang dilakukan oleh penulis selama magang yang berlangsung selama 10 bulan

Tabel 3.1. Tugas Kerja Magang

Minggu	Tugas yang dilakukan	Tanggal Mulai	Tanggal Selesai
1	Pengenalan situasi magang, Pengenalan lingkungan, Mempelajari alur tugas	01 Februari 2024	07 Februari 2024
2	Mempelajari konsep dasar flutter, Mempelajari slicing UI pada flutter.	08 Februari 2024	16 Februari 2024
3-5	Melakukan slicing UI pada flutter (login page dan home page).	19 Februari 2024	04 Maret 2024
6	Mempelajari Date & Time Handling.	05 Maret 2024	15 Maret 2024
7	Melakukan develop pembuatan date & time supaya pada saat user absen muncul tanggal dan jam absen masuk dan absen pulang.	18 Maret 2024	22 Maret 2024
8	Mempelajari Geolocation untuk membuat aplikasi absen hanya bisa melakukan absen sesuai	25 Maret 2024	29 Maret 2024

	dengan titik koordinat yang telah ditentukan, Mengkonsumsi API get presence.		
9-12	Melakukan develop geolocation supaya user hanya bisa melakukan absen sesuai dengan titik koordinat yang telah ditentukan, Testing aplikasi.	01 April 2024	22 April 2024
13-20	Membantu mengawasi teknisi instalasi lapangan.	23 April 2024	05 Juni 2024
21-37	Pembahasan tugas baru untuk membuat website company untuk brand baru, dan pembuatan website.	06 Juni 2024	23 Sepetember 2024
39	Deploy ke hosting.	24 September 2024	24 September 2024

3.2.2 Penjelasan Tugas

Pada aplikasi absen ini terdapat empat halaman yaitu login page, home page, permohonan perbaikan absen, dan profile page. Setiap halaman mempunyai fungsi dan fitur masing-masing. Untuk menggambarkan fungsi dan fitur yang terdapat di masing-masing page yang telah disebutkan diatas, penulis akan menjelaskan secara singkat melalui tabel 3.2 dibawah ini:

Tabel 3.2. Penjelasan Tugas Singkat

Halaman	Fungsi	Fitur
Login	Supaya user dapat masuk ke akun mereka yang telah terdaftar sebelumnya dengan memasukkan user ID dan password, supaya user dapat melakukan absen.	Login menggunakan user id dan password menggunakan API.
Home	User dapat melakukan absen.	<ul style="list-style-type: none"> - Mendeteksi lokasi terkini user - Perhitungan jarak user berada sekarang dengan titik koordinat - Melakukan absen dengan mengambil titik longitude dan latitude pada maps
Permohonan Perbaikan Absen	User dapat melakukan perbaikan absen dengan cara mengisi formulir dan diteruskan ke Pak Zulkifli.	<ul style="list-style-type: none"> - Formulir perbaikan absen - Pengiriman pesan otomatis ke whatsapp

Untuk menjelaskan lebih detail per fitur yang terdapat pada aplikasi absensi ini, penulis akan menjelaskan secara detail termasuk kode yang digunakan untuk membuat fitur tersebut beserta langkah-langkah pengembangannya.

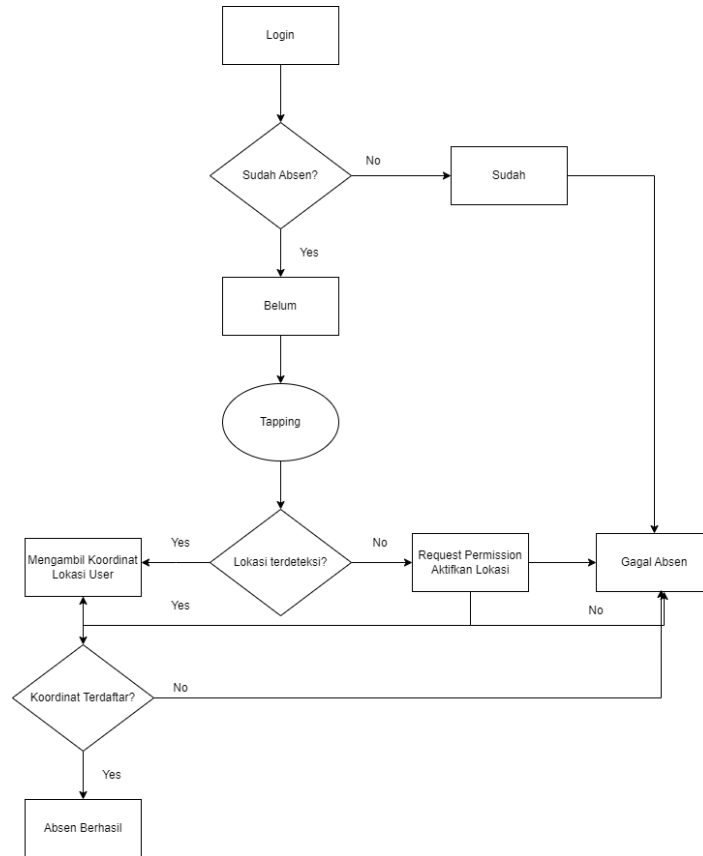
Mulai dari perancangan antarmuka pengguna hingga implementasi logika yang mendukung setiap fitur.

Penjelasan ini mencakup proses pembuatan antarmuka pengguna (UI), pengaturan database untuk menyimpan data absensi, integrasi dengan API untuk validasi lokasi pengguna, serta penerapan notifikasi otomatis untuk mengingatkan karyawan tentang absensi. Setiap bagian kode akan dijelaskan fungsinya untuk memudahkan pemahaman alur kerja aplikasi dan bagaimana setiap fitur saling berhubungan untuk memberikan pengalaman pengguna yang optimal.

Membuat fitur absensi

Fitur utama pada aplikasi ini ialah fitur untuk melakukan absensi, untuk diagram alur proses cara user melakukan absen dapat dilihat pada gambar 3.2. Pertama, user akan login terlebih dahulu ke akun mereka saat membuka aplikasi absen tersebut. Lalu, aplikasi akan mengecek apakah user telah melakukan absen atau belum.

Jika sudah maka user tidak dapat melakukan absen. Akan tetapi jika user belum melakukan absen maka user dapat melakukan tapping absen. Ketika user melakukan tapping absen maka aplikasi akan mengecek lokasi user sekarang, apakah user tersebut berada di lokasi yang terdaftar. Jika tidak maka user tidak dapat melakukan absen. Sedangkan, jika user berada di lokasi yang terdaftar maka user dapat melakukan absen.



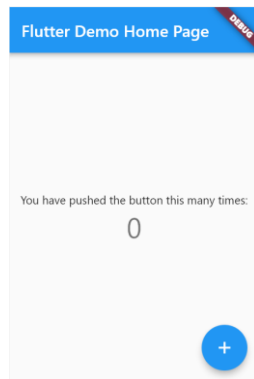
Gambar 3.2 Diagram Alur Proses User Melakukan Absen

Membuat tampilan aplikasi absensi dan fitur pada aplikasi

Pada pengembangan aplikasi android untuk membuat user interface (UI) diperlukan membuat slicing UI terlebih dahulu. Slicing UI ini bertujuan untuk kita membagi menjadi komponen-komponen kecil untuk mengimplementasikan desain user interface (UI) ke dalam bentuk kode bahasa pemrograman. Proses ini memungkinkan pengembang untuk memastikan bahwa setiap bagian dari antarmuka pengguna, mulai dari tombol, ikon, gambar, hingga tata letak teks, dapat diimplementasikan dengan presisi sesuai dengan desain awal.

Untuk mengimplementasikan *slicing* UI, kita perlu terlebih dahulu membentuk kerangka dasar dari aplikasi tersebut, yang umumnya dilakukan dengan menggunakan *scaffold* sebagai struktur utama. Dalam bahasa pemrograman Flutter, untuk membuat tampilan antarmuka pengguna (UI) yang menarik dan responsif, kita dapat menambahkan berbagai elemen seperti teks, gambar, ikon, dan elemen lainnya, yang semuanya diwakili dalam bentuk *widget*. *Widget* ini berfungsi sebagai komponen dasar yang digunakan untuk menyusun dan mengatur tampilan, serta memberikan interaksi yang baik bagi pengguna aplikasi. Setiap elemen visual dalam Flutter merupakan bagian dari hierarki *widget* yang memungkinkan kita untuk membangun UI yang fleksibel dan dinamis. Ada beberapa jenis *widget* dalam Flutter, di antaranya:

1. **Stateless Widget:** *Widget* yang tidak memiliki state atau kondisi internal yang dapat berubah selama siklus hidupnya. Dengan kata lain, setelah *widget* tersebut dirender, ia akan tetap sama sepanjang waktu dan tidak akan mengalami perubahan data atau tampilan. *Stateless widget* biasanya digunakan untuk elemen yang statis dan tidak membutuhkan interaksi atau pembaruan konten secara dinamis. Beberapa contoh umum dari *stateless widget* termasuk *Text*, *Icon*, dan *Container*, yang tampilannya ditentukan oleh parameter awal tanpa dipengaruhi oleh perubahan data. Berikut merupakan implementasi dari *stateless widget* pada gambar 3.3:



Gambar 3.3 Stateless Widget

Pada saat awal kita membuat proyek baru di Flutter, framework ini secara otomatis akan menghasilkan kerangka dasar atau struktur awal aplikasi yang sudah siap digunakan. Kerangka ini mencakup pengaturan dasar dari proyek Flutter, seperti folder *lib*, file *main.dart* sebagai titik awal aplikasi, serta beberapa konfigurasi penting yang diperlukan untuk menjalankan proyek. Dengan adanya kerangka ini, kita tidak perlu memulai semuanya dari nol. Kita hanya perlu menambahkan kode-kode tertentu untuk membangun tampilan antarmuka pengguna (UI) sesuai dengan kebutuhan dan desain yang kita inginkan.

2. **Stateful Widget:** Widget yang memiliki state atau kondisi internal yang dapat berubah selama siklus hidupnya, memungkinkan pembaruan tampilan ketika ada perubahan pada data atau kondisi tertentu. Berbeda dengan stateless widget, stateful widget dapat beradaptasi dan menampilkan perubahan secara dinamis berdasarkan interaksi pengguna atau modifikasi data lainnya. Penggunaan stateful widget sangat tepat untuk elemen yang membutuhkan interaksi berkelanjutan, seperti saat pengguna melakukan input atau memanipulasi data yang mempengaruhi tampilan.

Beberapa contoh umum dari stateful widget adalah *Checkbox*, yang memungkinkan pengguna memilih atau menghapus pilihan, *TextField*, yang menerima masukan teks secara dinamis, dan *AnimatedContainer*, yang memungkinkan perubahan animasi berdasarkan kondisi atau state yang berbeda.

Pada aplikasi absensi ini diimplementasikan di halaman login, saat user sudah login ke akun user maka tampilan akan berubah ke halaman home page. Nah perubahan tampilan ini kita butuhkan stateful widget. Contoh penerapan stateful widget pada aplikasi absensi pada gambar 3.4:

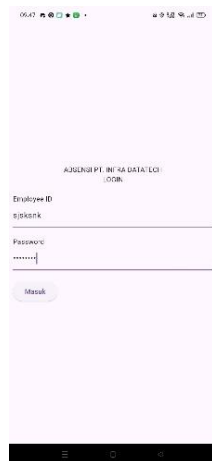


```
lib > login-page.dart > _LoginPageState
9 class LoginPage extends StatefulWidget {
10   const LoginPage({Key? key}) : super(key: key);
11
12   @override
13   State<LoginPage> createState() => _LoginPageState();
14 }
15
16 class _LoginPageState extends State<LoginPage> {
17   final Future<SharedPreferences> _prefs = SharedPreferences.getInstance();
18   TextEditingController emailController = TextEditingController();
19   TextEditingController passwordController = TextEditingController();
20   late Future<String> _name, _token;
21
22   @override
23   void initState() {
24     // TODO: implement initState
25     super.initState();
26     _token = _prefs.then((SharedPreferences prefs) {
27       return prefs.getString("token") ?? "";
28     });
29
30     _name = _prefs.then((SharedPreferences prefs) {
31       return prefs.getString("name") ?? "";
32     });
33     checkToken(_token, _name);
34   }
35 }
```

Gambar 3.4 Contoh Penerapan Stateful Widget Pada Aplikasi Absensi

Penjelasan kode:

Untuk membuat halaman login kita membuat sebuah class bernama LoginPage, class LoginPage ini merupakan bagian dari StatefulWidget dikarenakan tampilan halaman aplikasi akan berubah ketika user sudah melakukan login. Lalu, kita membuat sebuah class LoginPageState untuk membuat statement atau hal yang akan dilakukan oleh class LoginPage itu sendiri. Di class LoginPageState ini kita akan membuat TextEditingController yang berguna untuk user dapat memasukkan email dan password saat melakukan login. Lalu dari email dan password yang sudah dimasukkan oleh user melalui TextEditingController selanjutnya email dan password tersebut akan di simpan oleh SharedPreferences untuk di cek apakah string dari token dan nama kosong atau tidak (user telah memasukkan email dan password atau belum). Jika tidak kosong maka string dari token dan nama akan diperiksa. Berikut tampilan halaman login yang string tidak kosong pada gambar 3.5:



Gambar 3.5 Tampilan Halaman Login

```

//membuat halaman login
@override
Codeium: Refactor | Explain | Generate Function Comment | X
Widget build(BuildContext context) {
  return Scaffold(
    body: SafeArea(
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Center(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Center(child: Text("ABSENSI_PT_INFRA_DATATECH")),
              Center(child: Text("LOGIN")),
              SizedBox(height: 20),
              Text("Employee ID"),
              TextField(
                controller: emailController,
              ), // TextField
              SizedBox(height: 20),
              Text("Password"),
              TextField(
                controller: passwordController,
                obscureText: true,
              ), // TextField
              SizedBox(height: 20),
              ElevatedButton(
                onPressed: () {
                  login(emailController.text, passwordController.text);
                },
                child: Text("Masuk")) // ElevatedButton
            ],
          ), // Column
        ), // Center
      ), // Padding // SafeArea
    ); // Scaffold
  }

```

Gambar 3.6 Kode Untuk Membuat Tampilan Halaman Login

Pada gambar 3.5, kode di atas merupakan kode untuk membuat tampilan dari halaman login itu sendiri. Kode Widget build(BuildContext context) berfungsi untuk membangun tampilan UI di flutter. Pada kode tersebut artinya kita akan membangun tampilan UI dengan mengimplementasikan struktur tata letak visual Desain Material dasar. Selanjutnya kita memastikan area yang kita akan membangun tampilan UI tidak oleh bagian pengganggu sehingga tidak terjadi pemupukkan dan bisa dijalankan dengan baik. Untuk memastikan area aman kita memerlukan padding untuk memberi jarak kosong di dalam widget, pada kode kita mengatur padding sebesar 8 piksel (8.0). Lalu kita gunakan center untuk membuat tulisan berada ditengah. Terakhir kita gunakan emailController dan passwordController untuk membuat kotak untuk user dapat memasukkan email dan password.

Memanggil API Login

Pada saat user belum memasukkan employee id dan password maka string masih bernilai kosong. Akan tetapi setelah user memasukkan employee id dan password pada halaman login aplikasi, maka string tidak bernilai kosong. Setelah string pada text input tidak kosong, maka input data yang berupa email (employee id) dan password akan di cek apakah terdaftar di database yang tersedia atau tidak menggunakan kode pada gambar 3.7 dibawah ini:

```
48 | }
49 | //untuk fetch data login sesuai yang ada di api
50 | Future login(String email, String password) async {
51 |   LoginResponseModel? loginResponseModel;
52 |   Map<String, String> body = {"email": email, "password": password};
53 |
54 |   try {
55 |     var response = await myHttp.post(
56 |       Uri.parse('https://datatech.co.id/indexLaravel.php/api/login'),
57 |       body: body,
58 |     );
59 |
60 |     if (response.statusCode == 200) {
61 |       loginResponseModel =
62 |         LoginResponseModel.fromJson(json.decode(response.body));
63 |       saveUser(loginResponseModel.data.token, loginResponseModel.data.name);
64 |     } else if (response.statusCode == 401) {
65 |       showErrorSnackBar("Email atau password salah");
```

Gambar 3.7 Kode Untuk Menarik API Login

Pada saat user melakukan login, aplikasi akan mengecek username dan password user tersebut apakah terdaftar atau tidak. Jika user tersebut tidak terdaftar maka terdapat tulisan email atau password salah. Jika user terdaftar maka muncul tulisan welcome pada terminal vs code. Selain error user tidak ditemukan, juga terdapat error-error lainnya, pada gambar 3.8 merupakan kode jika berhasil login dan terdapat error.

```

82     }
83     //jika berhasil login data user disimpan supaya user ketika keluar dari aplikasi tidak perlu login ulang
84     Future saveUser(token, name) async {
85         try {
86             print("WELCOME " + token + " | " + name);
87             final SharedPreferences pref = await _prefs;
88             pref.setString("name", name);
89             pref.setString("token", token);
90             Navigator.of(context)
91                 .push(MaterialPageRoute(builder: (context) => HomePage()))
92                 .then((value) {
93                     setState(() {});
94                 });
95         } catch (err) {
96             print('ERROR : ' + err.toString());
97             ScaffoldMessenger.of(context)
98                 .showSnackBar(SnackBar(content: Text(err.toString())));
99         }
100     }

```

Gambar 3.8 Kode Ketika Berhasil Login

Pada saat user login dengan cara memasukkan employee id dan password yang sesuai dengan user id dan password yang terdaftar maka user berhasil melakukan login. Setelah user berhasil melakukan login user akan diarahkan ke home page. Kode pada gambar 3.9 dibawah ini merupakan respon jika email dan password sesuai dengan database yang terdaftar di API dan mengarahkan user ke home page setelah user berhasil melakukan login.

```

Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) => HomePage()))
    .then((value) {
        setState(() {});
    });

```

Gambar 3.9 Kode Untuk Mengarahkan Ke Halaman Home Page

Membuat tampilan Home Page

Setelah membuat tampilan halaman login, selanjutnya kita akan membuat tampilan home page. Ketika user telah melakukan login dan masuk ke home page maka aplikasi akan mengecek status presensi masuk, status presensi keluar, titik koordinat user sekarang (longitude dan latitude), dan versi aplikasi yang digunakan oleh user. Untuk memeriksa semua hal yang disebutkan sebelumnya, kita akan gunakan kode dibawah ini. Kode progress dialog digunakan untuk membuat kotak progress loading untuk update aplikasi jika versi aplikasi belum menggunakan versi terbaru. Untuk kode nya kita dapat gunakan kode pada gambar 3.10 dibawah ini:

```
class _HomePageState extends State<HomePage> {
  final Future<SharedPreferences> _prefs = SharedPreferences.getInstance();
  late Future<String> _name, _token;
  HomeResponseModel? homeResponseModel;
  Datum? hariIni;
  List<Datum> riwayat = [];
  List<dynamic> _notifications = [];
  List<dynamic> _usernotifications = [];
  bool isPresensiMasuk = false;
  bool isPresensiKeluar = false;
  String? previousLatitude;
  String? previousLongitude;
  String _latestVersion = '';
  String _updateUrl = '';

  bool isMockLocation = false;
  Location location = Location();
  double? _latitude;
  double? _longitude;
  ProgressDialog? _progressDialog;
}
```

Gambar 3.10 Kode Untuk Membuat Update Versi Aplikasi

Pada saat user telah melakukan login dan masuk ke halaman home page. Maka aplikasi akan mengecek apakah user telah melakukan absen atau belum melalui token dan nama user. Untuk mendapatkan token dan nama dibutuhkan shared preference yang akan mengirimkan token dan nama yang terdaftar ke aplikasi untuk di cek. Untuk mengecek status absensi user dapat gunakan kode dibawah ini, pada gambar 3.11:

```
@override
Codeium: Refactor | Explain | Generate Function Comment | X
void initState() {
    super.initState();
    _checkForUpdate();
    _token = _prefs.then((SharedPreferences prefs) {
        return prefs.getString("token") ?? "";
    });
    _name = _prefs.then((SharedPreferences prefs) {
        return prefs.getString("name") ?? "";
    });
    _checkPresensiStatus();
}
```

Gambar 3.11 Kode Untuk Mengecek Status Absensi User

Jika aplikasi telah melakukan pengecekan versi aplikasi yang digunakan oleh user, dan user belum menggunakan versi aplikasi terbaru. Maka pada home page aplikasi absen user tersebut akan menampilkan kotak yang berisi permintaan untuk update aplikasi. Untuk mengecek versi aplikasi terbaru kita menggunakan kode pada gambar 3.12 untuk menarik API.

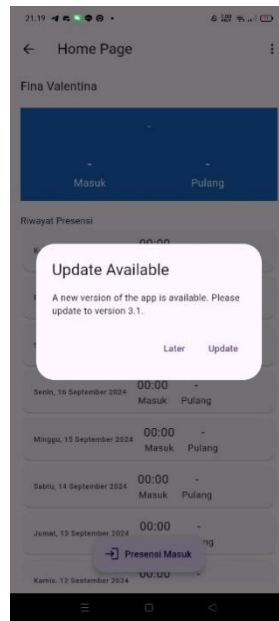

```

void _showUpdateDialog() {
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text('Update Available'),
      content: Text(
        'A new version of the app is available. Please update to version $_latestVersion.'), // Text
      actions: [
        TextButton(
          onPressed: () => Navigator.of(context).pop(),
          child: Text('Later'),
        ), // TextButton
        TextButton(
          onPressed: _downloadAndInstallApk,
          child: Text('Update'),
        ), // TextButton
      ],
    ), // AlertDialog
  );
}

```

Gambar 3.12 Kode Untuk Mengecek Versi Aplikasi Terbaru

Setelah menarik API, aplikasi akan melakukan pengecekan untuk memastikan apakah terdapat versi terbaru dari aplikasi tersebut atau tidak. Jika versi terbaru terdeteksi, maka aplikasi akan menampilkan sebuah kotak dialog atau notifikasi yang meminta pengguna untuk memperbarui aplikasi ke versi terbaru. Kotak permintaan ini biasanya berisi informasi mengenai perubahan atau fitur baru yang tersedia dalam versi terbaru, serta opsi untuk segera mengunduh dan menginstal pembaruan. Dengan demikian, pengguna dapat dengan mudah memperbarui aplikasi agar tetap mendapatkan performa terbaik, fitur terbaru, serta peningkatan keamanan yang mungkin disertakan dalam update tersebut, seperti tampilan gambar 3.13:



Gambar 3.13 Tampilan Kotak Permintaan Update Aplikasi

Jika user klik button update pada kotak permintaan update ke versi terbaru. Maka aplikasi akan update ke versi terbaru, dan ketika user klik update aplikasi akan install ke versi terbaru. Kode pada gambar 3.14 dibawah ini untuk melakukan install update ke versi aplikasi terbaru:

```

Future<void> _downloadAndInstallApk() async {
  _progressDialog = ProgressDialog(
    context,
    type: ProgressDialogType.download,
    textDirection: TextDirection.rtl,
    isDismissible: true,
  );
  _progressDialog!.style(
    message: 'Downloading Update...',
    borderRadius: 10.0,
    backgroundColor: Colors.white,
    elevation: 10.0,
    messageTextStyle: const TextStyle(
      color: Colors.black, fontSize: 19.0, fontWeight: FontWeight.w600),
  );
  _progressDialog!.show();
  final directory = await getExternalStorageDirectory();
  final savePath = '${directory!.path}/presensi.apk';

  try {
    await Dio().download(
      _updateUrl,
      savePath,
      onReceiveProgress: (received, total) {
        print('Received: $received, Total: $total');
      },
    );
    _progressDialog!.hide();
    print('Downloaded APK to: $savePath');
  } catch (e) {
    _progressDialog!.hide();
    print('Failed to download APK: $e');
    ScaffoldMessenger.of(context).showSnackBar(

```

Gambar 3.14 Kode Untuk Downloading Versi Aplikasi Terbaru

Pada saat user masuk ke aplikasi absensi aplikasi akan mengecek apakah user sudah melakukan absen atau belum setiap harinya. Jika user telah melakukan absensi maka akan ditampilkan hari dan jam absen hari ini. Untuk mengecek apakah user telah melakukan absen atau belum dengan menggunakan kode pada gambar 3.15 dibawah ini:

```

Future _checkPresensiStatus() async {
  await getData();

  if (homeResponseModel != null && homeResponseModel!.data.isNotEmpty) {
    final hariIniData =
      homeResponseModel!.data.firstOrNull((presensi) => presensi.isHariIni);
    if (hariIniData != null) {
      if (hariIniData.masuk != null) {
        setState(() {
          isPresensiMasuk = true;
          isPresensiKeluar = true;
        });
      }
    }
  }
}

```

Gambar 3.15 Kode Untuk Mengecek Apakah User Telah Melakukan Absen

Jika belum, supaya user dapat melakukan absensi masuk pada aplikasi tersebut dengan menggunakan kode pada gambar 3.16 dibawah ini. Ketika user melakukan absensi masuk aplikasi akan mengecek lokasi user berada sekarang. Jika user tidak berada di titik koordinat yang sudah di daftarkan maka user tidak dapat melakukan absensi masuk.

```

Future<void> _presensiMasuk() async {
  final LocationData? locationData = await _currentLocation();
  if (locationData != null) {
    _latitude = locationData.latitude;
    _longitude = locationData.longitude;

    if (_latitude != null && _longitude != null && !isMockLocation) {
      final String token = await _token;
      final result =
        await PresensiUtil.savePresensi(_latitude!, _longitude!, token);
      if (result['success']) {
        ScaffoldMessenger.of(context)
          .showSnackBar(SnackBar(content: Text(result['message'])));
        setState(() {
          isPresensiMasuk = true;
        });
      } else {
        ScaffoldMessenger.of(context)
          .showSnackBar(SnackBar(content: Text(result['message'])));
      }
    } else {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        content: Text('Tidak dapat mendapatkan lokasi saat ini.'))); // SnackBar
    }
  }
}

```

Gambar 3.16 Kode Untuk Melakukan Absen Masuk

Begitu pula dengan absen keluar, aplikasi akan mengecek kembali lokasi user sekarang dan apakah lokasi tersebut palsu atau tidak. Jika user berada di lokasi yang terdaftar maka user tidak dapat melakukan absen keluar. Untuk mengecek absen keluar menggunakan kode pada gambar 3.17 dibawah ini:

```
Future<void> _presensiKeluar() async {
  final LocationData? locationData = await _currentLocation();
  if (locationData != null) {
    _latitude = locationData.latitude;
    _longitude = locationData.longitude;

    if (_latitude != null && _longitude != null && !isMockLocation) {
      final String token = await _token;
      final result =
        await PresensiUtil.savePresensi(_latitude!, _longitude!, token);
      if (result['success']) {
        ScaffoldMessenger.of(context)
          .showSnackBar(SnackBar(content: Text(result['message'])));
        setState(() {
          isPresensiKeluar = true;
        });
      } else {
        ScaffoldMessenger.of(context)
          .showSnackBar(SnackBar(content: Text(result['message'])));
      }
    } else {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        content: Text('Tidak dapat mendapatkan lokasi saat ini.')); // SnackBar
    }
  }
}
```

Gambar 3.17 Kode Untuk Mengecek Absen Keluar

Untuk dapat melakukan absensi, aplikasi harus mendapatkan titik koordinat lokasi user berada sekarang dengan cara memanggil api daftar koordinat melalui kode pada gambar 3.18 dibawah ini. API daftar koordinat ini merupakan daftar titik koordinat lokasi mana saja yang user dapat melakukan absensi. Jika lokasi tidak terdaftar maka user tersebut tidak dapat melakukan absensi.

```

class PresensiUtil {
    static List<double> targetLatitudes = [];
    static List<double> targetLongitudes = [];

Codeium: Refactor | Explain | Generate Function Comment | X
    static Future<void> fetchTargetCoordinates() async {
        var response = await myHttp.get(
            Uri.parse('https://datatech.co.id/indexLaravel.php/get-coordina
        var data = json.decode(response.body);
        print(data);
        if (data['success']) {
            targetLatitudes = List<double>.from(
                data['data'].map((item) => double.parse(item['latitude'])));
            targetLongitudes = List<double>.from(
                data['data'].map((item) => double.parse(item['longitude'])));
        } else {
            throw Exception('Failed to load target coordinates');
        }
    }
}

```

Gambar 3.18 Kode Untuk Mengambil Titik Koordinat yang sudah di daftar

Jika lokasi user tidak dapat terdeteksi maka aplikasi akan meminta user untuk menyalakan lokasi di handphone nya. User menyalakan lokasi supaya aplikasi dapat mendeteksi lokasi user apakah sesuai atau tidak. Dari kode gambar 3.19 dibawah ini akan membuat kotak permission menyalakan lokasi user:

```
//jika Lokasi tidak terdeteksi maka aplikasi meminta user untuk menyalakan Lokasi
Codeium: Refactor | Explain | Generate Function Comment | X
Future<LocationData?> _currentLocation() async {
  bool serviceEnabled;
  PermissionStatus permissionGranted;

  serviceEnabled = await location.serviceEnabled();
  if (!serviceEnabled) {
    serviceEnabled = await location.requestService();
    if (!serviceEnabled) {
      return null;
    }
  }

  permissionGranted = await location.hasPermission();
  if (permissionGranted == PermissionStatus.denied) {
    permissionGranted = await location.requestPermission();
    if (permissionGranted != PermissionStatus.granted) {
      return null;
    }
  }
}
```

Gambar 3.19 Kode Jika Lokasi Tidak Terdeteksi

Setelah user melakukan absen, maka absen tersebut akan disimpan melalui kode pada gambar 3.20 dibawah ini. Nantinya data riwayat absensi akan tersimpan di handphone user dan disimpan di website hasil rekap presensi yang hanya bisa diakses oleh Pak Zulfikli. Pada tampilan website tersebut akan menampilkan nama, hari, jam absen masuk, jam absen pulang, lama kerja, lama lembur, dan lokasi. Seperti pada tabel gambar 3.21 dibawah ini:

Nama	Waktu	Masuk	Pulang	Lama Kerja	Lama Lembur	Lokasi (Latitude, Longitude)
A. Edi Suwardi	Senin, 13 Mei 2024	07:25:16	-	-	-	-6.24309, 106.86593
A. Edi Suwardi	Kamis, 16 Mei 2024	08:43:14	21:35:56	08:16	03:35	-6.24307, 106.86591
A. Edi Suwardi	Jumat, 17 Mei 2024	07:49:26	21:40:53	09:10	03:40	-6.24321, 106.86591
A. Edi Suwardi	Sabtu, 18 Mei 2024	07:49:01	21:18:21	09:10	03:18	-6.24345, 106.86797
A. Edi Suwardi	Minggu, 19 Mei 2024	08:30:42	15:34:53	07:04	-	-6.24371, 106.86702
A. Edi Suwardi	Rabu, 22 Mei 2024	07:45:51	21:37:12	09:14	03:37	-6.24267, 106.86467
A. Edi Suwardi	Kamis, 23 Mei 2024	07:52:44	21:37:31	09:07	03:37	-6.24379, 106.86716
A. Edi Suwardi	Jumat, 24 Mei 2024	08:02:01	21:38:05	08:57	03:38	-6.24369, 106.86713
A. Edi Suwardi	Sabtu, 25 Mei 2024	07:57:08	16:56:47	08:59	-	-6.24368, 106.86715
A. Edi Suwardi	Minggu, 26 Mei 2024	08:08:57	17:22:30	08:51	-	-6.24278, 106.86518

Gambar 3.20 Data Absen Pada Web Owner

```
Codeium | Refactor | Explain | Generate Function | Comment | ...  
static Future savePresensi(  
    double latitude, double longitude, String token) async {  
    if (targetLatitudes.isEmpty || targetLongitudes.isEmpty) {  
        await fetchTargetCoordinates();  
    }  
}
```

Gambar 3.21 Kode Untuk Menyimpan Absensi

Maksimal jarak yang diizinkan untuk melakukan absen adalah sejauh 500 meter dari titik koordinat yang terdaftar di API, untuk menghitung jarak maksimal pada aplikasi menggunakan kode pada gambar 3.22 sampai dengan gambar 3.24 dibawah ini. User baru dapat melakukan absen yaitu pada jarak paling jauh dari titik koordinat yang sudah terdaftar dengan lokasi user sekarang. Jika jarak melebihi 500 meter maka user tersebut tidak dapat melakukan absen.


```
double maxDistance = 0.5; // Jarak maksimum yang diizinkan  
double nearestDistance = double.infinity;  
int nearestIndex = -1; // Jarak maksimum yang diizinkan
```

Gambar 3.22 Kode Untuk Menghitung Jarak

Untuk menentukan jarak maksimal yang diizinkan untuk karyawan dapat melakukan absen pertama kita buat variabel `maxDistance` dan kira berikan nilai 0.5 karena dalam satuan km. Variabel `nearestDistance` diinisialisasi dengan nilai `double.infinity`, yang berarti jaraknya diatur ke nilai tak hingga. Ini berfungsi sebagai nilai awal karena program akan mencoba mencari jarak yang lebih kecil dari ini dalam proses perhitungan. Setelah perhitungan, `nearestDistance` akan menyimpan jarak terdekat yang ditemukan. Variabel ini menginisialisasi `nearestIndex` dengan nilai -1. Nilai ini biasanya digunakan sebagai penanda bahwa belum ada indeks.

```
for (int i = 0; i < targetLatitudes.length; i++) {  
    double targetLatitude = targetLatitudes[i];  
    double targetLongitude = targetLongitudes[i];  
  
    double distance = calculateDistance(  
        latitude, longitude, targetLatitude, targetLongitude);  
  
    if (distance <= maxDistance && distance < nearestDistance) {  
        nearestDistance = distance;  
        nearestIndex = i;  
    }  
}
```

Gambar 3.23 Kode Untuk Menghitung Jarak

Kode `for (int i = 0; i < targetLatitudes.length; i++)` berfungsi menghitung jarak dari jarak user ke target titik koordinat yang terdaftar, jadi ketika user bergerak maka indeks `i` nya bertambah. Pada fungsi loop tersebut juga diambil nilai lokasi user sekarang. Apakah jarak yang dihitung (`distance`) lebih kecil atau sama dengan `maxDistance` (jarak maksimum yang diizinkan). Apakah jarak yang dihitung lebih kecil dari `nearestDistance`, yang berarti jarak tersebut adalah yang paling dekat sejauh ini. Jika kedua syarat tersebut terpenuhi, artinya titik target tersebut berada dalam jarak yang diizinkan dan merupakan jarak terdekat yang ditemukan.

```
static double calculateDistance(  
    double lat1, double lon1, double lat2, double lon2) {  
    const double radiusEarth = 6371.0; // Radius Bumi dalam  
  
    double dLat = degreesToRadians(lat2 - lat1);  
    double dLon = degreesToRadians(lon2 - lon1);  
    double a = sin(dLat / 2) * sin(dLat / 2) +  
        cos(degreesToRadians(lat1)) *  
        cos(degreesToRadians(lat2)) *  
        sin(dLon / 2) *  
        sin(dLon / 2);  
    double c = 2 * atan2(sqrt(a), sqrt(1 - a));  
    double distance = radiusEarth * c;  
  
    return distance;  
}  
  
Codeium: Refactor | Explain | Generate Function Comment | X  
static double degreesToRadians(double degrees) {  
    return degrees * pi / 180.0;  
}  
}
```

Gambar 3.24 Kode Untuk Menghitung Jarak

Kode `static double calculateDistance(double lat1, double lon1, double lat2, double lon2)` ini mendefinisikan fungsi `calculateDistance` yang menerima empat parameter:

- lat1 dan lon1: lintang dan bujur titik pertama (misalnya, lokasi user).
- lat2 dan lon2: lintang dan bujur titik kedua (lokasi target absensi)

Kode **const double radiusEarth = 6371.0;** berfungsi untuk mendefinisikan konstanta radiusEarth, yaitu jari-jari Bumi dalam kilometer (6371 km). Kode **double dLat = degreesToRadians(lat2 - lat1)** digunakan untuk menyimpan selisih antara lintang kedua titik (lat2 - lat1), dan hasilnya diubah dari derajat ke radian menggunakan fungsi degreesToRadians. Hal ini diperlukan karena perhitungan trigonometri menggunakan radian. Fungsi $\sin(dLat / 2)$ dan $\sin(dLon / 2)$ menghitung setengah dari sudut lintang dan bujur yang telah dikonversi ke radian. Kode $\cos(\text{degreesToRadians}(\text{lat1}))$ dan $\cos(\text{degreesToRadians}(\text{lat2}))$ menghitung kosinus dari lintang kedua titik. Lalu, semua hasil pada kode tadi dikalikan untuk mendapatkan nilai variabel a, yang merupakan bagian dari rumus Haversine untuk menghitung sudut jarak antara dua titik.

Pada kode **double c = 2 * atan2(sqrt(a), sqrt(1 - a))** variabel c digunakan untuk menghitung jarak sudut antara dua titik dengan menggunakan fungsi atan2, yang menghitung arctangent dari dua variabel (yaitu \sqrt{a} dan $\sqrt{1 - a}$), yang juga merupakan bagian dari rumus Haversine. Kode **double distance = radiusEarth * c** variabel distance digunakan untuk menyimpan hasil akhir jarak antara dua titik. Nilai c dikalikan dengan jari-jari Bumi (radiusEarth) untuk mendapatkan jarak sebenarnya dalam kilometer. Terakhir, **return distance** ini digunakan untuk mengembalikan jarak yang dihitung dalam bentuk double. Fungsi calculateDistance menghitung jarak "garis lurus" (atau lebih tepatnya, jarak "geodesik" yang mengikuti kelengkungan Bumi) antara dua titik yang diberikan, menggunakan rumus Haversine.

Jika user sudah berada sesuai maksimal jarak dari titik koordinat yang terdaftar maka absensi akan tersimpan, jika user tidak berada di lokasi yang benar maka user tidak dapat melakukan absen. Akan tetapi, jika titik koordinat user sudah sesuai dengan titik koordinat yang terdaftar, ketika user melakukan tapping absen tersebut akan tersimpan dan diarahkan ke website yang hanya bisa di akses sama Pak Zulfikli dengan kode pada gambar 3.25 dibawah ini:

```
if (nearestIndex != -1) {
    Map<String, String> body = {
        "latitude": latitude.toString(),
        "longitude": longitude.toString()
    };
    Map<String, String> headers = {'Authorization': 'Bearer $token'};

    var response = await myHttp.post(
        Uri.parse(
            "https://datatech.co.id/indexLaravel.php/api/save-presensi"),
        body: body,
        headers: headers);
    return json.decode(response.body);
} else {
    return {
        'success': false,
        'message': 'Anda tidak berada di lokasi yang diizinkan untuk absen.'
    };
}
```

Gambar 3.25 Kode Untuk Bisa Atau Tidaknya Absen

Ketika absen telah berhasil di simpan, aplikasi akan mengambil data absen yang dilakukan oleh user dengan cara menarik API menggunakan kode pada gambar 3.26 dibawah ini. Data absen dibedakan menjadi dua yaitu data absen hari ini dan data absen hari-hari sebelumnya. Jika data absen merupakan data absen hari ini maka data absen tersebut akan ditampilkan di kotak paling atas, sedangkan data absen tersebut merupakan data absen pada hari-hari sebelumnya, maka data absen tersebut dimasukkan ke list kotak riwayat absensi.

```

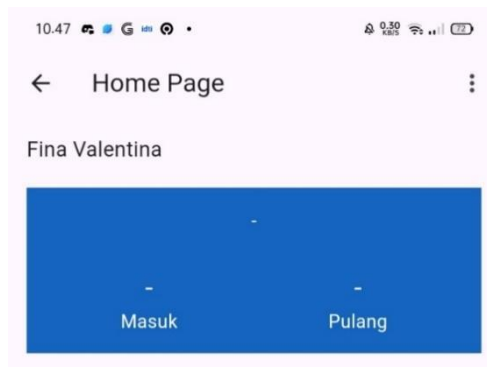
Future getData() async {
  final Map<String, String> headres = {
    'Authorization': 'Bearer ' + await token
  };
  var response = await myHttp.get(
    Uri.parse('https://datatech.co.id/indexLaravel.php/api/get-presensi'),
    headers: headres);

  homeResponseModel = HomeResponseModel.fromJson(json.decode(response.body));
  riwayat.clear();
  homeResponseModel!.data.forEach((element) {
    if (element.isHariIni) {
      hariIni = element;
    } else {
      riwayat.add(element);
    }
  });
}

```

Gambar 3.26 Kode Untuk Menyimpan Dan Mengarahkan Ke Kotak Aplikasi

Untuk menampilkan absen hari ini kita akan membuat kotak biru seperti dibawah ini, kita akan menggunakan kode dibawah ini. Disini kita akan membuat widget container yang berfungsi menjadi pembungkus dari elemen pada tampilan home page. Tampilan kotak untuk menampilkan jam masuk dan jam pulang karyawan hari ini seperti pada gambar 3.27:



Gambar 3.27 Tampilan Kotak Absen Hari Ini Pada Aplikasi

Pada home page kita akan membuat kotak yang berfungsi untuk menampilkan jam masuk dan jam pulang karyawan pada hari ini. Untuk membuat kotak tersebut kita gunakan class children. Kita akan membuat Row dan Column untuk mendefinisikan kolom dan row. Lalu kita gunakan text style untuk membuat tulisan yang akan muncul di kotak tersebut. Untuk mengatur tata letak tulisan tersebut kita akan gunakan kolom dan row. Kita akan gunakan kode pada gambar 3.28 dibawah ini:

```

Container(
  width: 400,
  decoration: BoxDecoration(color: Colors.blue[800]),
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      children: [
        Text(hariIni?.tanggal ?? '-',
          style: TextStyle(
            color: Colors.white, fontSize: 16)), // Text
        SizedBox(
          height: 30,
        ), // SizedBox
        Row(
          mainAxisAlignment:
            MainAxisAlignment.spaceAround,
          children: [
            Column(
              children: [
                Text(hariIni?.masuk ?? '-',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 24)), // TextStyle // Text
                Text("Masuk",
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 16)), // TextStyle // Text
              ],
            ), // Column

```

Gambar 3.28 Kode Untuk Membuat Tampilan Kotak Absen Hari Ini

Selanjutnya kita akan membuat list riwayat absen seperti dibawah ini. List riwayat absen ini merupakan list absen yang dilakukan oleh user di hari-hari sebelumnya. Pada riwayat presensi ini akan ditampilkan hari, tanggal, jam absen masuk dan jam absen pulang. Gambar 3.29 dibawah ini merupakan daftar riwayat presensi:



Gambar 3.29 Tampilan Kotak Riwayat Absen

Kode dibawah ini untuk membuat tampilan dari kotak list riwayat presensi. Disini kita membuat kelas child nanti nya di kelas child tersebut akan dibuat list view untuk menampilkan list Riwayat presensi yang dilakukan oleh user. Di kelas children juga kita akan membuat box yang berisi tanggal, hari, jam absen masuk dan jam absen pulang. Kita akan gunakan kode pada gambar 3.30 dibawah ini:


```
Codeium: Refactor | Explain
class ProfilePage extends StatefulWidget {
  @override
  ProfilePageState createState() => _ProfilePageState();
}
```

Gambar 3.31 Kode Membuat Class Profile

Selanjutnya kita akan membuat state dari kelas profile itu yang berfungsi untuk membuat halaman profile, dimana pada halaman profile terdapat nama user tersebut dan user dapat mengubah foto profile mereka. Untuk menampilkan nama user diperlukan nameController, dan supaya user bisa mengubah foto profile mereka, dan juga load profile digunakan untuk menampilkan semua hasil yang terdapat di halaman Profile. Untuk membuat halaman profile kita akan gunakan kode pada gambar 3.32 dibawah ini:

```
Codeium: Refactor | Explain
class _ProfilePageState extends State<ProfilePage> {
  final _nameController = TextEditingController();
  String? _imagePath;

  @override
  Codeium: Refactor | Explain | Generate Function Comment | X
  void initState() {
    super.initState();
    _loadProfile();
  }
}
```

Gambar 3.32 Kode State Membuat Halaman Profile

Untuk memasukkan foto profil di halaman profil pada aplikasi, user dapat memilih foto dengan cara memilih foto dari galeri handphone mereka. Kode dibawah ini merupakan kode supaya user dapat mengakses galeri di handphone mereka supaya dapat memilih foto dari galeri mereka. Untuk dapat memilih foto dari galeri kita akan gunakan `ImageSource.gallery` seperti kode pada gambar 3.33 dibawah ini:

```
Future<void> _pickImage() async {  
  final picker = ImagePicker();  
  final pickedFile = await picker.pickImage(source: ImageSource.gallery);  
  
  if (pickedFile != null) {  
    setState(() {  
      _imagePath = pickedFile.path;  
    });  
    _saveProfile(); // Simpan foto profil yang dipilih  
  }  
}
```

Gambar 3.33 Kode Untuk Dapat Memilih Foto Dan Simpan

Setelah user memilih foto untuk dijadikan foto profile mereka, maka foto tersebut akan muncul di foto profile mereka menggunakan kode pada gambar 3.34 dan gambar 3.35 dibawah ini. Kode dibawah ini akan memanggil path gambar dari gambar yang dipilih oleh user dari galeri handphone mereka. Setelah path gambar dipanggil dan gambar ditemukan maka gambar tersebut akan tampil. Selain gambar, pada halaman profile juga dapat menampilkan nama user tersebut.

```
Codeium: Refactor | Explain | Generate Function Comment | X
Future<void> _saveProfile() async {
    final prefs = await SharedPreferences.getInstance();
    await prefs.setString('imagePath', _imagePath ?? '');
}
```

Gambar 3.34 Kode Untuk Memanggil Path Gambar

```
Future<void> _loadProfile() async {
    final prefs = await SharedPreferences.getInstance();
    setState(() {
        _nameController.text = prefs.getString('name') ?? '';
        _imagePath = prefs.getString('imagePath');
    });
}
```

Gambar 3.35 Kode Untuk Mengambil Nama User

User juga dapat menghapus foto profile mereka jika mereka tidak menginginkan memakai foto profile tersebut. Setelah menghapus foto profile mereka, user dapat memilih foto baru. Supaya user dapat menghapus foto profile mereka, gunakan kode pada gambar 3.36 dibawah ini untuk membuat fitur menghapus foto profile.

```

Future<void> _deleteImage() async {
  final prefs = await SharedPreferences.getInstance();
  setState(() {
    _imagePath = null;
  });
  await prefs.remove('imagePath'); // Hapus path gambar dari SharedPreferences
}

```

Gambar 3.36 Kode Untuk Menghapus Foto

Kode pada gambar 3.37 dan gambar 3.38 dibawah ini untuk membuat tampilan halaman profile tersebut. Pada halaman profile terdapat lingkaran untuk menampilkan foto profile dengan cara user mengambil foto dari galeri handphone mereka. Lalu, pada halaman profile juga terdapat nama, dimana nama tersebut diambil dari data pada saat login.

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Profile'),
      actions: [
        IconButton(
          icon: Icon(Icons.logout),
          onPressed: _logout,
        ),
      ],
    ),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          Center(
            child: Stack(
              clipBehavior: Clip.none,
              children: [
                GestureDetector(
                  onTap: _pickImage,
                  child: CircleAvatar(
                    radius: 50,
                    backgroundImage:
                      _imagePath != null ? FileImage(File(_imagePath!)) : null,
                    child: _imagePath == null
                      ? Icon(
                          Icons.add_a_photo,
                          size: 50,
                        )

```

Gambar 3.37 Kode Untuk Membuat Tampilan Halaman Profile


```

void scheduleDailyNotifications() {
    // setup notification schedule
    List<TimeOfDay> notificationTimes = [
        const TimeOfDay(hour: 7, minute: 0),
        const TimeOfDay(hour: 7, minute: 30),
        const TimeOfDay(hour: 8, minute: 0),
        const TimeOfDay(hour: 8, minute: 30),
        const TimeOfDay(hour: 9, minute: 0),
        const TimeOfDay(hour: 16, minute: 30),
        const TimeOfDay(hour: 16, minute: 45),
        const TimeOfDay(hour: 17, minute: 0),
    ];

    for (int i = 0; i < notificationTimes.length; i++) {
        DateTime scheduledDate = _nextInstanceofTime(notificationTimes[i]);
        NotificationService.scheduleNotification(
            i,
            "Absensi",
            "Jangan lupa untuk melakukan absen",
            scheduledDate,
        );
    }
}

Codeium: Refactor | Explain | Generate Function Comment | X
DateTime _nextInstanceofTime(TimeOfDay time) {
    final now = DateTime.now();
    DateTime scheduledDate =
        DateTime(now.year, now.month, now.day, time.hour, time.minute);

    // If the scheduled time is before now, schedule for the next day
    if (scheduledDate.isBefore(now)) {
        scheduledDate = scheduledDate.add(const Duration(days: 1));
    }

    return scheduledDate;
}

```

Gambar 3.39 Kode Untuk Membuat Notifikasi Peringatan Absen

```

Future<void> createScheduleNotification() async {
    try {
        await awesomeNotifications.createNotification(
            content: NotificationContent(
                id: -1,
                channelKey: 'basic_channel',
                title: 'Absence',
                body: 'Jangan lupa absen masuk!');
    } catch (e) {}
}

```

Gambar 3.40 Kode Untuk Membuat Notifikasi Peringatan Absen

Membuat fitur perbaikan absen

Dikarenakan honor karyawan teknisi instalasi lapangan di PT. Infra Datatech dibayar per minggu, oleh karena itu absensi kehadiran sangat berpengaruh terhadap honor karyawan. Maka dari itu penulis ingin membuat fitur request permission bagi yang salah melakukan absen di aplikasi dengan syarat permission tersebut telah disetujui oleh pimpinan di PT. Infra Datatech. Permission yang dikirim akan di cek terlebih dahulu oleh supervisor lapangan, dan jika telah di cek oleh supervisor lapangan nantinya supervisor lapangan akan menyampaikan ke Pak Zulfikli sebagai pimpinan tertinggi di PT. Infra Datatech

Untuk melakukan permintaan perbaikan absensi, karyawan perlu mengisi formulir terlebih dahulu di aplikasi absen tersebut. Supaya user dapat mengisi formulir diperlukan kode dibawah ini. Kita akan membuat class request form dengan membuat text editing supaya user dapat memasukkan text ke formulir tersebut dan name, id, day, start time, end time controller untuk menyimpan data yang dimasukkan user pada saat memasukkan formulir. Berikut kode pada gambar 3.41 untuk membuat formulirnya:


```

class RequestForm extends StatefulWidget {
  @override
  _RequestFormState createState() => _RequestFormState();
}

class _RequestFormState extends State<RequestForm> {
  final TextEditingController _nameController = TextEditingController();
  final TextEditingController _idController = TextEditingController();
  final TextEditingController _dayController = TextEditingController();
  final TextEditingController _startTimeController = TextEditingController();
  final TextEditingController _endTimeController = TextEditingController();

  bool _approveRequest = false;
  bool _rejectRequest = false;
  bool _isSubmitted = false;
  String _status = 'Menunggu verifikasi...';

  @override
  void initState() {
    super.initState();
    _loadFormData();
  }
}

```

Gambar 3.41 Kode Untuk Membuat Formulir Perbaikan Absen

Jika karyawan telah mengisi formulir, maka akan terdapat pesan yang mengarahkan ke WhatsApp untuk di cek lebih lanjut oleh Pak Zulfikli. Kode dibawah ini berfungsi untuk mengarahkan formular yang telah diisi oleh user diarahkan ke aplikasi whatsapp. Disini juga saya membuat template pesan sehingga user tidak perlu mengetik ulang dan hanya mengirim pesan tersebut ke pak zulfikli. Pada gambar 3.42 dibawah ini kode dan template pesan yang mengarahkan ke whatsapp Pak Zulkifli untuk perbaikan absensi:

```

Future<void> _submitRequest() async {
  final name = _nameController.text;
  final id = _idController.text;
  final day = _dayController.text;
  final startTime = _startTimeController.text;
  final endTime = _endTimeController.text;

  if (name.isEmpty || id.isEmpty || day.isEmpty || startTime.isEmpty || endTime.isEmpty)
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Please fill in all the fields')),
    );
  return;
}

final message = 'Halo pak, saya $name dengan id $id terdapat kesalahan absen untuk hari
                'Saya masuk jam $startTime dan pulang jam $endTime.'
                'Ingin melakukan konfirmasi';

final encodedMessage = Uri.encodeComponent(message);
final url = 'https://api.whatsapp.com/send?text=$encodedMessage';

if (await canLaunch(url)) {
  await launch(url);
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Opening WhatsApp with your message')),
  );
  setState(() {
    _isSubmitted = true;
    _status = 'Menunggu verifikasi...';
  });
  await _saveFormData();
} else {

```

Gambar 3.42 Kode Untuk Mengirim Formulir Ke WhatsApp

Pada halaman permission request terdapat formulir untuk karyawan melakukan request perbaikan absen. Pada formulir request perbaikan absen akan terdapat nama, id, hari, jam masuk, dan jam keluar. Untuk membuat tampilan halaman permission request menggunakan kode pada gambar 3.43 dibawah ini:

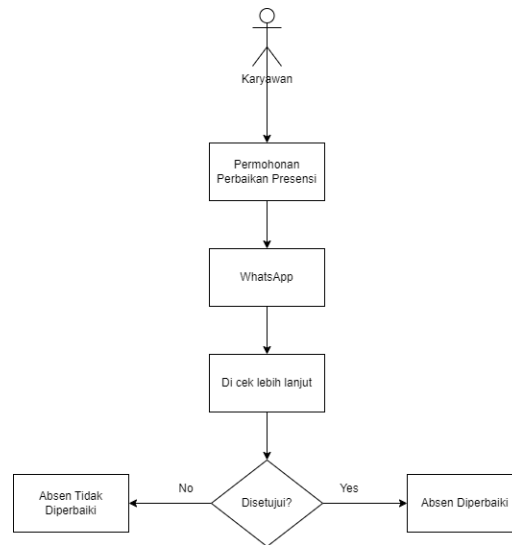
```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Permission Request Form'),
    ),
    body: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        children: [
          TextField(
            controller: _nameController,
            decoration: InputDecoration(labelText: 'Name'),
            enabled: !_isSubmitted,
          ),
          TextField(
            controller: _idController,
            decoration: InputDecoration(labelText: 'ID'),
            onChanged: (value) => _loadFormData(),
            enabled: !_isSubmitted,
          ),
          TextField(
            controller: _dayController,
            decoration: InputDecoration(labelText: 'Day'),
            enabled: !_isSubmitted,
          ),
          TextField(
            controller: _startTimeController,
            decoration: InputDecoration(labelText: 'Start Time'),
            enabled: !_isSubmitted,
          ),
        ],
      ),
    ),
  );
}

```

Gambar 3.43 Kode Untuk Membuat Template Pesan

Karyawan yang mengalami kesalahan absen di aplikasi absen mereka akan mengisi formulir terlebih dahulu pada aplikasi absen mereka, supaya Pak Zulfikli dapat mengecek kebenaran atas kesalahan tersebut. Setelah user mengisi formulir, maka formulir tersebut akan mengarahkan ke whatsapp untuk mengirimkan template pesan ke whatsapp Pak Zulfikli yang telah dibuat pada kode sebelumnya. Selanjutnya, pak Zulfikli akan mengecek dan memutuskan apakah perbaikannya disetujui atau tidak. Supaya lebih jelas bagaimana alur jika user mengalami kesalahan terhadap absensi yang dilakukan oleh user, penulis menggambarannya dengan flowchart pada gambar 3.44 dibawah ini:



Gambar 3.44 Diagram Alur Perbaikan Absen

Build To APK

Setelah kita selesai mengembangkan aplikasi absensi tersebut, langkah selanjutnya yang penting adalah memastikan bahwa aplikasi tersebut dapat digunakan oleh seluruh karyawan di PT. Infra Datatech. Untuk mencapai tujuan ini, aplikasi harus di-*build* menjadi format yang sesuai, yaitu file dengan ekstensi .apk (Android Package) yang dapat dijalankan di perangkat Android. Dengan di-*build* menjadi .apk, aplikasi dapat diinstal langsung di ponsel masing-masing karyawan, sehingga mereka dapat mengakses fitur absensi dengan mudah. Cara build to apk adalah sebagai berikut

1. Kita bisa gunakan command "flutter build apk --release" jika menggunakan VS Code, atau jika menggunakan Android Studio ke menu titik tiga -> Build -> Flutter -> Build APK

2. Jika mengalami bug tidak bisa build to apk secara langsung seperti yang disebutkan di nomor 1. Bisa menggunakan cara dibawah ini:
- jalankan "flutter build appbundle" di terminal vs code, seperti gambar 3.45 dibawah ini:

```
nardi@ima MINGW64 ~/Downloads/presence-app-master/presence-app-master
$ flutter build appbundle

Running Gradle task 'bundleRelease'...
  9.6s
✓ Built build\app\outputs\bundle\release\app-release.aab
(23.0MB)
```

Gambar 3.45 Command Untuk Menjadikan APK

- jika build sudah selesai, arahkan kursor mouse ke Built build\app sambil menekan ctrl
- setelah itu akan terbuka file app-release.aab di vscode
- scroll di menu bagian kiri sampai menemukan file app-release.aab, klik kanan terus reveal in file explorer
- copy file app-release.aab tersebut, lalu pergi ke folder bundletool di penyimpanan C: , lalu paste di folder bundletool
- klik kanan di folder bundletool, pilih open in terminal
- paste command berikut:
java -jar bundletool.jar build-apks --bundle=app-release.aab --output=app-release.apks --mode=universal --ks="C:\Keystore\upload-keystore.jks" --ks-pass=pass:presence1234 --ks-key-alias=upload --key-pass=pass:presence1234
- tunggu hingga proses build to apks selesai
- ubah format file dari app-release.apks, dengan cara klik kanan -> show more -> rename . ubah .apks jadi .zip supaya bisa di ekstrak
- ekstrak file tersebut

- k. di dalam folder tersebut terdapat file yang bernama universal.apk, itu merupakan hasil aplikasi yang di build to apk sebelumnya

Pembuatan Website Company PT. Infra Datatech Indonesia

Latar Belakang

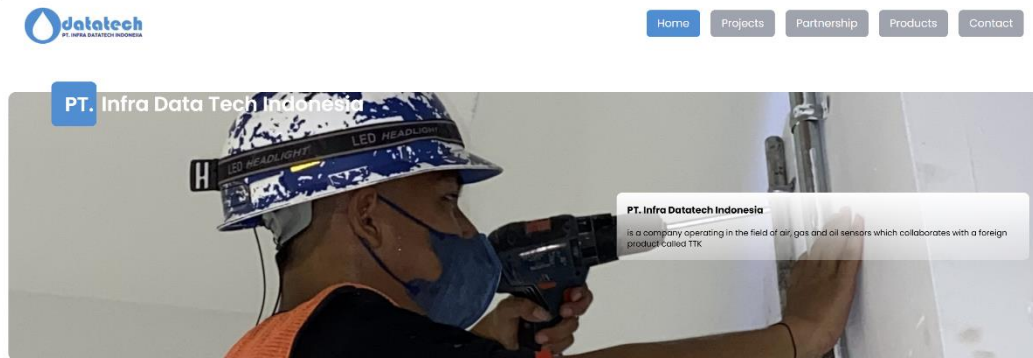
PT. Infra Datatech Indonesia merupakan perusahaan menengah yang sedang mengembangkan perusahaannya. Supaya dapat mengembangkan perusahaannya sehingga dapat dikenal oleh lebih banyak orang, dan client dapat mengenal dan mempercayai kualitas dari PT. Infra Datatech maka dibutuhkan website company yang berfungsi sebagai company branding PT. Infra Datatech Indonesia.

Proses Pembuatan

Untuk membuat website company PT. Infra Datatech Indonesia penulis menggunakan bahasa pemrograman type script. Website company ini terdiri dari 5 bagian navigation yaitu home, projects, partnership, products, contact. Pada halaman home website tersebut berisi konten mengenai pengenalan singkat mengenai PT. Infra Datatech Indonesia. Pada halaman project berisi konten project yang PT. Infra Datatech pernah kerjakan dan bentuk aplikasi jasa dan product yang dijual di PT. Infra Datatech. Pada halaman partnership tersebut berisi konten perusahaan yang bekerja sama dengan PT. Infra Datatech atau TTK, dan juga berisi sertifikat internasional yang di dapatkan oleh PT. Infra Datatech. Pada halaman produk tersebut berisi konten produk yang PT. Infra Datatech jual. Dan pada halaman kontak berisi konten tentang formulir untuk mengarahkan ke kontak PT. Infra Datatech supaya client dapat menghubunginya.

Untuk membuat halaman home, pertama penulis menampilkan gambar teknisi lapangan bekerja seperti pada gambar 3.46 dengan menggunakan

coursel. Coursel merupakan sebuah teknik untuk membuat gambar bisa di swipe. Untuk membuat coursel pada type script kita perlu import dengan menggunakan kode pada gambar 3.47



Gambar 3.46 Coursel Pada Home Page

```
import { Swiper, SwiperSlide } from "swiper/react";
import 'swiper/css';
import 'swiper/css/navigation';
import * as Image from "../../assets/images/home";
import { Autoplay } from 'swiper/modules';
```

Gambar 3.47 Kode Import Coursel

Selanjutnya, pada halaman About Us, penulis membuat bagian *About Us* untuk menjelaskan PT. Infra Datatech secara lebih detail, seperti yang ditampilkan pada gambar 3.48. Bagian ini bertujuan untuk memberikan informasi singkat mengenai visi, misi, dan nilai-nilai yang dianut oleh perusahaan. Untuk membuat bagian *About Us* tersebut, kita akan menggunakan kode yang ditunjukkan pada gambar 3.49.

About Us

PT. Infra Datatech Indonesia was founded in 2019. We focus on the provisioning and total solution for hardware and technical services in Data Center Industries.

Our team consists of experienced people in Infrastructure Data Center.

Since 2019, we have actively cooperated with global vendors and operators by supporting and maintaining their products and Data Centers.

We continuously create long-term value for customers by being responsive to their needs and requirements.



Gambar 3.48 About Us

```
import * as Image from ".../assets/images/home";

const AboutUs = () => {
  return (
    <
      <section className="my [100px] flex flex-col lg:flex-row lg:items-center lg:justify-between">
        {/* About Us Text */}
        <div className="lg:w-1/2 px-4 py-8">
          <h2 className="text-4xl font-sembold mb-4 text-blue-500">About Us</h2>
          <p className="text-base mb-4">
            PT. Infra Datatech Indonesia was founded in 2019. We focus on the provisioning and total solution for hardware and technical servit
          </p>
          <p className="text-base mb-4">
            Our team consists of experienced people in Infrastructure Data Center.
          </p>
          <p className="text-base mb-4">
            Since 2019, we have actively cooperated with global vendors and operators by supporting and maintaining their products and Data Cer
          </p>
          <p className="text-base">
            We continuously create long-term value for customers by being responsive to their needs and requirements.
          </p>
        </div>
        {/* Image */}
        <div className="lg:w-1/2 mt-8 lg:mt-0 px-4">
          <img
            className="w-full rounded-xl object-cover lg:h-[400px]"
            src={Image.ContactBackground}
            alt="People looking at a laptop"
          </img>
        </div>
      </section>
    </
  )
}

export default AboutUs;
```

Gambar 3.49 Kode Untuk Membuat About Us

Selanjutnya, pada halaman home juga terdapat bagian visi dan misi dari PT. Infra Datatech yang disusun untuk menjelaskan tujuan serta nilai-nilai perusahaan, seperti yang terlihat pada gambar 3.50. Untuk membuat bagian visi dan misi tersebut, penulis menggunakan kode yang ditunjukkan pada gambar 3.51. Bagian ini diimplementasikan untuk memberikan gambaran jelas mengenai arah perusahaan dan prinsip-prinsip yang dijunjung dalam operasional sehari-hari.



Gambar 3.50 Visi dan Misi

```
const Mission = () => {
  return (
    <section className="my-[100px] px-4 lg:px-0">
      <div className="lg:w-4/4 w-full px-4 py-8 mx-auto">
        { /* Vision */ }
        <div className="bg-blue-500 p-6 rounded-md shadow-lg text-white text-center">
          <h2 className="text-4xl font-semibold mb-4">Vision</h2>
          <p className="text-base">
            To become a leading company for Data Center Infrastructure services in Indonesia.
          </p>
        </div>
        { /* Mission */ }
        <div className="bg-blue-500 p-6 rounded-md shadow-lg text-white text-center">
          <h2 className="text-4xl font-semibold mb-4">Mission</h2>
          <ul className="list-disc list-inside text-base space-y-2">
            <li>
              Supports Data Center operators and vendors in building digital infrastructure that unites the entire
            </li>
            <li>
              Actively involved in the development of digital infrastructure throughout the Southeast Asia region.
            </li>
          </ul>
        </div>
      </div>
    </section>
  )
}
export default Mission;
```

Gambar 3.51 Kode Visi dan Misi

Selanjutnya, untuk memberitahukan kepada klien bahwa PT. Infra Datatech Indonesia fokus pada bidang tertentu, penulis menambahkan tiga ikon yang merepresentasikan jenis produk sensor yang dijual oleh PT. Infra Datatech, seperti yang ditunjukkan pada gambar 3.52. Ikon-ikon ini dirancang untuk memperjelas kategori produk yang ditawarkan kepada pelanggan. Untuk membuat tampilan tersebut, penulis menggunakan kode yang tertera pada gambar 3.53, dengan desain yang menarik dan informatif agar mudah dipahami oleh pengunjung website.

Applications

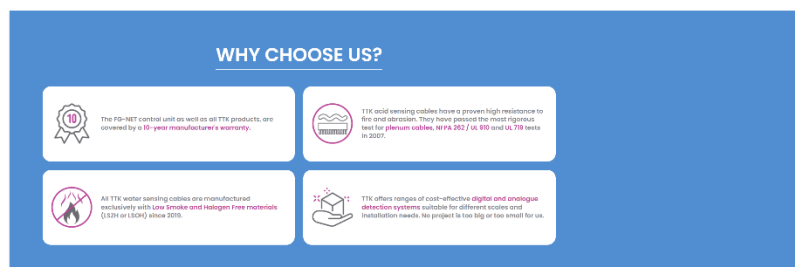


Gambar 3.52 Applications

```
function WeCareForYou() {
  return (
    <section className="my-8 px-4 lg:px-0">
      <div className="container mx-auto mb-12">
        <h1 className="text-3xl sm:text-4xl font-semibold mb-8 text-blue-500">Applications</h1>
        <div className="flex flex-wrap justify-center gap-8">
          <div className="flex flex-col items-center text-center">
            <div className="text-2xl text-gray-700 mb-2">Specialist</div>
            <div className="h-1 w-full bg-gray-700 mb-2" />
          </div>
          <img
            className="h-32 object-contain ml-4"
            src={Image.WeCare1}
            alt="Specialist"
          />
        </div>
        <div className="flex-1 min-w-[200px] h-auto p-4 flex items-center justify-center">
          <img
            className="h-32 object-contain"
            src={Image.WeCare2}
            alt="Image 2"
          />
        </div>
        <div className="flex-1 min-w-[200px] h-auto p-4 flex items-center justify-center">
          <img
            className="h-32 object-contain"
            src={Image.WeCare3}
            alt="Image 3"
          />
        </div>
      </div>
      <div className="mt-8 flex justify-around text-center">
        <div>
          <h2 className="text-xl text-blue-500 ml-14">Water Detection</h2>
        </div>
        <div>
          <h2 className="text-xl text-purple-500 ml-[-2cm]">Acid Detection</h2>
        </div>
      </div>
    </section>
  )
}
```

Gambar 3.53 Kode Untuk Membuat Applications

Berikutnya, untuk memberitahukan alasan mengapa PT. Infra Datatech lebih unggul di bidangnya dibandingkan perusahaan lain, penulis menambahkan bagian *Why Choose Us* seperti yang terlihat pada gambar 3.54. Bagian ini dirancang untuk menyoroti keunggulan, keahlian, dan nilai tambah yang ditawarkan oleh PT. Infra Datatech kepada klien. Untuk membuat tampilan tersebut, penulis menggunakan kode yang ditampilkan pada gambar 3.55, dengan tujuan menyampaikan poin-poin utama yang membedakan perusahaan dari kompetitor.



Gambar 3.54 Why Choose Us

```
import as image from ".../assets/images/home";

Codium: Refactor | Explain | Generate JSDoc | X
const GoodToKnow = () => {
  return (
    <div className="flex flex-col gap-16 bg-blue-datatech p-8 font-sembold text-white lg:flex-row lg:p-16">
      /* Left Section */

      /* Right Section */
      <div className="flex flex-col items-center gap-8 lg:w-[70%]">
        /* Title Section */
        <div className="text-center lg:text-left">
          <h1 className="text-2xl lg:text-4xl">WHY CHOOSE US?</h1>
          <hr className="mt-2 w-full border-1 border-white" />
        </div>

        /* Grid Section */
        <div className="grid grid-cols-1 gap-4 sm:grid-cols-2">
          <div className="flex items-center justify-center gap-4 rounded-2xl bg-white p-4 py-8">
            <img className="h-[60px] lg:h-[80px]" src={image.GoodToKnow2} alt="Warranty" />
            <p className="text-xs text-[#88888E]">
              The FG-NET control unit as well as all TTK products, are covered by a(" ")
              <span className="text-[#04389D]">10-year</span>
              manufacturer's warranty.
            </p>
          </div>

          <div className="flex items-center justify-center gap-4 rounded-2xl bg-white p-4 py-8">
            <img className="h-[60px] lg:h-[80px]" src={image.GoodToKnow3} alt="Cable Test" />
            <p className="text-xs text-[#88888E]">
              TTK acid sensing cables have a proven high resistance to fire and abrasion.
              They have passed the most rigorous test for(" ")
              <span className="text-[#04389D]">plenum cables, NFPA 262 / UL 910</span> and(" ")
              <span className="text-[#04389D]">UL 719</span> tests in 2007.
            </p>
          </div>
        </div>
      </div>
    </div>
  );
};
```

Gambar 3.55 Kode Untuk Membuat Why Choose Us

Karena PT. Infra Datatech ini merupakan perusahaan distributor dari produk luar negeri yang bernama TTK, oleh karena penting untuk menjelaskan produk TTK ini sudah terjual di negara mana saja. Sehingga penulis membuat bagian global contact untuk menjelaskan kepada klient TTK ini sudah tersebar luas di berbagai negara, dan menyebarkan produknya ke negara indonesia melalui PT. Infra Datatech. Berikut tampilan dari global contact dapat dilihat pada gambar 3.56. Untuk membuat tampilan dari global contact penulis menggunakan kode pada gambar 3.57.



Gambar 3.56 Global Contact

```

import * as Image from "../../assets/images/home";

// Maps Leaflet Import
// imports { MapControl, TileLayer, Marker, Popup } from "react-leaflet";
import "leaflet/dist/leaflet.css";

Codeium: Refactor | Explain | Generate JS Doc | X
const GlobalContact = () => {
  return (
    <>
      <section className="p-4 lg:p-16">
        <div className="flex flex-col lg:flex-row">
          {/ Globe Section /}
          <div className="relative w-full lg:w-2/5 overflow-hidden mb-8 lg:mb-0">
            <div className="absolute top-4 left-4 lg:top-16 lg:left-16 z-10 px-4 lg:px-0">
              <h1 className="text-2xl lg:text-4xl font-semantic text-[#A5AEB3]">
                GLOBAL CONTACT
              </h1>
              <p className="mt-4 lg:mt-8 font-semantic text-white">
                With offices in Europe, Asia, the Middle East and North America,
                we are perfectly placed to offer you tailored solutions for water,
                acid or hydrocarbon leak detection. Contact us for more information.
              </p>
            </div>
            <img
              className="absolute left-0 top-0 w-full h-auto lg:left-[100px] lg:w-[600px] lg:min-w-[600px]"
              src={Image.Fartha}
              alt="globe"
            />
          </div>
          {/ Content Section /}
          <div className="w-full lg:w-3/5">
            <img className="w-full h-auto" src={Image.Maps} alt="Global Map" />
            <div className="overflow-x-auto mt-8">
              <table className="min-w-full table-auto">
                <thead>
                  <tr className="bg-blue-datatech text-white">

```

Gambar 3.57 Kode Untuk Membuat Global Contact

Supaya semua bagian yang sudah disebutkan pada gambar-gambar sebelumnya untuk ditampilkan menjadi satu kesatuan pada halaman home, penulis gunakan kode pada gambar 3.58.

```

import * as Image from "../assets/images/home";

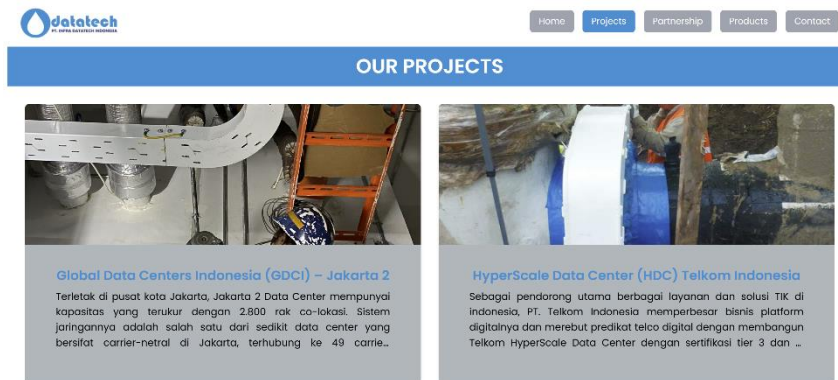
// Layout Import
import Section from "../layout/Section";
import Footer from "../layout/Footer";
import CarouselHome from "../components/home/CarouselHome";
import AboutUs from "../components/home/AboutUs";
import Mission from "../components/home/Mission";
import DrilledMen from "../components/home/DrilledMen";
import GoodToKnow from "../components/home/GoodToKnow";
import GlobalContact from "../components/home/GlobalContact";

Codeium: Refactor | Explain | Generate JSDoc | X
export default function Home() {
  return (
    <>
      <Section>
        <CarouselHome />
        <AboutUs />
        <Mission />
        <WeCareForYou />
        <GoodToKnow />
        <DrilledMen />
        <GlobalContact />
      </Section>
      <Footer />
    </>
  );
}

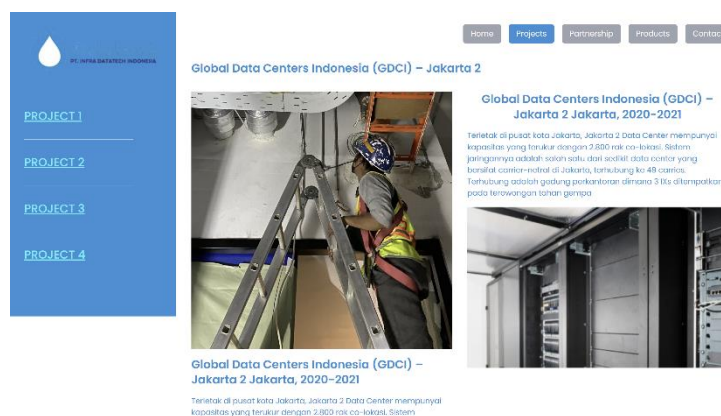
```

Gambar 3.58 Kode Untuk Menghubungkan

Berikutnya, terdapat halaman *Project*, di mana halaman ini berfungsi untuk menampilkan proyek-proyek yang telah dikerjakan oleh PT. Infra Datatech, beserta deskripsi singkat setiap proyek, seperti yang ditunjukkan pada gambar 3.59 dan gambar 3.60. Halaman ini bertujuan untuk memberikan gambaran kepada klien tentang pengalaman dan keahlian perusahaan dalam menangani berbagai proyek. Untuk membuat halaman *Project* ini, penulis menggunakan kode yang terlihat pada gambar 3.61, dengan penataan yang rapi dan informatif agar mudah diakses oleh pengunjung website.



Gambar 3.59 Our Projects



Gambar 3.60 Our Project full deskripsi

```
const OurProjectData = [
  {
    id: 1,
    link: "our-project",
    img: imgProject1,
    title: "Global Data Centers Indonesia (GDCI) Jakarta 2",
    description:
      "Terletak di pusat kota Jakarta, Jakarta 2 Data Center mempunyai kapasitas yang terukur dengan 2.800 rak co-lokasi. Sist",
  },
  {
    id: 2,
    link: "our-project",
    img: imgProject2,
    title: "SuperScale Data Center (HXC) Telkom Indonesia",
    description:
      "Sebagai pendorong utama berbagai layanan dan solusi IKT di Indonesia, PT. Telkom Indonesia memperbesar bisnis platform",
  },
  {
    id: 3,
    link: "our-project",
    img: imgProject1,
    title: "PROJECT 3",
    description:
      "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna ali",
  },
  {
    id: 4,
    link: "our-project",
    img: imgProject2,
    title: "PROJECT 4",
    description:
      "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna ali",
  },
  {
    id: 5,
    link: "our-project",
    img: imgProject1,
    title: "PROJECT 5",
  },
];
```

Gambar 3.61 Kode Untuk Membuat Our Projects

Selanjutnya, pada bagian partnership dibuat bertujuan untuk menjelaskan kepada klient perusahaan ini sudah bekerja sama dengan perusahaan mana saja sehingga dapat meningkatkan rasa kepercayaan kepada klient terhadap perusahaan. Untuk tampilan partnership dapat dilihat pada gambar 3.62. Untuk membuat tampilan partnership penulis menggunakan kode yang dapat dilihat pada gambar 3.63.

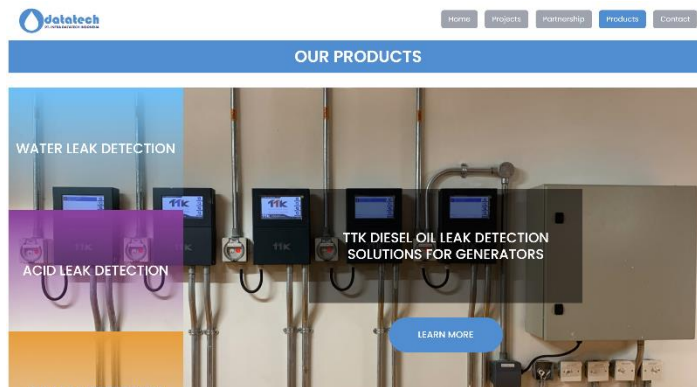


Gambar 3.62 Partnership

```
const Partner = () => {
  return (
    <div className="relative mt-[50px] flex w-full justify-center">
      <div className="relative">
        {/* Heading Text */}
        <h1 className="absolute left-4 sm:left-8 lg:left-16 top-4 sm:top-6 lg:top-8 z-10 text-2xl sm:text-3xl lg:text-4xl"
          <span className="rounded-xl bg-blue-datatech py-1 sm:py-2 pl-2 sm:pl-3 lg:py-2 lg:pl-4">PARTNERSHIP</span>
        </h1>
        {/* Background Image */}
        <img
          className="h-[200px] sm:h-[300px] lg:h-[400px] w-full object-cover"
          src={imgbackground}
          alt="Partnership background"
        />
      </div>
      {/* Subheading Text */}
      <h2 className="absolute -bottom-6 sm:-bottom-8 lg:-bottom-8 rounded-full bg-blue-datatech px-12 sm:px-20 lg:px-32 py"
        They Trust Us:
      </h2>
    </div>
  );
}
export default Partner;
```

Gambar 3.63 Kode Untuk Membuat Partnership

Selanjutnya, terdapat halaman Our Products untuk menjelaskan product yang dijual di PT. Infra Datatech. Dimana terdapat 3 bagian yaitu water leak detection, oil leak detection dan acid leak detection yang penulis bedakan warnanya supaya lebih menarik. Dimana ketika user klik kotak warna gradient muncul penjelasan singkat apa itu dan fungsinya untuk apa. Tampilan Our Products dapat dilihat pada gambar 3.64. Untuk membuat tampilan Our Products penulis menggunakan kode pada gambar 3.65.



Gambar 3.64 Our Products


```

import headline from "../../components/banner";
import imgService01 from "../../assets/images/service/service-image-01.png";
import imgService02 from "../../assets/images/service/service-image-02.png";
import imgService03 from "../../assets/images/service/service-image-03.png";

import { Swiper, SwiperSlide } from "swiper/react";
import { Autoplay, EffectSlide } from "swiper/modules";

import "swiper/css";
import "swiper/css/pagination";
import "swiper/css/navigation";
import { useState } from "react";
import { Link } from "react-router-dom";

const data = [
  {
    backgroundColor: "bg-gradient-to-b from-[#05C3FF]",
    image: imgService01,
    headline:
      "DEVELOPING PRODUCTS THAT CONTRIBUTE TO A GREENER, MORE ECO-FRIENDLY WORLD",
    title: "WATER LEAK DETECTION",
    content:
      "The addressable sense cables are fitted with patented micro-controllers that allow pin-pointing of faults on multiple sense c",
    link: "/products/water-leak-detection-any-product",
  },
  {
    backgroundColor: "bg-gradient-to-b from-[#9A37A2]",
    image: imgService02,
    headline:
      "DIGITAL LEAK DETECTION SYSTEMS FOR HYDROCARBON & WATER IN A DATA CENTRE",
    title: "ACID LEAK DETECTION",
    content:
      "The corrosive resistant acid sense cable can detect and locate the presence of acid and chemicals, making it the ideal solution",
    link: "#",
  },
  {
    backgroundColor: "bg-gradient-to-b from-[#1F9980]",
    image: imgService03,
  }
];

```

Gambar 3.65 Kode Our Products

Ketika Halaman produk di scroll ke bawah kita akan melihat 3 kotak berwarna beda lagi, akan tetapi 3 kotak berwarna itu fungsinya berbeda dengan sebelumnya. Jika tadi hanya untuk menjelaskan fungsi dan pengertian dari water leak, oil leak dan acid leak. Tiga kotak berwarna dibawah ini berfungsi untuk menampilkan produk-produk berdasarkan kategorinya (tiga kotak itu kategorinya). Tampilannya dapat dilihat pada gambar 3.66. Untuk membuat tampilan tersebut penulis menggunakan kode yang dapat dilihat pada gambar 3.67.



Gambar 3.66 Tampilan Produk Berdasarkan Kategori

```
import imgService01 from "../../assets/images/service/service-image-01.png";
import imgService02 from "../../assets/images/service/service-image-02.png";
import imgService03 from "../../assets/images/service/service-image-03.png";
import imgProduct01 from "../../assets/images/product-01.png";
import imgProduct02 from "../../assets/images/product-02.png";
import imgProduct03 from "../../assets/images/product-03.png";
import imgProduct04 from "../../assets/images/product-04.png";

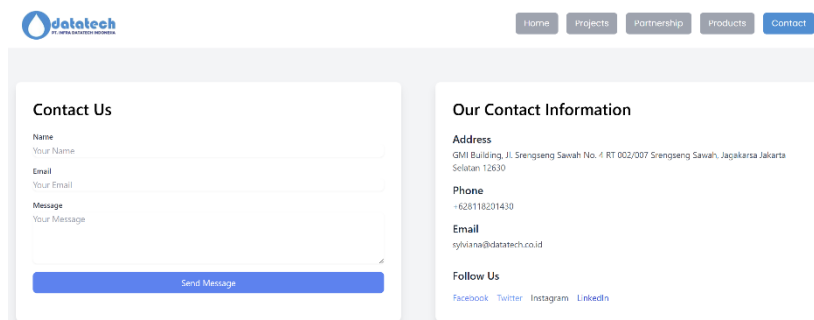
import { Swiper, SwiperSlide } from "swiper/react";
// Import required modules
import { Autoplay, Pagination, Navigation } from "swiper/modules";

import "swiper/css";
import "swiper/css/pagination";
import "swiper/css/navigation";
import { useState } from "react";
import { Link } from "react-router-dom";

const data = [
  {
    backgroundColor: "bg-gradient-to-b from-[#85C3FF]",
    image: imgService01,
    headline: "DEVELOPING PRODUCTS THAT CONTRIBUTE TO A GREENER, MORE ECO-FRIENDLY WORLD",
    title: "WATER LEAK DETECTION",
    content: "The addressable sense cables are fitted with patented micro-controllers that allow pin-pointing of faults on multiple link: /products/water-leak-detection-any-product",
  },
  {
    backgroundColor: "bg-gradient-to-b from-[#9A37A2]",
    image: imgService02,
    headline: "DIGITAL LEAK DETECTION SYSTEMS FOR HYDROCARBON & WATER IN A DATA CENTRE",
    title: "ACID LEAK DETECTION",
    content:
  },
];
```

Gambar 3.67 Kode Untuk Membuat Tampilan Produk Berdasarkan Kategori

Terakhir, terdapat halaman *Kontak* yang berfungsi untuk memberikan informasi mengenai cara menghubungi PT. Infra Datatech, seperti alamat kantor, nomor telepon, email, dan formulir kontak. Halaman ini bertujuan memudahkan klien atau calon klien dalam mengajukan pertanyaan, meminta penawaran, atau menjalin kerja sama dengan perusahaan. Tampilan halaman kontak ini dapat dilihat pada gambar 3.68. Untuk membuat tampilan halaman kontak ini penulis menggunakan kode yang dapat dilihat pada gambar 3.69.



Gambar 3.68 Halaman Kontak

```
import Footer from "../layout/footer"
<div className="container mx-auto">
  <div className="flex flex-col lg:flex-row lg:space-x-16">
    <div className="flex-1 bg-white shadow-lg rounded-lg p:8 md:8 lg:mb-0">
      <h2 className="text-3xl font-sans font-weight-medium mb-6">Contact Us</h2>
      <form>
        <div className="mb-4">
          <label htmlFor="name" className="block text-sm font-medium text-gray-700">Name</label>
          <input
            type="text"
            id="name"
            className="mt-1 block w-full border-gray-300 rounded-lg shadow-sm focus:border-blue-500 focus:placeholder-gray-500"
            placeholder="Your Name"
          />
        </div>
        <div className="mb-4">
          <label htmlFor="email" className="block text-sm font-medium text-gray-700">Email</label>
          <input
            type="email"
            id="email"
            className="mt-1 block w-full border-gray-300 rounded-lg shadow-sm focus:border-blue-500 focus:placeholder-gray-500"
            placeholder="Your Email"
          />
        </div>
        <div className="mb-4">
          <label htmlFor="message" className="block text-sm font-medium text-gray-700">Message</label>
          <textarea
            id="message"
            rows={4}
            className="mt-1 block w-full border-gray-300 rounded-lg shadow-sm focus:border-blue-500 focus:placeholder-gray-500"
            placeholder="Your Message"
          />
        </div>
      </form>
    </div>
  </div>
</div>
```

Gambar 3.69 Kode Untuk Membuat Halaman Kontak

Pembuatan Website reLuna

Latar Belakang

Dikarenakan PT. Infra Datatech ingin membuat brand produk baru yang bernama reLuna, yang memiliki perbedaan signifikan dibandingkan produk sebelumnya, penulis diminta untuk membuatkan website company khusus untuk produk tersebut. Produk reLuna ini berbeda karena, tidak seperti produk sebelumnya yang merupakan hasil kerja sama dengan supplier luar negeri bernama TTK, reLuna dikembangkan secara independen tanpa bekerja sama dengan pihak luar. Oleh karena itu, atasan saya meminta saya untuk merancang dan membangun sebuah website yang merepresentasikan brand dan produk reLuna secara profesional.

Proses Pembuatan

Untuk membuat website reLuna ini sendiri, penulis menggunakan bahasa pemrograman vue.js. Dimana website tersebut akan memiliki enam navbar yaitu home, industrial automation, kostel management system, about us dan contact us. Pada halaman home berisi konten penjelasan dari reLuna itu sendiri, dan product andalan yang reLuna jual ke konsumen nantinya. Halaman industrial automation berisi konten mengenai produk yang termasuk ke golongan sistem kontrol. Kostel Management System berisi konten produk sistem management hotel menggunakan website serta produk yang telah terinstallasi ke sistem management hotel tersebut. Halaman about us berisi konten penjelasan mengenai reLuna dan keunggulan dari reLuna. Terakhir, terdapat halaman *Kontak* yang berfungsi untuk memberikan informasi mengenai cara menghubungi reLuna, seperti alamat kantor, nomor telepon, email, dan formulir kontak. Halaman ini bertujuan memudahkan klien atau calon klien dalam mengajukan pertanyaan, meminta penawaran, atau menjalin kerja sama dengan perusahaan.

Pada halaman home, penulis membuat bagian *About Us* untuk menjelaskan reLuna secara lebih detail, seperti yang ditampilkan pada gambar 3.70. Bagian ini bertujuan untuk memberikan informasi singkat mengenai visi, misi, dan nilai-nilai yang dianut oleh perusahaan. Untuk membuat bagian *About Us* tersebut, kita akan menggunakan kode yang ditunjukkan pada gambar 3.71.



Gambar 3.70 About Us Singkat reLuna

```
<div class="container mx-auto">
  <div class="py-8">
    <div class="flex justify-between">
      <div>
        <br />
        <br />
        <p class="text-4xl font-bold">Your Smart Automation Partner</p>
        <p class="text-2xl font-bold">Smart Automation Made Simple and Affordable</p>
      </div>
      <p>Integrate smart technologies into your home and company, and live better. reLur</p>
    </div>
  </div>
  <br />
  <br />
</div>
</div>
```

Gambar 3.71 Kode Untuk Membuat About Us reLuna

Selanjutnya, kita akan membuat bagian Our Costumer untuk memberitahukan konsumen yang sudah pernah membeli produk dari reLuna. Tampilan Our Costumer dapat dilihat pada gambar 3.72. Untuk membuat tampilan tersebut penulis menggunakan kode yang dapat dilihat pada gambar 3.73.

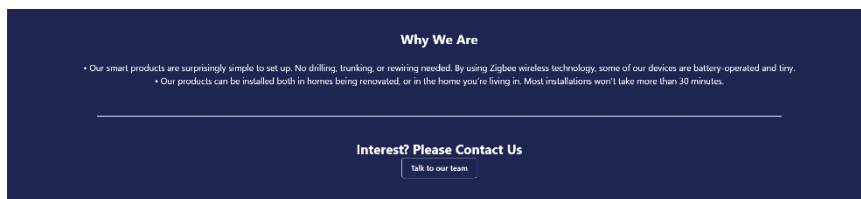


Gambar 3.72 Our Costumer

```
<div class="flex flex-col justify-center mt-20">
  <p class="text-4xl text-center font-bold">Our Customer</p>
  <br />
</div>
<div class="w-full mb-20">
  <div class="grid grid-cols-4">
    <div class="flex flex-col items-center gap-2">
      <svg
        xmlns="http://www.w3.org/2000/svg"
        width="32"
        height="32"
        viewBox="0 0 256 256"
      >
        <path
          fill="#0078D4"
          d="M0 0h121.329v121.329H0zm134.671 0h121.329v121.329H134.671z"
        />
      </svg>
      <p>Microsoft</p>
    </div>
    <div class="flex flex-col items-center gap-2">
      <svg
        xmlns="http://www.w3.org/2000/svg"
        width="26.01"
        height="32"
        viewBox="0 0 256 315"
      >
        <path
          d="M213.803 167.03c.442 47.58 41.74 63.413 42.197 63.615c-.35 1.116-6.599 22.563-21.
        />
      </svg>
      <p>Apple</p>
    </div>
    <div class="flex flex-col items-center gap-2">
      <svg
```

Gambar 3.73 Kode Untuk Membuat Our Costumer

Berikutnya, untuk memberitahukan alasan mengapa PT. Infra Datatech lebih unggul di bidangnya dibandingkan perusahaan lain, penulis menambahkan bagian *Why We Are* seperti yang terlihat pada gambar 3.74. Bagian ini dirancang untuk menyoroti keunggulan, keahlian, dan nilai tambah yang ditawarkan oleh reLuna kepada klien. Untuk membuat tampilan tersebut, penulis menggunakan kode yang ditampilkan pada gambar 3.75, dengan tujuan menyampaikan poin-poin utama yang membedakan perusahaan dari kompetitor.



Gambar 3.74 Why We Are

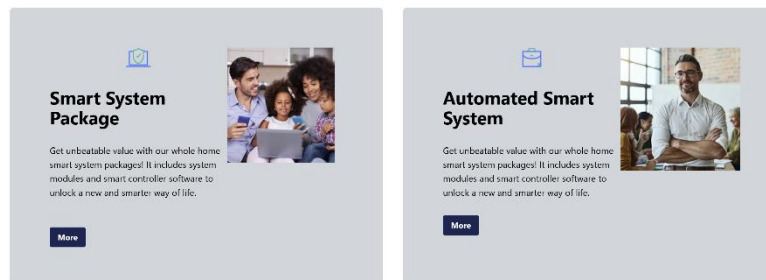
```

<div class="bg-blue-950">
  <div class="container mx-auto text-center text-white">
    <p class="text-2xl font-bold pt-10">Why We Are</p>
    <br />
    <ul class="pb-8">
      <li>
        • Our smart products are surprisingly simple to set up. No drilling, trunking, or rewiring.
      </li>
      <li>
        • Our products can be installed both in homes being renovated, or in the home you're living in.
      </li>
    </ul>
    <p>
    <p class="text-2xl font-bold pt-10">Interest? Please Contact Us</p>
    <div class="pb-12">
      <button
        class="inline-block text-sm bg-transparent hover:bg-white text-white font-semibold hover:opacity-75">
        <NuxtLink to="contactUs">Talk to our team</NuxtLink>
      </button>
    </div>
  </div>
</div>

```

Gambar 3.75 Kode Untuk Membuat Why We Are

Berikutnya ada halaman kategori produk, dimana halaman kategori produk ini untuk menampilkan kategori produk pada reLuna, deskripsi proyek yang dapat dilihat pada gambar 3.76. Untuk membuat halaman project penulis menggunakan kode pada gambar 3.77



Gambar 3.76 Kategori Produk

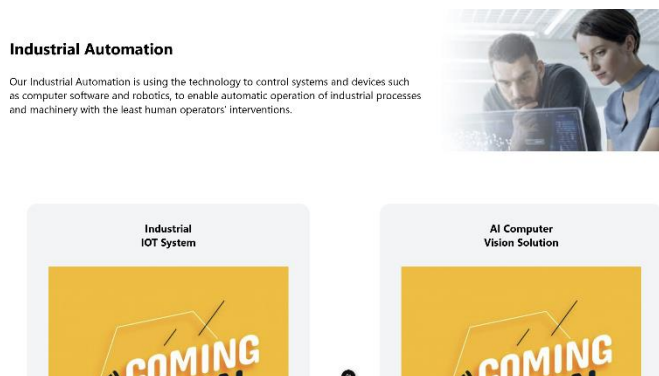
```

<!-- Card A -->
<div class="grid grid-cols-12 bg-gray-300 text-black rounded-lg">
  <div class="col-span-7">
    <div class="flex flex-col py-20 pl-20">
      <div class="self-center">
        
      </div>
      <h1 class="text-4xl font-bold mt-10 mb-8">Smart System Package</h1>
      <p class="text-lg mb-14">
        Get unbeatable value with our whole home smart system packages! It includes system m
      </p>
      <div class="flex">
        <button
          class="bg-blue-950 hover:bg-blue-900 text-white font-bold py-2 px-4 rounded mr-4"
        >
          <NuxtLink to="product">More</NuxtLink>
        </button>
      </div>
    </div>
  </div>
  <div class="col-span-5 mt-20">
    
  </div>
</div>
<!-- Card B -->
<div class="grid grid-cols-12 bg-gray-300 text-black rounded-lg">
  <div class="col-span-7">
    <div class="flex flex-col py-20 pl-20">
      <div class="self-center">
        
      </div>
      <h1 class="text-4xl font-bold mt-10 mb-8">Automated Smart System</h1>
      <p class="text-lg mb-8">
        Get unbeatable value with our whole home smart system packages! It includes system n
      </p>
      <div class="flex">

```

Gambar 3.77 Kode Untuk Membuat Kategori Produk

Selanjutnya, terdapat halaman Industrial Automation untuk menjelaskan product yang dijual di reLuna pada kategori sistem kontrol. Tampilan Our Products dapat dilihat pada gambar 3.78. Untuk membuat tampilan Our Products penulis menggunakan kode pada gambar 3.79



Gambar 3.78 Industrial Automation

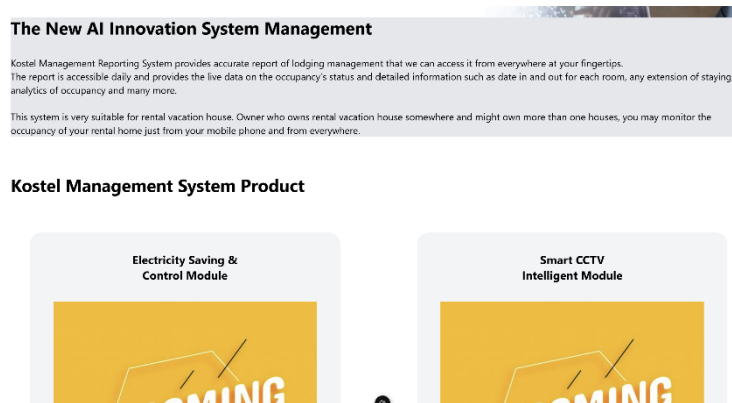

```

<div class="">
  <div class="container mx-auto">
    <div class="flex flex-col">
      <div class="py-20">
        <p class="text-4xl font-bold">Industrial Automation <br />
        <br />
        <p class="text-2xl">Our Industrial Automation is using the technology to control systems as computer software and robotics, to enable automatic operation of industrial processes and machinery with the least human operators' interventions. <br />
      </p>
    </div>
  </div>
</div>
</div>
</div>
</div>
<div class="w-full">
  <div class="container mx-auto mt-20" id="produk">
    <div class="grid grid-cols-1 sm:grid-cols-2 gap-20">
      <div class="flex flex-col bg-gray-100 mx-10 my-10 rounded-3xl">
        <p class="self-center text-2xl font-bold mt-10">Industrial</p>
        <p class="self-center text-2xl font-bold">IoT System</p>
        
        <button
          class="bg-white hover:bg-black outline outline-black hover:outline-black text-black"
        >
          <NuxtLink to="/totalSecurity">Lihat lebih detail</NuxtLink>
        </button>
      <div class="m-8 ml-8">
        <hr class="w-full h-px my-8 border-0 dark:bg-gray-700" />
      </div>
    </div>
  </div>
</div>

```

Gambar 3.79 Kode Untuk Membuat Industrial Automation

Selanjutnya, terdapat halaman Kostel Management System untuk menjelaskan product sistem management hotel dan produk yang dijual. Tampilan Our Products dapat dilihat pada gambar 3.80. Untuk membuat tampilan Our Products penulis menggunakan kode pada gambar 3.81.

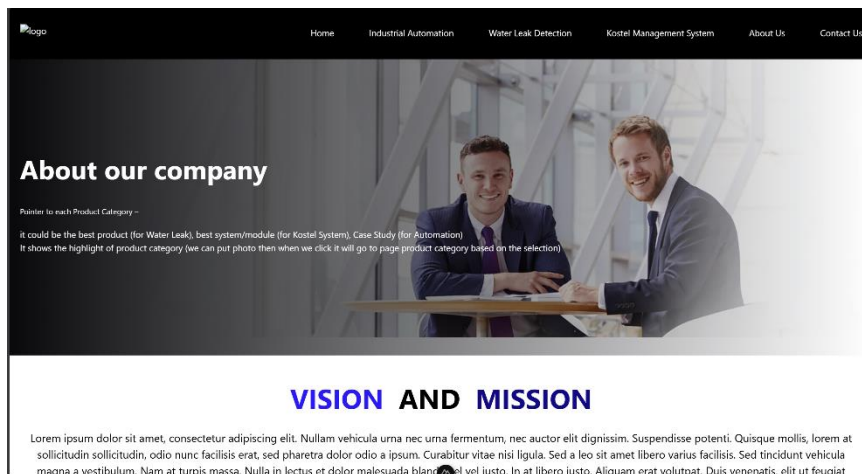


Gambar 3.80 Sistem Management Hotel

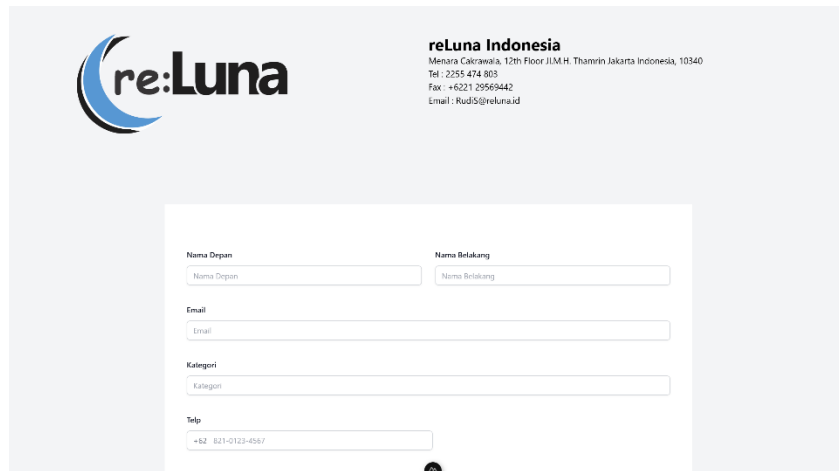
```
3 <div class="py-20">
4 <p class="text-4xl font-bold">Industrial Automation <br />
5 <br />
6 </p>
7 <p class="text-2xl">Our Industrial Automation is using the technology to control sy
8 as computer software and robotics, to enable automatic operation of industrial prod
9 and machinery with the least human operators' interventions. <br />
10 </p>
11 </div>
12 </div>
13 </div>
14 </div>
15 </div>
16 </div>
17 <div class="w-full">
18 <div class="container mx-auto mt-20" id="produk">
19 <div class="grid grid-cols-1 sm:grid-cols-2 gap-20">
20 <div class="flex flex-col bg-gray-100 mx-10 my-10 rounded-3xl">
21 <p class="self-center text-2xl font-bold mt-10">Industrial</p>
22 <p class="self-center text-2xl font-bold">IoT System</p>
23 </div>
24 </div>
25 </div>
26 </div>
```

Gambar 3.81 Kode Untuk Membuat Sistem Management Hotel

Selanjutnya, pada halaman About Us, penulis membuat bagian *About Us* untuk menjelaskan reLuna secara lebih detail, seperti yang ditampilkan pada gambar 3.82. Bagian ini bertujuan untuk memberikan informasi singkat mengenai visi, misi, dan nilai-nilai yang dianut oleh perusahaan. Untuk membuat bagian *About Us* tersebut, kita akan menggunakan kode yang ditunjukkan pada gambar 3.83.



Gambar 3.82 About Us



Gambar 3.84 Contact Us

```

<div className="mb-4">
  <label htmlFor="name" className="block text-sm font-medium text-gray-700">Name</label>
  <input
    type="text"
    id="name"
    className="mt-1 block w-full border-gray-300 rounded-lg shadow-sm focus:border-blue-500 focus:ring-blue-500"
    placeholder="Your Name"
  />
</div>
<div className="mb-4">
  <label htmlFor="email" className="block text-sm font-medium text-gray-700">Email</label>
  <input
    type="email"
    id="email"
    className="mt-1 block w-full border-gray-300 rounded-lg shadow-sm focus:border-blue-500 focus:ring-blue-500"
    placeholder="Your Email"
  />
</div>
<div className="mb-4">
  <label htmlFor="message" className="block text-sm font-medium text-gray-700">Message</label>
  <textarea
    id="message"
    rows="4"
    className="mt-1 block w-full border-gray-300 rounded-lg shadow-sm focus:border-blue-500 focus:ring-blue-500"
    placeholder="Your Message"
  ></textarea>
</div>
<button
  type="submit"
  className="w-full py-2 px-4 bg-blue-500 text-white rounded-lg hover:bg-blue-600 focus:outline-none"
  >
  Send Message
</button>
</Form>
</div>
<div className="flex-1 bg-white shadow-lg rounded-lg p-8">
  <h2 className="text-2xl font-medium mb-6">Our Contact Information</h2>

```

Gambar 3.85 Kode Untuk Membuat Contact Us

3.3 Kendala yang Ditemukan

Selama saya melaksanakan program magang ini, saya mengalami berbagai pengalaman, baik suka maupun tantangan. Di satu sisi, banyak hal positif yang saya rasakan, seperti kesempatan untuk belajar langsung dari para profesional di bidangnya, mengembangkan keterampilan teknis, serta terlibat dalam berbagai proyek yang menambah wawasan saya. Namun, di sisi lain, saya juga menghadapi beberapa kendala selama menjalani magang di PT. Infra Datatech Indonesia, antara lain:

1. Kurangnya pengalaman develop aplikasi android.
2. Tidak adanya mentor divisi IT.
3. Fitur yang kurang relevan terhadap aplikasi absen tersebut.

3.4 Solusi atas Kendala yang Ditemukan

Dikarenakan terdapat beberapa masalah yang dihadapi. Berikut merupakan Solusi yang dapat dilakukan untuk permasalahan-permasalahan tersebut:

1. Belajar dari beberapa referensi sumber pembelajaran yang berkaitan dengan projek yang sedang dikerjakan. Seperti melalui youtube dan platform lainnya.

Menggali dan mempelajari fitur-fitur yang lebih relevan supaya dapat meningkatkan aplikasi tersebut