Laporan Penetration Testing

Coffee Portal

Mobile Application

Maret 2025

Kelompok 6

Nama:

1. 2702356261 - David Christiano Kumala
2. 2702349395 - Bryant Junius
3. 2702222812 - Made Abhirama Adwitya Kartika
4. 2702322916 - Willy
5. 2702309800 - Brigitteney Yuanevida Hutabarat
6. 2702355473 - Nicolaus Willy Simatupang

Aplikasi: Coffee Portal
Platform: Android

1. Executive Summary

A. Ringkasan

Aplikasi mobile **Coffee Portal** yang dianalisis dalam proyek ini menunjukkan adanya beberapa kelemahan signifikan yang berpotensi membahayakan keamanan data dan layanan backend. Salah satu temuan paling kritis adalah adanya **eksposur informasi sensitif terkait endpoint API** yang digunakan oleh aplikasi. Informasi tersebut terlihat jelas dan dapat diakses tanpa adanya mekanisme pelindung tambahan, seperti otentikasi atau enkripsi yang memadai.

Eksposur semacam ini membuka peluang besar bagi aktor jahat untuk **mengakses data sensitif**, menyusup ke dalam komunikasi antara aplikasi dan server, serta **memanipulasi fungsi server-side** seperti proses pemesanan, login, hingga pengambilan data pengguna. Celah ini juga menandakan bahwa struktur keamanan saat ini belum cukup kuat untuk menahan upaya eksploitasi berbasis API injection, unauthorized access, atau sniffing data melalui man-in-the-middle attack.

Kombinasi dari kelemahan ini menunjukkan bahwa aplikasi **masih berada dalam risiko eksploitasi tingkat menengah hingga tinggi**, dan oleh karena itu memerlukan:

- **Peninjauan ulang konfigurasi keamanan jaringan (network security config)**,
- **Implementasi otentikasi API yang kuat**, seperti token-based authentication (JWT/OAuth2),
- **Pembersihan informasi sensitif dari file log dan kode sumber**, serta
- **Penerapan enkripsi menyeluruh** baik saat penyimpanan data maupun saat transmisi (data at rest dan data in transit).

Dengan segera dilakukan perbaikan pada celah-celah tersebut, maka tidak hanya risiko keamanan dapat ditekan secara signifikan, namun juga akan meningkatkan kepercayaan pengguna terhadap keamanan layanan Coffee Portal.

2.Tujuan dan Ruang Lingkup

A. Tujuan Pengujian

Tujuan utama dari pengujian ini adalah untuk **mengevaluasi aspek keamanan dari aplikasi mobile Coffee Portal**, dengan fokus pada analisis *client-side* (aplikasi Android dalam format `.aab`) dan komunikasi yang dilakukan dengan layanan API (*Application Programming Interface*) pada sisi server.

Pengujian ini bertujuan untuk:

- Mengidentifikasi potensi kerentanan pada aplikasi dari sisi distribusi aplikasi (.aab), kode sumber hasil dekompilasi, dan komunikasi jaringan.
- Mengkaji konfigurasi keamanan aplikasi, seperti permission, komponen yang diekspor, serta manajemen data sensitif.
- Melakukan observasi terhadap interaksi API untuk mengidentifikasi kelemahan seperti komunikasi tidak terenkripsi, kontrol akses yang lemah, atau penyimpanan data yang tidak aman.
- Menyusun dokumentasi bukti teknis (*proof of concept*) dan memberikan rekomendasi mitigasi.

B. Ruang Lingkup Pengujian

Pengujian dilakukan terhadap dua komponen utama berikut:

1. **Aplikasi Android – Format `.aab` (Android App Bundle)**
    - File aplikasi yang dianalisis adalah `coffee-portal.aab`.
    - File `.aab` dikonversi menjadi `.apk` terlebih dahulu untuk keperluan pengujian menggunakan tools melalui website https://aabtoapk.online/
    - Setelah konversi, analisis dilakukan terhadap konten APK hasil ekstraksi menggunakan tools seperti **Jadx** dan **Android Studio**.
    - Fokus pengujian mencakup analisis statis terhadap kode hasil dekompilasi, konfigurasi manifest, permission, dan potensi kerentanan klien.
2. **Layanan API Backend – `coffee-portal-api.climawan.com`**
    - Observasi dan analisis komunikasi antara aplikasi dan API backend.
    - Pengujian mencakup validasi autentikasi, integritas data, enkripsi, dan kontrol akses API.
    - Analisis dilakukan dengan dukungan emulator dan alat debug seperti **Android Studio**, serta eksplorasi endpoint dengan bantuan **Visual Studio** atau HTTP client lain.

C. Batasan Pengujian

- Pengujian dilakukan dalam model **grey-box**, hanya berdasarkan informasi dari file `.aab` yang diberikan dan observasi dari interaksi API.

- Tidak dilakukan pengujian langsung terhadap sistem produksi atau server internal.
- File `.aab` dikonversi terlebih dahulu ke `.apk`, sehingga beberapa perilaku runtime yang hanya aktif dalam *dynamic delivery modules* mungkin tidak terpantau.

3. Tools dan Metodologi

Pengujian dilakukan dengan pendekatan **static analysis** dan **manual review**, menggunakan kombinasi tools berikut:

- **Jadx GUI**: Untuk *reverse engineering* file APK dan menganalisis kode sumber Java dari aplikasi.
  
- **Android Studio**: Untuk memahami struktur proyek dan menjalankan proses debugging serta eksplorasi behavior aplikasi.
  

- **Visual Studio**: Digunakan untuk membantu debugging backend/API (jika ada proyek server-side yang tersedia), atau untuk pengujian jaringan via emulator/debug tool jika terhubung dengan back-end yang dikembangkan dalam .NET.
  

4. Hasil Pengujian

4.1. Root & Emulator Detection Bypass

- **ID Temuan:** M-001
- **Kategori:** Security Mechanism Bypass
- **Severity:** High
- **Deskripsi:**
  Security Mechanism yang didesain/dibentuk dalam code aplikasi Coffee Portal dapat diubah/dihapus ketika membuat APK serupa tanpa mekanisme keamanan. Serangan ini dapat menggantungkan serangan lain yang lebih kompleks dikarenakan attack vectornya semakin luas, yaitu device yang menggunakan emulator dapat menyerang aplikasi.
- **Langkah Reproduksi:**
  1. Install aplikasi
  2. Mengamati dan meneliti cara kerja aplikasi dengan membaca Activity yang muncul pertama kali

3. Terdapat pengecekan dari sisi aplikasi. Pengecekan ini didesain dalam aplikasi sehingga bisa diubah secara manual dalam file .smali menggunakan text editor
4. Menghapus cara kerja keamanan dari aplikasi
5. Build kembali file .apk dengan file .smali yang telah diubah
6. Buka aplikasi

- **Bukti:**

```
    if (!initialCheck()) {
        Toast.makeText(this, "Invalid environment to execute application...", 0).show();
        finish();
    } else {
        startActivity(new Intent(this, (Class<?>) HomeActivity.class));
    }
}
```

```
536
537     if-nez v0, :cond_0
538
539     .line 51
540     const-string v0, "Invalid environment to execute application..."
541
542     const/4 v1, 0x0
543
544     invoke-static {p0, v0, v1}, Landroid/widget/Toast;->makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid/widget/Toast;
545
```

- **Rekomendasi:**
  Pakai API Integrity untuk memastikan keamanan aplikasi dari perubahan internal (perubahan code)

4.2. Banner diganti (BOPLA)

- **ID Temuan:** M-002

- **Kategori:**
  - **Broken Object Property Level Authorization (BOPLA)**
  - **Insufficient UI Tampering Protection**
- **Severity:** Medium
- **Deskripsi:**

  Masalah **Broken Object Property** ini tidak hanya memungkinkan penyerang untuk mengakses data milik pengguna lain, tetapi juga memberikan potensi untuk **melakukan manipulasi data pada objek yang seharusnya tidak dapat diakses atau diubah oleh pengguna biasa**.

  Dalam konteks aplikasi *Coffee Portal*, penyerang yang memahami struktur payload API dapat memodifikasi properti penting dalam request JSON seperti `product_id`, `image_url`, `product_name`, atau `description` dengan mengganti nilai

properti tersebut dan mengirimkannya ke endpoint yang tidak memverifikasi hak akses objek.

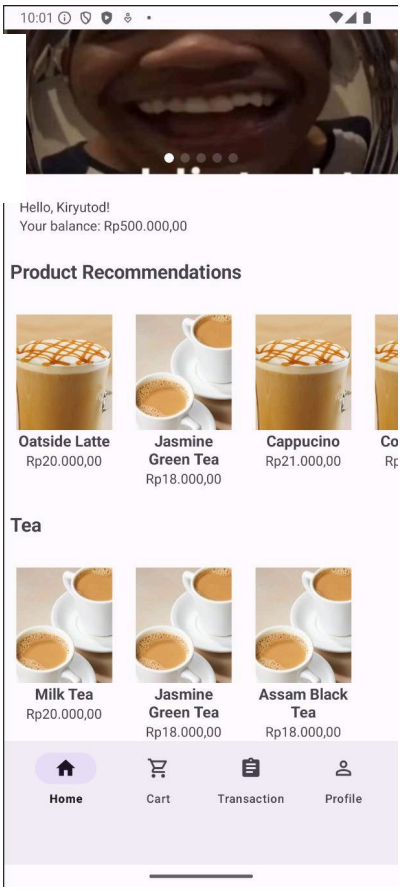
- **Langkah-langkah:**

  1. Menganalisis cara kerja HomeActivity
  2. Menganalisis atribut yang ada dalam HomeActivity
  3. Mengubah value sebuah atribut/variabel yang ingin diubah sesuai dengan keinginan di file .smali dengan teks editor
  4. Build kembali file .apk dengan .smali yang telah diperbarui
  5. Buka Aplikasi dan trigger/picu atribut supaya dapat dikerjakan aplikasi

- **Bukti:**

```
private void prepareSlideshows(View view) {
    this.svShowcase = (SliderView) view.findViewById(R.id.sv_showcase);
    Vector vector = new Vector();
    vector.add(new ShowcaseSlideEntity("Slide 1", "http://coffee-portal-api.climawan.com/assets/images/slide-1.webp"));
    vector.add(new ShowcaseSlideEntity("Slide 2", "http://coffee-portal-api.climawan.com/assets/images/slide-2.webp"));
    vector.add(new ShowcaseSlideEntity("Slide 3", "http://coffee-portal-api.climawan.com/assets/images/slide-3.webp"));
    vector.add(new ShowcaseSlideEntity("Slide 4", "http://coffee-portal-api.climawan.com/assets/images/slide-4.webp"));
    vector.add(new ShowcaseSlideEntity("Slide 5", "http://coffee-portal-api.climawan.com/assets/images/slide-5.webp"));
    this.showcaseSliderAdapter = new ShowcaseSliderAdapter(vector);
    SliderView sliderView = (SliderView) view.findViewById(R.id.sv_showcase);
    this.svShowcase = sliderView;
    sliderView.setSliderAdapter(this.showcaseSliderAdapter);
    this.svShowcase.startAutoCycle();
}
```

```
const-string v2, "Slide 1"

const-string v3, "http://coffee-portal-api.climawan.com/assets/images/slide-1.webp"

invoke-direct {v1, v2, v3}, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;-><init>(Ljava/lang/String;Ljava/lang/String;)V

invoke-virtual {v0, v1}, Ljava/util/Vector;->add(Ljava/lang/Object;)Z

.line 110
new-instance v1, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;

const-string v2, "Slide 2"

const-string v3, "http://coffee-portal-api.climawan.com/assets/images/slide-2.webp"

invoke-direct {v1, v2, v3}, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;-><init>(Ljava/lang/String;Ljava/lang/String;)V

invoke-virtual {v0, v1}, Ljava/util/Vector;->add(Ljava/lang/Object;)Z

.line 111
new-instance v1, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;

const-string v2, "Slide 3"

const-string v3, "http://coffee-portal-api.climawan.com/assets/images/slide-3.webp"

invoke-direct {v1, v2, v3}, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;-><init>(Ljava/lang/String;Ljava/lang/String;)V

invoke-virtual {v0, v1}, Ljava/util/Vector;->add(Ljava/lang/Object;)Z

.line 112
new-instance v1, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;

const-string v2, "Slide 4"

const-string v3, "http://coffee-portal-api.climawan.com/assets/images/slide-4.webp"

invoke-direct {v1, v2, v3}, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;-><init>(Ljava/lang/String;Ljava/lang/String;)V

invoke-virtual {v0, v1}, Ljava/util/Vector;->add(Ljava/lang/Object;)Z

.line 113
new-instance v1, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;

const-string v2, "Slide 5"

const-string v3, "http://coffee-portal-api.climawan.com/assets/images/slide-5.webp"

invoke-direct {v1, v2, v3}, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/ShowcaseSlideEntity;-><init>(Ljava/lang/String;Ljava/lang/String;)V

invoke-virtual {v0, v1}, Ljava/util/Vector;->add(Ljava/lang/Object;)Z
```



- **Rekomendasi:**

Gambar banner dijadikan sebagai resource/request seperti gambar item kopi yang menggunakan API. Dengan ini, gambar yang ditampilkan adalah data yang direquest dari API sehingga bisa memastikan proses memunculkan gambar tidak dapat diubah oleh client (pengguna)

4.3. Ubah Checkout OnClickListener untuk menambahkan balance

- **ID Temuan:** M-003

- **Kategori:**
  - **Insecure Client-Side Business Logic**
  - **Data Tampering**
  - **Broken Function Level Authorization** *(jika sistem tidak membedakan hak pengguna saat memproses request)*
- **Severity:** Medium
- **Deskripsi:**

  Aplikasi *Coffee Portal* melakukan validasi logika checkout dan transaksi secara eksklusif di sisi klien (client-side). Berdasarkan hasil analisis statis dari laporan MobSF dan dekompilasi kode, terlihat bahwa proses seperti perhitungan harga total, kuantitas produk, dan validasi keranjang dilakukan di dalam aplikasi tanpa adanya pengamanan tambahan atau verifikasi silang ke server.

  Hal ini memungkinkan pengguna atau penyerang untuk **memodifikasi alur checkout** melalui aplikasi yang telah diubah, misalnya dengan:

  - Menurunkan total harga sebelum dikirim ke server,

  - Menambahkan produk secara gratis,

  - Melakukan manipulasi diskon.

  Karena tidak ada *server-side validation* yang memverifikasi data akhir dari transaksi, maka sistem backend menjadi rentan terhadap data palsu yang dikirim dari klien.

- **Langkah-langkah:**

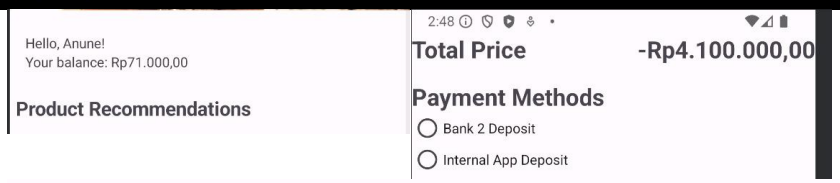
  1. Menganalisis CheckoutOnClickListener lewat JADX
  2. Mencocokkan hasil analisis CheckoutOnClickListener dengan file .smali
  3. Mengubah logika aritmatika nya
  4. Build kembali file .apk dengan file .smali yang telah diubah
  5. Lakukan transaksi seperti biasa

- **Bukti:**

```
public class CheckoutOnClickListener implements View.OnClickListener {
    private CheckoutOnClickListener() {
    }

    @Override // android.view.View.OnClickListener
    public void onClick(View view) {
        List<CartProductEntity> products = CartFragment.this.adapter.getProducts();
        if (products.isEmpty()) {
            Toast.makeText(CartFragment.this.view.getContext(), "You need to add product to cart!", 0).show();
            return;
        }
        Integer num = 0;
        for (CartProductEntity cartProductEntity : products) {
            num = Integer.valueOf(num.intValue() + (cartProductEntity.getPrice().intValue() * cartProductEntity.getCount().intValue()));
        }
        Bundle bundle = new Bundle();
        bundle.putInt(CartFragment.this.getString(R.string.ib_total_price), num.intValue());
        Intent intent = new Intent(CartFragment.this.view.getContext(), CartDetailActivity.class);
        intent.putExtra(CartFragment.this.getString(R.string.i_cart_info), bundle);
        CartFragment.this.startActivity(intent);
    }
```

```
.line 186
invoke-virtual {v0}, Ljava/lang/Integer;->intValue()I

move-result v0

invoke-virtual {v1}, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/CartProductEntity;->getPrice()Ljava/lang/Integer;

move-result-object v2

invoke-virtual {v2}, Ljava/lang/Integer;->intValue()I

move-result v2

invoke-virtual {v1}, Lcom/climawan/comp6844001_firsthalfproject/coffeeportal/app/models/entities/CartProductEntity;->getCount()Ljava/lang/Integer;

move-result-object v1

invoke-virtual {v1}, Ljava/lang/Integer;->intValue()I

move-result v1

mul-int/2addr v2, v1

sub-int/2addr v0, v2
```

You have Docker installed



- **Rekomendasi:**

Buatlah transaksi dalam bentuk API request untuk menghindari perubahan proses transaksi sehingga proses transaksi tidak dapat diubah dari sisi client (pengguna)

4.4. Ganti Description Item

- **ID Temuan:** M-004

- **Kategori:**
  - Data Tampering
  - Broken Object Level Authorization
  - Insecure Direct Object References (IDOR) *(jika dilakukan dengan mengganti ID produk)*

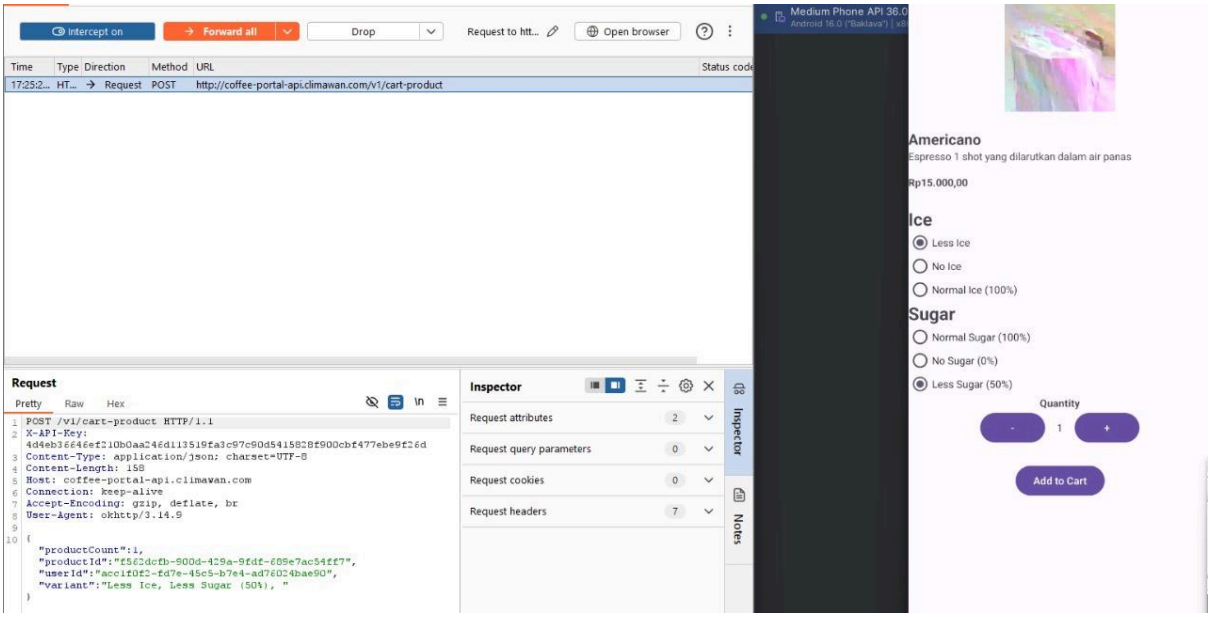
- **Severity:** Medium
- **Deskripsi:**

  Dalam pengecekan di burpsuite, kita menemukan bahwa ada cara untuk mengubah deskripsi item yang kita beli

- **Langkah-langkah:**

  1. Buka BurpSuite
  2. Lakukan pemesanan pada aplikasi
  3. Lakukan Intercept ketika merequest data (Klik menu yang tersedia di app)
  4. Saat data ter-intercept, ganti description yang ditangkap oleh BurpSuite, lalu send ke Servernya

- **Bukti:**



- **Rekomendasi:**

4.5. Authentication Bypass dengan mengubah fungsi validation

- **ID Temuan:** M-005

- **Kategori:**
  - **Credential Validation**

- **Severity:** Medium
- **Deskripsi:**

  Dalam pengujian terhadap aplikasi *Coffee Portal*, ditemukan kerentanan serius berupa **Authentication Bypass** yang terjadi akibat lemahnya mekanisme validasi email dalam proses login atau pendaftaran pengguna. Penyerang dapat memodifikasi kriteria validasi email seperti pola regex, domain, atau flag validasi melalui manipulasi sisi klien, baik dengan cara dekompilasi dan edit kode menggunakan Jadx, maupun hook runtime menggunakan tools seperti Frida. Karena validasi hanya terjadi di sisi aplikasi tanpa verifikasi tambahan dari server, attacker dapat menyisipkan alamat email yang tidak sah atau memalsukan format email agar lolos pengecekan dan langsung diproses oleh sistem backend. Akibatnya, penyerang dapat melewati proses otentikasi normal, mendapatkan akses tidak sah, atau membuat akun baru dengan identitas palsu tanpa proses verifikasi yang semestinya. Celah ini sangat berbahaya karena dapat dimanfaatkan untuk mengakses layanan internal, menghindari filter keamanan, atau

bahkan menyamar sebagai pengguna lain. Kerentanan ini menunjukkan kurangnya validasi sisi server (server-side validation) dan kontrol autentikasi yang kuat, sehingga perlu segera ditangani melalui penerapan validasi yang konsisten di backend, penggunaan token otentikasi yang kuat, dan pembatasan akses berdasarkan domain atau pola email yang telah disetujui.

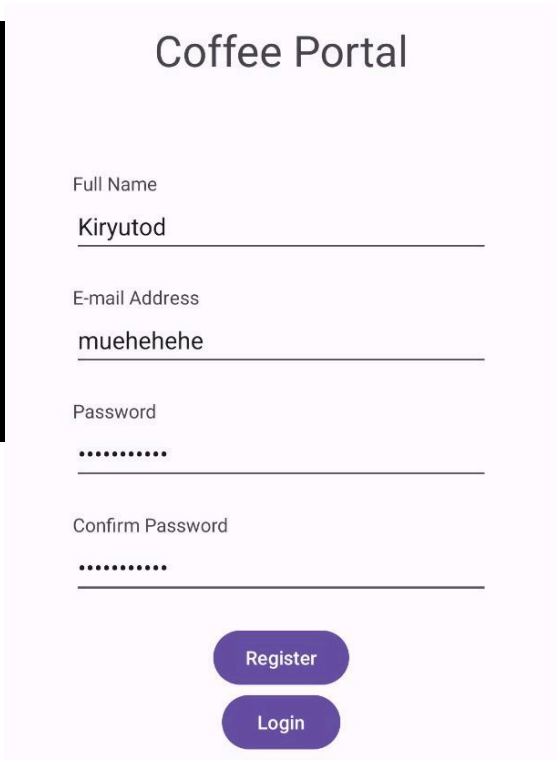
- **Langkah-langkah:**

  1. Menganalisis function registration
  2. Mencocokkan kembali hasil analisis dengan file .smali
  3. Mengubah/menghapus syarat/ketentuan validasi dari aplikasi di file .smali dengan teks editor
  4. Build kembali file .apk dengan .smali yang telah diubah
  5. Melakukan registrasi akun seperti biasa

- **Bukti:**

```
if (!str2.endsWith("@climawan.com") && !str2.endsWith("@vi.com")) {
    Toast.makeText(this, "E-mail address must ends with @climawan.com or @vi.com", 0).show();
    return false;
}
```

```
    .line 72
    :cond_1
    const-string p1, "@climawan.com"

    invoke-virtual {p2, p1}, Ljava/lang/String;->endsWith(Ljava/lang/String;)Z

    move-result p1

    if-nez p1, :cond_2

    const-string p1, "@vi.com"

    invoke-virtual {p2, p1}, Ljava/lang/String;->endsWith(Ljava/lang/String;)Z

    move-result p1

    if-nez p1, :cond_2

    .line 73
    const-string p1, "E-mail address must ends with @climawan.com or @vi.com"

    invoke-static {p0, p1, v0}, Landroid/widget/Toast;->makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid/widget/Toast;

    move-result-object p0

    invoke-virtual {p0}, Landroid/widget/Toast;->show()V

    return v0
```



- **Rekomendasi:**
  Proses administration didesain menjadi sebuah request ke API sehingga proses administrasi berjalan di server, bukan di client. Dengan ini, client (pengguna) tidak dapat mengubah proses validasi authentication dari aplikasi Coffee Portal

4.6. Admin Dummy

- **ID Temuan:** M-006

- **Kategori:**
  - Brute Force Attack / Insecure Authentication
  - Lack of Rate Limiting
  - Weak Account Lockout Mechanism

- **Severity:** High

- **Deskripsi:**

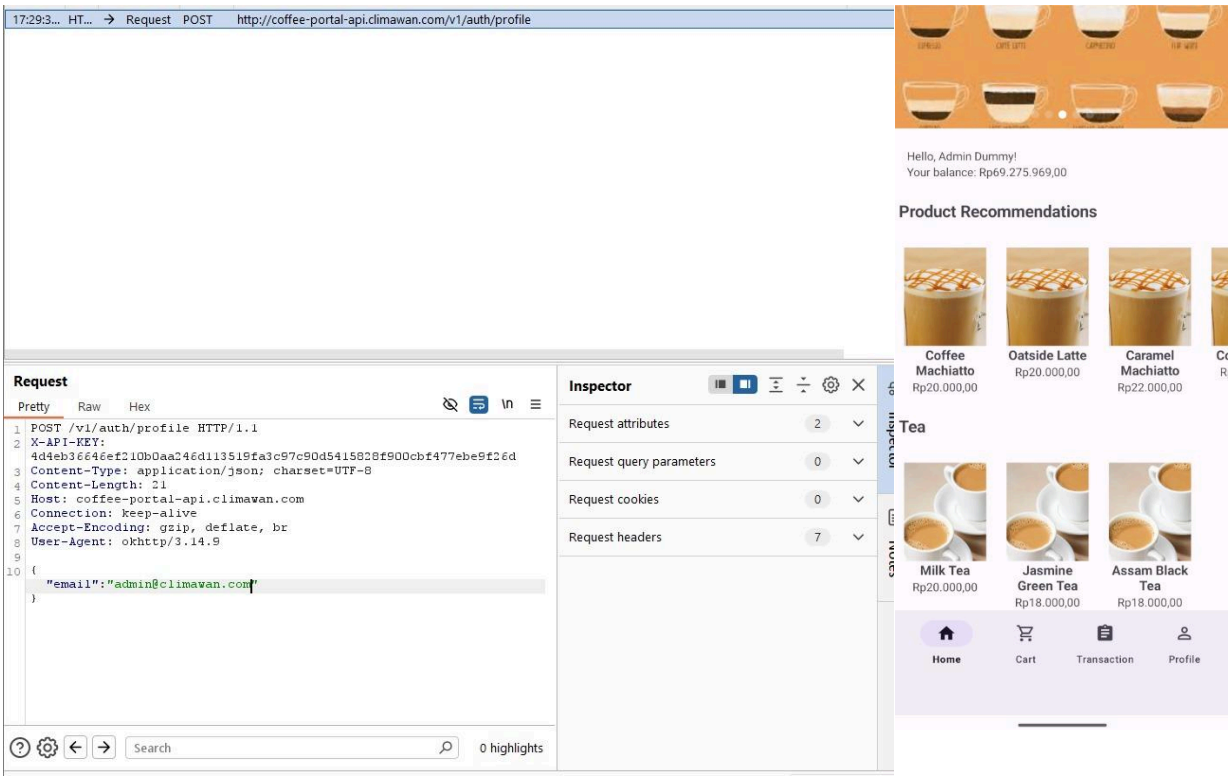
  Dalam pengujian terhadap aplikasi *Coffee Portal*, ditemukan bahwa attacker dapat mengubah deskripsi produk yang ditampilkan kepada pengguna lain dengan memanipulasi properti data dalam request yang dikirim ke server. Hal ini terjadi karena tidak adanya validasi yang memadai terhadap kepemilikan objek di sisi server, sehingga penyerang cukup mengetahui product_id untuk dapat mengubah isi deskripsi, nama produk, harga, atau informasi lainnya, meskipun bukan pemilik resmi data tersebut. Serangan ini tergolong sebagai Data Tampering, dan sangat berbahaya karena deskripsi produk merupakan komponen utama dalam proses pengambilan keputusan pembelian. Dengan memanfaatkan celah ini, attacker dapat menyisipkan informasi palsu, menipu pengguna dengan promo yang tidak nyata, atau mencemarkan nama baik penjual lain, yang pada akhirnya mengganggu integritas sistem dan kepercayaan pengguna terhadap platform.

- **Langkah-langkah:**

  1. Buka aplikasi target melalui browser (yang sudah disambungkan ke Burp).
  2. Buat akun pengguna biasa menggunakan form registrasi.
  3. Saat proses ini terjadi, **Burp Suite akan mencatat HTTP request** di tab **HTTP history** (Proxy → HTTP history).
  4. Saat melakukan intercept terdapat body JSON “email:[user]”

5. Dengan bruteforce, mencoba common username/email dengan menambahkan @climawan.com
6. Berhasil Login ke User tersebut

- **Bukti:**



- **Rekomendasi:**
  Membuat agar account admin tidak mudah diretas dengan merubah nama admin menjadi yang lain, misalnya nama owner atau nama staff atau nama yang tidak mudah diprediksi