**LAPORAN**

**TUGAS AKHIR**

**APLIKASI PELANGGARAN SISWA (APES)**

**BERBASIS WEB & MOBILE**

Dibuat sebagai salah satu prasyarat sebelum mengikuti Program Praktik Kerja Lapangan Rekayasa Perangkat Lunak SMK Negeri 1 Cimahi

Oleh:

|  |  |
| --- | --- |
| AZKHA NAZZALA PRASADHA DIES | (211115694) |
| IRNI DWI ADRYANI | (211115706) |



**SEKOLAH MENENGAH KEJURUAN NEGERI 1 CIMAHI**

**KOTA CIMAHI**

**2023**

**LEMBAR PENGESAHAN APLIKASI PELANGGARAN SISWA BERBASIS MOBILE DAN WEBSITE**

Laporan ini telah disetujui dan diperiksa oleh :

|  |  |  |  |
| --- | --- | --- | --- |
| Ketua Kompetensi Keahlian, |  | Pembimbing, | |
| Agus Rahmawan, S.T. |  | Chandra Hardiawan S.pd, MT. | |
| NIP. 197512022006041010 |  | NIP. 198403272010011005 |

# 

# KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena dengan rahmat, karunia, taufik, inayah, dan hidayah-Nya sehingga kami dapat menyelesaikan laporan “Aplikasi Pelanggaran Siswa (APES) Berbasis Web dan Mobile” tepat pada waktunya. Tak lupa solawat serta salam semoga selalu tercurahkan kepada junjungan alam Nabi Muhammad SAW. Kepada keluarga, sahabat, dan kita selaku umatnya hingga akhir zaman.

Adapun maksud dan tujuan dari penulisan laporan ini adalah sebagai kewajiban para siswa tingkat III di Kompetensi Keahlian Rekayasa Perangkat Lunak yang telah menyelesaikan tugas akhir. Kami mengucapkan terima kasih kepada:

1. Orang tua dan segenap keluarga penulis yang senantiasa memberikan semangat, dukungan dan do’a kepada penulis.
2. Agus Priyatmono Nugroho, S.Pd, M.Si, selaku Kepala SMK Negeri 1 Cimahi.
3. Agus Rahmawan, S.T, selaku Ketua Kompetensi Keahlian Rekayasa Perangkat Lunak.
4. Chandra Hardiawan S.pd, MT, selaku Wali Kelas tingkat III Kompetensi Keahlian Rekayasa Perangkat Lunak A.
5. Seluruh staff, guru dan karyawan SMK Negeri 1 Cimahi.
6. Seluruh sahabat dan teman penulis baik di lingkungan SMK Negeri 1 Cimahi maupun di luar lingkungan sekolah.

Akhir kata penulis harap laporan ini dapat bermanfaat dan berguna bagi pembaca. Penulis menyadari atas ketidaksempurnaan dalam pembuatan laporan ini, maka dari itu penulis mengharapkan adanya masukan berupa kritik dan saran agar laporan ini bisa lebih sempurna. Bimbingan dan perhatian yang diberikan semoga senantiasa mendapatkan pahala serta rahmat dari Allah SWT.

Cimahi, Februari 2023

Penulis

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc152741579)

[DAFTAR ISI iii](#_Toc152741580)

[DAFTAR GAMBAR vi](#_Toc152741581)

[DAFTAR TABEL viii](#_Toc152741582)

[BAB I PENDAHULUAN 1](#_Toc152741583)

[1.1. Latar Belakang 1](#_Toc152741584)

[1.2. Tujuan 1](#_Toc152741585)

[1.3. Pembatasan Masalah 1](#_Toc152741586)

[1.4. Sistematika Pembahasan 2](#_Toc152741587)

[BAB II LANDASAN TEORI 3](#_Toc152741588)

[2.1. Aplikasi Mobile 3](#_Toc152741589)

[2.1.1. Pengertian Aplikasi Mobile 3](#_Toc152741590)

[2.1.2. Manfaat Aplikasi Mobile 3](#_Toc152741591)

[2.1.3. Keunggulan Aplikasi Mobile 4](#_Toc152741592)

[2.2. Android Studio 4](#_Toc152741593)

[2.2.1. Pengertian 4](#_Toc152741594)

[2.2.2. Instalasi 5](#_Toc152741595)

[2.2.3. Keunggulan 9](#_Toc152741596)

[2.3. Java 9](#_Toc152741597)

[2.3.1. Pengertian 9](#_Toc152741598)

[2.3.2. Sejarah 9](#_Toc152741599)

[2.3.3. Keunggulan 11](#_Toc152741600)

[2.4. XML 11](#_Toc152741601)

[2.4.1. Pengertian 11](#_Toc152741602)

[2.4.2. Sejarah 12](#_Toc152741603)

[2.4.3. Keunggulan 12](#_Toc152741604)

[2.4.4. Kegunaan 12](#_Toc152741605)

[2.5. Figma 13](#_Toc152741606)

[2.5.1. Pengertian 13](#_Toc152741607)

[2.5.2. Sejarah 13](#_Toc152741608)

[2.5.3. Keunggulan 14](#_Toc152741609)

[2.5.4. Fitur-fitur Figma 14](#_Toc152741610)

[2.6. System Operasi Android 14](#_Toc152741611)

[2.6.1. Pengertian 14](#_Toc152741612)

[2.6.2. Sejarah Android 15](#_Toc152741613)

[2.6.3. Perkembangan Seri Android 16](#_Toc152741614)

[2.7. ReactJS 21](#_Toc152741615)

[2.7.1. Pengertian 21](#_Toc152741616)

[2.7.2. Fitur React 22](#_Toc152741617)

[2.7.3. Kelebihan 23](#_Toc152741618)

[2.7.4. Cara Menginstall Reactjs 24](#_Toc152741619)

[2.8. Laravel 24](#_Toc152741620)

[2.8.1. Pengertian 24](#_Toc152741621)

[2.8.2. Fungsi Laravel 25](#_Toc152741622)

[2.8.3. Kelebihan Laravel 25](#_Toc152741623)

[2.8.4. Fitur-fitur Laravel 25](#_Toc152741624)

[2.9. Tailwind CSS 26](#_Toc152741625)

[2.9.1. Pengertian 26](#_Toc152741626)

[2.9.2. Sejarah Singkat 26](#_Toc152741627)

[2.9.3. Kelebihan 27](#_Toc152741628)

[2.9.4. Kekurangan Tailwind CSS 27](#_Toc152741629)

[2.10. PHPMyAdmin 27](#_Toc152741630)

[2.10.1. Pengertian 27](#_Toc152741631)

[2.10.2. Fungsi PHPMyAdmin 27](#_Toc152741632)

[2.10.3. Kelebihan 28](#_Toc152741633)

[2.11. API 28](#_Toc152741634)

[2.11.1. Pengertian 29](#_Toc152741635)

[2.11.2. Jenis 29](#_Toc152741636)

[2.12. Postman 30](#_Toc152741637)

[BAB III PERENCANAAN 31](#_Toc152741638)

[3.1. Alur Sistem Berjalan 31](#_Toc152741639)

[3.1.1. Alur Sistem Berjalan Aplikasi 31](#_Toc152741640)

[3.2. Alur Sistem Aplikasi 31](#_Toc152741641)

[3.3. Data Flow Diagram (DFD) 32](#_Toc152741642)

[3.4. Entity Relation Diagram 33](#_Toc152741643)

[3.5. Struktur Tabel 33](#_Toc152741644)

[3.6. Mockup 34](#_Toc152741645)

[BAB IV PEMBUATAN 38](#_Toc152741646)

[4.1. Pembuatan Aplikasi Mobile 38](#_Toc152741647)

[4.1.1. Pembuatan Layout 38](#_Toc152741648)

[4.1.2. Syntax Java 60](#_Toc152741649)

[4.1.3. Pembuatan API 88](#_Toc152741650)

[4.2. Pembuatan Aplikasi Web 95](#_Toc152741651)

[4.2.1. Pembuatan Backend di Laravel 95](#_Toc152741652)

[4.2.2. Pembuatan Backend di React js 101](#_Toc152741653)

[BAB V PENUTUP 130](#_Toc152741654)

[5.1. Kesimpulan 130](#_Toc152741655)

[DAFTAR PUSTAKA 131](#_Toc152741656)

[LAMPIRAN 133](#_Toc152741657)

# DAFTAR GAMBAR

[Gambar 2. 1 Logo Android Studio 4](#_Toc152757796)

[Gambar 2. 2 Web Android Studio 5](#_Toc152757797)

[Gambar 2. 3 Opsi Download Android Studio 5](#_Toc152757798)

[Gambar 2. 4 License Agreement Android Studio 5](#_Toc152757799)

[Gambar 2. 5 *Setup* Android Studio 6](#_Toc152757800)

[Gambar 2. 6 *Choose Components* Android Studio 6](#_Toc152757801)

[Gambar 2. 7 Pemilihan lokasi *Directory* 7](#_Toc152757802)

[Gambar 2. 8 Menu *Instalasi* 7](#_Toc152757803)

[Gambar 2. 9 Proses *Instalasi* Selesai 8](#_Toc152757804)

[Gambar 2. 10 Instalasi Android Studio Selesai 8](#_Toc152757805)

[Gambar 2. 11 Logo Java 9](#_Toc152757806)

[Gambar 2. 12 Logo XML 11](#_Toc152757807)

[Gambar 2. 13 Logo Figma 13](#_Toc152757808)

[Gambar 2. 14 Logo Android 14](#_Toc152757809)

[Gambar 2. 15 *Create Project on React* 24](#_Toc152757810)

[Gambar 2. 16 Cara Kerja API 29](#_Toc152757811)

[Gambar 2. 17 Logo Postman 30](#_Toc152757812)

[Gambar 3. 1 Alur Sistem Berjalan Aplikasi 31](#_Toc152758334)

[Gambar 3. 2 Alur Sistem Aplikasi 32](#_Toc152758335)

[Gambar 3. 3 *Data Flow Diagram* 33](#_Toc152758336)

[Gambar 3. 4 *Entity Relation Diagram* 33](#_Toc152758337)

[Gambar 3. 5 Struktur Tabel 34](#_Toc152758338)

[Gambar 3. 6 *Splash Screen Mobile* 34](#_Toc152758339)

[Gambar 3. 7 Halaman Login *Mobile* 35](#_Toc152758340)

[Gambar 3. 8 *Main Page Mobile* 36](#_Toc152758341)

[Gambar 3. 9 *Input Data Mobile* 36](#_Toc152758342)

[Gambar 3. 10 *History Mobile* 37](#_Toc152758343)

[Gambar 3. 11 *Login Page Web* 38](#_Toc152758344)

[Gambar 3. 12 *Home Page Web* 38](#_Toc152758345)

[Gambar 3. 13 Tabel Pelnaggaran Web 39](#_Toc152758346)

[Gambar 3. 14 Kategori Pelanggaran Web 39](#_Toc152758347)

[Gambar 4. 1 *Splash Screen Page* 40](#_Toc152759802)

[Gambar 4. 2 Halaman Login 43](#_Toc152759803)

[Gambar 4. 3 Halaman Dashboard 46](#_Toc152759804)

[Gambar 4. 4 Halaman Input Pelanggar 51](#_Toc152759805)

[Gambar 4. 5 Halaman History 58](#_Toc152759806)

[Gambar 4. 6 *Card View* 61](#_Toc152759807)

[Gambar 4. 7 *Sidebar* 114](#_Toc152759808)

[Gambar 4. 8 Halaman Login Web 125](#_Toc152759809)

[Gambar 4. 9 Halaman Dashboard Web 129](#_Toc152759810)

[Gambar 4. 10 Halaman Dashboard Web 129](#_Toc152759811)

[Gambar 4. 11 Tabel Pelanggaran 136](#_Toc152759812)

# DAFTAR TABEL

[Tabel 4. 1 Pengujian Login Admin 140](#_Toc152767875)

[Tabel 4. 2 Pengujian Halaman Dashboard 140](#_Toc152767876)

[Tabel 4. 3 Pengujian Halaman Tambah Data Pelanggaran 141](#_Toc152767877)

[Tabel 4. 4 Pengujian Halaman History 141](#_Toc152767878)

[Tabel 4. 5 Pengujian Login Superadmin 141](#_Toc152767879)

[Tabel 4. 6 Pengujian Halaman Dashboard 142](#_Toc152767880)

[Tabel 4. 7 Pengujian Data Perjurusan 142](#_Toc152767881)

[Tabel 4. 8 Pengujian Hapus Data Pelanggaran 142](#_Toc152767882)

[Tabel 4. 9 Pengujian Sistem Logout 142](#_Toc152767883)

# PENDAHULUAN

## Latar Belakang

Pentingnya pengelolaan pelanggaran siswa di lingkungan pendidikan menantang sekolah untuk meningkatkan efisiensi dan akurasi dalam pencatatan serta pelaporan pelanggaran. Sistem manajemen manual seringkali menyulitkan proses tersebut, dan inovasi diperlukan untuk memperbaiki kelemahan ini. Dengan memanfaatkan aplikasi pelanggaran siswa, sekolah dapat merespons kebutuhan ini dengan lebih baik. Aplikasi tersebut tidak hanya memungkinkan pencatatan yang lebih cepat dan akurat, tetapi juga memberikan akses yang lebih mudah ke data pelanggaran. Penerapan teknologi ini juga sejalan dengan tren penggunaan teknologi dalam dunia pendidikan, memastikan bahwa sekolah terus berinovasi untuk memberikan pengalaman yang lebih baik bagi siswa. Tentu saja, penting untuk memperhatikan keamanan data dan privasi agar aplikasi ini memberikan manfaat tanpa mengorbankan integritas dan keamanan informasi. Seiring dengan perkembangan regulasi pendidikan, aplikasi pelanggaran siswa dapat membantu sekolah memenuhi standar dan meningkatkan akuntabilitas. Dengan demikian, pengembangan dan implementasi aplikasi pelanggaran siswa menjadi langkah positif menuju manajemen pendidikan yang lebih efisien dan terukur.

## Tujuan

Dibuatnya aplikasi ini bertujuan untuk:

1. Mempermudah guru untuk mendata pelanggaran yang dilakukan oleh siswa
2. Menampilkan data pelanggaran yang dilakukan oleh siswa

## Pembatasan Masalah

Dalam menyusun laporan ini, pembahasan materi diberikan batasan sebagai berikut:

1. Merancang kerangka tampilan aplikasi pelanggaran siswa*.*
2. Menyusun kode untuk merangkai tampilan aplikasi menggunakan Java untuk mobile serta ReactJs dan Laravel untuk web.
3. Pembahasan hanya bertuju pada web dan mobile.

## Sistematika Pembahasan

Penyusunan laporan ini dibagi menjadi lima bab ditambah bagian awal yang terdiri dari lembar pengesahaan dari Pihak Sekolah, Kata Pengantar, Daftar Isi, Daftar Gambar, dan Daftar Tabel. Pada Bagian Akhir, dilampirkan daftar pustaka sebagai referensi dan didapatkan penulis untuk pembuatan Laporan ini :

**BAB I: PENDAHULUAN**

Dalam bab ini, penulis membahas mengenai latar belakang masalah, tujuan dari pemilihan judul, pembatasan masalah, dan sistematika pembahasan.

**BAB II: LANDASAN TEORI**

Pada bab ini berisi tentang teori-teori yang menjadi acuan, landasan terhadap masalah yang dibahas.

**BAB III: PERENCANAAN**

Pada bab ini berisi tentang perencanaan dari aplikasi pelanggaran siswa

**BAB IV: PEMBAHASAN**

Pada bab ini berisi tentang penguraian hasil pembuatan aplikasi pelanggaran siswa

**BAB V: PENUTUP**

Pada bab ini berisi kesimpulan dan uraian masalah yang telah dibahas serta saran-saran teknis dan masukan kepada pihak sekolah dan perusahaan dalam pembuatan aplikasi pelanggaran siswa

# LANDASAN TEORI

### Pengertian Aplikasi Mobile

Mobile apps merupakan perangkat lunak berupa aplikasi yang dikembangkan menggunakan program komputerisasi untuk disematkan pada perangkat mobile seperti ponsel, tablet dan jam tangan digital.

Awal kemunculan mobile apps di tahun 2000-an, tepatnya sekitar tahun 2009. Aplikasi mobile ini merupakan pengembangan dari rancangan aplikasi yang sebelumnya belum terlalu canggih.

Mobile apps di masa kini banyak dikenal sebagai aplikasi smartphone yang rancangannya didukung dengan pemrograman yang sudah elit.

Untuk membuat sebuah aplikasi, perlu membutuhkan keahlian khusus, yang mana dalam prosesnya dibantu dengan program computer. Setiap tahun, aplikasi mobile ini selalu dikembangkan atau di upgrade untuk meningkatkan kinerjanya.

Mobile apps yang sudah dirakit dengan sempurna biasanya akan di upload di situs khusus yang berisi sekumpulan aplikasi mobile. Misal untuk smartphone android bisa mengunduh mobile apps di Playstore yang dimiliki pihak Google Corporation.

Sedangkan untuk iPhone bisa mengunduhnya di AppStore yang merupakan besutan perusahaan Apple Inc. Namun beberapa mobile apps yang belum resmi akan dirilis di situs website lain di internet.

### Manfaat Aplikasi Mobile

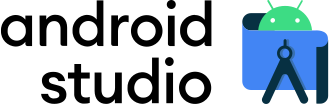
1. Sebagai sarana penyebar informasi
2. Menjalin komunikasi jarak jauh
3. Memudahkan pekerjaan manusia
4. Meningkatkan Proses Bisnis

### Keunggulan Aplikasi Mobile

Selain adanya manfaat yang bisa dinikmati pengguna, mobile apps juga memiliki beberapa keunggulan. Berikut adalah keunggulan-keunggulan mobile apps.

1. Memiliki tampilan awal atau user interface (UI) yang menarik. Tampilan UI bisa dirancang dengan desain layout, tema dan icon yang menarik. Yang mana biasanya disesuaikan dengan bidang aplikasi tersebut.
2. Pengembangan aplikasi tidak semua bisa di akses dengan internet saja. banyak aplikasi yang bisa diakses tanpa menggunakan internet atau dalam keadaan offline. Salah satunya aplikasi game.
3. Mobile apps bisa diakses kapanpun tanpa batasan waktu. Selama 24 jam, pengguna bisa mengakses aplikasi mobile dengan lancar melalui perangkat mobile yang dimiliki.

## Android Studio



Gambar 2. 1 Logo Android Studio

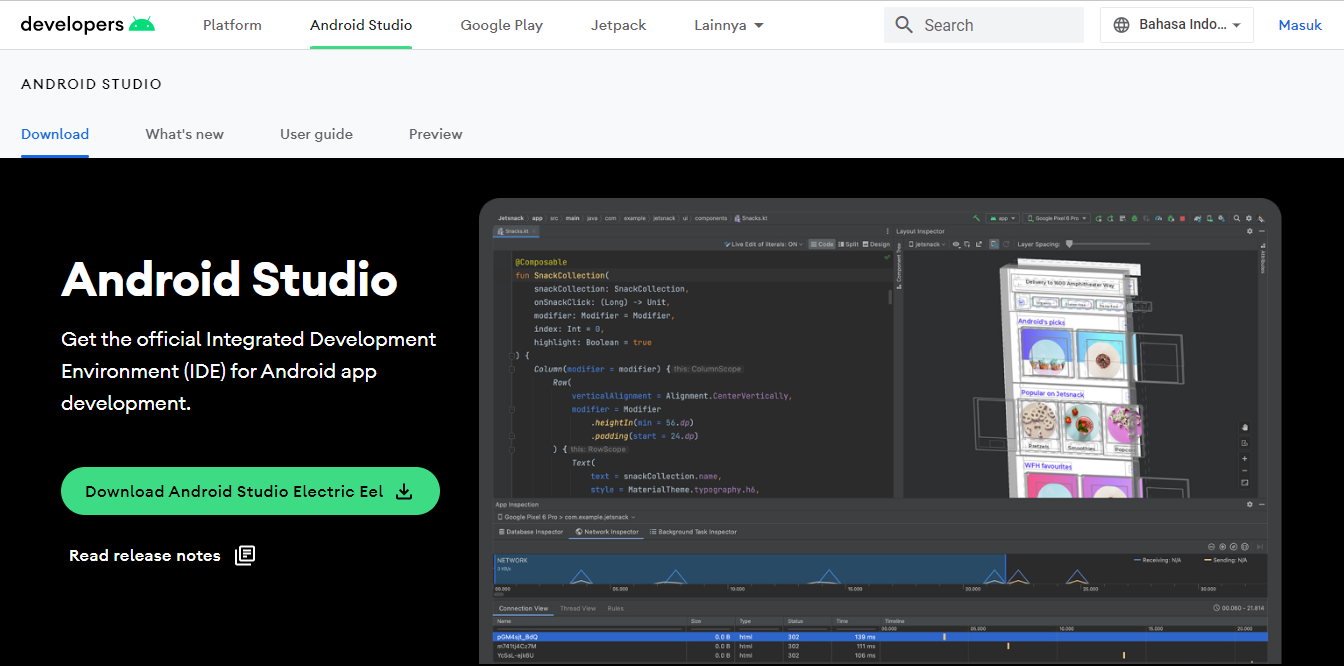
### Pengertian

Android Studio adalah lingkungan pengembangan terintegrasi (IDE) untuk mengembangkan aplikasi Android berdasarkan *IntelliJ IDEA*. Android Studio tidak hanya editor kode dan alat pengembang *IntelliJ* yang kuat, tetapi juga menawarkan banyak fitur lainnya.

Android Studio adalah perangkat lunak resmi yang sepenuhnya disetujui oleh Google sebagai perusahaan induk dari sistem operasi Android. Sebelum Google resmi merilis Android Studio dan mendukung penuh, Google sudah mendukung Eclipse. Tempat dimana Eclipse pernah ada adalah perangkat lunak atau IDE yang digunakan pengembang Android untuk mengembangkan aplikasi Android.

### Instalasi

1. Buka web <https://developer.android.com/studio>



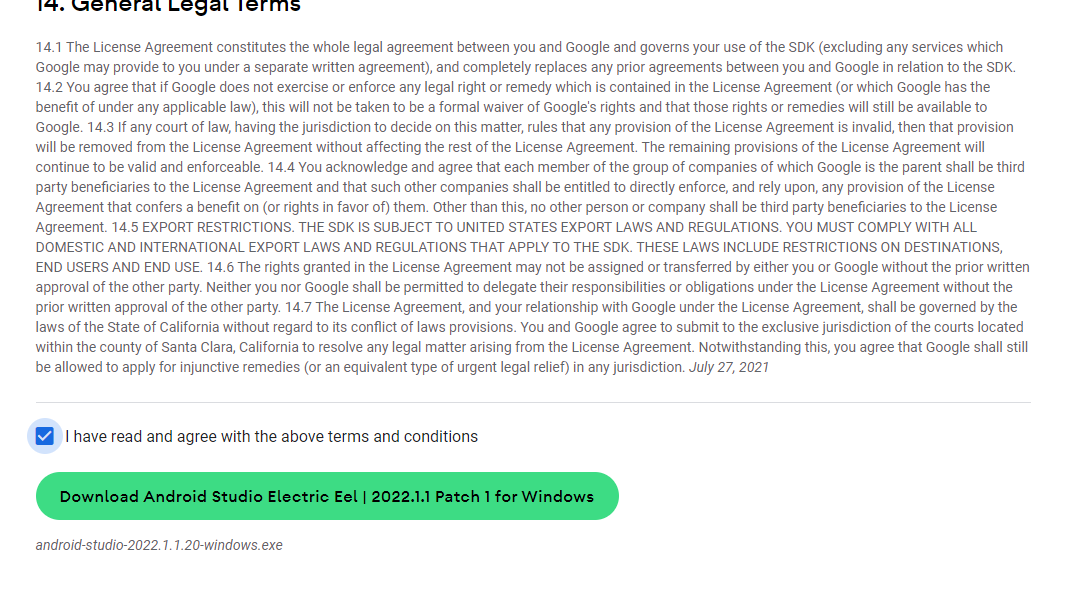
Gambar 2. 2 Web Android Studio

1. Pilih opsi Download Android Studio



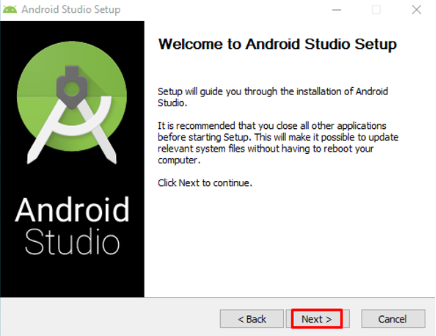
Gambar 2. 3 Opsi Download Android Studio

1. Scroll sampai bawah dan ceklis kolom “I have read and agree”



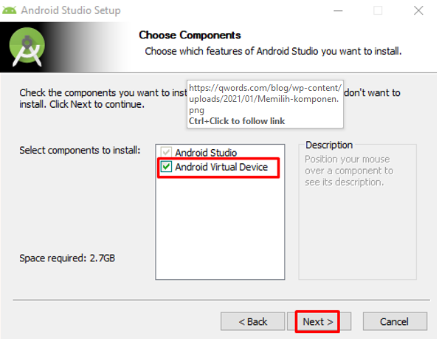
Gambar 2. 4 License Agreement Android Studio

1. Klik *download*
2. Tunggu sampai unduhan selesai
3. Buka file unduhan
4. Klik *next*



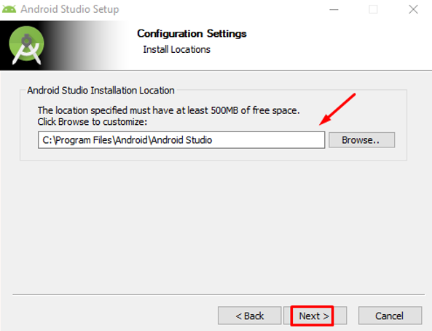
Gambar 2. 5 *Setup* Android Studio

1. Klik *next*



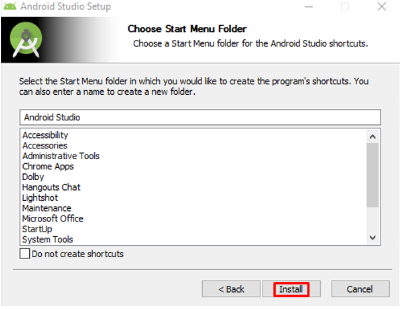
Gambar 2. 6 *Choose Components* Android Studio

1. Klik *next*



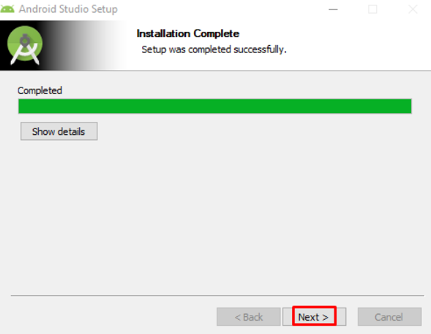
Gambar 2. 7 Pemilihan lokasi *Directory*

1. Klik *install*



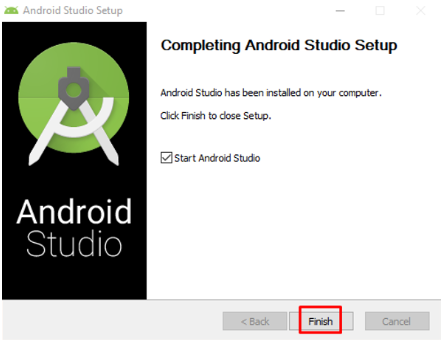
Gambar 2. 8 Menu *Instalasi*

1. Klik *next*



Gambar 2. 9 Proses *Instalasi* Selesai

1. Klik *finish*



Gambar 2. 10 Instalasi Android Studio Selesai

### Keunggulan

Android Studio memiliki keunggulan dibandingkan dengan *text editor* yang lain, yaitu :

1. *Text editor* gratis.
2. Sudah terintegrasi GIT
3. Memiliki layout editor
4. Emulator yang cepat
5. Memiliki Intelligence Code Editor

## Java



Gambar 2. 11 Logo Java

### Pengertian

Java adalah sebuah bahasa pemrograman yang diciptakan pada tahun 1995 oleh James Gosling dari Sun Microsystems (sekarang Oracle Corporation). Bahasa pemrograman ini diciptakan dengan tujuan untuk membantu dalam pengembangan perangkat lunak yang bersifat portabel, aman, dan juga dapat diandalkan. Java menggunakan konsep pemrograman berorientasi objek dan platform-independent, yang memungkinkannya untuk dapat digunakan dalam berbagai jenis perangkat seperti komputer, telepon genggam, dan perangkat elektronik lainnya.

### Sejarah

[Bahasa pemrograman](https://id.wikipedia.org/wiki/Bahasa_pemrograman) Java terlahir dari [The Green Project](https://id.wikipedia.org/w/index.php?title=The_Green_Project&action=edit&redlink=1" \o "The Green Project (halaman belum tersedia)), yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan [Oak](https://id.wikipedia.org/wiki/Oak" \o "Oak). Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, dan [James Gosling](https://id.wikipedia.org/wiki/James_Gosling" \o "James Gosling), beserta sembilan pemrogram lainnya dari [Sun Microsystems](https://id.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems). Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh [Joe Palrang](https://id.wikipedia.org/w/index.php?title=Joe_Palrang&action=edit&redlink=1" \o "Joe Palrang (halaman belum tersedia)).

Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road* di [Menlo Park](https://id.wikipedia.org/wiki/Menlo_Park,_California). Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program *Java Oak* pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "\*7" (*Star Seven*).

Setelah era *Star Seven* selesai, sebuah anak perusahaan [TV kabel](https://id.wikipedia.org/wiki/TV_kabel" \o "TV kabel) tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, [Palo Alto](https://id.wikipedia.org/wiki/Palo_Alto" \o "Palo Alto).

Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian [Internet](https://id.wikipedia.org/wiki/Internet" \o "Internet) sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan [akademisi](https://id.wikipedia.org/wiki/Akademisi" \o "Akademisi) dan [militer](https://id.wikipedia.org/wiki/Militer).

Mereka menjadikan [peramban](https://id.wikipedia.org/wiki/Peramban" \o "Peramban) (*browser*) [Mosaic](https://id.wikipedia.org/w/index.php?title=Mosaic&action=edit&redlink=1) sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinsipirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995.

Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel [Sheraton Palace](https://id.wikipedia.org/w/index.php?title=Sheraton_Palace&action=edit&redlink=1" \o "Sheraton Palace (halaman belum tersedia)). Tiga dari pimpinan utama proyek, [Eric Schmidt](https://id.wikipedia.org/wiki/Eric_Schmidt" \o "Eric Schmidt) dan [George Paolini](https://id.wikipedia.org/w/index.php?title=George_Paolini&action=edit&redlink=1) dari [Sun Microsystems](https://id.wikipedia.org/wiki/Sun_Microsystems) bersama [Marc Andreessen](https://id.wikipedia.org/w/index.php?title=Marc_Andreessen&action=edit&redlink=1), membentuk [Netscape](https://id.wikipedia.org/wiki/Netscape" \o "Netscape).

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", [James Gosling](https://id.wikipedia.org/wiki/James_Gosling" \o "James Gosling). Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau [Jawa](https://id.wikipedia.org/wiki/Jawa" \o "Jawa). Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java).

### Keunggulan

1. Membuat aplikasi yang fleksibel
2. Memiliki library yang lengkap
3. Berorientasi pada objek
4. Mirip dengan Bahasa c++
5. Menulis coding lebih sederhana
6. Mendukung native method

## XML



Gambar 2. 12 Logo XML

### Pengertian

Extensible Markup Language (XML) adalah bahasa komputer yang dibuat oleh World Wide Web Consortium (W3C) untuk menyederhanakan proses pertukaran dan penyimpanan data. Hal ini disebabkan keunikan dan perbedaan sistem yang digunakan oleh masing-masing server yang terhubung ke internet. Maka dari itu, diperlukan adanya standardisasi proses transfer data antar server.

XML akan menyimpan data dalam format teks yang sederhana. Jadi, data tersebut akan dapat dimengerti oleh server yang menerima data tanpa perlu perubahan atau modifikasi apapun. Hal ini tentunya sangat memudahkan dalam penyimpanan dan pengiriman data antar server.

### Sejarah

XML memiliki kepanjangan yaitu eXtensible Markup, XML yang biasanya digunakan untuk menyimpan data atau informasi. Karena XML didesain untuk bisa menyimpan data secara ringkas dan mudah diatur.

XML diperkenalkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan oleh XML sebenarnya bukan teknologi baru, namun turunan dari SGML yang telah dikembangkan di awal tahuan 1980-an dan telah banyak digunakan pada dokumentasi teknis proyek dengan berskala besar. Pada saat HTML dikembangkan tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML akhrinya bisa menghasilkan markup language yang tidak kalah hebatnya dengan SGML.

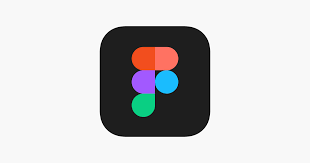
### Keunggulan

1. Pintar. XML dapat menangani berbagai tingkat (level) kompleksitas.
2. Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri.
3. Seperti Microsoft membuaat bahasa MSXML atau Macromedia mengembangkan MXML.
4. Mudah dalam pemeliharaannya.
5. Mudah dipindah-pindahkan (Portability). XML mempunyai kemudahan perpindahan pada yang lebih bagus.

### Kegunaan

1. XML digunakan untuk membawa data pada API seperti SOAP.
2. XML digunakan untuk membentuk struktur form pada pemrograman desktop seperti delphi, lazarus, vb, gtk, dan sebagainya.
3. XML digunakan untuk membuat layout aplikasi android.
4. XML bertugas untuk membentuk struktur proyek pada java yang menggunakan Maven.
5. XML digunakan sebagai format file SVG, DOCX, dan sebagainya.

## Figma



Gambar 2. 13 Logo Figma

### Pengertian

Figma adalah editor grafis vektor dan alat prototyping dengan berbasis web serta fitur offline tambahan yang diaktifkan oleh aplikasi desktop untuk Mac OS dan Windows. Aplikasi pendamping Figma Mirror untuk Android dan iOS memungkinkan untuk melihat prototype Figma pada perangkat seluler. Rangkaian fitur Figma berfokus pada penggunaan dalam antarmuka pengguna dan desain pengalaman pengguna dengan penekanan pada kolaborasi waktu nyata (*real- time*).

### Sejarah

Dylan Field dan Evan Wallace mulai mengerjakan Figma pada tahun 2012 saat belajar ilmu komputer di Universitas Brown. Wallace mempelajari grafik dan pernah menjadi seorang asisten dosen di Departemen Ilmu Komputer, sementara Field mengepalai CS Departmental Undergraduate Group. Tujuan awal di balik Figma adalah untuk memungkinkan siapa pun menjadi kreatif dengan menciptakan peralatan kreatif yang sederhana dan gratis pada browser. Field dan Wallace bereksperimen dengan berbagai ide, termasuk perangkat lunak untuk drone dan sebuah pembuat meme, sebelum mantap dengan perangkat lunak editor grafik.

Pada bulan Juni 2022, Google for Education milik Google mengumumkan bahwa mereka akan bekerja sama dengan Figma untuk menyertakan platform desain dan prototyping, dan juga FigJam milik Figma, ke dalam Chromebook edukasi.

### Keunggulan

1. Bekerja bersamaan dan berkolaborasi secara realtime
2. File sharing yang mudah dan cepat
3. Aplikasi desain yang lengkap dan serba bisa
4. Berbasis cloud tanpa perlu instalasi

### Fitur-fitur Figma

1. *Modern pen tool*
2. *Instant arc designs*
3. *Open type*
4. *Less manual resizing*
5. *Stretch to fill*

## System Operasi Android

### Pengertian

Sistem operasi Android adalah salah satu sistem operasi *platform* terbuka untuk perangkat *mobile.* Sistem operasi ini dikembangkan oleh Google dan dikepalai oleh *Open Handset Alliance*. Sistem operasi ini menggunakan Linux sebagai inti sistem operasi.



Gambar 2. 14 Logo Android

Saat ini, penggunaan aplikasi *Mobile* berbasis Android banyak digunakan oleh sebagian besar pengguna ponsel pintar (*smartphone*). Hampir sebagian besar pengguna menggunakan aplikasi *mobile* berbasis Android dikarekan kemudahan dalam penggunaan aplikasi dan perantinya sendiri terjangkau oleh masyarakat luas.

Berikut ini adalah kelebihan-kelebihan dari sistem operasi Android:

1. Mendukung proses *multitasking*, yaitu memungkinkan pengguna Android dapat membuka banyak aplikasi dalam satu waktu.
2. Akses unduh aplikasi menjadi lebih mudah menggunakan Google Play Store.
3. Notifikasi yang mudah, setiap ada *Short Message Service* (SMS), *E-mail* ataupun artikel dari *Really Simple Syndication Reader* (RSS Reader) akan berlangsung muncul di *home screen* serta lampu *Light-Emitting Diode* (LED) yang berkedip-kedip.
4. Didukung berbagai macam ponsel, sekarang sudah banyak ponsel yang memakai sistem operasi ini mulai dari HTC, Samsung, Nokia, sampai ponsel Cina seperti Evercoss dan Mito.
5. Dapat mengakses seluruh produk dari Google, mulai dari Google Map, Google Reader, Google Translate, Youtube, Blogger, dan lain-lain.
6. Modifikasi *Read Only Memory* (ROM), jika kurang puas dengan tampilan standar Android, ada banyak sekali modifikasi ROM yang bisa digunakan di ponsel sistem Android.

Berikut ini adalah kekurangan-kekurangan dari sistem operasi Android:

1. Harus selalu terhubung ke internet untuk mengakses konten yang terdapat dalam ponsel Android.
2. Terdapat banyak iklan di setiap aplikasi dari Android.

### Sejarah Android

Android sendiri terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia.

Pada saat rilis perdana Android, 5 November 2007, Android Bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

### Perkembangan Seri Android

Saat ini, perkembangan sistem operasi Android berkembang secara cepat. Berikut ini macam-macam versi Android yang menyertai sejarah perkembangan sistem operasi Android sejak diperkenalkan ke publik pada tanggal 5 November 2007:

1. Android Beta

Android Beta merupakan versi pertama sistem operasi Android yang dirilis oleh Google; tepatnya pada tanggal 5 November 2007. Ini merupakan versi resmi yang diperkenalkan ke publik setelah Google mengakuisisi Android, Inc pada tahun 2005.

1. Android versi 1.0 Astro

Sistem operasi Android resmi yang digunakan pada perangkat telepon seluler berbasis Android pertama adalah Android 1.0 Astro. Android 1.0 Astro pertama kali dipasang pada telepon seluler HTC Dream; sistem operasi ini pertama kali diperkenalkan pada 23 September 2008. Nama Astro kemudian dihilangkan karena masalah hak cipta.

1. Android 1.1 Bender

Android 1.1 Bender merupakan versi *update* yang dirilis pada 9 Februari 2009; nama Bender kemudian juga dihilangkan karena masalah hak cipta. Ponsel T-Mobile G1 merupakan perangkat yang menggunakan sistem operasi ini.

1. Android 1.5 Cupcake

Android 1.5 Cupcake merupakan versi Android selanjutnya dari sistem operasi Android; ini merupakan awal dari era penamaan sistem operasi Android dengan menggunakan nama *dessert* atau makanan pencuci mulut.

Sebagai versi resmi yang ketiga; dipilih nama *Cupcake* yang menggunakan abjad awalan C. Sistem operasi Android ini dibuat berdasarkan *kernel* Linux 2.6.27 dan pertama kali dirilis pada 30 April 2009. Terdapat beberapa fitur baru termasuk *widget* yang dapat diubah ukurannya serta kemampuan mengunggah video dan gambar ke Youtube dan Picasa.

1. Android 1.6 Donut

Android 1.6 Donut dirilis pada 15 September 2009; fitur pencarian ditambahkan pada *update* sistem operasi ini serta desain *interface* yang lebih mudah dipahami. Versi Android ini juga merupakan sistem operasi Android pertama yang dapat dikombinasikan dengan teknologi CDMA/EVDO, 802.1x, VPNs dan juga layar WVGA.

1. Android 2.0/2.1 Éclair

Android 2.0/2.1 Eclair pertama kali dirilis pada 9 Desember 2009; terdapat peningkatan pada optimasi perangkat keras serta adanya Google Maps 3.1.2 serta penambahan browser dengan basis HTML5. Fitur lampu kilat untuk kamera hingga 3,2 MP serta *digital zoom* dan juga koneksi Bluetooth 2.1 telah ditambahkan pada versi ini.

1. Android 2.2 Froyo

Android 2.2 Froyo atau *Frozen Yoghurt* dirilis pada tanggal 20 Mei 2012 dan terpasang pada *smartphone* Google Nexus One. Versi ini dapat menjalankan aplikasi Adobe Flash Player 10.1 serta menyertakan kemampuan menggunakan kartu memori SD untuk menyimpan aplikasi. Versi ini sempat mengalami dua kali *update* dengan dirilisnya Android versi 2.2.1 hingga versi 2.2.3.

1. Android 2.3 Gingerbread

Android 2.3 Gingerbread pertama kali dirilis pada 6 Desember 2010; perkembangan pesat terjadi pada *update* kali ini dengan optimasi kemampuan aplikasi serta *game* serta disertakannya *Near Field Communication* (NFC). Dukungan untuk penggunaan layar WXVGA serta perkembangan hingga dirilisnya versi 2.3.7 membuat era Gingerbread menjadi era perkembangan pesat Android yang disertai dengan jumlah pengguna terbanyak dibandingkan versi - versi Android sebelumnya.

1. Android 3.0 Honeycomb

Sistem operasi Android 3.0 Honeycomb dan Android 3.1 Honeycomb dirilis untuk pertama kali pada 22 Februari 2011; sistem operasi Android ini pertama kali dipasang pada Motorola Xoom. Tidak seperti sistem operasi Android yang sebelumnya; Versi Android 3.x Honeycomb dikembangkan secara khusus untuk melakukan optimasi fungsi dan aplikasi pada PC *tablet*.

1. Android 4.0 Ice Cream Sandwich

Sistem operasi Android 4.0 Ice Cream Sandwich merupakan generasi Android selanjutnya; sistem operasi Android ini pertama kali diperkenalkan pada 19 Oktober 2011 dan Samsung Galaxy Nexus menjadi *smartphone* pertama yang menggunakan sistem operasi ini.

1. Android 4.1 Jelly Bean

Sistem operasi Android 4.1 Jelly Bean merupakan lanjutan dari pengembangan sistem operasi Android sebelumnya; sistem operasi ini memiliki beberapa keunggulan yang tidak dimiliki oleh sistem operasi versi sebelumnya serta menawarkan beberapa fitur baru. Pembaruan pada sistem *input keyboard*, tampilan baru pada fitur pencarian dan *Voice Search* serta tampilan *interface* yang lebih mudah dipahami dan digunakan.

Penambahan fitur Google Now menjadi sebuah lompatan besar pada sistem operasi Android 4.1 Jelly Bean ini; Google Now dapat memberikan berbagai macam informasi terkini seperti kondisi cuaca, kepadatan lalu lintas hingga informasi populer seperti hasil pertandingan bola basket dan sepak bola. Sistem operasi ini pertama kali digunakan pada tablet buatan Asus yaitu Google Nexus7.

1. Android 4.2 Jelly Bean

Android 4.2 Jelly Bean merupakan versi *update* dari versi Android 4.1. Terdapat beberapa perbaikan dan penambahan fitur seperti adanya fitur *photosphere* untuk menangkap gambar *panorama* serta *screensaver* *Daydream*. Beberapa fitur fungsional seperti *power control*, mengunci layar *widget* serta menjalankan beberapa *user* pada PC *tablet* menjadi beberapa penambahan pada *update* sistem operasi Android ini. Android 4.2 Jelly Bean pertama kali dipasang pada LG Google Nexus 4.

Versi Android ini pertama kali dirilis pada 24 Juli 2013 dan pertama kali dipasang pada tablet Nexus 7 versi update dengan perbaikan pada beberapa *bug* kemudian dirilis pada 22 Agustus.

1. Android 4.4 KitKat

Android 4.4 KitKat pertama kali dirilis pada 3 September 2013; sebelumnya sistem operasi Android ini diberi nama Key Lime Pie dan kemudian diubah karena nama tersebut kurang populer. Sistem operasi ini sempat dikabarkan menjadi Android versi 5.0 namun ternyata hanya menggunakan nama baru KitKat; sistem operasi Android ini pertama kali digunakan pada Nexus 5.

1. Android 5.0 Lollipop

Sistem operasi Android 5.0 Lollipop pertama kali dirilis pada 25 Juni 2014; sebelumnya sistem operasi ini diperkenalkan dengan kode Android L. *Update* penggunaan sistem operasi Android dari versi sebelumnya ke versi 5.0 pertama kali dibuka pada 12 November 2014.

*Interface* baru digunakan serta beberapa perbaikan pada akses notifikasi yang lebih mudah. Perubahan *platform* dengan Android *Runtime* menggantikan *Dalvik* dilakukan guna meningkatkan performa aplikasi dan juga mengoptimalkan konsumsi energi dari baterai.

1. Android 6.0 Marshmallow

Pertama kali dikenalkan pada Mei 2015 dengan *code name* Android M yang kemudian dirilis secara resmi pada Oktober 2015. Marshmallow merupakan pengembangan dari Android Lollipop salah satu kelebihannya mampu menjaga konsumsi baterai yang lebih hemat.

1. Android 7.0 Nougat

Beragam peningkatan fitur dan kemampuan menghiasi kehadiran Android N yang menjadi penerus Versi Android Marshmallow. Salah satunya adalah *Multi-Window* yang memungkinkan pengguna *handphone* Android membuka dua aplikasi sekaligus secara *split screen*. Nama Nougat diambil dari makanan penutup khas dari daratan Eropa yang berupa campuran telur kocok, gula/madu, dan kacang (almond, pistachio, atau hazelnut), biasanya berbentuk batangan seperti coklat kacang kemasan.

Android 7.0 Nougat sendiri mulai dikembangkan pada Maret 2016 dan Rilis Akhir tahun 2016. Peningkatan kemampuan untuk menghemat baterai menjadi salah satu fitur yang paling dibutuhkan sebagian besar pengguna *handphone* Android. Fitur *Doze Mode* dapat mengatur penggunaan baterai dengan cara membatasi aktivitas aplikasi yang bekerja di belakang layar ketika layar mulai padam.

1. Android 8.0 Oreo

Android Oreo hadir dengan keunggulan yang *smarter*, *faster*, dan lebih *Powerfull* dari versi sebelum-sebelumnya. Fitur baru yang diunggulkan adalah *picture-in-picture* dan *autofill* yang dapat membantu pengguna *smartphone* Android menavigasi tugas dengan mulus dan ringan. Selain itu Android 8.0 Oreo telah hadir fitur *shortcut* yang akan mempercepat penggunanya dalam melakukan aktivitas.

1. Android 9.0 Pie

Android versi 9.0 dikembangkan untuk ponsel pintar yang memiliki desain Layar minim bingkai atau yang lebih di kenal smartphone desain bezelless. Dengan kata lain tampilan UI yang ada Android ini lebih menyatu dengan pada ponsel bezelless, baik dari segi icon, dan lain-lain. selain beberapa fitur di atas, tentu saja performa, privasi serta optimasi penghematan baterai juga terjadi pada versi OS ini, karena ini hal biasa di lakukan oleh developer OS Android, mereka terus berusaha membuat OS ini hemat baterai dan memiliki keamanan tingkat tinggi namun tetap memiliki desain UI yang sempurna.

1. Android 10 Q

Android 10 dirilis pertama kali pada 3 September 2019. Sebulan sebelum dirilis, Google sudah mengumumkan bahwa versi Android terbaru di 2019 ini tidak lagi menggunakan nama makanan manis atau pencuci mulut seperti sebelumya. Namun secara internal, Android 10 diberi julukan nama Android Quince Tart.

1. Android 11 Red Velvet Cake

Versi Android dengan nama makanan (versi internal) Red Velvet Cake ini dirilis resmi pada 8 September 2020. OS Android 11 bisa mengelompokkan aplikasi dalam status notifikasi. Misal semua aplikasi pesan digabung dalam notifikasi Conversation.

1. Android 12 Snow Cone

Android 12 akhirnya kembali menggunakan nama makanan manis, yaitu Snow Cone. Android versi 12 ini dirilis perdana pada 4 Oktober 2021. Urutan Android versi terbaru ini memiliki update besar pada Desain Material yang kemudian disebut sebagai “Material You.”. Sistem operasi bisa secara otomatis menghasilkan tema warna untuk menu sistem dan aplikasi yang didukung menggunakan warna wallpaper pengguna. Android 12 juga dilengkapi fitur untuk ambil screenshot tampilan web secara utuh atau bergulir. Pengguna juga bisa mencegah aplikasi untuk menggunakan aplikasi dan mikrofon melalui pengaturan cepat.

1. Android 13 Tiramisu

OS Android 13 pertama kali diperkenalkan pada 10 Februari 2022. Urutan Android terbaru ini dirilis sekitar 4 bulan setelah versi stabil Android 12 dirilis. Android 13 fokus pada penyempurnaan fitur pada Android 12L, meningkatkan privasi, keamanan, dan optimalisasi UI.

Android 13 punya fitur pemilih foto baru. Fitur ini memungkinkan Anda bisa atur foto atau video tertentu yang bisa dibagikan dengan aplikasi. Berbeda seperti versi sebelumnya yang memberi akses ke semua foto dan video di library. Di Android 13 pengguna juga bisa kustomisasi ikon aplikasi dengan warna yang sama seperti tema atau wallpaper.

## ReactJS

### Pengertian

[ReactJS](https://reactjs.org/) atau React adalah library [JavaScript](https://www.hostinger.co.id/tutorial/apa-itu-javascript) populer buatan Facebook yang digunakan dalam proses pengembangan aplikasi mobile dan web. React berisi kumpulan snippet kode JavaScript (disebut ‘komponen’) yang bisa digunakan berulang kali untuk mendesain antarmuka pengguna.

Perlu diketahui bahwa ReactJS bukanlah framework JavaScript, karena hanya bertugas untuk untuk merender komponen area tampilan aplikasi. React adalah alternatif untuk framework seperti [Angular](https://www.hostinger.co.id/tutorial/apa-itu-angularjs) dan [Vue](https://vuejs.org/), yang sama-sama bisa digunakan untuk membuat fungsi yang kompleks.

### Fitur React

React memiliki beberapa fitur utama yang menjadikannya lebih unggul dibandingkan dengan library JavaScript lainnya.

1. JSX

JSX adalah ekstensi syntax JavaScript yang digunakan dalam pembuatan elemen React. Developer menggunakannya untuk menyematkan (embed) kode [HTML](https://www.hostinger.com/tutorials/html-cheat-sheet" \t "_blank) pada objek JavaScript. Karena bisa memproses embedding ekspresi dan fungsi JavaScript yang valid, JSX pun membantu mempersingkat struktur kode yang kompleks.

1. DOM Virtual

Document Object Model (**DOM**) berfungsi untuk menyajikan halaman web dalam tampilan struktur data (tree/pohon). ReactJS menyimpan struktur data DOM Virtual ini dalam memorinya, sehingga kalau ada perubahan pada bagian tertentu dalam struktur data tersebut, Anda tidak perlu merender ulang semuanya.

Setiap kali ada perubahan data, ReactJS akan membuat satu struktur data DOM Virtual baru dan membandingkannya dengan yang sebelumnya, lalu mencari cara tercepat untuk menerapkan perubahan tersebut pada DOM yang asli. Proses ini disebut [diffing](https://dev.to/koolkishan/what-is-virtual-dom-how-virtual-dom-works-what-is-reconciliation-what-is-diffing-algorithm-what-makes-react-so-fast-327a" \t "_blank).

Jadi, perubahan UI hanya akan memengaruhi bagian tertentu dalam struktur data DOM yang asli, sehingga proses render perubahan pun bisa lebih cepat dan lebih hemat resource.

1. Komponen dan Properti React

ReactJS memisahkan antarmuka pengguna menjadi beberapa potong kode tersendiri yang nantinya bisa digunakan lagi, yang disebut **komponen**. Cara kerja komponen React mirip dengan fungsi JavaScript, yaitu sama-sama menerima input arbitrer yang disebut properti atau **props**. Elemen React yang dihasilkan (return) akan menentukan bagaimana tampilan antarmuka pengguna yang dilihat oleh user (sisi klien).

1. Manajemen State

**State** (keadaan) adalah objek JavaScript yang mewakili satu bagian dari sebuah komponen. Setiap kali user berinteraksi dengan aplikasi, state juga akan berubah, dengan merender UI baru guna menampilkan perubahan dari interaksi tersebut. Fitur manajemen state ini mengacu pada prosedur yang dilakukan untuk mengelola keadaan sebuah aplikasi React. Contohnya seperti menyimpan data di library manajemen state pihak ketiga, dan memicu proses rendering ulang setiap kali data diubah. Library manajemen state berfungsi untuk mempermudah komunikasi dan proses berbagi data di antara komponen React. Ada beberapa library manajemen state pihak ketiga yang tersedia saat ini, yang paling populer adalah **Redux** dan **Recoil**.

1. **Navigasi Terprogram**

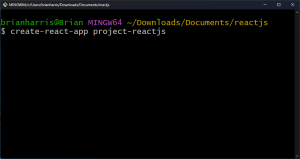
Navigasi terprogram adalah peristiwa di mana rangkaian kode menghasilkan sebuah tindakan yang mengarahkan user ke area tertentu. Contohnya adalah proses login dan mendaftar akun, yang secara terprogram akan mengarahkan user ke halaman baru. [React Router](https://reactrouter.com/" \t "_blank), library standar React untuk perutean, menyediakan berbagai metode navigasi terprogram yang aman antarkomponen tanpa mengharuskan user mengklik link apa pun. Metode utama yang digunakan dalam navigasi terprogram biasanya adalah komponen **Redirect**, atau **history.push()** sebagai opsi lainnya. Singkatnya, package React Router menyinkronkan antarmuka pengguna dengan URL sehingga Anda bisa mengontrol tampilan aplikasi React tanpa bergantung pada link.

### Kelebihan

1. Mudah digunakan
2. Mendukung komponen java yang reusable
3. Mudah untuk menulis komponen
4. Performa tinggi
5. SEO-Friendly

### Cara Menginstall Reactjs

1. Pastikan anda telah menginstall nodejs terlebih dahulu pada komputer anda dengan mengunduh installer yang telah disediakan untuk Windows. Buka halaman [download Node.js](https://nodejs.org/en/download/" \t "_blank) dan klik tautan unduhan untuk. Jalankan installer dan ikuti panduannya seperti menginstal aplikasi lainnya pada komputer anda.
2. Buka command prompt atau terminal pada komputer Anda
3. Masukkan perintah “npm install -g create-react-app” dan tunggu hingga selesai
4. Setelah selesai, buatlah direktori baru untuk proyek React JS Anda
5. Masuk ke direktori tersebut dengan perintah “cd nama-direktori”
6. Jalankan perintah “create-react-app nama-project” untuk membuat proyek React pertama Anda



Gambar 2. 15 *Create Project on React*

1. Tunggu hingga selesai hingga anda melihat pesan sukses terinstall.

## Laravel

### Pengertian

Laravel adalah *framework*berbasis bahasa pemrograman [PHP](https://www.dewaweb.com/blog/apa-itu-php/) yang bisa digunakan untuk membantu proses pengembangan sebuah website agar lebih maksimal. Dengan menggunakan Laravel, website yang dihasilkan akan lebih dinamis.

Kehadiran *framework*Laravel menjadikan [bahasa pemrograman](https://www.dewaweb.com/blog/belajar-coding-untuk-pemula/" \o "bahasa pemrograman) PHP menjadi lebih powerful*.*Perlu kita ketahui bahwa kehadiran *framework*Laravel selalu menghadirkan fitur-fitur terbaru dibandingkan *framework*lainnya.

*Framework*Laravel menggunakan struktur MVC (*Model View Controller)*. MVC merupakan model aplikasi yang memisahkan antara data dan tampilan berdasarkan komponen aplikasi. Dengan adanya model MVC, pengguna Laravel menjadi lebih mudah dalam mempelajari Laravel. Serta menjadikan proses pembuatan aplikasi berbasis website menjadi lebih cepat.

### Fungsi Laravel

Sebagai *framework*yang memiliki fitur yang lengkap, Laravel berfungsi untuk membantu proses pembuatan website menjadi lebih maksimal. Selain itu, dengan menggunakan Laravel, perusahaan juga akan menghemat waktu dan biaya. Hal ini karena nantinya hanya fitur yang sedang dibutuhkanlah yang akan dibuat. Sedangkan fitur yang belum dibutuhkan akan dibuat pada saat masa pengembangan website tersebut.

### Kelebihan Laravel

1. Template ringan
2. Library yang lengkap
3. Menggunakan model MVC
4. Tool artisan
5. Modul bersifat individu dan indepeden

### Fitur-fitur Laravel

1. Authentication

Fitur ini merupakan fitur yang dapat kamu gunakan untuk membuat website dengan otentikasi. Kamu tidak perlu menulis kode dari awal untuk menambahkan fitur otentikasi di website kamu.

1. Testing and Debugging

Dalam Laravel sudah disediakan fitur untuk testing dan debugging sebuah website. Kamu tidak perlu khawatir lagi untuk melakukan testing dan debugging.

1. Routing

Dengan fitur yang satu ini, kamu dapat membuat aplikasi yang dengan lebih mudah. Hal ini karena dalam framework ini, semua request akan dipetakan dengan menggunakan bantuan rute. Kamu bisa memberikan nama routing, mengelompokan, memfilter routing yang ada sesuai kebutuhan kamu.

1. Blade Template Engine

Framework ini dilengkapi dengan fitur blade yang bisa digunakan untuk mendesain layout yang konsisten.

1. Composer

Fitur ini akan membantu dalam menyempurnakan dan meningkatkan proses pengembangan website, serta mempermudah proses update-nya.

## Tailwind CSS

### Pengertian

Tailwind CSS adalah kerangka kerja framework css yang didalamnya terdapat sekumpulan ultility classes untuk membangun antarmuka custom.CSS framework juga bersifat ultility-first yang bisa digunakan untuk membangun tampilan antarmuka sebuah website menjadi lebih cepat.

Tailwind menyediakan class-class kecil yang bisa digunakan dengan sangat fleksibel untuk membuat interface sesuai dengan keinginan tanpa perlu override styless seperti framework pada umumnya. Jadi, ketika anda ingin membuat komponen, maka anda perlu menggunakan beberapa class kecil hingga tercipta komponen yang anda inginkan.

### Sejarah Singkat

Seperti namanya Tailwind CSS atau sebuah framework css, yang dikembangkan oleh tim project Adam Wathan, Jonathan Reinink, David Hemphill dan Steve Schoger. Project ini dirilis pertamakali pada 22 Oktober 2019 silam.Framework ini diciptakan oleh Adam Wathan dengan desain visual komponen nya yang dibuat oleh Steve Schoger serta diluncurkan sekitar tahun 2020 lalu.

Walaupun masih terbilang baru namun untuk saat ini tailwind sendiri sudah menempu versi yang ke v.3.0.23, framework ini digambarkan seperti kerangka css yang mengutamakan utilitas dan mereka mengklaim pembuatan website dengan menggunakan framework ini dapat lebih cepat karena mereka tidak perlu lagi berpindah pindah file untuk mengatur style pada website mereka.

### Kelebihan

1. Lebih sederhana dan efisien
2. Responsive
3. Customizable

### Kekurangan Tailwind CSS

1. File TailwindCSS cukup besar dibanding yang lain
2. Belum adanya komponen untuk form
3. Nama class sangat Panjang

## PHPMyAdmin

### Pengertian

PHPMyAdmin merupakan sebuah aplikasi berbasis web yang berfungsi untuk mengelola database [MySQL](https://makinrajin.com/blog/mysql-adalah/) atau bisa disebut juga sebagai tool database. Siapapun sebenarnya tidak salah jika mempelajarinya, karena aplikasi ini akan sangat berguna dalam pengembangan situs web yang saat ini semakin populer, misalnya [WordPress](https://makinrajin.com/blog/wordpress-adalah/) yang memerlukan akses ke database.

Software berbasis web ini akan memudahkan Anda untuk melakukan manipulasi database MySQL tanpa harus mengetikkan perintah pada command line. Aplikasi ini memiliki tampilan yang mudah dipahami dengan fitur lengkap sesuai kebutuhan para pengguna.

### Fungsi PHPMyAdmin

PHPMyAdmin berfungsi untuk membuat, mengedit, menghapus database, tabel, serta membuat atau menghapus relasi antar tabel, mensortir data, dan lain-lain sesuai dengan kebutuhan Anda. Saat menggunakannya, Anda akan mendapatkan kemudahan dengan cara yang lebih efektif dalam pembuatan database menuju web server.

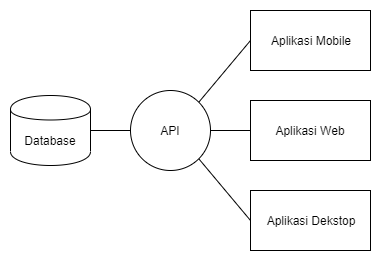
PHPMyAdmin adalah software yang mempunyai fasilitas import yang bisa Anda manfaatkan untuk membuat database dengan ekstensi .SQL. Lalu pada versi offline, Anda hanya cukup memindahkan ke versi web yang tersedia. Selain itu, saat membuat tabel-tabel pada [database](https://makinrajin.com/blog/database-adalah/), pertimbangkan juga tentang primary key dan foreign key serta relasi datanya.

### Kelebihan

1. Memiliki antarmuka atau interface berbasis web yang user friendly.
2. Terdiri dari berbagai fitur yang dapat Anda gunakan untuk menambah, menghapus, menyalin, maupun memperbaharui database, view, tabel, index, fields. Kemudian melakukan pemeliharaan database melalui menu konfigurasi, melakukan eksekusi terhadap query MySQL, serta mengelola akun pengguna MySQL.
3. Melakukan import data yang berformat MySQL atau [SVG](https://makinrajin.com/blog/svg-adalah/).
4. Ekspor data ke berbagai format lain seperti [XML](https://makinrajin.com/blog/xml-adalah/), CSV, PDF, dan lain sebagainya.
5. Mengelola banyak server.
6. Dapat membuat perintah query dengan fitur QBE (Query by Example)
7. Mampu melakukan pencarian global pada database.
8. Mengubah data dalam format apapun.

## API

### Pengertian



Gambar 2. 16 Cara Kerja API

API adalah singkatan dari Application Programming Interface. API sendiri merupakan interface yang dapat menghubungkan satu aplikasi dengan aplikasi lainnya. Dengan kata lain, peran API adalah sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau pun lintas platform.

Perumpamaan yang bisa digunakan untuk menjelaskan API adalah seorang pelayan di restoran. Tugas pelayan tersebut adalah menghubungkan tamu restoran dengan juru masak.

Jadi, tamu cukup memesan makanan sesuai daftar menu yang ada dan pelayan memberitahukannya ke juru masak. Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan.

### Jenis

API memiliki beberapa jenis, yaitu:

1. *Public* API

*Public* API sering disebut dengan *Open* API. Sesuai dengan namanya, *Public* API adalah jenis API yang bisa digunakan siapa saja dalam lintas platform.

1. *Private* API

Berbeda dengan public API, Private API adalah jenis API yang tidak terbuka untuk penggunaan umum. Biasanya, API jenis ini dibuat untuk keperluan internal dalam pengembangan aplikasi tertentu.

1. *Partner* API

*Partner* API bisa digunakan untuk kepentingan umum tapi sebatas pihak yang sudah memiliki izin penggunaannya. Seperti halnya public API, hanya perlu mendaftar kepada penyedia API terlebih dahulu. Kemudian, menggunakannya hanya di aplikasi tertentu sesuai perjanjian.

1. *Composite* API

*Composite* API merupakan API yang menyimpan data dari berbagai server atau *hosting* dalam satu tempat. Tentunya, hal tersebut sangat menghemat waktu bagi pengguna. Itu karena pengguna bisa mendapatkan berbagai jenis data hanya dalam sekali akses.

## Postman



Gambar 2. 17 Logo Postman

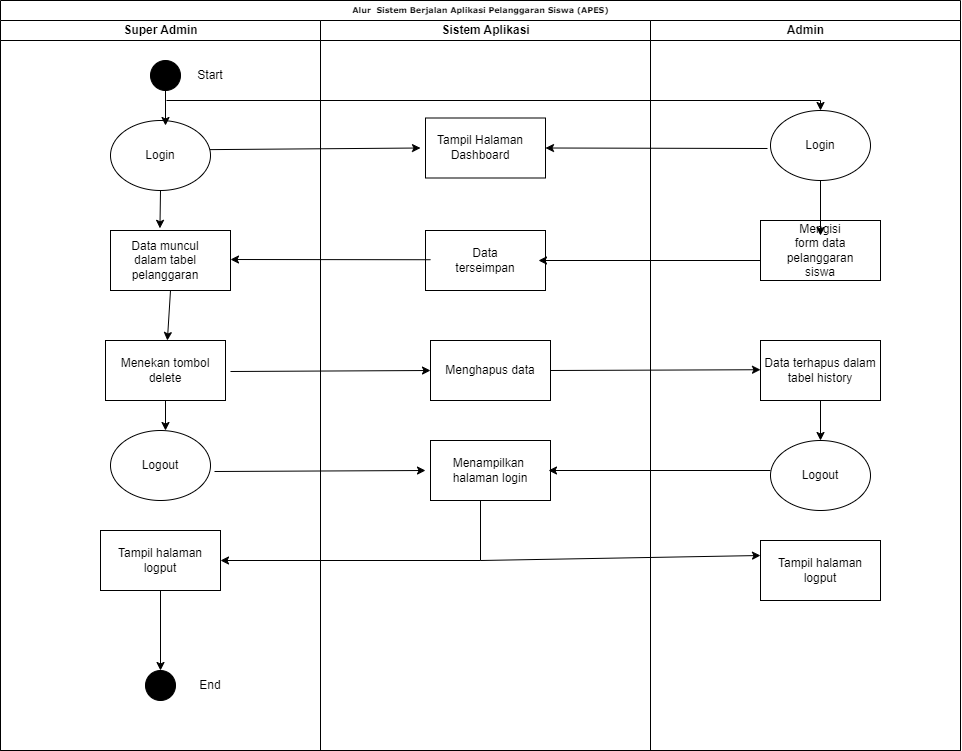
Postman merupakan aplikasi yang berfungsi untuk melakukan pengujian REST API yang dipasang pada *Web Browser* Chrome. Namun seiringnya perkembangan teknologi, kini Postman tersedia pada aplikasi *desktop*. Postman membuat pengembangan API lebih cepat, lebih mudah, dan lebih baik. Postman dirancang dengan mempertimbangkan pengembang, dan dikemas dengan fitur dan opsi.

# PERENCANAAN

## Alur Sistem Berjalan

### Alur Sistem Berjalan Aplikasi

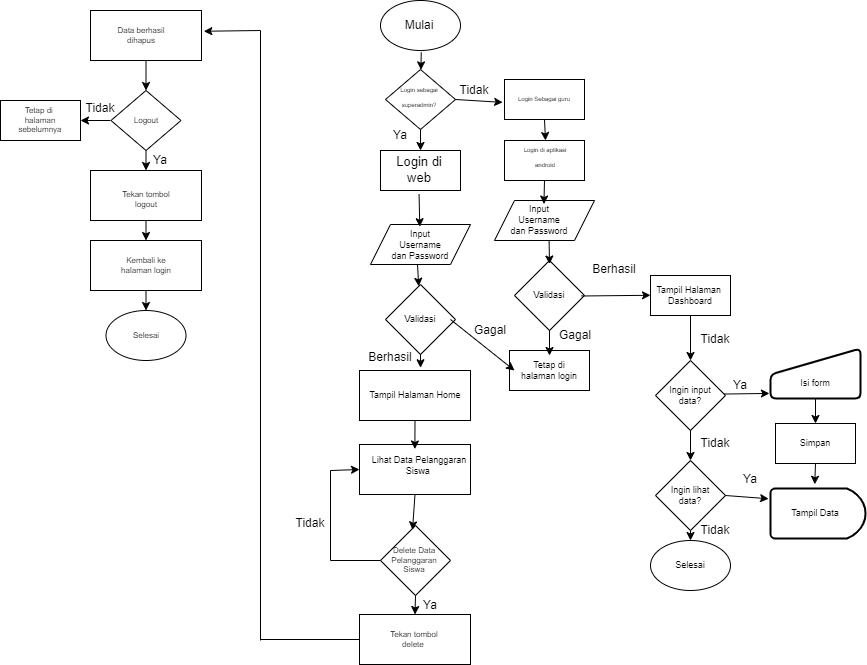
**Alur sistem** merujuk pada urutan langkah-langkah atau tahapan yang harus diikuti dalam menjalankan sebuah sistem atau aplikasi. Ini melibatkan pergerakan data, informasi, atau instruksi dari satu komponen ke komponen lain dalam rangkaian sistem. Berikut merupakan alur sistem berjalan dari aplikasi kami



Gambar 3. 1 Alur Sistem Berjalan Aplikasi

## Alur Sistem Aplikasi

Alur sistem aplikasi atau yang biasa disebut *flowmap* merupakan campuran peta dan *flow chart,*yang menunjukan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket  dalam jaringan. *Flowmap* menolong analisis dan *programmer* untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternaitf-alternatif lain dalam pengoprasian. Berikut merupakan alur sistem atau *flowmap* dari *aplikasi pelanggaran siswa*

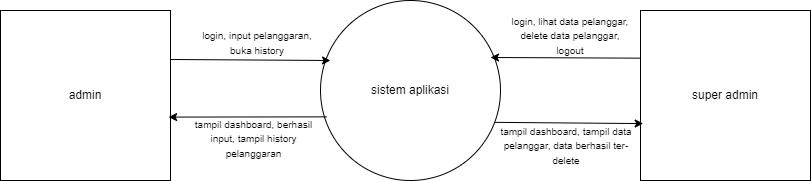


Gambar 3. 2 Alur Sistem Aplikasi

## Data Flow Diagram (DFD)

DFD adalah suatu diagram yang menggambarkan aliran data dari sebuah proses yang sering disebut dengan sistem informasi. Di dalam *data flow diagram*juga menyediakan informasi mengenai *input*dan *output* dari tiap entitas dan proses itu sendiri.

Dalam diagram alir data juga tidak mempunyai kontrol terhadap *flow*-nya, sehingga tidak adanya aturan terkait keputusan atau pengulangan. Bentuk penggambaran berupa data *[flowchart](https://www.sekawanmedia.co.id/blog/pengertian-dan-simbol-flowchart/)*dengan skema yang lebih spesifik. Menurut Kenneth Kozar, tujuan dari adanya DFD sendiri adalah sebagai penyedia atau menjembatani antara pengguna dengan sistem. Berikut merupakan DFD dari *aplikasi pelanggaran siswa*

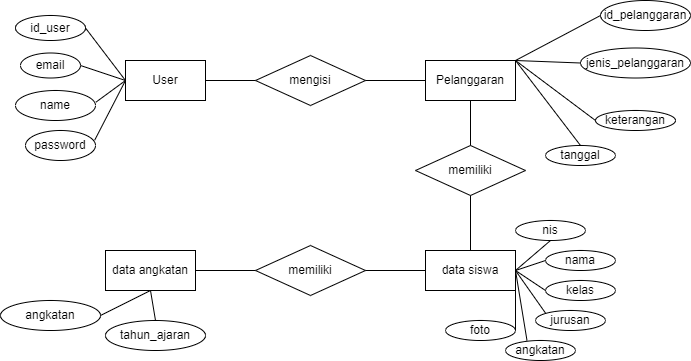


Gambar 3. 3 *Data Flow Diagram*

## Entity Relation Diagram

ERD atau entity relationship diagram adalah istilah dari bahasa Inggris yang artinya diagram hubungan entitas. ERD pun sering disebut dengan model ER atau ER Diagram.

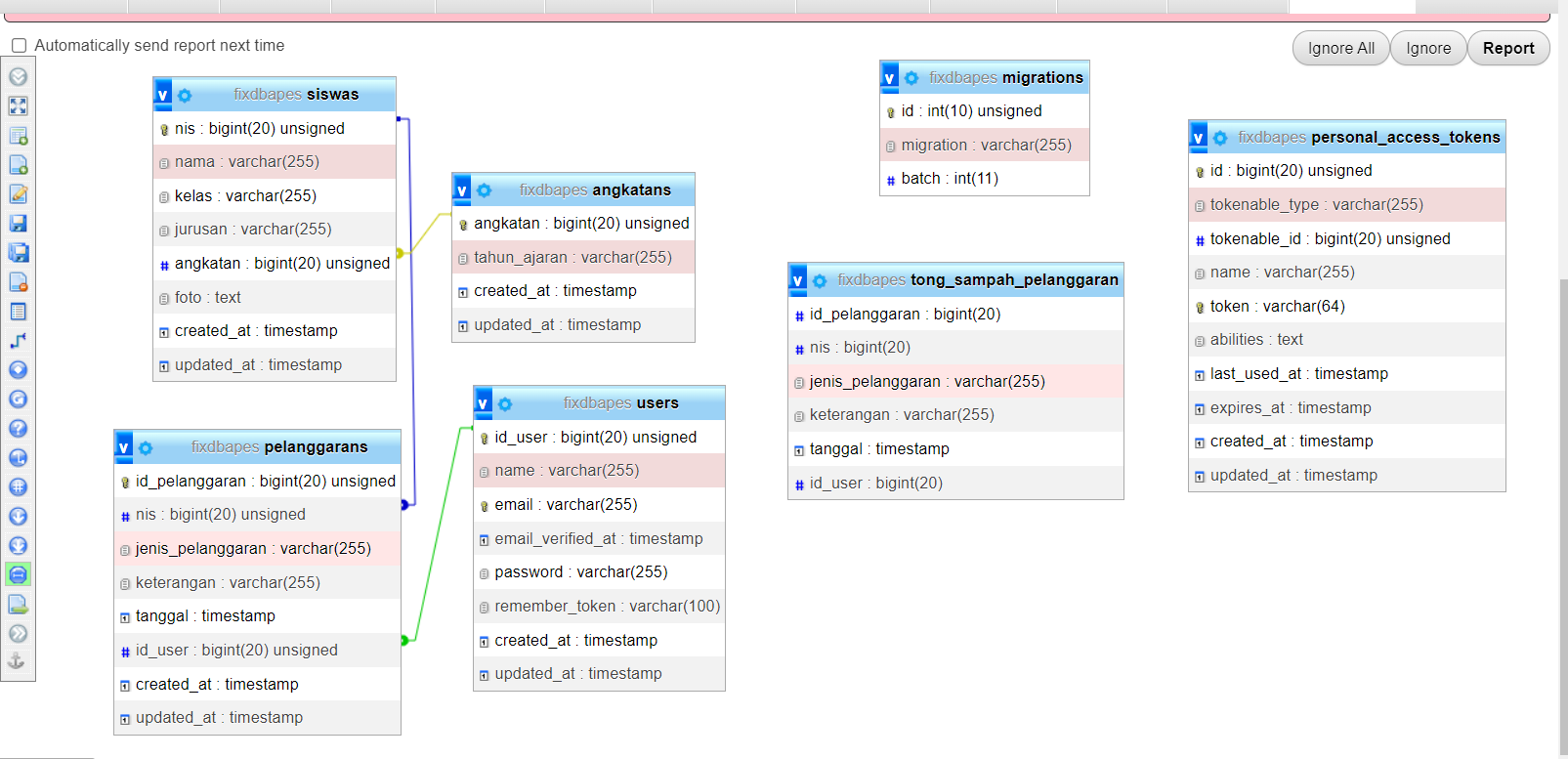
Sederhananya, entity relationship diagram adalah salah satu jenis diagram yang sifatnya lebih struktural dan bisa digunakan untuk dimanfaatkan dalam suatu desain pada suatu database ataupun pada sebuah business plan. Berikut merupakan ERD dari aplikasi pelanggaran siswa



Gambar 3. 4 *Entity Relation Diagram*

## Struktur Tabel

Berikut merupakan struktur table dari *aplikasi pelanggaran siswa*



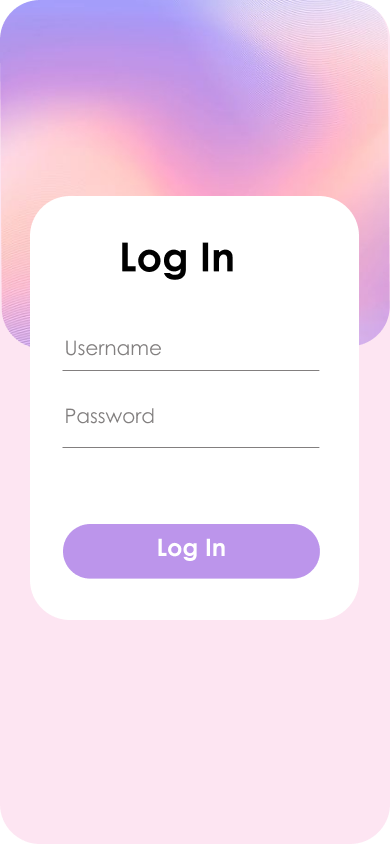
Gambar 3. 5 Struktur Tabel

## Mockup

1. Mockup aplikasi berbasis mobile



Gambar 3. 6 *Splash Screen Mobile*



Gambar 3. 7 Halaman Login *Mobile*



Gambar 3. 8 *Main Page Mobile*

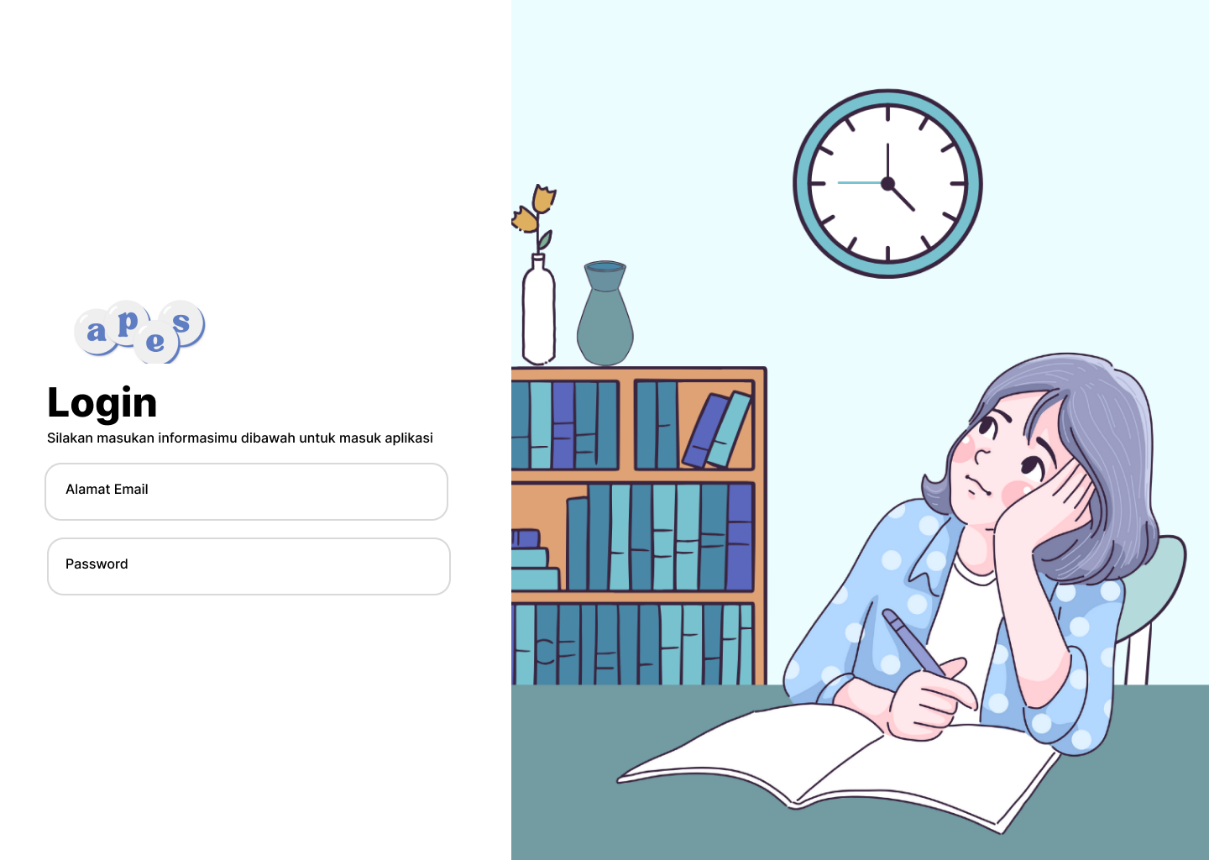


Gambar 3. 9 *Input Data Mobile*



Gambar 3. 10 *History Mobile*

1. Mockup aplikasi berbasis web



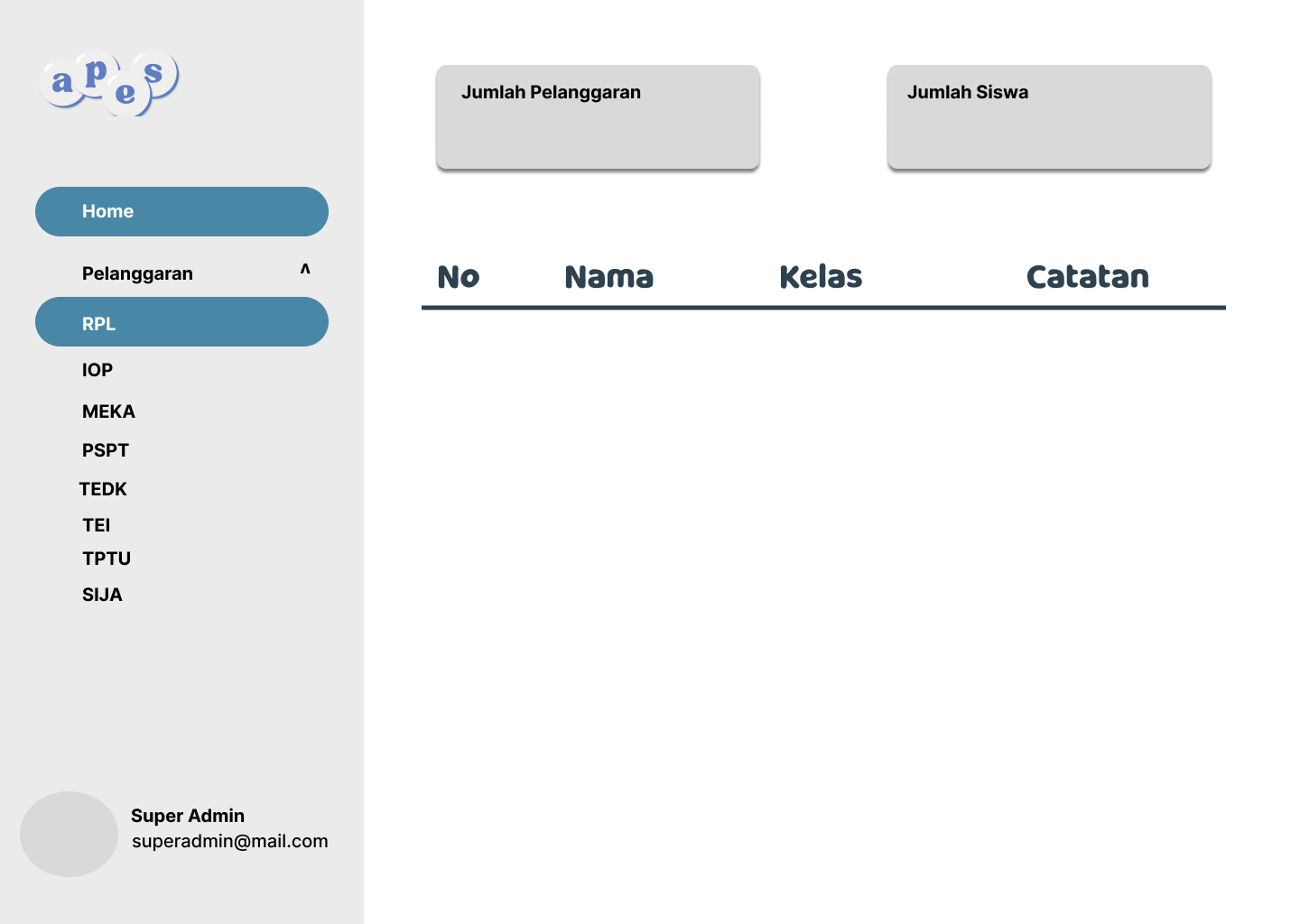
Gambar 3. 11 *Login Page Web*



Gambar 3. 12 *Home Page Web*



Gambar 3. 13 Tabel Pelnaggaran Web



Gambar 3. 14 Kategori Pelanggaran Web

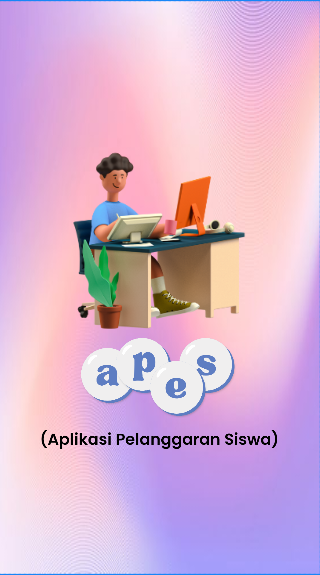
# PEMBUATAN

## Pembuatan Aplikasi Mobile

### Pembuatan Layout

Pada sub bab ini akan diberikan gambar-gambar hasil dari implementasi perancangan mockup yang sudah dilakukan pada tahap perancangan. Berikut merupakan hasil screenshoot dari layout yang sudah dibuat

1. Tampilan halaman splash screen



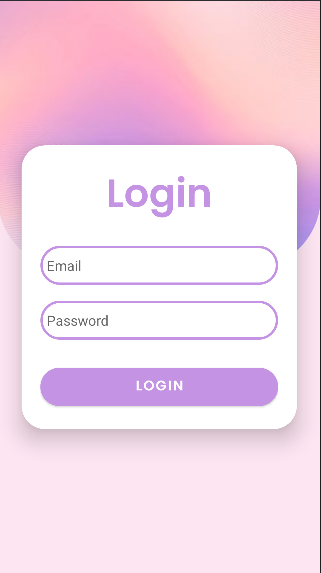
Gambar 4. 1 *Splash Screen Page*

Gambar diatas merupakan layout *splash screen*. Layout ini muncul ketika aplikasi dibuka. Dibawah ini merupakan sintaks dari layout *splash screen*

**activity\_splash\_screen.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/bg\_apk"  android:orientation="vertical"  tools:context=".SplashScreen">  <ImageView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:src="@drawable/splash\_aset"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.497"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintVertical\_bias="0.381" />  <ImageView  android:layout\_width="250dp"  android:layout\_height="wrap\_content"  android:layout\_marginTop="250dp"  android:src="@drawable/apes\_logo"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.498"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintVertical\_bias="0.555"/>  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:fontFamily="@font/poppins\_semibold"  android:text="(Aplikasi Pelanggaran Siswa)"  android:textColor="@color/black"  android:textSize="20sp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintVertical\_bias="0.775" />  </androidx.constraintlayout.widget.ConstraintLayout> |

1. Tampilan halaman login

****

Gambar 4. 2 Halaman Login

Gambar diatas merupakan tampilan dari halaman login. Setelah muncul halaman splashscreen, user akan diarahkan untuk login terlebih dahulu. Dibawah ini merupakan sintaks dari halaman login

**activity\_login.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  xmlns:card\_view="http://schemas.android.com/apk/res-auto"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:gravity="center"  android:background="@drawable/login\_bg"  tools:context=".LoginActivity">  <androidx.cardview.widget.CardView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  app:cardCornerRadius="30dp"  app:cardElevation="20dp"  android:background="@drawable/custom\_edittext">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:layout\_gravity="center\_horizontal"  android:padding="24dp">  <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Login"  android:id="@+id/loginText"  android:textSize="50sp"  android:textAlignment="center"  android:textStyle="bold"  android:fontFamily="@font/poppins\_semibold"  android:textColor="#C493E4"/>  <EditText  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:id="@+id/email"  android:background="@drawable/custom\_edittext"  android:drawablePadding="8dp"  android:hint="Email"  android:padding="8dp"  android:inputType="text"  android:textColor="@color/black"  android:textColorHighlight="@color/cardview\_dark\_background"  android:layout\_marginTop="20dp"/>  <EditText  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:id="@+id/password"  android:background="@drawable/custom\_edittext"  android:drawablePadding="8dp"  android:hint="Password"  android:padding="8dp"  android:inputType="textPassword"  android:textColor="@color/black"  android:textColorHighlight="@color/cardview\_dark\_background"  android:layout\_marginTop="20dp"/>  <Button  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:id="@+id/loginButton"  android:text="Login"  android:textSize="18sp"  android:fontFamily="@font/poppins\_semibold"  android:layout\_marginTop="30dp"  android:backgroundTint="#C493E4"  app:cornerRadius = "30dp"/>  </LinearLayout>  </androidx.cardview.widget.CardView>  </LinearLayout> |

1. Tampilan halaman dashboard



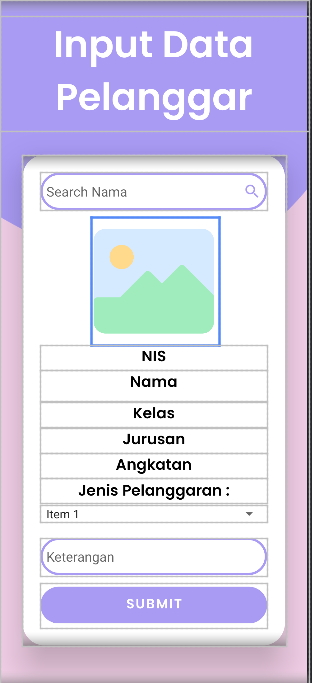
Gambar 4. 3 Halaman Dashboard

Gambar diatas merupakan tampilan dari gambar dashboard. Tampilan tersebut muncul ketika user berhasil login. Berikut merupakan sintaks pada halaman dashboard

**activity\_main.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  xmlns:card\_view="http://schemas.android.com/apk/res-auto"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:gravity="center"  android:background="#FFE1E1"  tools:context=".LoginActivity">  <androidx.constraintlayout.widget.ConstraintLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/constraintLayout"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent">  <TextView  android:layout\_width="0dp"  android:layout\_height="70dp"  android:id="@+id/apesName"  android:text="Hello !"  android:textSize="60sp"  android:textColor="@color/black"  android:layout\_marginStart="20dp"  android:layout\_marginTop="0dp"  android:fontFamily="@font/poppins\_semibold"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toStartOf="@+id/helloimg"/>  <TextView  android:layout\_width="170dp"  android:layout\_height="wrap\_content"  android:id="@+id/desc"  android:text="Welcome to Aplikasi Pelanggaran Siswa !"  android:textSize="20sp"  android:layout\_marginTop="10dp"  android:layout\_marginStart="10dp"  android:textColor="@color/black"  android:fontFamily="@font/poppins\_regular"  android:textStyle="bold"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@id/apesName"  app:layout\_constraintEnd\_toStartOf="@+id/helloimg"/>  <ImageView  android:layout\_width="190dp"  android:layout\_height="190dp"  android:layout\_marginTop="0dp"  android:id="@+id/helloimg"  android:src="@drawable/hello"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  android:layout\_marginEnd="13dp"  app:layout\_constraintStart\_toEndOf="@id/apesName"/>  </androidx.constraintlayout.widget.ConstraintLayout>  <androidx.cardview.widget.CardView  android:id="@+id/CVaddpelanggar"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  app:cardCornerRadius="20dp"  app:cardElevation="40dp"  android:background="@drawable/custom\_edittext">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <ImageView  android:id="@+id/addpelanggaran"  android:layout\_width="110dp"  android:layout\_height="120dp"  android:src="@drawable/add\_img"  android:layout\_alignParentStart="true"  android:layout\_centerVertical="true" />  <TextView  android:id="@+id/addtulisan"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:text="Tambah Pelanggaran"  android:textColor="@color/black"  android:textSize="20sp"  android:fontFamily="@font/poppins\_semibold"  android:layout\_toEndOf="@+id/addpelanggaran"  android:layout\_centerVertical="true" />  </RelativeLayout>  </androidx.cardview.widget.CardView>  <androidx.cardview.widget.CardView  android:id="@+id/CVHistoryPelanggar"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  app:cardCornerRadius="20dp"  app:cardElevation="40dp"  android:background="@drawable/custom\_edittext">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <ImageView  android:id="@+id/IvLihatPelanggar"  android:layout\_width="110dp"  android:layout\_height="120dp"  android:src="@drawable/history"  android:layout\_alignParentStart="true"  android:layout\_centerVertical="true" />  <TextView  android:id="@+id/TvLihatPelanggar"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="10dp"  android:textSize="20sp"  android:textColor="@color/black"  android:fontFamily="@font/poppins\_semibold"  android:text="Lihat History Pelanggaran"  android:layout\_toEndOf="@+id/IvLihatPelanggar"  android:layout\_centerVertical="true" />  </RelativeLayout>  </androidx.cardview.widget.CardView>  </LinearLayout> |

1. Tampilan halaman input pelanggaran



Gambar 4. 4 Halaman Input Pelanggar

Gambar diatas merupakan tampilan dari halaman input pelanggaran. Jika pada halaman dashboard user memencet input pelanggaran maka akan pindah ke halaman input pelanggaran dengan gambar seperti diatas. Berikut merupakan sintaks dari halaman input pelanggaran

**activity\_input.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/bg\_input"  android:gravity="center"  android:orientation="vertical"  tools:context=".InputActivity">  <TextView  android:id="@+id/loginText"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:fontFamily="@font/poppins\_semibold"  android:text="Input Data Pelanggar"  android:textAlignment="center"  android:textColor="@color/white"  android:textSize="50sp"  android:textStyle="bold" />  <androidx.cardview.widget.CardView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="30dp"  android:background="@drawable/custom\_edittext"  app:cardCornerRadius="30dp"  app:cardElevation="30dp">  <androidx.core.widget.NestedScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_gravity="center\_horizontal"  android:orientation="vertical"  android:padding="24dp">  <EditText  android:id="@+id/searchNama"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:background="@drawable/edittext"  android:drawableEnd="@drawable/search"  android:drawablePadding="8dp"  android:drawableTint="#A99AF4"  android:hint="Search Nama"  android:imeOptions="actionSearch"  android:inputType="text"  android:padding="8dp"  android:textColor="@color/black"  android:textColorHighlight="@color/cardview\_dark\_background" />  <ImageView  android:id="@+id/fotoSiswa"  android:layout\_width="170dp"  android:layout\_height="170dp"  android:layout\_gravity="center"  android:layout\_marginTop="10dp"  android:src="@drawable/img" />  <TextView  android:id="@+id/txtNis"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:fontFamily="@font/poppins\_semibold"  android:text="NIS"  android:textAlignment="center"  android:textColor="@color/black"  android:textSize="20sp"  android:textStyle="bold" />  <TextView  android:id="@+id/txtNama"  android:layout\_width="match\_parent"  android:layout\_height="42dp"  android:fontFamily="@font/poppins\_semibold"  android:text="Nama"  android:textAlignment="center"  android:textColor="@color/black"  android:textSize="20sp"  android:textStyle="bold" />  <TextView  android:id="@+id/txtKelas"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:fontFamily="@font/poppins\_semibold"  android:text="Kelas"  android:textAlignment="center"  android:textColor="@color/black"  android:textSize="20sp"  android:textStyle="bold" />  <TextView  android:id="@+id/txtJurusan"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:fontFamily="@font/poppins\_semibold"  android:text="Jurusan"  android:textAlignment="center"  android:textColor="@color/black"  android:textSize="20sp"  android:textStyle="bold" />  <TextView  android:id="@+id/txtAngkatan"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:fontFamily="@font/poppins\_semibold"  android:text="Angkatan"  android:textAlignment="center"  android:textColor="@color/black"  android:textSize="20sp"  android:textStyle="bold" />  <TextView  android:id="@+id/jenis"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:fontFamily="@font/poppins\_semibold"  android:text="Jenis Pelanggaran :"  android:textAlignment="center"  android:textColor="@color/black"  android:textSize="20sp"  android:textStyle="bold" />  <Spinner  android:id="@+id/jenisPelanggaran"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:outlineSpotShadowColor="@color/black" />  <EditText  android:id="@+id/Etketerangan"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:layout\_marginTop="20dp"  android:background="@drawable/edittext"  android:drawablePadding="8dp"  android:hint="Keterangan"  android:inputType="text"  android:padding="8dp"  android:textColor="@color/black"  android:textColorHighlight="@color/cardview\_dark\_background" />  <Button  android:id="@+id/submitButton"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:layout\_marginTop="10dp"  android:backgroundTint="#A99AF4"  android:fontFamily="@font/poppins\_semibold"  android:text="SUBMIT"  android:textSize="18sp"  app:cornerRadius="30dp" />  </LinearLayout>  </androidx.core.widget.NestedScrollView>  </androidx.cardview.widget.CardView>  </LinearLayout> |

1. Tampilan halaman history pelanggaran



Gambar 4. 5 Halaman History

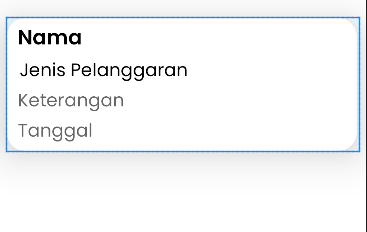
Gambar diatas merupakan tampilan dari halaman history pelanggaran. Ketika user meng-input data pada halaman input pelanggaran lalu berhasil, maka akan kembali ke halaman dashboard. Lalu jika meng-klik navigasi history pelanggaran, maka akan berpindah halaman ke halaman history pelanggaran. Berikut merupakan sintaks dari halaman history pelanggaran

**activity\_history.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#C493E4"  tools:context=".HistoryActivity">  <com.google.android.material.appbar.AppBarLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:background="#C493E4"  app:elevation="0dp">  <androidx.appcompat.widget.Toolbar  android:id="@+id/toolbar"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">  <ImageView  android:layout\_width="50dp"  android:layout\_height="50dp"  android:src="@drawable/ic\_back" />  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentStart="true"  android:layout\_centerInParent="true"  android:layout\_marginStart="60dp"  android:fontFamily="@font/poppins\_semibold"  android:text="History Pelanggaran"  android:textColor="@color/white"  android:textSize="30sp"  android:textStyle="bold" />  </RelativeLayout>  </androidx.appcompat.widget.Toolbar>  </com.google.android.material.appbar.AppBarLayout>  <androidx.core.widget.NestedScrollView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <androidx.recyclerview.widget.RecyclerView  android:id="@+id/rvPelanggaran"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  tools:listitem="@layout/pelanggaran\_item" />  </LinearLayout>  </androidx.core.widget.NestedScrollView>  <ProgressBar  android:id="@+id/pbData"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent"  android:visibility="invisible" />  </androidx.coordinatorlayout.widget.CoordinatorLayout> |

1. Tampilan card view

Tampilan cardview merupakan sebuah card yang membungkus textview yang berisikan pelanggaran siswa.



Gambar 4. 6 *Card View*

Gambar diatas merupakan cardview yang digunakan untuk menampilkan pelanggaran yang dilakukan oleh siswa. Berikut merupakan sintaks dari cardview.

**pelanggaran\_item.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="4dp"  android:layout\_marginTop="20dp"  android:layout\_marginStart="10dp"  android:layout\_marginEnd="10dp"  app:cardElevation="30dp"  app:cardCornerRadius="20dp">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_margin="4dp">  <TextView  android:id="@+id/tv\_nama"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentTop="true"  android:layout\_marginStart="8dp"  android:fontFamily="@font/poppins\_semibold"  android:textColor="@color/black"  android:text="Nama"  android:textSize="23sp"  android:textStyle="bold" />  <TextView  android:id="@+id/tv\_jenisPelanggaran"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textColor="@color/black"  android:layout\_below="@+id/tv\_nama"  android:layout\_marginStart="10dp"  android:fontFamily="@font/poppins\_regular"  android:text="Jenis Pelanggaran"  android:textSize="20sp" />  <TextView  android:id="@+id/tv\_tanggal"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="8dp"  android:textColor="@color/black"  android:layout\_below="@+id/tv\_keterangan"  android:fontFamily="@font/poppins\_regular"  android:hint="Tanggal"  android:textSize="20sp" />  <TextView  android:id="@+id/tv\_keterangan"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="8dp"  android:textColor="@color/black"  android:layout\_below="@+id/tv\_jenisPelanggaran"  android:fontFamily="@font/poppins\_regular"  android:hint="Keterangan"  android:textSize="20sp" />  <TextView  android:id="@+id/tvId"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Id"  android:textSize="20sp"  android:textStyle="bold"  android:fontFamily="@font/poppins\_regular"  android:visibility="gone"/>  </RelativeLayout>  </androidx.cardview.widget.CardView> |

### Syntax Java

1. Syntax java splashscreen

SplashScreen.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Context;  import android.content.Intent;  import android.content.SharedPreferences;  import android.os.Bundle;  import android.os.Handler;  public class SplashScreen extends AppCompatActivity {  SharedPreferences sharedPreferences;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_splash\_screen);  sharedPreferences = getSharedPreferences("MyPreferences", Context.MODE\_PRIVATE);  final Handler handler = new Handler();  handler.postDelayed(new Runnable() {  @Override  public void run() {  if (sharedPreferences.getBoolean("isLogin", false)) {  startActivity(new Intent(getApplicationContext(), MainActivity.class));  } else {  startActivity(new Intent(getApplicationContext(), LoginActivity.class));  }  finish();  }  }, 3000L); //3000=detik  }  } |

1. Syntax java login

LoginActivity.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk;  import android.content.Context;  import android.content.Intent;  import android.content.SharedPreferences;  import android.os.Bundle;  import android.text.TextUtils;  import android.util.Log;  import android.view.Gravity;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.Toast;  import androidx.appcompat.app.AppCompatActivity;  import com.azkhanazzala.pelanggaranapk.response.GetLogin;  import com.azkhanazzala.pelanggaranapk.rest.ApiClient;  import com.azkhanazzala.pelanggaranapk.rest.ApiInterface;  import retrofit2.Call;  import retrofit2.Callback;  import retrofit2.Response;  public class LoginActivity extends AppCompatActivity {  EditText Email, Password;  Button LogIn;  ApiInterface mApiInterface;  String PasswordHolder, EmailHolder;  SharedPreferences sharedPreferences;  SharedPreferences.Editor editor;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_login);  sharedPreferences = getSharedPreferences("MyPreferences", Context.MODE\_PRIVATE);  editor = sharedPreferences.edit();  Email = (EditText) findViewById(R.id.email);  Password = (EditText) findViewById(R.id.password);  LogIn = (Button) findViewById(R.id.loginButton);  mApiInterface = ApiClient.getClient().create(ApiInterface.class);  LogIn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (CheckEditTextIsEmptyOrNot()) {  handleLoginUSer();  } else {  // Toast.makeText(UserLoginActivity.this, "Please fill all form fields.", Toast.LENGTH\_LONG).show();  // Get the custom layout view.  View toastView = getLayoutInflater().inflate(R.layout.activity\_toast\_custorm\_view, null);  // Initiate the Toast instance.  Toast toast = new Toast(getApplicationContext());  // Set custom view in toast.  toast.setView(toastView);  toast.setDuration(Toast.LENGTH\_LONG);  toast.setGravity(Gravity.CENTER, 0,0);  toast.show();  }  }  });  }  public Boolean CheckEditTextIsEmptyOrNot() {  EmailHolder = Email.getText().toString();  PasswordHolder = Password.getText().toString();  return !TextUtils.isEmpty(EmailHolder) && !TextUtils.isEmpty(PasswordHolder);  }  private void handleLoginUSer(){  Email = findViewById(R.id.email);  Password = findViewById(R.id.password);  String email = Email.getText().toString();  String password = Password.getText().toString();  try {  Call<GetLogin> emailUser = mApiInterface.login("login\_user",email, password);  emailUser.enqueue(new Callback<GetLogin>() {  @Override  public void onResponse(Call<GetLogin> call, Response<GetLogin> response) {  if (response.isSuccessful()) {  if (response.body().getStatus()) {  handleSaveIdUser(response.body().getData().getIdUser(), true);  Intent i = new Intent(LoginActivity.this,MainActivity.class);  startActivity(i);  }  else Log.e("Error login ", response.body().getMessage());  } else {  Log.e(response.message() + " " + response.code(), "Response body is null or request not successful.");  }  }  @Override  public void onFailure(Call<GetLogin> call, Throwable t) {  Log.e("Retrofit Post Login", t.toString());  }  });  }catch (Exception e){  Log.e("Error", e.toString());  }  }  private void handleSaveIdUser(String idUser, boolean isLogin) {  editor.putString("idUser", idUser);  editor.putBoolean("isLogin", isLogin);  editor.apply();  }  } |

1. Syntax java dashboard

**MainActivity.java**

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk;  import androidx.appcompat.app.AppCompatActivity;  import androidx.cardview.widget.CardView;  import android.content.Intent;  import android.os.Bundle;  import android.view.View;  public class MainActivity extends AppCompatActivity {  CardView add, history;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  CardView add = (CardView) findViewById(R.id.CVaddpelanggar);  add.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent ad = new Intent(MainActivity.this, InputActivity.class);  startActivity(ad);  }  });  CardView history = (CardView) findViewById(R.id.CVHistoryPelanggar);  history.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent hi = new Intent(MainActivity.this, HistoryActivity.class);  startActivity(hi);  }  });  }  } |

1. Syntax java input pelanggaran

**InputActivity.java**

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk;  import androidx.appcompat.app.AppCompatActivity;  import androidx.cardview.widget.CardView;  import android.content.Intent;  import android.os.Bundle;  import android.view.View;  public class MainActivity extends AppCompatActivity {  CardView add, history;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  CardView add = (CardView) findViewById(R.id.CVaddpelanggar);  add.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent ad = new Intent(MainActivity.this, InputActivity.class);  startActivity(ad);  }  });  CardView history = (CardView) findViewById(R.id.CVHistoryPelanggar);  history.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent hi = new Intent(MainActivity.this, HistoryActivity.class);  startActivity(hi);  }  });  }  } |

1. Syntax java history

**HistoryActivity.java**

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk;  import androidx.appcompat.app.AppCompatActivity;  import androidx.recyclerview.widget.LinearLayoutManager;  import androidx.recyclerview.widget.RecyclerView;  import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;  import android.content.Intent;  import android.os.Bundle;  import android.view.View;  import android.widget.ProgressBar;  import android.widget.Toast;  import com.azkhanazzala.pelanggaranapk.response.GetPelanggaran;  import com.azkhanazzala.pelanggaranapk.response.Pelanggaran;  import com.azkhanazzala.pelanggaranapk.rest.ApiClient;  import com.azkhanazzala.pelanggaranapk.rest.ApiInterface;  import com.google.android.material.floatingactionbutton.FloatingActionButton;  import java.util.ArrayList;  import java.util.List;  import retrofit2.Call;  import retrofit2.Callback;  import retrofit2.Response;  public class HistoryActivity extends AppCompatActivity {  private RecyclerView rvData;  private RecyclerView.Adapter adData;  private RecyclerView.LayoutManager lmData;  private List<Pelanggaran> listData = new ArrayList<>();  private ProgressBar pbData;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_history);  rvData = findViewById(R.id.rvPelanggaran);  pbData = findViewById(R.id.pbData);  lmData = new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);  rvData.setLayoutManager(lmData);  }  @Override  protected void onResume() {  super.onResume();  retrieveData();  }  public void retrieveData() {  pbData.setVisibility(View.VISIBLE);  ApiInterface ardData = ApiClient.getClient().create(ApiInterface.class);  Call<GetPelanggaran> tampilData = ardData.ardRetrieveData();  tampilData.enqueue(new Callback<GetPelanggaran>() {  @Override  public void onResponse(Call<GetPelanggaran> call, Response<GetPelanggaran> response) {  String kode = response.body().getStatus();  String pesan = response.body().getMessage();  Toast.makeText(HistoryActivity.this, "Kode : " + kode + " | Pesan : " + pesan, Toast.LENGTH\_LONG).show();  listData = response.body().getListDataPelanggaran();  adData = new PelanggarAdapter(HistoryActivity.this, listData);  rvData.setAdapter(adData);  adData.notifyDataSetChanged();  pbData.setVisibility(View.INVISIBLE);  }  @Override  public void onFailure(Call<GetPelanggaran> call, Throwable t) {  Toast.makeText(HistoryActivity.this, "Gagal Konek Ke Server : " + t.getMessage(), Toast.LENGTH\_LONG).show();  pbData.setVisibility(View.INVISIBLE);  }  });  }  public void onBackPressed() {  Intent list\_act = new Intent(getApplicationContext(), MainActivity.class);  startActivity(list\_act);  finish();  }  } |

1. Syntax java adapter

**PelanggarAdapter.java**

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk;  import android.content.Context;  import android.content.DialogInterface;  import android.os.Handler;  import android.util.Log;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.ImageView;  import android.widget.TextView;  import android.widget.Toast;  import androidx.annotation.NonNull;  import androidx.appcompat.app.AlertDialog;  import androidx.recyclerview.widget.RecyclerView;  import com.azkhanazzala.pelanggaranapk.response.DataSiswa;  import com.azkhanazzala.pelanggaranapk.response.GetPelanggaran;  import com.azkhanazzala.pelanggaranapk.response.Pelanggaran;  import com.azkhanazzala.pelanggaranapk.rest.ApiClient;  import com.azkhanazzala.pelanggaranapk.rest.ApiInterface;  import com.bumptech.glide.Glide;  import com.bumptech.glide.request.RequestOptions;  import java.util.List;  import retrofit2.Call;  import retrofit2.Callback;  import retrofit2.Response;  public class PelanggarAdapter extends RecyclerView.Adapter<PelanggarAdapter.HolderData> {  private Context ctx;  private List<Pelanggaran> listData;  private int idPelanggaran;  public PelanggarAdapter(Context ctx, List<Pelanggaran> listData) {  this.ctx = ctx;  this.listData = listData;  }  @NonNull  @Override  public PelanggarAdapter.HolderData onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  View layout = LayoutInflater.from(parent.getContext()).inflate(R.layout.pelanggaran\_item, parent, false);  HolderData holder = new HolderData(layout);  return holder;  }  @Override  public void onBindViewHolder(@NonNull PelanggarAdapter.HolderData holder, int position) {  Pelanggaran dm = listData.get(position);  holder.tvNama.setText(dm.getNama());  holder.tvJenisPelanggaran.setText(dm.getJenis\_pelanggaran());  holder.tvKeterangan.setText(dm.getKeterangan());  holder.tvTanggal.setText(dm.getTanggal());  holder.tvId.setText(String.valueOf(dm.getId()));  }  @Override  public int getItemCount() {  return listData.size();  }  public class HolderData extends RecyclerView.ViewHolder {  TextView tvId, tvNama, tvJenisPelanggaran, tvKeterangan, tvTanggal;  public HolderData(@NonNull View itemView) {  super(itemView);  tvNama = itemView.findViewById(R.id.tv\_nama);  tvJenisPelanggaran = itemView.findViewById(R.id.tv\_jenisPelanggaran);  tvKeterangan = itemView.findViewById(R.id.tv\_keterangan);  tvTanggal = itemView.findViewById(R.id.tv\_tanggal);  tvId = itemView.findViewById(R.id.tvId);  }  }  } |

1. Syntax java config

Config.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk;  public class Config {  public static final String BASE\_URL = "http://192.168.3.43/api\_pelanggaran\_siswa\_fix/";  public static final String IMAGES\_URL = "http://192.168.3.43/api\_pelanggaran\_siswa\_fix/images/";  } |

1. Syntax login

Login.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  public class Login {  @SerializedName("id\_user")  private String idUser;  @SerializedName("username")  private String username;  @SerializedName("password")  private String password;  @SerializedName("email")  private String email;  public Login(  String idUser,  String username,  String password,  String email  ) {  this.idUser = idUser;  this.username = username;  this.password = password;  this.email = email;  }  public String getIdUser() { return idUser; }  public void setIdUser(String idUser) {this.idUser = idUser; }  public String getUsername() { return username; }  public void setUsername(String username) {this.username = username; }  public String getPassword() { return password; }  public void setPassword(String password) {this.password = password; }  public String getEmail() { return email; }  public void setEmail(String email) {this.email = email; }  } |

1. Syntax java pelanggaran

Pelanggaran.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  public class Pelanggaran {  @SerializedName("id")  private String id;  @SerializedName("nama")  private String nama;  @SerializedName("jenis\_pelanggaran")  private String jenis\_pelanggaran;  @SerializedName("keterangan")  private String keterangan;  @SerializedName("tanggal")  private String tanggal;  public String getId() {  return id;  }  public void setId(String id) {  this.id = id;  }  public String getNama() {  return nama;  }  public void setNama(String nama) {  this.nama = nama;  }  public String getJenis\_pelanggaran() {  return jenis\_pelanggaran;  }  public void setJenis\_pelanggaran(String jenis\_pelanggaran) {  this.jenis\_pelanggaran = jenis\_pelanggaran;  }  public String getKeterangan() {  return keterangan;  }  public void setKeterangan(String keterangan) {  this.keterangan = keterangan;  }  public String getTanggal() {  return tanggal;  }  public void setTanggal(String tanggal) {  this.tanggal = tanggal;  }  } |

1. Syntax java input pelanggaran

InputPelanggaran.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  public class InputPelanggaran {  @SerializedName("status")  int status;  @SerializedName("message")  String message;  public int getStatus() {  return status;  }  public void setStatus(int status) {  this.status = status;  }  public String getMessage() {  return message;  }  public void setMessage(String message) {  this.message = message;  }  } |

1. Syntax java get pelanggaran

GetPelanggaran.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  import java.util.List;  public class GetPelanggaran {  @SerializedName("status")  String status;  @SerializedName("message")  String message;  @SerializedName("data")  List<Pelanggaran> listDataPelanggaran;  public String getStatus() {  return status;  }  public void setStatus(String status) {  this.status = status;  }  public String getMessage() {  return message;  }  public void setMessage(String message) {  this.message = message;  }  public List<Pelanggaran> getListDataPelanggaran() {  return listDataPelanggaran;  }  public void setListDataPelanggaran(List<Pelanggaran> listDataPelanggaran) {  this.listDataPelanggaran = listDataPelanggaran;  }  } |

1. Syntax java get login

GetLogin.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  public class GetLogin {  @SerializedName("status")  boolean status;  @SerializedName("message")  String message;  @SerializedName("data")  Login dataLogin;  public boolean getStatus() {return status;}  public void setStatus(boolean status) {  this.status = status;  }  public String getMessage() {  return message;  }  public void setMessage(String message) {  this.message = message;  }  public Login getData() {  return dataLogin;  }  public void setData(Login loginUser) {this.dataLogin = loginUser;}  } |

1. Syntax java get foto siswa

GetFotoSiswa.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  public class GetFotoSiswa {  @SerializedName("status")  int status;  @SerializedName("message")  String message;  @SerializedName("data")  String dataFotoSiswa;  public int getStatus() {  return status;  }  public void setStatus(int status) {  this.status = status;  }  public String getMessage() {  return message;  }  public void setMessage(String message) {  this.message = message;  }  public String getData() {  return dataFotoSiswa;  }  public void setData(String fotoSiswa) {  this.dataFotoSiswa = fotoSiswa;  }  } |

1. Syntax java get data siswa

GetDataSiswa.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  public class GetDataSiswa {  @SerializedName("status")  int status;  @SerializedName("message")  String message;  @SerializedName("data")  DataSiswa dataSiswa;  //List<Pelanggaran> ListDataPelanggaran;  public int getStatus() {  return status;  }  public void setStatus(int status) {  this.status = status;  }  public String getMessage() {  return message;  }  public void setMessage(String message) {  this.message = message;  }  public DataSiswa getData() {  return dataSiswa;  }  public void setData(DataSiswa pelanggaran) {  this.dataSiswa = pelanggaran;  }  /\*public List<Pelanggaran> getListDataPelanggaran(){  return ListDataPelanggaran;  }\*/  } |

1. Syntax java foto siswa

FotoSiswa.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  public class FotoSiswa {  @SerializedName("imageData")  private String fotoSiswa;  public FotoSiswa(String fotoSiswa) {  this.fotoSiswa = fotoSiswa;  }  public String getFotoSiswa() {  return fotoSiswa;  }  public void setFotoSiswa(String fotoSiswa) {  this.fotoSiswa = fotoSiswa;  }  } |

1. Syntax java data siswa

DataSiswa.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.response;  import com.google.gson.annotations.SerializedName;  public class DataSiswa {  @SerializedName("nis")  private String nis;  @SerializedName("nama")  private String nama;  @SerializedName("kelas")  private String kelas;  @SerializedName("jurusan")  private String jurusan;  @SerializedName("angkatan")  private String angkatan;  @SerializedName("foto")  private String foto;  /\*@SerializedName("date")  private String date;\*/  public DataSiswa(String nis, String nama, String kelas, String jurusan, String angkatan, String foto) {  this.nis = nis;  this.nama = nama;  this.kelas = kelas;  this.jurusan = jurusan;  this.angkatan = angkatan;  this.foto = foto;  }  public String getNis() {  return nis;  }  public void setNis(String nis) {  this.nis = nis;  }  public String getNama() { return nama; }  public void setNama(String nama) {this.nama = nama; }  public String getKelas() { return kelas; }  public void setKelas(String kelas) {this.kelas = kelas; }  public String getJurusan() { return jurusan; }  public void setJurusan(String jurusan) {this.jurusan = jurusan; }  public String getAngkatan() { return angkatan; }  public void setAngkatan(String angkatan) {this.angkatan = angkatan; }  public String getFoto() { return foto; }  public void setFoto(String foto) {this.foto = foto; }  /\*public String getDate(){  return date;  }\*/  } |

1. Syntax java api client

ApiClient.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.rest;  import com.azkhanazzala.pelanggaranapk.Config;  import com.google.gson.Gson;  import com.google.gson.GsonBuilder;  import retrofit2.Retrofit;  import retrofit2.converter.gson.GsonConverterFactory;  public class ApiClient {  public static final String BASE\_URL = Config.BASE\_URL;  private static Retrofit retrofit = null;  public static Retrofit getClient() {  Gson gson = new GsonBuilder().setLenient().create();  if (retrofit == null) {  retrofit = new Retrofit.Builder()  .baseUrl(BASE\_URL)  .addConverterFactory(GsonConverterFactory.create(gson))  .build();  }  return retrofit;  }  } |

1. Syntax java interface Api Interface

ApiInterface.java

|  |
| --- |
| package com.azkhanazzala.pelanggaranapk.rest;  import com.azkhanazzala.pelanggaranapk.response.GetFotoSiswa;  import com.azkhanazzala.pelanggaranapk.response.GetLogin;  import com.azkhanazzala.pelanggaranapk.response.GetDataSiswa;  import com.azkhanazzala.pelanggaranapk.response.GetPelanggaran;  import com.azkhanazzala.pelanggaranapk.response.InputPelanggaran;  import retrofit2.Call;  import retrofit2.http.Field;  import retrofit2.http.FormUrlEncoded;  import retrofit2.http.GET;  import retrofit2.http.POST;  import retrofit2.http.Query;  public interface ApiInterface {  @FormUrlEncoded  @POST("inputpelanggaran.php")  Call<GetLogin> login(  @Query("function") String function,  @Field("email") String email,  @Field("password") String password  );  @GET("inputpelanggaran.php")  Call<GetDataSiswa> getDataSiswaByName(  @Query("function") String function,  @Query("nama\_siswa") String name  );  @GET("inputpelanggaran.php")  Call<GetFotoSiswa> getFotoSiswa(  @Query("function") String function  );  @FormUrlEncoded  @POST("inputpelanggaran.php")  Call<InputPelanggaran> inputPelanggaran(  @Query("function") String function,  @Field("nis") String nis,  @Field("jenis\_pelanggaran") String jenisPelanggaran,  @Field("keterangan") String keterangan,  @Field("tanggal") String tanggal,  @Field("id\_user") String idUser  );  @GET("retrieve.php")  Call<GetPelanggaran> ardRetrieveData();  @FormUrlEncoded  @POST("get.php")  Call<GetPelanggaran> ardGetData(  @Field("id") int id  );  @FormUrlEncoded  @POST("delete.php")  Call<GetPelanggaran> ardDeleteData(  @Field("id") int id  );  } |

### Pembuatan API

1. Koneksi

Koneksi.php

|  |
| --- |
| <?php  $hostname = "localhost";  $database = "fixdbapes";  $username = "root";  $password = "";  $connect = mysqli\_connect($hostname, $username, $password, $database);  // script cek koneksi  if(!$connect){  die("Koneksi tidak berhasil : ".mysqli\_connect\_error());  }  ?> |

1. Get

Get.php

|  |
| --- |
| <?php  require ("koneksi.php");  $response = array();  if($\_SERVER['REQUEST\_METHOD'] == 'POST'){  $id = $\_POST["id"];  $perintah = "SELECT p.\*, s.nis, s.nama as nama  FROM pelanggarans p  JOIN siswas s ON p.nis = s.nis";  $eksekusi = mysqli\_query($connect, $perintah);  $cek = mysqli\_affected\_rows($connect);  if($cek > 0){  $response["kode"] = 1;  $response["pesan"] = "Data Tersedia";  $response["data"] = array();  while($ambil = mysqli\_fetch\_object($eksekusi)){  $F["id\_pelanggaran"] = $ambil->id\_pelanggaran;  $F["nis"] = $ambil->nis;  $F["jenis\_pelanggaran"] = $ambil->jenis\_pelanggaran;  $F["keterangan"] = $ambil->keterangan;  $F["tanggal"] = $ambil->tanggal;  $F["id\_user"] = $ambil->id\_user;  array\_push($response["data"], $F);  }  } else {  $response["kode"] = 11;  $response["pesan"] = "Data Tidak Tersedia";  }  } else {  $response["kode"] = 11;  $response["pesan"] = "Tidak Ada Post Data";  }  echo json\_encode($response);  mysqli\_close($connect); |

1. Delete

Delete.php

|  |
| --- |
| <?php  require ("koneksi.php");  $response = array();  if($\_SERVER['REQUEST\_METHOD'] == 'POST'){  $id = $\_POST["id\_pelanggaran"];  $perintah = "DELETE FROM pelanggaran WHERE id\_pelanggaran = '$id'";  $eksekusi = mysqli\_query($connect, $perintah);  $cek = mysqli\_affected\_rows($connect);  if($cek > 0){  $response["kode"] = 1;  $response["pesan"] = "Data Berhasil Dihapus";  } else {  $response["kode"] = 11;  $response["pesan"] = "Data Gagal Dihapus";  }  } else {  $response["kode"] = 11;  $response["pesan"] = "Tidak Ada Post Data";  }  echo json\_encode($response);  mysqli\_close($connect); |

1. Input

Inputpelanggaran.php

|  |
| --- |
| <?php  require\_once "koneksi.php";  if (isset($\_GET['function']) && function\_exists($\_GET['function'])) {  $\_GET['function']();  }  function get\_siswa\_by\_name() {  global $connect;  $nama\_siswa = $\_GET['nama\_siswa'];  $query = $connect->query("SELECT \* FROM siswas WHERE `nama` LIKE '%$nama\_siswa%'");  $data = mysqli\_fetch\_object($query);  $response = array(  'status' => 1,  'message' => 'success',  'data' => $data  );  header('Content-Type: application/json');  echo json\_encode($response);  }  function get\_siswa\_image() {  global $connect;  $imagePath = 'fotosiswa/abel.jpg';    if (file\_exists($imagePath)) {  $imageData = base64\_encode(file\_get\_contents($imagePath));    $response = array(  'status' => 1,  'message' => 'success',  'data' => $imageData  );  header('Content-Type: application/json');  echo json\_encode($response);  } else {  http\_response\_code(404);  header('Content-Type: application/json');  echo json\_encode(['error' => 'Image not found']);  }  }  function login\_user()  {  if(isset($\_POST)){  global $connect;  $email = $\_POST['email'];  $password = $\_POST['password'];  $data;    $query = $connect->query("SELECT \* FROM users WHERE `email` = '". $email . "' AND `password` = '". $password ."' LIMIT 1");  $data = mysqli\_fetch\_object($query);  if(isset($data) && (count((array)$data) > 0)){  $message = "Data Matched";  $returnStatus = true;  } else {  $message = "Invalid Username or Password Please Try Again";  $returnStatus = false;  }  } else {  $message = "Check Again";  $returnStatus = false;  }  $response = array(  'status' => $returnStatus,  'message' => $message,  'data' => $data  );  header('Content-Type: application/json');  echo json\_encode($response);  }  function insert\_pelanggaran()  {  global $connect;  $check = array('nis' => '', 'jenis\_pelanggaran' => '', 'keterangan' => '', 'tanggal' => '', 'id\_user' => '');  $check\_match = count(array\_intersect\_key($\_POST, $check));  if($check\_match == count($check)){  $query = "INSERT INTO pelanggarans (nis, jenis\_pelanggaran, keterangan, tanggal, id\_user) VALUES ('".$\_POST["nis"]."','".$\_POST["jenis\_pelanggaran"]."','".$\_POST["keterangan"]."','".$\_POST["tanggal"]."','".$\_POST["id\_user"]."')";  $result = mysqli\_query($connect, $query);    if($result)  {  $response=array(  'status' => 1,  'message' =>'Insert Success'  );  }  else  {  $response=array(  'status' => 0,  'message' =>'Insert Failed.'  );  }  }else{  $response=array(  'status' => 0,  'message' =>'Wrong Parameter'  );  }  header('Content-Type: application/json');  echo json\_encode($response);  }  ?> |

1. Retrieve

Retrieve.php

|  |
| --- |
| <?php  require("koneksi.php");  $perintah = "SELECT p.\*, s.nis, s.nama as nama  FROM pelanggarans p  JOIN siswas s ON p.nis = s.nis";  $eksekusi = mysqli\_query($connect, $perintah);  $cek = mysqli\_affected\_rows($connect);  if($cek > 0){  $response["kode"] = 1;  $response["pesan"] = "Data Tersedia";  $response["data"] = array();  while($ambil = mysqli\_fetch\_object($eksekusi)){  $F["id\_pelanggaran"] = $ambil->id\_pelanggaran;  $F["nama"] = $ambil->nama;  $F["jenis\_pelanggaran"] = $ambil->jenis\_pelanggaran;  $F["keterangan"] = $ambil->keterangan;  $F["tanggal"] = $ambil->tanggal;  $F["id\_user"] = $ambil->id\_user;  array\_push($response["data"], $F);  }  } else {  $response["kode"] = 0;  $response["pesan"] = "Data Tidak Tersedia";  }  echo json\_encode($response);  mysqli\_close($connect); |

## Pembuatan Aplikasi Web

### Pembuatan Backend di Laravel

1. Membuat fungsi login

|  |
| --- |
| public function login(Request $request)  {  $request->validate([  'email' => 'required|string|email',  'password' => 'required|string',  ]);  $credentials = $request->only('email', 'password');  if (Auth::attempt($credentials)) {  $user = Auth::user();  return response()->json([  'data' => [  'user' => $user,  'authorization' => [  'token' => $user->createToken('ApiToken')->plainTextToken,  'type' => 'bearer',  ]  ]  ]);  }  return response()->json([  'message' => 'Invalid credentials',  ], 401);  } |

1. Fungsi mengambil data seluruh pelanggaran

|  |
| --- |
| function getPelanggaran($id = null)  {  if (isset($id)) {  $pelanggaran = Pelanggaran::join('siswas', 'pelanggarans.nis', '=', 'siswas.nis')  ->join('users', 'pelanggarans.id\_user', '=', 'users.id\_user')  ->where('pelanggarans.id\_pelanggaran', $id)  ->select('siswas.nama as nama\_siswa', 'siswas.kelas', 'siswas.angkatan', 'siswas.jurusan', 'users.name as nama\_pendata', 'pelanggarans.\*', 'pelanggarans.jenis\_pelanggaran')  ->get();  return response()->json(['msg' => 'Data retrieved', 'data' => $pelanggaran], 200);  } else {  $pelanggaran = Pelanggaran::join('siswas', 'pelanggarans.nis', '=', 'siswas.nis')  ->join('users', 'pelanggarans.id\_user', '=', 'users.id\_user')  ->select('siswas.nama as nama\_siswa', 'siswas.kelas', 'siswas.angkatan', 'siswas.jurusan', 'users.name as nama\_pendata', 'pelanggarans.\*')  ->get();  return response()->json(['msg' => 'Data retrieved', 'data' => $pelanggaran], 200);  }  } |

1. Fungsi menghapus data seluruh jurusan

|  |
| --- |
| function destroy($id)  {  $pelanggaran = Pelanggaran::findOrFail($id);  $pelanggaran->delete();  return response()->json(['msg' => 'Data deleted'], 200);  } |

1. Fungsi mengambil data per jurusan

|  |
| --- |
| public function getAllRpl()  {  $rpl = RPL::join('siswas', 'pelanggarans.nis', '=', 'siswas.nis')  ->join('users', 'pelanggarans.id\_user', '=', 'users.id\_user')  ->where('siswas.jurusan', 'Rekayasa Perangkat Lunak')  ->select('siswas.nama as nama\_siswa', 'siswas.kelas', 'siswas.angkatan', 'siswas.jurusan', 'users.name as nama\_pendata', 'pelanggarans.\*', 'pelanggarans.jenis\_pelanggaran')  ->get();    return response()->json(['msg' => 'Data retrieved', 'data' => $rpl], 200);  } |

1. Fungsi menghapus data per jurusan

|  |
| --- |
| public function destroyrpl($id)  {  $rpl = RPL::find($id);  if (!$rpl) {  return response()->json(['message' => 'Pelanggaran tidak ditemukan'], 404);  }  $pelanggaran->delete();  return response()->json(['message' => 'Pelanggaran berhasil dihapus']);  } |

1. Isi dari model yang mencakup seluruh pelanggaran

|  |
| --- |
| lass Pelanggaran extends Model  {  use HasFactory;    protected $table = 'pelanggarans';  protected $primaryKey = 'id\_pelanggaran';  protected $keyType = 'string';  public $incrementing = false;  protected $fillable = ['nis', 'jenis\_pelanggaran', 'keterangan', 'id\_user', 'foto'];  protected $hidden = [  'created\_at',  'updated\_at',  ];  public function siswa()  {  return $this->belongsTo(Siswa::class, 'nis', 'nis');  }  public function user()  {  return $this->belongsTo(User::class, 'id\_user', 'id\_user');  }  } |

1. Isi dari model yang mencakup per jurusan

|  |
| --- |
| class RPL extends Model  {  use HasFactory;    protected $table = 'pelanggarans';  protected $primaryKey = 'id\_pelanggaran';  protected $keyType = 'string';  public $incrementing = false;  protected $fillable = ['nis', 'jenis\_pelanggaran', 'keterangan', 'id\_user', 'foto'];  protected $hidden = [  'created\_at',  'updated\_at',  ];  public function siswa()  {  return $this->belongsTo(Siswa::class, 'nis', 'nis');  }  public function user()  {  return $this->belongsTo(User::class, 'id\_user', 'id\_user');  }  } |

1. Route agar fungsi bisa dipanggil

|  |
| --- |
| Route::middleware('auth:api')->get('/user', function (Request $request) {  return $request->user();  });  Route::controller(AuthController::class)->group(function () {  Route::post('login', 'login');  Route::post('register', 'register');  });  Route::get('/search', 'SearchController@search');  Route::middleware('auth:sanctum')->group(function () {  // Current Logged In User  Route::get('current-user', [AuthController::class, 'currentUser']);  // Current Logged In User  Route::get('user', [AuthController::class, 'allUser']);  // Logout  Route::post('logout', [AuthController::class, 'logout']);  // Angkatan  Route::post('create-angkatan', [AngkatanController::class, 'createAngkatan']);  Route::get('angkatan', [AngkatanController::class, 'showAllAngkatan']);  // Siswa  Route::post('create-siswa', [SiswaController::class, 'createSiswa']);  Route::get('siswa', [SiswaController::class, 'showAllSiswa']);  // Pelanggaran  Route::post('create-pelanggaran', [PelanggaranController::class, 'createPelanggaran']);  Route::get('pelanggarans', [PelanggaranController::class, 'getPelanggaran']);  Route::get('pelanggarans/{id}', [PelanggaranController::class, 'getPelanggaran']);  // RPL  Route::get('pelanggaran/rpl', [AllJurusanController::class, 'getAllRpl']); |

### Pembuatan Backend di React js

1. Menghubungkan API dari laravel dalam file env

|  |
| --- |
| VITE\_BACKEND\_HOST=http://localhost:8000/api/ |

1. Fungsi memanggil url API dari file env dalam folder src/lib/backend.js

|  |
| --- |
| export function getBackendHost() {  return (  import.meta.env?.VITE\_BACKEND\_HOST?.replace(/\/$/, "") ||  window.location.origin  );  } |

1. Konfigurasi URL API menggunakan AXIOS dalam folder src/lib/common/api.js

|  |
| --- |
| // Konfigurasi Headers  const headersReg = {  "Content-Type": "application/json",  Authorization: "",  Accept: "\*/\*"  };  // Konfigurasi URL API menggunakan AXIOS  // getBackendHost() manggil dari .env  export const Api = axios.create({  baseURL: getBackendHost(), // http://localhost:8000/api/  headers: headersReg  });  Api.interceptors.request.use(  async (config) => {  // token diambil dari store  const { userToken } = useStore.getState();  if (config?.headers) {  if (userToken) {  config.headers.Authorization = `Bearer ${userToken}`;  }  }  return config;  },  (error) => {  return Promise.reject(error);  }  ); |

1. Membuat Api, classNames, dan errorNotification tersedia untuk diimpor di tempat lain dalam aplikasi Anda dalam file index.js dalam folder src/lib/common/index.js

|  |
| --- |
| export { Api } from './api'  export { classNames, errorNotification } from './utils' |

1. Buat file utils.js dalam folder src/lib/common/utils.js

|  |
| --- |
| import { toast } from 'react-toastify'  export const classNames = (...classes) => {  return classes.filter(Boolean).join(' ')  }  export const errorNotification = (error) => {  if (Array.isArray(error.response.data.message)) {  error.response.data.message.forEach((el) =>  toast.error(el, {  position: 'top-right',  })  )  } else {  toast.error(error.response.data.message, {  position: 'top-right',  })  }  } |

1. Berikut fungsi untuk logout dan login user dalam folder src/api/auth.js

|  |
| --- |
| import { Api } from '@/lib/common'  export const loginUserFn = async (user) => {  const response = await Api.post('login', user)  return response.data  }  export const logoutUserFn = async () => {  const response = await Api.post('logout', )  return response.data  }  export const currentUserFn = async () => {  const response = await Api.get('current-user')  return response.data  } |

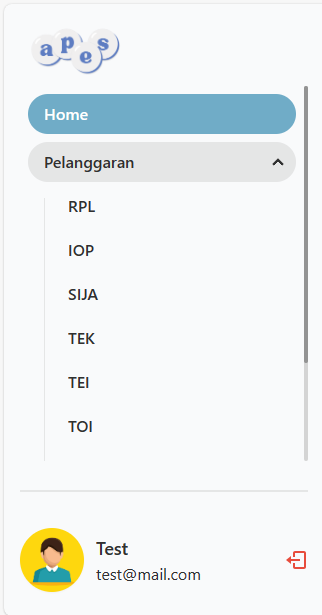
1. Fungsi untuk menyimpan data pelanggaran dalam fokder src/api/pelanggaran.js

|  |
| --- |
| // Pelanggaran All  export const allPelanggaranFn = async () => {  const response = await Api.get('pelanggarans')  return response.data  }  // Siswa All  export const allSiswasFn = async () => {  const response = await Api.get('siswa')  return response.data  }  // User All  export const allUsersFn = async () => {  const response = await Api.get('user')  return response.data  }  // Pelanggaran RPL  export const allPelanggaranRplFn = async () => {  const response = await Api.get('pelanggaran/rpl')  return response.data  } |

1. Agar tampilan rapih buat file AutorizeLayout.jsx dalam folder src/components/layout

|  |
| --- |
| export default function AuthorizeLayout(props) {  const { content } = props    //outlet biar tampilan rapih tersusun nya  return (  <main>  <Sidebar>  <div className='p-4'>  <div>{content ?? <Outlet />}</div>  </div>  </Sidebar>  </main>  )  } |

1. Untuk membuat tampilan navigasi buat folder dalam src/Components/Navigation



Gambar 4. 7 *Sidebar*

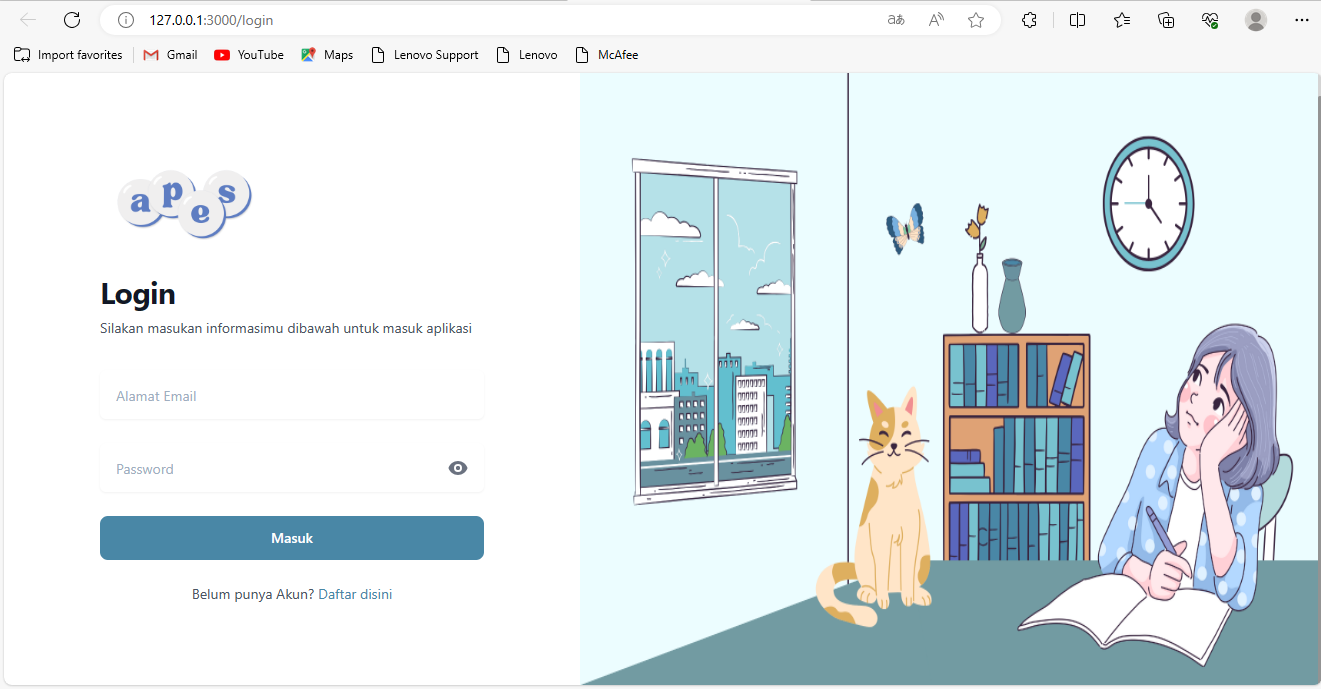
* 1. Buat file NavigationItem.js

|  |
| --- |
| export const navigation = [  { name: 'Home', href: '/', current: false },  {  name: 'Pelanggaran',  sub\_item: [  {  name: 'RPL',  href: '/pelanggaran/rpl',  current: false,  },  {  name: 'IOP',  href: '/pelanggaran/iop',  current: false,  },  {  name: 'SIJA',  href: '/pelanggaran/sija',  current: false,  },  {  name: 'TEK',  href: '/pelanggaran/tek',  current: false,  },  {  name: 'TEI',  href: '/pelanggaran/tei',  current: false,  },  {  name: 'TOI',  href: '/pelanggaran/toi',  current: false,  },  {  name: 'PSPT',  href: '/pelanggaran/pspt',  current: false,  },  {  name: 'TPTU',  href: '/pelanggaran/tptu',  current: false,  },  {  name: 'MEKA',  href: '/pelanggaran/meka',  current: false,  },  ],  },  ] |

* 1. Buat file Sidebar.jsx

|  |
| --- |
| <div className='drawer !h-[100vh] lg:drawer-open'>  <input id='my-drawer-2' type='checkbox' className='drawer-toggle' />  <div className='drawer-content overflow-y-auto flex flex-col'>  <div  className={`${navbarClass} w-full lg:hidden navbar navbar-transition fixed top-0 bg-base`}  >  <div className='flex-none lg:hidden'>  <label  htmlFor='my-drawer-2'  aria-label='open sidebar'  className='btn btn-square btn-ghost drawer-button lg:hidden'  >  <HiOutlineBars3CenterLeft size={24} />  </label>  </div>  <div className='flex-1 px-2 mx-2'>{pageTitle}</div>  </div>  {children}  </div>  <div className='drawer-side !h-[100vh]'>  <label  htmlFor='my-drawer-2'  aria-label='close sidebar'  className='drawer-overlay'  ></label>  <div className='flex flex-col p-4 w-full sm:w-80 h-[100%] bg-base-200 text-base-content'>  <div className='h-[10%] mb-2 flex justify-between items-center'>  <img src={ApesLogo} className='w-28' alt='' />  <label  htmlFor='my-drawer-2'  aria-label='open sidebar'  className='btn btn-square btn-ghost drawer-button lg:hidden'  >  <LiaTimesSolid size={18} />  </label>  </div>  <ul className='!block text-base font-medium menu h-[70%] overflow-y-auto gap-1 bg-base-200 w-full'>  {updatedNavigation.map((item, index) => (  <li key={index} className='mb-2'>  {item.sub\_item ? ( // Check if there are sub-items  <details>  <summary className='mb-1'>  {item.name}  </summary>  <ul className='flex gap-1 flex-col'>  {item.sub\_item.map(  (subItem, subIndex) => (  <li key={subIndex}>  <Link  className={  subItem.current  ? 'active !text-white'  : ''  }  to={subItem.href}  >  {subItem.name}  </Link>  </li>  )  )}  </ul>  </details>  ) : (  // Render basic li for items with no sub-items  <Link  className={  item.current  ? 'active !text-white'  : '!bg-transparent'  }  href={item.href}  >  {item.name}  </Link>  )}  </li>  ))}  </ul>  <div className='divider h-[5%]'></div>  {!loadingLoggedInUser && (  <div className='flex h-[15%] items-center justify-between'>  <div className='flex items-center gap-3'>  <div className='rounded-full shadow-sm'>  <img  className='rounded-full w-16'  src={dummyAvatarImg}  alt=''  />  </div>  <div>  <p className='text-lg font-semibold capitalize'>  {dataLoggedInUser.user.name}  </p>  <p className=''>  {dataLoggedInUser.user.email}  </p>  </div>  </div>  <Button  className='tooltip tooltip-secondary h-fit'  variant='solid'  color='transparent'  data-tip='Logout'  onClick={() =>  document  .getElementById('confirm-logout-modal')  .showModal()  }  >  <BiLogOut size={24} color='#E74C3C' />  </Button>  </div>  )}  </div>  </div>  <dialog id='confirm-logout-modal' className='modal'>  <div className='modal-box'>  <h3 className='font-bold text-lg text-[#E74C3C]'>Logout</h3>  <p className='py-4'>Are you sure you want to logout?</p>  <div className='modal-action'>  <form method='dialog'>  <Button className='btn !rounded-lg'>Close</Button>  <Button  onClick={logoutUser}  className='btn !rounded-lg text-[#E74C3C]'  >  Logout  </Button>  </form>  </div>  </div> |

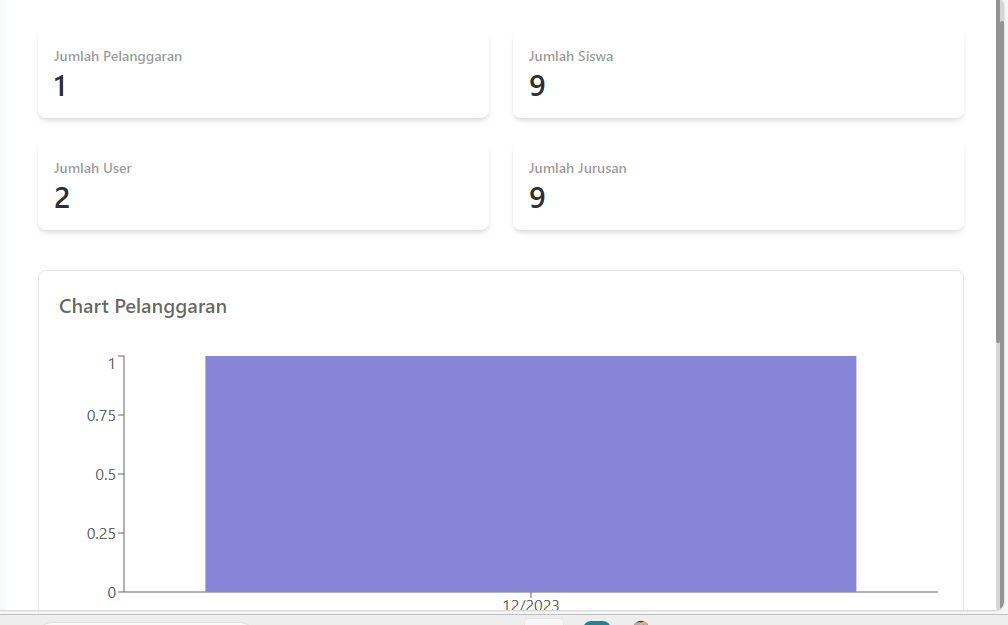
1. Berikut kode untuk menampilkan tampilan login pada web



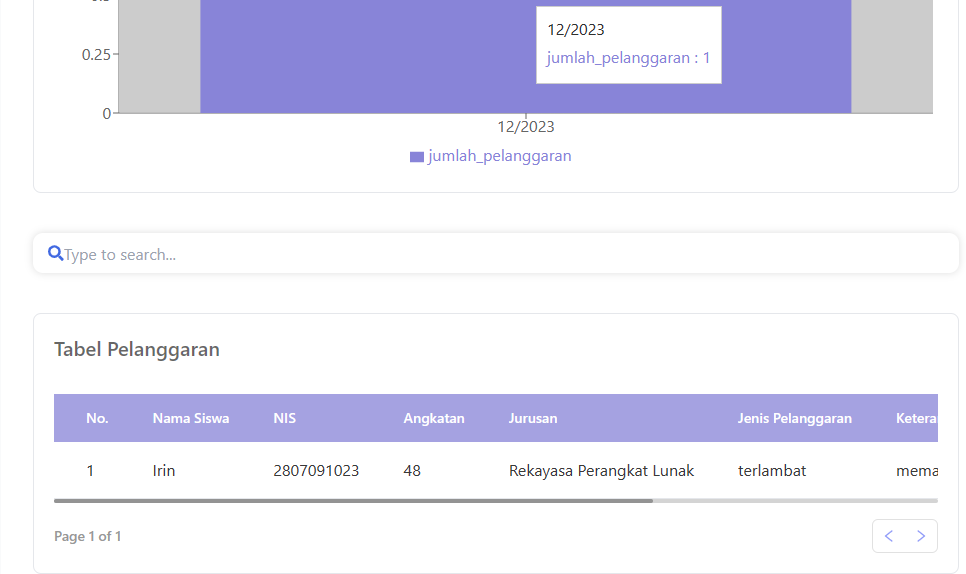
Gambar 4. 8 Halaman Login Web

|  |
| --- |
| <div className='flex min-h-full'>  <div className='flex flex-1 flex-col justify-center px-4 py-12 sm:px-6 lg:flex-none lg:px-20 xl:px-24'>  <div className='mx-auto w-full max-w-sm lg:w-96'>  <div>  <img  className='h-24 w-auto '  src={ApesLogo}  alt='Cimahi Logo'  />  <h2 className='mt-6 text-3xl font-bold tracking-tight text-gray-900'>  Login  </h2>  <p className='mt-2 text-sm text-gray-600'>  Silakan masukan informasimu dibawah untuk masuk  aplikasi  </p>  </div>  <div className='mt-8'>  <div className='mt-6'>  <FormProvider {...methods}>  <form  className='space-y-6'  onSubmit={handleSubmit(onSubmitHandler)}  >  <Input  id='email'  label='Alamat Email'  className='text-base font-normal py-2'  />  <PasswordInput  id='password'  label='Password'  className='text-base font-normal py-2'  />  <div>  <Button  className='w-full px-5 py-3 rounded-lg'  type='submit'  variant='solid'  color='blue'  >  Masuk  </Button>  </div>  <div>  <p className='mt-2 text-center text-sm text-gray-600'>  Belum punya Akun?{' '}  <Link to={'/register'}>  Daftar disini  </Link>  </p>  </div>  </form>  </FormProvider>  </div>  </div>  </div>  </div>  <div className='relative hidden w-0 flex-1 lg:block'>  <img  className='absolute inset-0 h-full w-full object-center'  src={LoginBG}  alt=''  />  </div>  </div> |

1. Untuk membuat tampilan data seluruh pelanggaran buat folder dalam src/Components/Dashboard



Gambar 4. 9 Halaman Dashboard Web



Gambar 4. 10 Halaman Dashboard Web

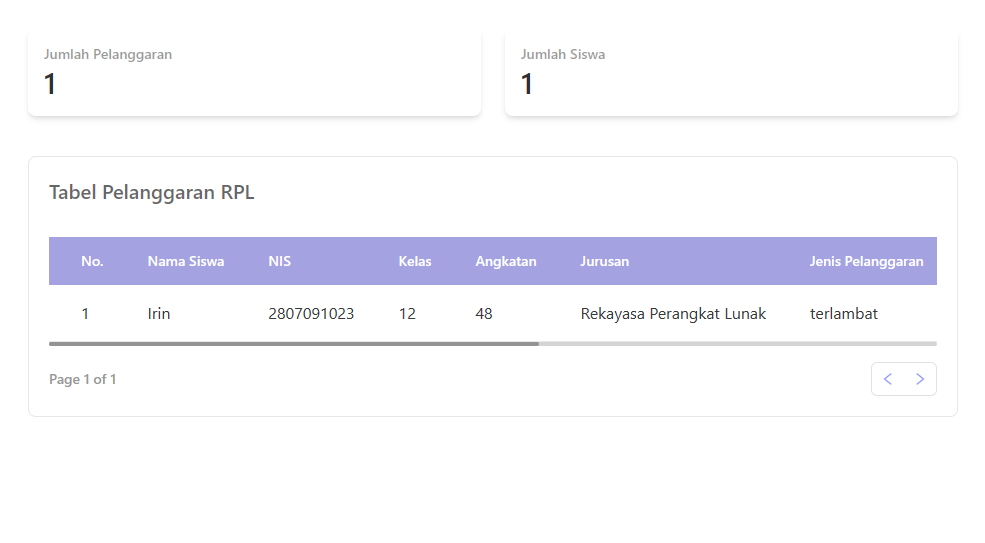
1. Buat file DashboardTable.jsx

|  |
| --- |
| const columns = [  columnHelper.accessor((row, index) => index + 1, {  header: 'No.',  }),  columnHelper.accessor('nama\_siswa', {  header: 'Nama Siswa',  cell: (info) => info.getValue(),  footer: (info) => info.getValue().nama\_siswa,  }),  columnHelper.accessor('nis', {  header: 'NIS',  cell: (info) => info.getValue(),  footer: (info) => info.getValue().nis,  }),  columnHelper.accessor('angkatan', {  header: 'Angkatan',  cell: (info) => info.getValue(),  footer: (info) => info.getValue().angkatan,  }),  columnHelper.accessor('jurusan', {  header: 'Jurusan',  cell: (info) => info.getValue(),  footer: (info) => info.getValue().jurusan,  }),  columnHelper.accessor('jenis\_pelanggaran', {  header: 'Jenis Pelanggaran',  cell: (info) => info.getValue(),  footer: (info) => info.getValue().jenis\_pelanggaran,  }),  columnHelper.accessor('keterangan', {  header: 'Keterangan',  cell: (info) => info.getValue(),  footer: (info) => info.getValue().keterangan,  }),  columnHelper.accessor('tanggal', {  header: 'Tanggal',  cell: (info) => info.getValue(),  footer: (info) => info.getValue().tanggal,  }),  columnHelper.accessor("id\_pelanggaran", {  header: "Aksi",  cell: (info) => (  <div className="flex flex-row gap-5">  <Button  key="delete-button"  variant="solid"  color="blue"  onClick={() => {  setSelectedId(info.getValue());  document  .getElementById('confirm-delete-modal')  .showModal()  }}  >  Delete  </Button>  </div>  ),  footer: (info) => info.getValue().id\_pelanggaran  }),  ]  return (  <>  <Table data={dataPelanggaran} columns={columns} />  <dialog id='confirm-delete-modal' className='modal'>  <div className='modal-box'>  <h3 className='font-bold text-lg text-[#E74C3C]'>Hapus Pelanggaran</h3>  <p className='py-4'>Yakin ingin menghapus data ini?</p>  <div className='modal-action'>  <form method='dialog'>  <Button className='btn !rounded-lg !me-4'>Tutup</Button>  <Button  onClick={() => deletePelanggaran(selectedId)}  className='btn !rounded-lg text-[#E74C3C]'  >  Hapus  </Button>  </form>  </div>  </div>  </dialog>  </>  ) |

1. Buat file index.js

|  |
| --- |
| <div className='mt-14 lg:mt-0 lg:p-4'>  <div className='grid grid-cols-1 sm:grid-cols-2 gap-6'>  <div className='p-4 bg-white shadow-md rounded-lg'>  <p className='text-sm font-medium text-black/40'>  Jumlah Pelanggaran  </p>  {!loadingPelanggaran && (  <h1 className='font-semibold'>  {dataPelanggaran.data.length}  </h1>  )}  </div>  <div className='p-4 bg-white shadow-md rounded-lg'>  <p className='text-sm font-medium text-black/40'>  Jumlah Siswa  </p>  {!loadingSiswa && (  <h1 className='font-semibold'>  {dataSiswa?.data?.length}  </h1>  )}  </div>  <div className='p-4 bg-white shadow-md rounded-lg'>  <p className='text-sm font-medium text-black/40'>  Jumlah User  </p>  {!loadingUser && (  <h1 className='font-semibold'>  {dataUser.data.length}  </h1>  )}  </div>  <div className='p-4 bg-white shadow-md rounded-lg'>  <p className='text-sm font-medium text-black/40'>  Jumlah Jurusan  </p>  {!loadingUser && <h1 className='font-semibold'>9</h1>}  </div>  </div>  {!loadingPelanggaran && (  <>  <div className='mt-10 mb-10 border rounded-lg p-5 w-full'>  <h3 className='font-medium mb-8 text-black/60'>  Chart Pelanggaran  </h3>  <Chart dataPelanggaran={dataPelanggaran} />  </div>  <Search />  <div className='mt-10 border rounded-lg p-5'>  <h3 className='font-medium mb-8 text-black/60'>  Tabel Pelanggaran  </h3>  <DashboardTable  refetch={refetchPelanggaran}  dataPelanggaran={dataPelanggaran.data}  />  </div>  </>  )}  </div> |

1. Untuk membuat tampilan tabel pelanggaran per jurusan (contoh jurusan RPL) buat folder dalam src/Components/Pelanggaran/RPL



Gambar 4. 11 Tabel Pelanggaran

1. Buat file RPLTable.jsx

|  |
| --- |
| <Table data={dataPelanggaran} columns={columns} />  <dialog id='confirm-delete-modal' className='modal'>  <div className='modal-box'>  <h3 className='font-bold text-lg text-[#E74C3C]'>Hapus Pelanggaran</h3>  <p className='py-4'>Yakin ingin menghapus data ini?</p>  <div className='modal-action'>  <form method='dialog'>  <Button className='btn !rounded-lg !me-4'>Tutup</Button>  <Button  onClick={() => deletePelanggaran(selectedId)}  className='btn !rounded-lg text-[#E74C3C]'  >  Hapus  </Button>  </form>  </div>  </div>  </dialog> |

1. Buat file index.jsx

|  |
| --- |
| <div className='mt-14 lg:mt-0 lg:p-4'>  <div className='grid grid-cols-1 sm:grid-cols-2 gap-6'>  <div className='p-4 bg-white shadow-md rounded-lg'>  <p className='text-sm font-medium text-black/40'>  Jumlah Pelanggaran  </p>  {!loadingPelanggaranRpl && (  <h1 className='font-semibold'>  {dataPelanggaranRpl.data.length}  </h1>  )}  </div>  <div className='p-4 bg-white shadow-md rounded-lg'>  <p className='text-sm font-medium text-black/40'>  Jumlah Siswa  </p>  {siswaRpl && (  <h1 className='font-semibold'>  {siswaRpl.length}  </h1>  )}  </div>  </div>  <div className='mt-10 border rounded-lg p-5'>  <h3 className='font-medium mb-8 text-black/60'>  Tabel Pelanggaran RPL  </h3>  {  !loadingPelanggaranRpl && (  <RplTable  refetch={refetchPelanggaranRpl}  dataPelanggaran={dataPelanggaranRpl.data}  />  )  }  </div>  </div> |

1. Buatlah file route untuk menampilkan tampilan kode diatas dalam folder src/router/index.jsx

|  |
| --- |
| <Route index element={<Dashboard />} />  <Route path='/pelanggaran/rpl' element={<Rpl />} />  <Route path='/pelanggaran/iop' element={<Iop />} />  <Route path='/pelanggaran/sija' element={<Sija />} />  <Route path='/pelanggaran/tek' element={<Tek />} />  <Route path='/pelanggaran/tei' element={<Tei />} />  <Route path='/pelanggaran/toi' element={<Toi />} />  <Route path='/pelanggaran/pspt' element={<Pspt />} />  <Route path='/pelanggaran/tptu' element={<Tptu />} />  <Route path='/pelanggaran/meka' element={<Meka />} />  </Route>  {/\* route untuk halaman yang belum punya token / anonymous \*/}  <Route element={<NoAuthorizeLayout />}>  <Route path={'/login'} element={<Login />} />  <Route path={'/register'} element={<Register />} />  </Route> |

## Pengujian Sistem Mobile

### Pengujian Login Admin

Tabel 4. 1 Pengujian Login Admin

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Memasukkan email salah | Tidak masuk ke dalam halaman dashboard | Berfungsi |
| Memasukkan password salah | Tidak masuk ke dalam halaman dashboard | Berfungsi |
| Memasukkan emal dan password yang kosong | Tidak masuk ke dalam halaman dashboard | Berfungsi |
| Memasukkan email dan password yang benar | Masuk ke halaman dashboard | Berfungsi |

### Pengujian Halaman Dashboard

Tabel 4. 2 Pengujian Halaman Dashboard

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Menekan tombol tambah pelanggaran | Berpindah ke halaman tambah pelanggaran | Berfungsi |
| Menekan tombol history pelanggaran | Berpindah ke halaman history pelanggaran | Berfungsi |

### Pengujian Halaman Tambah Data Pelanggaran

Tabel 4. 3 Pengujian Halaman Tambah Data Pelanggaran

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Memasukkan data pelanggaran yang kosong | Data tidak tersimpan | Berfungsi |
| Memasukkan data pelanggaran yang benar | Data tersimpan | Berfungsi |

### Pengujian Halaman History

Tabel 4. 4 Pengujian Halaman History

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Data history pelanggaran tampil atau tidak | Data history pelanggaran tampil | berfungsi |

## Pengujian Sistem Web

### Pengujian Login Superadmin

Tabel 4. 5 Pengujian Login Superadmin

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Memasukkan email salah | Tidak masuk ke dalam halaman dashboard | Berfungsi |
| Memasukkan password salah | Tidak masuk ke dalam halaman dashboard | Berfungsi |
| Memasukkan emal dan password yang kosong | Tidak masuk ke dalam halaman dashboard | Berfungsi |
| Memasukkan email dan password yang benar | Masuk ke halaman dashboard | Berfungsi |

### Pengujian Halaman Dashboard

Tabel 4. 6 Pengujian Halaman Dashboard

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Data pelanggaran tampil atau tidak | Data pelanggaran tampil | Berfungsi |
| Grafik pelanggaran tampil atau tidak | Grafik pelanggaran tampil | Berfungsi |

### Pengujian Data Perjurusan

Tabel 4. 7 Pengujian Data Perjurusan

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Menampilkan halaman saat menekan jurusan pada sidebar | Data per jurusan tampil | Berfungsi |

### Pengujian Hapus Data Pelanggaran

Tabel 4. 8 Pengujian Hapus Data Pelanggaran

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Menghapus data pelanggaran | Data yang duharapkan terhapus | Berfungsi |

### Pengujian Sistem Logout

Tabel 4. 9 Pengujian Sistem Logout

|  |  |  |
| --- | --- | --- |
| **Pengujian** | **Hasil yang diharapkan** | **Keterangan** |
| Berhasil logout dari website | Logout sukes dan kembali ke halaman login | Befungsi |

# PENUTUP

## Kesimpulan

Aplikasi pelanggaran siswa berbasis web dan mobile berhasil memberikan solusi efektif untuk pengelolaan pelanggaran di lingkungan pendidikan. Integrasi platform meningkatkan aksesibilitas, memberikan keuntungan administratif, dan meningkatkan transparansi. Meskipun menghadapi tantangan seperti validasi data dan keamanan, langkah-langkah keamanan diterapkan dengan sukses. Aplikasi diharapkan memberikan kontribusi positif terhadap pengelolaan disiplin dan meningkatkan kualitas pendidikan, dengan potensi pengembangan lebih lanjut melalui umpan balik pengguna.

## Saran

Pembuatan Aplikasi Pelanggaran Siswa Berbasis Android dan Web tentunya masih dapat dikembangkan lagi. Adapun yang bisa dikembangkan dari aplikasi ini antara lain:

1. Fitur dalam aplikasi masih dapat dikembangkan dengan menambahkan beberapa fitur seperti mencetak langsung data pelanggaran menjadi dokumen.
2. Cakupan penggunaan aplikasi ke depannya dapat digunakan di lebih dari satu tempat destinasi.

# DAFTAR PUSTAKA

Agus Suratna(2021). *Tutorial Android Studio : #2 Cara Install Android Studio di Windows.* Dari [https://agussuratna.net/2021/02/13/pengenalan-dan- instalasi-android-studio/](https://agussuratna.net/2021/02/13/pengenalan-dan-%09instalasi-android-studio/), 27 November 2023.

Alexandromeo Lawrence(2020). *API: Pengertian, Fungsi, dan Cara Kerjanya*.  
 Dari <https://www.niagahoster.co.id/blog/api-adalah/>, 27 November 2023.

Amelia Shinta(2022). *Apa itu XML? Pengertian, Contoh & Cara Membuka XML.*

Dari <https://www.dewaweb.com/blog/apa-itu-xml/>, 27 November 2023.

Feradhita(2019). *Kelebihan Android Studio dalam Pengembangan Aplikasi Android.* Dari [https://www.logique.co.id/blog/2019/05/03/kelebihan- android-studio/](https://www.logique.co.id/blog/2019/05/03/kelebihan-%09android-studio/), 27 November 2023.

Tekno(2020).  *Pengertian Keunggulan & Manfaat Mobile Apps.*

Dari <https://majapahit.id/blog/2021/03/17/apa-itu-mobile-apps/>, 27 November 2023.

Novriadi(2019). *Apa Itu Pengertian XML: Sejarah, Manfaat, Tipe, & Keunggulan.* Dari <https://www.novriadi.com/apa-itu-pengertian-xml/>, 23 Februari 2023

Oriza(2022). *Mengenal Android Studio : Pengertian, Manfaat, fitur, dan Cara Install.* Dari [https://idmetafora.com/news/read/701/Mengenal-Android- Studio-Pengertian-Manfaat-fitur-dan-Cara-Install.html](https://idmetafora.com/news/read/701/Mengenal-Android-%09Studio-Pengertian-Manfaat-fitur-dan-Cara-Install.html),

27 November 2023.

Serba Serbi, *Technology*(2020).

*Mengenal Apa itu Figma : Fitur, Fungsi, Cara Kerja / Menggunakannya.*

Dari <https://idcloudhost.com/mengenal-apa-itu-figma-fitur-fungsi-cara-kerja-menggunakannya/>, 29 November 2023.

Yudhistira(2023). *Urutan Versi Android dari Awal Hingga Android 13.*

Dari <https://www.bhinneka.com/blog/urutan-android/>, 29 November 2023.

Faradilla(2023). *Apa itu react ? Pengertian dan cara kerja ReactJs.*

Dari <https://www.hostinger.co.id/tutorial/apa-itu-react>, 6 Desember 2023.

Alexandromeo(2022). *Apa itu PHPMyAdmin? Pengertian, Fungsi, dan -*

*Cara Menggunakannya*. Dari  [https://makinrajin.com/blog -/phpmyadmin-adalah/](%20https://makinrajin.com/blog%20-/phpmyadmin-adalah/). 6 Desember 2023.