

Perancangan Aplikasi Mobile Pengelolaan Data Mahasiswa Berbasis Kodular Dengan Integrasi Airtable

Muhamad Alda^{1*}, Aji dewo Pangestu², M Arif Rahmat Pasaribu³, Indra Putra Mahayudi⁴, Umar Muzammil⁵

^{1,2,3,4,5}Program Studi Sistem Informasi, Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara

*muhamadalda@uinsu.ac.id, ²ajidewo273@gmail.com, ³rahmatpsb12@gmail.com, ⁴mahayudi1@gmail.com,
⁵umarmuzammil168@gmail.com

Abstract

Manual student data management often leads to data duplication, delayed updates, and limited access to information, which reduces the effectiveness of academic administration. This research aims to design and implement a mobile-based student data management application using the no-code platform Kodular integrated with Airtable as a cloud database to support efficient digital data administration. The Waterfall methodology was applied, consisting of requirement analysis, system design, implementation, testing, and maintenance. The system was developed as a single-screen application equipped with input fields for student ID (NIM), name, major, and year of entry, as well as CRUD (Create, Read, Update, Delete) and Refresh functions connected to Airtable through API communication. Dummy student data were used during testing. The black-box testing results show that all functional features operate correctly and the application can synchronize data quickly and responsively. Therefore, this application is considered feasible as a digital solution to improve the efficiency of academic administration and serves as a reference for developing simple information systems using no-code platforms in educational environments.

Keywords: kodular ,airtable ,waterfall ,mobile pplication,student data.

Abstrak

Pengelolaan data mahasiswa pada sebagian institusi pendidikan masih dilakukan secara manual, sehingga sering menimbulkan duplikasi data, keterlambatan pembaruan informasi, dan rendahnya efisiensi administrasi akademik. Untuk mengatasi permasalahan tersebut, penelitian ini bertujuan merancang dan mengimplementasikan aplikasi mobile berbasis platform no-code Kodular yang terintegrasi dengan Airtable sebagai cloud database untuk mendukung pengelolaan data mahasiswa secara digital. Metode pengembangan yang digunakan adalah Waterfall melalui tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Sistem dibangun dengan rancangan *single screen application* yang menyediakan form input NIM, nama, jurusan, dan angkatan, serta dilengkapi fitur CRUD (Create, Read, Update, Delete) dan Refresh yang terhubung ke Airtable melalui API. Data yang digunakan dalam pengujian berupa data dummy mahasiswa yang diinput secara langsung ke dalam sistem. Pengujian menggunakan metode *black-box* menunjukkan bahwa seluruh fungsionalitas berjalan sesuai spesifikasi dan aplikasi mampu melakukan sinkronisasi data dengan cepat dan responsif. Berdasarkan hasil tersebut, aplikasi dinilai layak sebagai solusi digital untuk peningkatan efektivitas administrasi akademik serta dapat dijadikan model pengembangan sistem informasi sederhana berbasis platform no-code di lingkungan pendidikan.

Kata kunci: kodular ,airtable ,waterfall ,aplikasi mobile ,data mahasiswa.

©This work is licensed under a Creative Commons Attribution - ShareAlike 4.0 International License

1. Pendahuluan

Pengelolaan data mahasiswa yang masih banyak dilakukan secara manual atau terserak pada berbagai file/lembar kerja menimbulkan permasalahan operasional seperti duplikasi data, keterlambatan update, dan kesulitan akses informasi bagi staf maupun mahasiswa. Kondisi ini mengurangi efisiensi administrasi akademik[1] serta meningkatkan potensi kesalahan pencatatan yang berdampak pada layanan pendidikan[2]. Oleh karena itu, pengembangan sistem informasi akademik berbasis mobile yang mudah diakses dan dikelola menjadi solusi praktis untuk meningkatkan kualitas layanan dan efektivitas administrasi. Studi tentang platform low-code/no-code[3][4] menegaskan bahwa pendekatan ini mempercepat proses pengembangan aplikasi dan memungkinkan partisipasi pengembang non-teknis dalam pembuatan prototipe fungsional.

Dalam penelitian ini dipilih kombinasi Kodular sebagai platform front-end no-code untuk membangun aplikasi mobile dan Airtable sebagai layanan cloud database untuk menyimpan serta mengelola data mahasiswa[5][6]. Kodular memungkinkan perancangan antarmuka dan logika aplikasi tanpa penulisan kode yang kompleks, sedangkan Airtable menyediakan struktur tabel relasional yang mudah diakses lewat API sehingga integrasi CRUD (Create, Read, Update, Delete) menjadi lebih sederhana dan cepat diimplementasikan. Sistem yang dirancang berupa sebuah *single screen application* yang menampung form input data mahasiswa[7] (NIM, nama, jurusan melalui spinner dengan opsi Sistem Informasi dan Ilmu Komputer, serta angkatan) serta tombol operasi utama: Input, Edit, Delete, Refresh, dan View; seluruh operasi ini berkomunikasi dengan Airtable sebagai backend. Pendekatan arsitektur ini dipilih untuk meminimalkan kompleksitas navigasi dan

mempercepat alur kerja pengguna pada perangkat mobile.

Metode pengembangan yang digunakan adalah model Waterfall, karena ruang lingkup fitur relatif terdefinisi dan tahapan pengerjaan dapat dijalankan secara berurutan: analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Waterfall memudahkan dokumentasi teknis dan evaluasi fungsional aplikasi—khususnya untuk proyek pengembangan skala kecil hingga menengah seperti prototipe sistem administrasi mahasiswa ini[8][9]. Tujuan penelitian adalah merancang dan mengimplementasikan aplikasi mobile pengelolaan data mahasiswa berbasis Kodular dan Airtable, serta mengevaluasi kelayakan fungsionalnya melalui pengujian black-box pada operasi CRUD dan mekanisme refresh/view data. Kontribusi yang diharapkan adalah tersedianya prototipe aplikasi yang praktis untuk administrasi akademik dan pedoman implementasi integrasi Kodular–Airtable bagi institusi pendidikan.

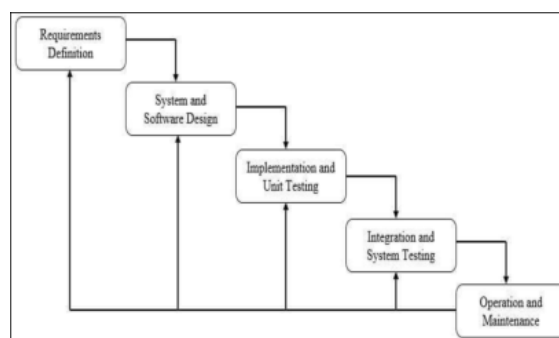
2. Metode Penelitian

Penelitian ini menggunakan metode pengembangan perangkat lunak Waterfall, yang terdiri dari lima tahapan utama: analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Model ini dipilih karena sesuai dengan karakteristik penelitian pengembangan aplikasi dengan ruang lingkup fitur yang telah terdefinisi dengan jelas serta memerlukan dokumentasi yang sistematis di setiap tahapannya.

Tabel 1. Analisis Kebutuhan

| No. | Jenis Kebutuhan | Deskripsi |
|-----|--------------------------|---|
| 1 | Input Data | Sistem dapat melakukan input data mahasiswa (NIM, Nama, Jurusan, Angkatan). |
| 2 | Edit Data | Sistem dapat memperbarui data mahasiswa yang sudah tersimpan. |
| 3 | Delete Data | Sistem dapat menghapus data mahasiswa dari database. |
| 4 | View Data | Sistem dapat menampilkan seluruh data mahasiswa dari Airtable. |
| 5 | Refresh Data | Sistem dapat memuat ulang data dari Airtable agar menampilkan data terbaru. |
| 6 | Kebutuhan Non-Fungsional | Aplikasi berjalan pada Android, terhubung internet, dan memiliki |

antarmuka sederhana.



Gambar 1. Metode Waterfall

2.1. Analisis Kebutuhan

Tahap ini bertujuan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional dari sistem yang akan dibangun[10]. Berdasarkan hasil analisis, aplikasi dirancang untuk membantu pengelolaan data mahasiswa yang meliputi penyimpanan, pembaruan, penghapusan, serta penampilan data.

Kebutuhan fungsional sistem meliputi:

1. Input data mahasiswa (NIM, Nama Mahasiswa, Jurusan, dan Angkatan).
2. Edit data mahasiswa yang sudah tersimpan.
3. Hapus data mahasiswa dari database.
4. Refresh untuk memperbarui tampilan data dari server.
5. View untuk menampilkan data mahasiswa secara keseluruhan.
6. Kebutuhan non-fungsional meliputi:
7. Aplikasi dapat dijalankan pada perangkat Android.
8. Data tersimpan pada layanan cloud database Airtable.
9. Antarmuka pengguna sederhana dan mudah digunakan oleh admin atau operator akademik.

2.2. Perancangan Sistem

Tahap ini mencakup desain arsitektur sistem, antarmuka pengguna, serta rancangan database. Aplikasi dirancang menggunakan satu layar utama (single screen) dengan form input berisi komponen TextBox untuk NIM, nama mahasiswa, dan angkatan, serta Spinner untuk memilih jurusan (opsi: Sistem Informasi dan Ilmu Komputer). Terdapat lima tombol utama yaitu: Input, Edit, Delete, Refresh, dan View yang masing-masing memiliki logika blok tersendiri di Kodular.

Pada sisi database, struktur tabel Airtable dirancang dengan field: NIM, Nama, Jurusan, dan Angkatan. Setiap field terhubung dengan API yang digunakan Kodular untuk melakukan operasi CRUD melalui Web Component[11]. Diagram arsitektur sistem menggambarkan alur komunikasi antara aplikasi Kodular dengan Airtable API menggunakan metode HTTP.

2.3. Implementasi

Tahap implementasi dilakukan dengan membangun aplikasi di platform Kodular. Setiap komponen visual (form, tombol, label, dan spinner) disusun sesuai rancangan antarmuka yang telah dibuat. Integrasi dengan Airtable dilakukan menggunakan Web Component di Kodular dengan memanfaatkan API Key dan Base ID yang diperoleh dari akun Airtable. Operasi CRUD[12] diimplementasikan melalui pemanggilan API dengan metode POST, PATCH, GET, dan DELETE[13] sesuai fungsi tombol. Contohnya, tombol Input mengirimkan data baru ke Airtable menggunakan metode POST, sementara tombol Edit memperbarui data berdasarkan NIM menggunakan PATCH. Tombol View menampilkan seluruh data mahasiswa dari tabel Airtable ke dalam daftar tampilan pada aplikasi.

2.4. Pengujian Sistem

Pengujian dilakukan untuk memastikan setiap fungsi dalam aplikasi berjalan sesuai dengan kebutuhan yang telah didefinisikan. Metode pengujian yang digunakan adalah black-box testing[14], yaitu pengujian berdasarkan keluaran (output) yang dihasilkan tanpa melihat kode program di baliknya.

Aspek yang diuji meliputi:

1. Fungsi input, edit, delete, refresh, dan view.
2. Koneksi dan sinkronisasi data antara aplikasi Kodular dengan database Airtable.
3. Kecepatan respon dan stabilitas aplikasi saat digunakan.

Hasil pengujian dinyatakan berhasil apabila setiap tombol menghasilkan keluaran sesuai dengan fungsinya dan data pada Airtable berubah sesuai operasi yang dilakukan.

2.5. Pemeliharaan

Tahap pemeliharaan dilakukan setelah aplikasi berhasil diuji dan dinyatakan berfungsi dengan baik. Kegiatan yang dilakukan meliputi perbaikan apabila ditemukan bug, penyesuaian pada tampilan antarmuka, serta pembaruan API Key atau struktur tabel Airtable jika diperlukan[15]. Tahap ini juga mencakup evaluasi performa sistem serta identifikasi fitur tambahan yang mungkin dikembangkan di masa mendatang, seperti sistem login pengguna, laporan otomatis, atau ekspor data ke format spreadsheet.

3. Hasil dan Pembahasan

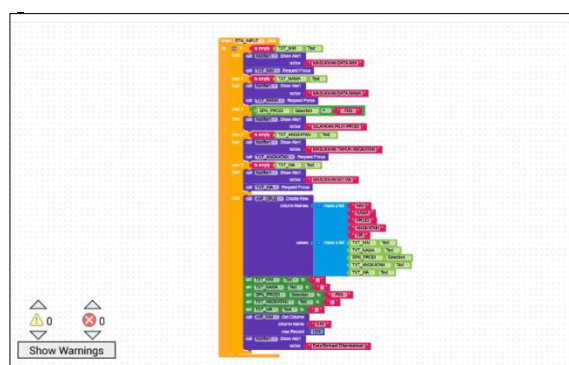
3.1. Tampilan Proses Input



Gambar 2. Tampilan Proses Input.

Ketika user sudah menginput data, maka akan keluar notifikasi “Data Berhasil Ditambahkan”. Tahap input data dalam pengembangan aplikasi mobile merupakan langkah krusial yang menjadi gerbang awal masuknya informasi ke dalam sistem. Dalam kegiatan praktikum ini, mahasiswa dikenalkan pada konsep dasar pengambilan data dari tampilan antarmuka, pemrosesan, hingga pengiriman ke basis data daring menggunakan platform Kodular dan Airtable. Melalui pendekatan visual ini, mahasiswa tidak hanya mempelajari cara merancang antarmuka yang interaktif, tetapi juga memahami dasar-dasar pertukaran data menggunakan API serta struktur penyimpanan data berbasis cloud.

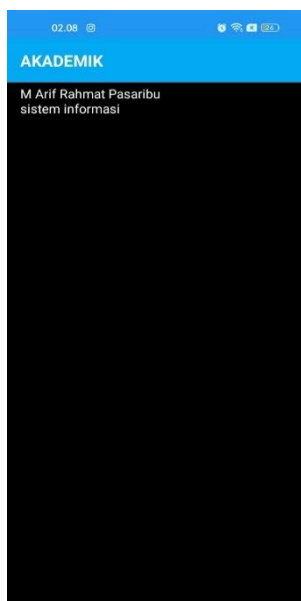
3.2. Tampilan Blok Input



Gambar 3. Tampilan Proses Input

Gambar diatas merupakan Block Input yang digunakan untuk mengambil atau menerima data dari pengguna melalui komponen antarmuka pengguna (UI) seperti TextBox, Button, Spinner, dan sebagainya. Sesuai dengan kebutuhan aplikasi sederhana Pendataan Mahasiswa.

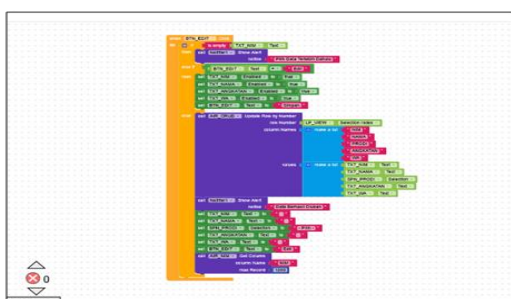
3.2 Tampilan View



Gambar 4. Tampilan View.

Gambar di atas merupakan tampilan *View* pada aplikasi mobile Pendataan Mahasiswa yang berfungsi untuk menampilkan data mahasiswa yang telah tersimpan dalam database *Airtable*. Informasi yang ditampilkan meliputi nama dan jurusan mahasiswa yang ditarik secara langsung dari server melalui pemanggilan *API*. Tampilan ini dibuat dengan antarmuka yang sederhana dan mudah dipahami sehingga memudahkan pengguna dalam melakukan pemantauan dan pengecekan data secara cepat. Selain itu, fitur *View* juga menjadi tahap verifikasi awal bagi pengguna sebelum melakukan proses lanjutan seperti *Edit* atau *Delete* data. Dengan adanya fitur ini, pengguna dapat memastikan bahwa data yang ditampilkan sesuai dengan data terbaru yang tersimpan pada database, sehingga mendukung kelancaran proses administrasi akademik dalam aplikasi.

3.4. Tampilan blok View

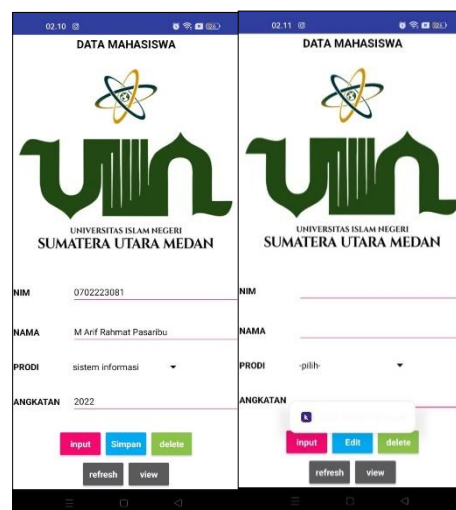


Gambar 5. Blok View

Gambar tersebut menampilkan kumpulan blok pemrograman di Kodular yang digunakan untuk mengatur logika sebuah aplikasi, kemungkinan berupa form input data ke *Airtable*. Blok-blok berwarna

oranye, ungu, hijau, dan biru mewakili berbagai fungsi seperti pengambilan data, pengecekan kondisi, serta penyimpanan ke database. Terlihat ada proses pengecekan input (seperti NIM atau data identitas), pengiriman data ke *Airtable*, dan pemberian notifikasi atau pesan tertentu kepada pengguna saat proses berhasil atau gagal. Struktur ini menunjukkan alur logika yang mengatur validasi, koneksi, dan pengelolaan data dalam aplikasi tersebut.

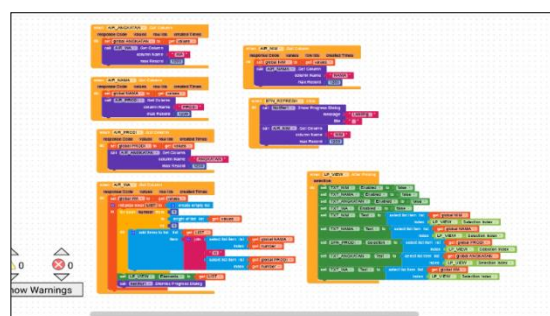
3.5. Tampilan Edit



Gambar 6. Tampilan Edit

Gambar diatas merupakan Tampilan Edit pada Aplikasi Pendataan Mahasiswa. Sebelum melakukan edit, kita harus memilih data dari botton view, kemudian mengganti data yang ingin kita ubah, setelah itu simpan kembali data. Maka data sudah berubah (ter edit).

3.6. Tampilan blok Edit



Gambar 7. Blok Edit

Gambar tersebut menampilkan blok program Kodular yang berfungsi untuk memperbarui (update) data di *Airtable* berdasarkan nomor atau ID tertentu. Blok oranye menunjukkan event utama yang memicu proses, sementara blok ungu mengatur koneksi dan perintah update ke database. Di dalamnya terdapat blok hijau yang berisi data dari komponen input (seperti nama, prodi, NIM, dan lainnya) yang akan dikirim ke *Airtable*. Secara keseluruhan, blok ini berfungsi untuk mengganti data lama dengan data baru pada baris

tertentu di Airtable ketika pengguna melakukan perubahan pada form aplikasi.

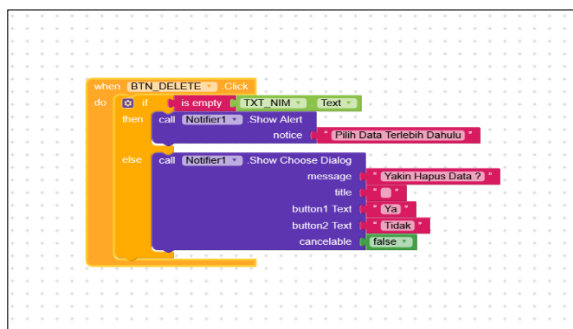
3.7. Tampilan Delete



Gambar 8. Tampilan Delete

Gambar diatas merupakan tampilan Proses Delete pada Aplikasi Pendataan Mahasiswa. Untuk menghapus data, pertama kita harus memilih data dari botton view kemudian memilih data yang ingin dihapus, setelah itu tekan botton Delete untuk menghapus data, Namun sebelum data terhapus akan muncul notifikasi yang berbunyi “Yakin Hapus Data? Ya atau Tidak”.

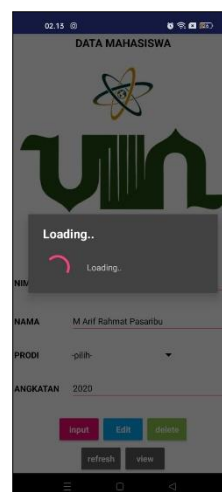
3.8. Tampilan blok Delete



Gambar 9. Block Delete

Gambar tersebut memperlihatkan blok program Kodular yang digunakan untuk mengatur aksi saat tombol Hapus (BTN_DELETE) diklik. Logikanya, jika kolom TXT_NIM masih kosong, maka sistem akan menampilkan peringatan melalui Notifier dengan pesan “Pilih Data Terlebih Dahulu”. Namun, jika kolom tersebut terisi, maka aplikasi akan menampilkan dialog konfirmasi dengan pertanyaan “Yakin Hapus Data?” dan dua pilihan tombol, yaitu “Ya” dan “Tidak”. Tujuan blok ini adalah untuk mencegah penghapusan data tanpa konfirmasi dan memastikan pengguna memilih data yang benar sebelum menghapusnya.

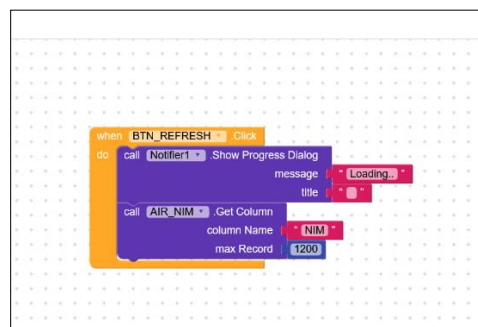
3.9. Tampilan Refresh



Gambar 10. Tampilan Refresh

Gambar diatas merupakan proses Refresh pada Aplikasi Mobile Sederhana Pendataan Mahasiswa. Untuk mengaktifkan Refresh kita harus menekan tombol Botton Refresh maka akan muncul pesan “Loading” untuk merefresh data mahasiswa agar menampilkan data terbaru.

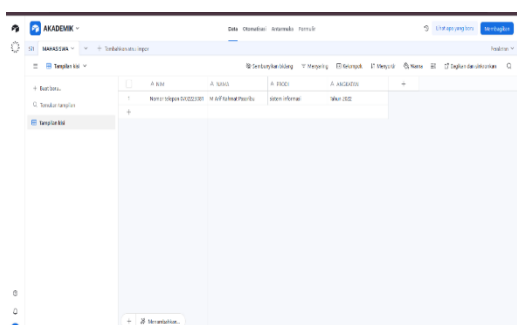
3.10. Tampilan Blok Refresh



Gambar 11. Block Refresh

Gambar tersebut menampilkan blok program Kodular yang mengatur fungsi tombol Refresh (BTN_REFRESH). Saat tombol diklik, aplikasi akan menampilkan pesan “Loading...” menggunakan Notifier untuk memberi tahu pengguna bahwa proses sedang berlangsung. Setelah itu, blok AIR_NIM.Get Column digunakan untuk mengambil data dari kolom “NIM” di Airtable, dengan batas maksimum 1200 record. Tujuan utama blok ini adalah untuk memperbarui atau memuat ulang daftar data NIM dari database Airtable agar aplikasi menampilkan data terbaru.

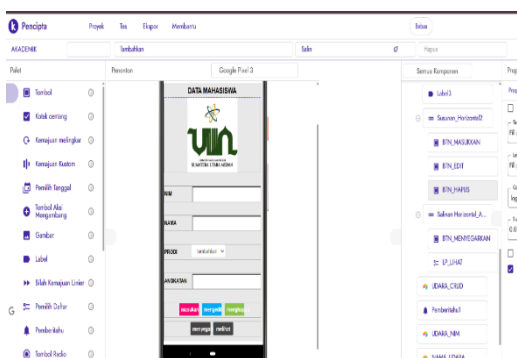
3.11. Tampilan database (airtable)



Gambar 12. Membuat Tabel Database

Tabel ini bisa diibaratkan seperti lembar kerja dalam Excel, dan Anda bisa mengganti nama tabel sesuai kebutuhan, misalnya “AKADEMIK”. Di dalam tabel tersebut, Anda dapat membuat dan mengatur kolom (fields) yang berfungsi sebagai struktur data.

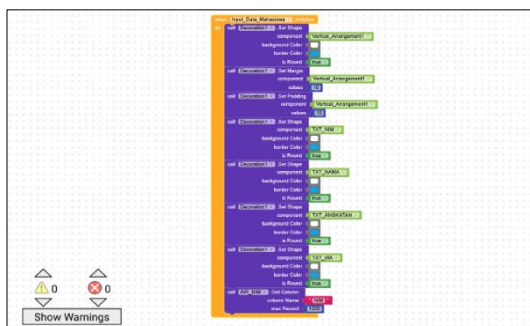
3.12. Tampilan User Interface



Gambar 13. Design User Interface

Pada gambar diatas, Anda akan melihat tiga bagian utama: komponen, area desain, dan properti. Untuk mulai membuat desain tampilan aplikasi, Anda dapat menambahkan komponen antarmuka pengguna (user interface) dari panel sebelah kiri ke layar desain.

3.13. Tampilan Design User Interface



Gambar 14. lock Design User Interface

Gambar diatas merupakan Block dekorasi yang berfungsi untuk mengubah tampilan pada Tabel Nim, Tabel Nama, Tabel Angkatan, dan Tabel WA, Sesuai dengan keinginan User.

4. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian, dapat disimpulkan bahwa aplikasi mobile pengelolaan data mahasiswa berbasis Kodular dengan integrasi Airtable berhasil dibangun dan berfungsi sesuai kebutuhan, terutama dalam menjalankan operasi CRUD serta fitur view dan refresh melalui API Airtable. Penggunaan platform no-code Kodular mempermudah proses pengembangan, sementara Airtable mendukung pengelolaan data secara daring dan terintegrasi, dengan metode Waterfall yang terbukti efektif melalui tahapan pengembangan yang sistematis. Hasil pengujian black-box dan white-box menunjukkan seluruh fungsi utama berjalan sesuai logika dan kebutuhan fungsional, termasuk validasi input, percabangan logika, serta penanganan kesalahan, sehingga aplikasi dinilai andal dan layak digunakan sebagai prototipe untuk mendukung efisiensi administrasi akademik di institusi pendidikan.

Daftar Rujukan

- [1] J. Nikukaran, A. Fallahzadeh, S. Hazhir, A. Hami, and S. A. Fatemi Aghda, “Design and evaluation of a mobile application for educational purposes and facilitating scientific literature search for students and researchers,” *BMC Med. Educ.*, vol. 25, no. 1, p. 1396, 2025, doi: 10.1186/s12909-025-07897-y.
- [2] M. Zulfikar and S. Putra, “Design and Development of a Student Assessment Application Based on Android Using the Waterfall Model,” *J. Informatics Manag. Inf. Technol.*, vol. 5, no. 3, pp. 302–310, 2025, doi: 10.47065/jimat.v5i3.694.
- [3] S. McHugh, N. Carroll, and C. Connolly, “Low-Code and No-Code in Secondary Education—Empowering Teachers to Embed Citizen Development in Schools,” *Comput. Sch.*, vol. 41, no. 4, pp. 399–424, 2024, doi: 10.1080/07380569.2023.2256729.
- [4] G. Lagogiannis and A. Chatzopoulos, “A No-Code Educational Platform for Introducing Internet of Things and Its Application to Agricultural Education,” *Internet of Things*, vol. 6, no. 3, pp. 1–26, 2025, doi: 10.3390/iot6030042.
- [5] Ira Murni, Putri Ratricia, and Muhammad Dedi Irawan, “Perancangan Aplikasi Pengelolaan Data Pegawai Berbasis Mobile Menggunakan Kodular dan Airtable,” *Bull. Comput. Sci. Res.*, vol. 4, no. 2, pp. 188–195, 2024, doi: 10.47065/bulletincsr.v4i2.335.
- [6] M. Alda, “Pengembangan Aplikasi Pengolahan Data Siswa Berbasis Android Menggunakan Metode Prototyping,” *J. Manaj. Inform.*, vol. 13, no. 1, pp. 11–23, 2023, doi: 10.34010/jamika.v13i1.8216.
- [7] M. Akbar and T. P. Ningsri, “Perancangan Aplikasi Input Laporan Data Industri Kecil Menengah Menggunakan Kodular dan Airtable Small and Medium Industry Data Report Input Application Design Using Kodular and Airtable,” *J. Comput. Sci. Informatics Eng.*, vol. 02, no. 3, pp. 105–115, 2023.
- [8] N. Ihza, D. Rahmawati, and S. Sukrim, “Sistem Informasi Aplikasi Kasir Berbasis Website Dengan Menggunakan Metode Waterfall,” *J. Ilm. Fak. Tek.*, vol. 3, no. 1, pp. 18–26, 2023, doi: 10.33592/jimtek.v3i1.3819.
- [9] Z. Ariza, “PERBANDINGAN MODEL WATERFALL DAN METODE PROTOTYPE UNTUK PENGEMBANGAN APLIKASI PADA SISTEM INFORMASI,” *J. Ilm. Metadata*, vol. 5, no. 1, pp. 83–95, 2023, doi: 10.62386/jised.v2i1.50.
- [10] S. Herawati, Y. D. P. Negara, H. F. Febriansyah, and D. A. Fatah, “Application of the Waterfall Method on a Web-Based Job Training Management Information System at Trunojoyo University Madura,” *E3S Web Conf.*, vol. 328, 2021, doi: 10.1051/e3sconf/202132804026.
- [11] F. Nurjanah and S. E. Y. Suprihatin, “The development of Android-based learning media using Kodular in making suit

- patterns subject,” *J. Pendidik. Vokasi*, vol. 13, no. 3, pp. 232–245, 2023, doi: 10.21831/jpv.v13i3.54542.
- [12] M. Alda, S. M. Tanjung, M. O. Abdillah, and L. Melsya, “Aplikasi Penginputan Data dan Nilai Siswa Pada SDS Nurul Huda Menggunakan Kodular dan Airtable,” vol. 8, no. 3, 2024.
- [13] M. A. A. W. Muhamad Alda, Ira Murni, Bagus Rizki Lubis, “Rancang Bangun Aplikasi Pengolahan Data Barang Inventaris Berbasis Mobile di Bidang Pemuda dan Olahraga,” *Reslaj Relig. Educ. Soc. Laa Roiba J.*, vol. 6, no. 3, pp. 950–959, 2024, doi: 10.47476/reslaj.v6i3.1483.
- [14] Suratno, A. Wahyudin, and D. Wahyudin, “Blackbox Testing: Teknik Equivalence Partitions untuk Menguji Aplikasi Monitoring Project-Based Learning,” *J. Pendidik. Tambusai*, vol. 9, no. 2614–3097, pp. 2190–2197, 2025.
- [15] P. Wibowo, M. F. A. Tarigan, and H. F. Siregar, “Implementasi Teknologi Mobile E-Digital dalam Meningkatkan Efisiensi Pengaduan Masyarakat,” vol. 3, pp. 1–10, 2025.