

LAPORAN FINAL PROJECT
MOBILE APLICATIONS FRAMEWORK PADA ADDICTIVE CLOTHING

Disusun Guna Memenuhi Ujian Akhir Semester IV
Mata Kuliah Workshop Mobile Apllications Framework

Dosen Pengampu:

Ahmad Fahriyanur Rosyady, S.Kom., M.MT.

Mochammad Rifki Ulil Albaab, S.T., M.Tr.T.



Disusun oleh:

Arfan Astaraja (E41212007)

Dinda Amalia Julyandri (E41212015)

Bilal Shandyarta Syamsudin (E41212037)

Lya Nurul Ulla (E41211962)

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

TAHUN 2023

KATA PENGANTAR

Puji syukur senantiasa kami panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan laporan final project ini dengan baik, guna untuk memenuhi salah satu syarat untuk dapat mengikuti ujian akhir semester, Laporan ini berisikan tentang *mobile applications framework* dari sistem penjualan pada addictive clothing untuk memenuhi mata kuliah Workshop.

Kami mengucapkan terima kasih kepada selaku dosen Mata Kuliah Workshop Mobile Applications Framework yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan sesuai dengan bidang studi. Terimakasih juga disampaikan kepada semua teman-teman anggota kelompok yang telah berpartisipasi dalam pengerjaan final project ini sehingga dapat terselesaikan.

Kami menyadari sepenuhnya bahwa laporan ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang kami miliki. Oleh karena itu, kami mengharapkan segala bentuk saran serta kritik yang membangun dari berbagai pihak. Kami berharap semoga laporan ini dapat bermanfaat bagi pembaca dan perkembangan dunia pendidikan.

Sidoarjo, Juni 2023

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iii
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
BAB II. TINJAUAN PUSTAKA	3
2.1 Aplikasi Mobile.....	3
2.2 Sistem Informasi Penjualan	3
2.3 Framework.....	3
2.4 Flutter	4
2.5 Digital Marketing	4
BAB III. METODOLOGI.....	5
3.1 Desain Sistem	5
3.2 <i>Flowchart</i>	6
3.3 ERD	7
BAB IV. PEMBAHASAN	8
4.1 Hasil Tampilan Aplikasi <i>Mobile</i>	8
3.2.1 Tampilan Login.....	8
3.2.2 Tampilan Register.....	8
3.2.3 Tampilan Home Pertama	9
3.2.4 Tampilan Home Kedua	10
3.2.5 Tampilan List Produk	10
3.2.6 Tampilan Detail Produk.....	11
3.2.7 Tampilan Checkout	12
3.2.8 Tampilan User Profile	12
BAB V. PENUTUP.....	14
5.1 Kesimpulan	14
5.2 Saran.....	15
DAFTAR PUSTAKA.....	16
LAMPIRAN.....	17

DAFTAR GAMBAR

Gambar 1. Desain Sistem	5
Gambar 2. Flowchart.....	6
Gambar 3. <i>ERD</i>	7
Gambar 4. <i>Login</i>	8
Gambar 5. <i>Register</i>	9
Gambar 6. <i>Home Pertama</i>	9
Gambar 7. <i>Home Kedua</i>	10
Gambar 8. <i>List Product</i>	11
Gambar 9. <i>Detail Product</i>	11
Gambar 10. <i>Checkout</i>	12
Gambar 11. <i>User Profile</i>	13
Gambar 12. <i>Database</i>	37
Gambar 13. Logo Mitra	37
Gambar 14. Pemilik Mitra	38
Gambar 15. Berdiskusi Pembuatan Aplikasi <i>Mobile</i>	38
Gambar 16. Berdiskusi Pembuatan <i>Website</i>	39

BAB I. PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi yang kini semakin pesat memacu kita untuk ikut terlibat secara langsung dalam pemanfaatannya, diciptakan sedemikian rupa bertujuan untuk mempermudah pekerjaan manusia dalam memperoleh informasi yang terperinci. Teknologi dengan cepat memberikan perubahan besar terhadap gaya hidup manusia saat ini. Salah satunya adalah semakin banyak pengguna *smartphone*, terutama yang berbasis *android*. Berbagai macam keunggulan dari sistem *android* adalah aplikasi sistem operasi yang didalamnya dapat diubah sesuai dengan keinginan kita sendiri dan banyak juga aplikasi komputer yang sudah tersedia untuk *smartphone android*. Sehingga teknologi ini memungkinkan masyarakat dapat dengan mudah untuk melakukan segala aktivitas (Setiaji 2020).

E-commerce adalah sistem jual beli yang berlangsung secara online, dimana pembeli berada tidak selalu harus datang ke toko untuk membeli produk. Saat ini e-commerce adalah alternatif untuk perusahaan yang berkomitmen khusus bidang kewirausahaan sebagai sarana informasi yang memudahkan interaksi penjual dan pembeli tanpa batasan ruang dan waktu. Didukung oleh pembangunan teknologi informasi menjadi lebih canggih dan lebih mudah tersedia, perkembangan perdagangan elektronik berkembang dan semakin diminati banyak perusahaan (Efriyanti, Garaika, and Irviani 2018).

Addictive Clothing merupakan suatu bisnis yang berada di kota Kediri, telah didirikan sejak tahun 2021 oleh beberapa remaja, yang menjual berbagai macam pakaian distro, tidak hanya itu *addictive clothing* juga melayani pemesanan Kaos lusinan, Kaos Jearsy, Sablon DTF, Sablon DTG, dan juga Sablon Plastisol. Akan tetapi, selama ini belum memiliki media khusus untuk mempromosikan produk-produknya dan konsumen masih harus langsung datang ke toko apabila ingin mendapatkan informasi produk dan memesan produk sesuai dengan desain yang diinginkan.

Berdasarkan kepentingan pemilik bisnis *Addictive Clothing* dengan melihat kemajuan teknologi informasi agar penjualan, pemesanan ataupun promosi produk menjadi lebih mudah dan dapat memperluas peluang penjualannya maka, muncul gagasan untuk membuat sistem informasi penjualan *online* yang diharapkan dapat memudahkan pelanggan dalam melihat produk yang tersedia, dan juga memudahkan pemesanan. Sehingga dapat membantu *Addictive Clothing* dalam meningkatkan penjualan bisnisnya (Aryanto and Tarigan 2015).

1.2 Rumusan Masalah

Dari latar belakang diatas, maka dapat dirumuskan masalah yang akan dibahas pada tugas akhir ini yaitu:

1. Bagaimana membangun sebuah sistem informasi penjualan berbasis *Mobile Applications Framework* untuk *Addictive Clothing Store* agar dapat mengembangkan bisnis usahanya?
2. Bagaimana merancang sistem informasi agar mempermudah konsumen untuk mendapatkan informasi produk terbaru dan melakukan transaksi tanpa harus mengunjungi toko?

1.3 Tujuan

Berdasarkan rumusan masalah diatas, maka tujuan dari tugas akhir ini sebagai berikut:

1. Merancang sistem informasi penjualan dan pemasaran berbasis *Mobile Applications Framework*
2. Membuat sistem informasi untuk membantu memperluas promosi bisnis sehingga mempermudah *branding Addictive Clothing*.
3. Membantu membuat sistem informasi ketersediaan pakaian untuk memudahkan pelanggan dalam pemesanan produk secara *detail*.

BAB II. TINJAUAN PUSTAKA

2.1 Aplikasi Mobile

Aplikasi *Mobile* adalah perangkat lunak yang bekerja pada perangkat seluler seperti *smartphone* atau *tablet*. aplikasi Mobile juga dikenal sebagai aplikasi penyimpanan dimuat dan memiliki fungsi tertentu meningkatkan fungsi perangkat ponsel itu sendiri menjadi ponsel aplikasi yang diinginkan, pengguna bisa Unduh melalui situs web yang diberikan dengan sistem operasi yang digunakan. GooglePlay dan iTunes adalah beberapa contohnya situs web yang menawarkan berbagai aplikasi untuk pengguna Android dan iOS Unduh aplikasi yang kita inginkan (Irsan 2015).

Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain tanpa terjadi pemutusan atau terputusnya komunikasi (Aulia 2023). *android* adalah sebuah sistem operasi untuk perangkat mobile berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

2.2 Sistem Informasi Penjualan

Sistem informasi penjualan merupakan salah satu pembuatan pernyataan penjualan atau kegiatan yang dijelaskan melalui prosedur-prosedur penjualan. Dengan adanya sistem informasi penjualan produk yang ditawarkan akan lebih dikenal. Selain itu informasi yang tersedia dapat dengan mudah dijangkau banyak orang, misalnya dalam pencarian data yang akan lebih cepat, karena data-data telah diorganisir dengan baik oleh sistem. Sehingga proses pengolahan data penjualan lebih cepat dan mudah (Jihan 2017).

2.3 Framework

Framework dalam konteks pengembangan perangkat lunak, menyediakan struktur, komponen, dan alat bantu untuk membangun sebuah aplikasi. Framework memberikan pendekatan atau aturan dalam pengembangan perangkat lunak yang membantu membangun aplikasi dengan lebih efisien.

Framework biasanya terdiri dari kumpulan komponen dan alat bantu yang ditujukan untuk memecahkan masalah umum dalam pengembangan perangkat lunak. Mereka menyediakan pondasi yang solid dan struktur yang terorganisir untuk membangun aplikasi, sehingga pengembang dapat fokus pada fitur yang ingin mereka kembangkan, dari pada menghabiskan waktu dan upaya untuk membangun semuanya dari awal (Al-Turany et al. 2012).

2.4 Flutter

Flutter adalah SDK untuk pengembangan aplikasi mobile dengan kinerja tinggi, aplikasi untuk iOS dan Android, dari satu codebase yang di buat oleh Google dengan lisensi open source. Dengan didukung oleh Google, flutter dapat berkembang sangat cepat dan dapat bersaing dengan Xamarin, React Native, dan framework mobile lainnya. Dengan flutter, aplikasi android dan IOS dapat dibuat menggunakan basis kode dan Bahasa pemrograman yang sama, yaitu dart, Bahasa pemrograman yang juga diproduksi oleh google pada tahun 2011. Sebelumnya, aplikasi murni native untuk android perlu dibuat menggunakan Bahasa Java atau Kotlin (Anon n.d.).

Untuk membuat aplikasi Flutter, diperlukan untuk mengerti bahasa Dart. Dart merupakan bahasa pemrograman yang dibuat oleh Google untuk menggantikan Javascript. Dart adalah sebuah bahasa pemrograman yang dikembangkan oleh Google dan merupakan bahasa pemrograman resmi untuk Flutter, sebuah UI toolkit dan aplikasi multiplatform dari Google. Flutter sendiri telah digunakan oleh berbagai perusahaan besar seperti Google, Alibaba.com, dan Tencent karena dapat menghemat waktu dan tenaga dengan cara cukup membutuhkan satu codebase untuk mengembangkan aplikasi di berbagai platform, daripada harus menghabiskan waktu untuk membuat codebase terpisah untuk masing-masing platform (Tjandra and Chandra 2020).

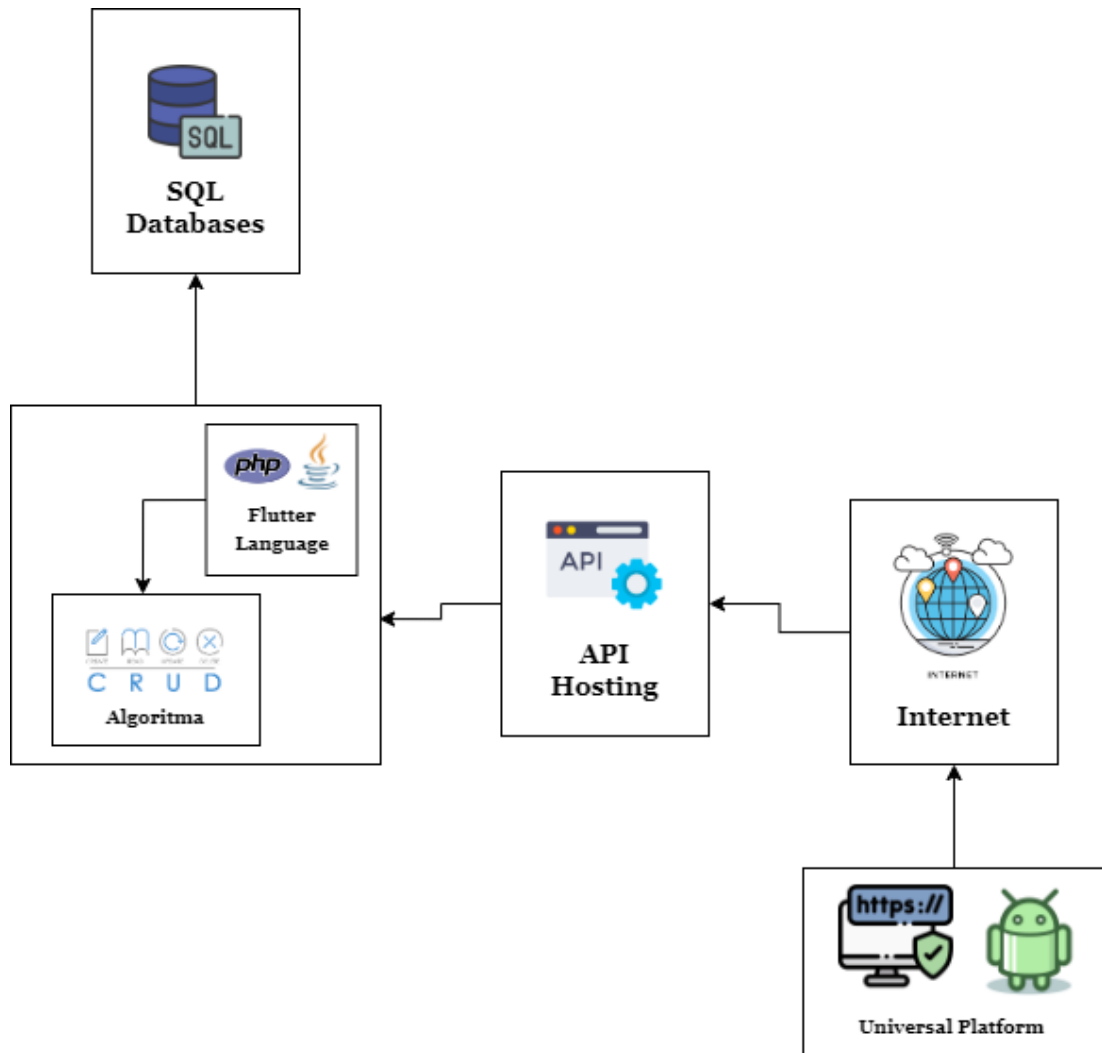
2.5 Digital Marketing

Dalam industri yang berkembang pesat dan kompetitif pada era seperti sekarang ini. Pemasaran yang massive dan active dibutuhkan agar dapat mendukung berbagai kegiatan bisnis baik besar maupun kecil agar dapat dikenal secara global. Adapun beberapa contoh teknik pemasaran yang termasuk dalam digital marketing dan membantunya untuk berkembang yaitu SEO (Search Engine Optimization). Dengan menggunakan media online, hasil dari kegiatan pemasaran dapat langsung diketahui. Informasi seperti berapa lama produk telah dilihat serta berapa banyak orang yang melihat produk. Setelah mengetahui informasi tersebut pengusaha dapat langsung melakukan evaluasi. Kelebihan digital marketing juga ialah memiliki jangkauan yang lebih luas dan dapat menyebarkan brand ataupun produk ke seluruh dunia hanya dengan memanfaatkan internet dan digital marketing (Rozinah and Meiriki 2020).

Perlu diketahui bahwa digital marketing adalah pilihan tepat untuk melakukan penjualan dan membantu usaha growth dan profitable dengan mudah. Digital marketing dapat menjangkau hampir semua pasar sesuai apa yang diinginkan. Digital marketing adalah metode yang tepat karena bisa menjangkau dan saling terhubung dengan pelanggan dari berbagai belahan dunia, kapan pun dan di mana pun. Hasilnya adalah jasa atau produk yang ditawarkan menjadi selalu tersedia serta dapat dibeli oleh pelanggan kapan saja (Pradiani 2017).

BAB III. METODOLOGI

3.1 Desain Sistem

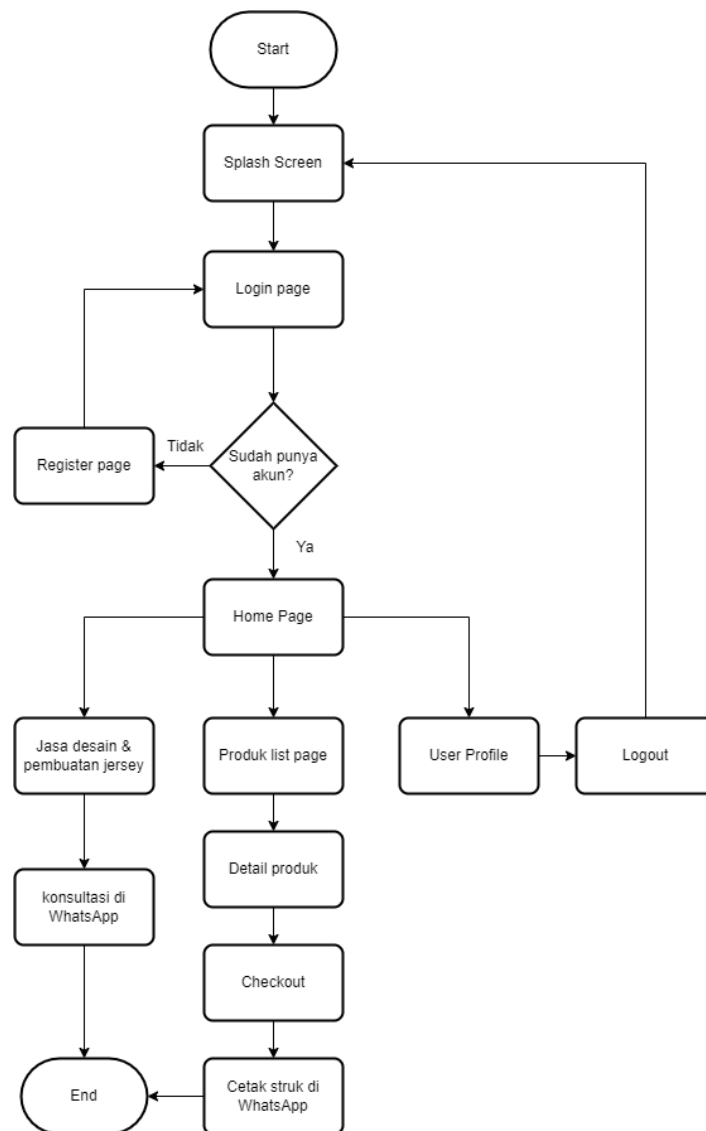


Gambar 1. Desain Sistem

Desain sistem dibawah menjelaskan Website dan Aplikasi Mobile terhubung ke internet dan mengambil API untuk mencapai integrasi antara keduanya. Hal ini memungkinkan pengguna untuk mengakses fitur dan data yang sama melalui platform yang berbeda. API Hosting berperan sebagai penghubung antara Website dan Aplikasi Mobile dengan menghasilkan algoritma CRUD (Create, Read, Update, Delete) yang terintegrasi dalam Aplikasi Mobile yang dikembangkan menggunakan Flutter. Algoritma CRUD ini memungkinkan pengguna untuk melakukan operasi dasar seperti membuat, membaca, memperbarui, dan menghapus data melalui aplikasi mobile. Namun untuk saat ini fitur mengupdate dan delete data hanya dapat dilakukan pada admin website

Hasil dari operasi CRUD yang dilakukan melalui admin pada Website akan mengubah data yang ada dalam SQL database. Ini berarti bahwa ketika admin melakukan perubahan data melalui antarmuka admin di Website, data yang terkait dalam database SQL akan diperbarui sesuai dengan perubahan yang dilakukan.

3.2 Flowchart

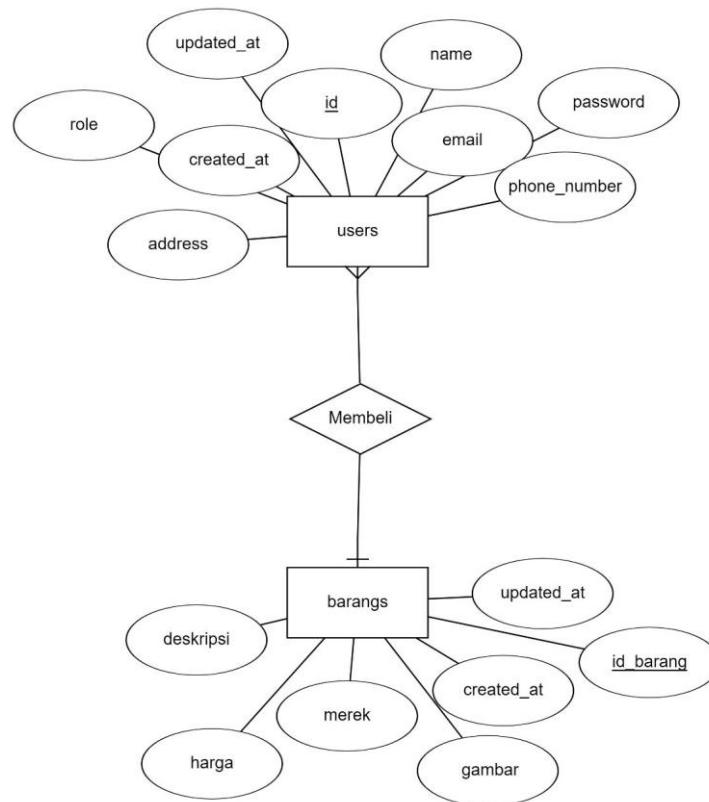


Gambar 2. Flowchart

Berikut adalah penjelasan singkat flowchart aplikasi *Addictive Clothing*:

1. *Start*: Aplikasi dimulai.
2. *Splash screen*: Tampilan awal dengan logo "Addictive Clothing".
3. *Login*: Masuk menggunakan akun dan kata sandi.
4. Jika belum punya akun: Daftar terlebih dahulu.
5. *Register*: Isi informasi untuk membuat akun baru.
6. Setelah login: Menu utama.
7. Menu *Home*: Ada 3 pilihan menu:
 - *User Profile*: Lihat data profil dan Logout akun.
 - Jasa Sablon dan Jersey: Konsultasi dan ajukan permintaan desain.
 - Produk List: Lihat daftar produk distro.
8. Pre-order produk: Hubungi admin melalui WhatsApp.
9. *End*: Selesai.

3.3 ERD



Gambar 3. ERD

ERD yang dibuat untuk sistem ini terdiri dari dua entitas utama: users dan barangs. Setiap entitas memiliki atribut-atribut yang relevan untuk menyimpan informasi yang diperlukan. Entitas users memiliki atribut seperti id, name, email, phone_number, address, role, created_at, dan updated_at. Atribut-atribut ini mencakup informasi tentang pengguna, seperti ID pengguna, nama, alamat email, nomor telepon, alamat, peran dalam sistem, serta tanggal dan waktu pembuatan dan pembaruan data pengguna.

Sementara itu, entitas barangs memiliki atribut seperti id_barang, gambar, merek, harga, deskripsi, created_at, dan updated_at. Atribut-atribut ini digunakan untuk menyimpan informasi tentang barang yang tersedia dalam sistem. Mulai dari ID barang, gambar, merek, harga, deskripsi, serta tanggal dan waktu pembuatan dan pembaruan data barang.

Selain atribut-atribut yang terdefinisi, ERD ini juga menunjukkan hubungan antara entitas users dan barangs. Hubungan yang terbentuk adalah "membeli" dengan relasi one-to-one, yang berarti setiap pengguna memiliki satu barang yang mereka beli. Hubungan ini memungkinkan pengguna untuk memiliki kepemilikan tunggal terhadap barang tertentu. ERD yang Anda buat memberikan pemodelan yang sederhana dan jelas tentang hubungan antara pengguna dan barang dalam sistem. Dengan informasi yang disimpan dalam atribut-atribut entitas, sistem dapat melakukan penelusuran data dan memfasilitasi proses transaksi antara pengguna dan barang dengan efisien.

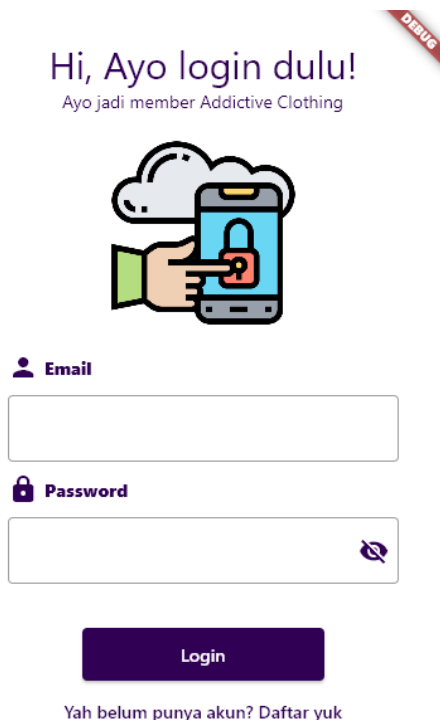
BAB IV. PEMBAHASAN

4.1 Hasil Tampilan Aplikasi *Mobile*

Berdasarkan hasil dari *running* sistem aplikasi *mobile* yang telah dibuat maka diperoleh tampilan-tampilan pada halaman aplikasi seperti pada gambar dibawah ini.

3.2.1 Tampilan Login

Form login adalah induk yang digunakan untuk keamanan agar pengguna lain tidak dapat masuk ke dalam halaman manipulasi. Pada menu ini akan aktif setelah pengoneksian *database* berhasil. Pada gambar Login dibawah tampilan untuk masuk ke dalam sistem, disini pengguna dapat memasukan *email* dan *password*. Dan akan dicek kedalam *database* apakah sesuai atau tidak jika sesuai maka akan langsung menuju halaman menu utama. Apabila belum pengguna belum memiliki akun untuk login maka harus melakukan registrasi terlebih dahulu dengan menekan *text* Daftar yuk akan langsung menuju halaman *register*.



Hi, Ayo login dulu!

Ayo jadi member Addictive Clothing

Email

Password

Login

Yah belum punya akun? Daftar yuk


Gambar 4. *Login*


3.2.2 Tampilan Register


Gambar 4 merupakan halaman registrasi, Pada halaman ini pengguna harus mengisi form-form beberapa identitas seperti email, nama lengkap, nomor telepon, dan alamat pengguna. Data tersebut akan tersimpan dalam *database* sehingga pengguna bisa langsung melakukan login. Data-data yang sudah diisi tersebut akan muncul pada profil pengguna.


Ayo daftar sekarang!


Isi datanya dengan benar ya



 **Email**

 **Nama Lengkap**

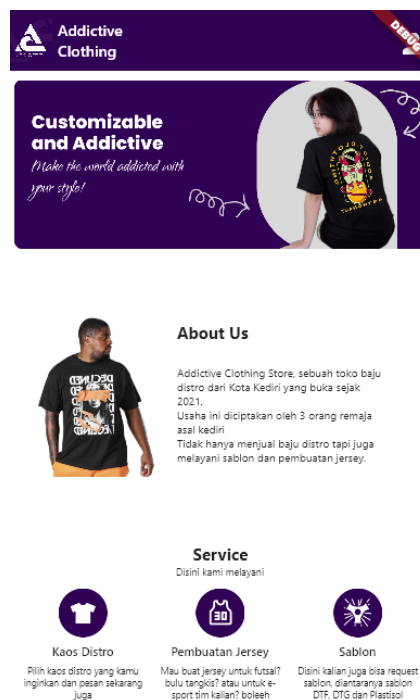
 **Nomor Telepon**

 **Alamat**

Gambar 5. Register

3.2.3 Tampilan Home Pertama


Pada halaman ini terdapat logo mitra pada pojok kiri atas, icon button pada kanan atas, terdapat informasi tentang *Addictive Clothing* dan juga terdapat beberapa layanan yang ada pada *Addictive Clothing* seperti kaos distro, pembuatan jersy, custom sablon.



Gambar 6. Home Pertama

3.2.4 Tampilan Home Kedua

Pada halaman berikut berisi tentang produk dari mitra yang ketika diklik menuju list produk dan terdapat button jasa sablon dan pembuatan jersay ketika diklik akan langsung menghubungi admin melalui *WhatsApp* untuk pemesan produk lebih detailnya. Terdapat juga ukuran baju yang digunakan *Addictive Clothing* seperti panjang dan lebarnya baju.

 Addictive Clothing

Produk Kami



T Shirt Addict Clothing V.3
Rp 135.000

Klik Disini Untuk Jasa Sablon & Jersey

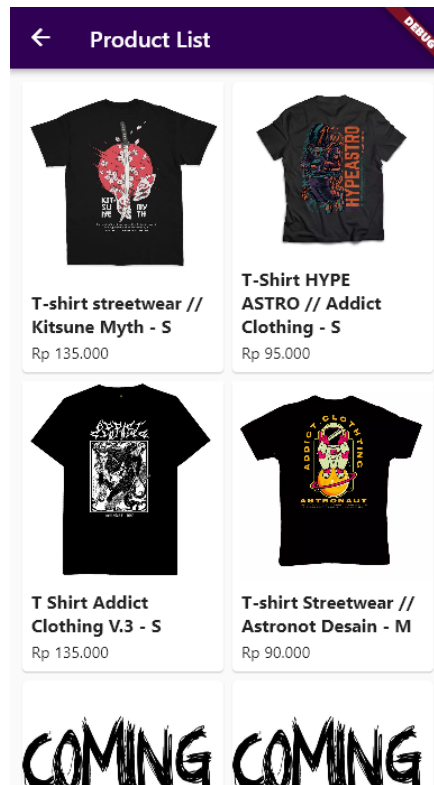
SIZE CHART

S	L: 47cm P: 68cm
M	L: 49cm P: 70cm
L	L: 51cm P: 72cm
XL	L: 53cm P: 74cm
XXL	L: 55cm P: 76cm
XXXL	L: 57cm P: 78cm

Gambar 7. Home Kedua

3.2.5 Tampilan List Produk

Halaman list produk berisi seluruh produk pakaian distro yang dijual *addictive clothing* terdapat berbagai jenis pakaian yang terseedia dan juga harga sehingga calon pembeli lebih mudah untuk memilih produk yang ingin dipesan. Pada list produk tersebut juga bisa diklik untuk menuju halaman detail produk.



Gambar 8. List Product

3.2.6 Tampilan Detail Produk

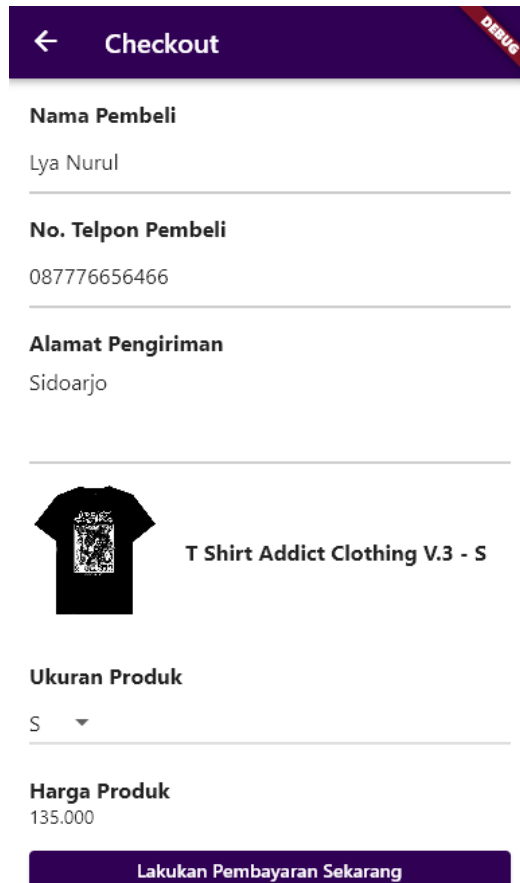
Pada halaman detail produk berisi tentang keterangan baju seperti warna baju, bahan baju yang digunakan, jenis sablon pada baju, metode pembayaran, dan pengiriman dan juga terdapat button *pre order* untuk melakukan pembelian.



Gambar 9. Detail Product

3.2.7 Tampilan Checkout

Pada halaman *checkout* berikut terdapat form-form yang wajib diisi untuk mempermudah pengiriman dan terdapat ukuran baju yang bisa dipilih jadi pembeli bisa memilih ukuran baju yang sesuai, terdapat juga metode pembayaran. Setelah selesai mengisi semua, langsung saja klik untuk melanjutkan pembayaran yang bukti *transfERNYA* dikirim melalui *whatsapp*.




← Checkout

Nama Pembeli
Lya Nurul

No. Telpn Pembeli
087776656466

Alamat Pengiriman
Sidoarjo

 T Shirt Addict Clothing V.3 - S

Ukuran Produk
S ▼

Harga Produk
135.000

Lakukan Pembayaran Sekarang

Gambar 10. *Checkout*


3.2.8 Tampilan User Profile

Pada tampilan ini berisi tentang informasi pengguna yang telah ada saat melakukan registrasi di awal jadi data tersebut diambil dari *database* dan data tersebut tidak bisa diganti atau dirubah terdapat juga button *logout* untuk keluar dan kembali login.

←

Profil Pengguna

DEBUG



Nama

Lya Nurul

Email

lya@gmail.com

Nomor Telepon

087776656466

Alamat

Sidoarjo

Logout

Gambar 11. *User Profile*

BAB V. PENUTUP

5.1 Kesimpulan

Aplikasi mobile Addictive Clothing merupakan sebuah solusi digital yang efektif dalam membangun dan memperkuat brand Addictive Clothing. Aplikasi ini memiliki beberapa fitur dan fungsionalitas yang secara konsisten mempromosikan brand dan menciptakan pengalaman belanja yang khas bagi pengguna. Berikut adalah penekanan utama yang menunjukkan bagaimana aplikasi ini berfungsi sebagai alat branding:

Desain Antarmuka yang Mencerminkan Identitas Brand:

Desain antarmuka pengguna aplikasi ini dibangun dengan berlandaskan identitas brand Addictive Clothing. Pengguna akan merasakan keselarasan antara tampilan aplikasi dengan citra brand melalui penggunaan elemen visual, warna, dan logo yang konsisten. Desain yang menarik dan kohesif akan memberikan kesan profesional dan membantu memperkuat brand Addictive Clothing.

Katalog Produk yang Memperlihatkan Nilai Brand:

Katalog produk yang ditampilkan dalam aplikasi secara konsisten mencerminkan nilai dan gaya brand Addictive Clothing. Pengguna dapat melihat produk-produk yang trendy, stylish, dan sesuai dengan identitas brand. Hal ini membantu membangun persepsi positif dan kepercayaan pengguna terhadap brand Addictive Clothing.

Proses Pemesanan yang Efisien:

Aplikasi Addictive Clothing menawarkan proses pemesanan yang mudah dan efisien. Pengguna dapat mengisi informasi pengiriman, memilih ukuran produk, dan memilih metode pembayaran dengan cepat dan praktis.

Fitur Komunikasi dengan Admin: Aplikasi ini menyediakan fitur komunikasi langsung antara pengguna dan admin melalui WhatsApp. Pengguna dapat melakukan pre-order, mengajukan pertanyaan, atau mengirim bukti transfer melalui fitur ini.

Dengan memanfaatkan aplikasi mobile Addictive Clothing, brand ini dapat meningkatkan eksposur, meningkatkan kesadaran, dan memperkuat citra brand mereka. Aplikasi ini memberikan kesempatan kepada pengguna untuk terlibat dalam pengalaman belanja yang khas dan konsisten dengan brand. Dalam jangka panjang, fokus pada branding melalui aplikasi ini akan membantu Addictive Clothing memperluas pangsa pasarnya, membangun loyalitas pengguna, dan mencapai keberhasilan yang lebih besar dalam industri fashion.

5.2 Saran

Sistem yang dihasilkan pada aplikasi *mobile* ini masih terdapat beberapa kekurangan. Diharapkan agar sistem yang dihasilkan dapat berkembang. Saran untuk selanjutnya adalah sebagai berikut:

1. Menambahkan metode pembayaran secara *online* seperti rekening bank, Dana, Ovo, gopay, maupun Qris
2. Menambah fitur lupa *password* untuk pengguna dan menghubungkan dengan *email* yang didaftarkan.
3. Menambahkan jasa kirim pada sistem seperti dengan menggunakan API Raja Ongkir.

DAFTAR PUSTAKA

- Al-Turany, M., D. Bertini, R. Karabowicz, and ... 2012. "The Fairroot Framework." *Journal of Physics*
- Anon. n.d. "Flutter - Build Apps for Any Screen." Retrieved June 12, 2023 (<https://flutter.dev/>).
- Aryanto, D., and F. A. Tarigan. 2015. "Aplikasi Penjualan Pakaian Secara Online (Studi Kasus: Tauko Medan)." *Jurnal TIMES*.
- Aulia, A. 2023. "SISTEM INFORMASI AKUNTANSI PERHITUNGAN HARGA POKOK PENJUALAN SAPI BERBASIS MOBILE." *Jurnal Teknologi Terkini*.
- Efriyanti, M., R. I. Garaika, and R. Irviani. 2018. "Analisis Implementasi Electronic Commerce Untuk Meningkatkan Omset Penjualan Butik Mery Berbasis Web Mobile." *Jurnal*
- Irsan, M. 2015. "Rancang Bangun Aplikasi Mobile Notifikasi Berbasis Android Untuk Mendukung Kinerja Di Instansi Pemerintahan." *JustIN (Jurnal Sistem Dan Teknologi Informasi)*.
- Jihan, H. 2017. "Analisa Dan Perancangan Sistem Informasi Penjualan Roti Manis Aplikasi Dekstop Pada Pt. Charlies Lestari Sentosa." *JIKA (Jurnal Informatika)*.
- Pradiani, T. 2017. "Pengaruh Sistem Pemasaran Digital Marketing Terhadap Peningkatan Volume Penjualan Hasil Industri Rumahan." *Jurnal Ilmiah Bisnis Dan Ekonomi Asia*.
- Rozinah, S., and A. Meiriki. 2020. "Pemanfaatan Digital Marketing Pada Usaha Mikro Kecil Dan Menengah (UMKM) Di Kota Tangerang Selatan." *Jurnal Doktor Manajemen (JDM)*.
- Setiaji, A. 2020. "Rancang Bangun Aplikasi Pemesanan Desain Jersey Berbasis Android Dengan Menggunakan Teknologi Firebase (Studi Kasus: Konfeksi Minister)." *Jurnal Sistem Informasi Dan Sains Teknologi*.
- Tjandra, S., and G. S. Chandra. 2020. "Pemanfaatan Flutter Dan Electron Framework Pada Aplikasi Inventori Dan Pengaturan Pengiriman Barang." *Journal of Information System, Graphics*

LAMPIRAN

1. Source Code

Kode sumber (*source code*) merupakan kumpulan instruksi tertulis dalam pemrograman perangkat lunak. Ini digunakan untuk memahami, mengedit, dan memperbaiki kode yang memungkinkan kolaborasi antara para pengembang. Dalam pengembangan aplikasi *mobile Addictive Clothing*, pengembang menggunakan bahasa pemrograman *Dart* dan mengimplementasikannya menggunakan *Flutter*. Tujuan utama dari penggunaan *source code* yang terstruktur, terorganisasi, dan efisien adalah untuk mencapai hasil yang optimal dalam pembangunan aplikasi mobile ini. Berikut adalah beberapa contoh *source code* yang digunakan dalam pengembangan aplikasi *Addictive Clothing*:

➤ Splash Screen

Splash screen adalah tampilan awal yang muncul ketika aplikasi sedang dimuat. Pada source code splash screen, umumnya terdapat instruksi untuk menampilkan logo atau animasi yang mewakili merek *Addictive Clothing*. Durasi tampilan splash screen dapat dikonfigurasi sebelum aplikasi beralih ke halaman login atau halaman utama.

```
class SplashScreen extends StatefulWidget {
  const SplashScreen({super.key});

  @override
  // ignore: library_private_types_in_public_api
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
    super.initState();
    Timer(const Duration(seconds: 3), () {
      // Navigasi ke halaman utama setelah splash screen selesai
      Navigator.pushReplacementNamed(context, '/home');
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFF36004E),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
```

```

width: 120,
height: 120,
decoration: const BoxDecoration(
  color: Color(0xCCFFFFFF),
  shape: BoxShape.circle,
),
child: Padding(
  padding: const EdgeInsetsDirectional.fromSTEB(8, 8, 8, 8),
  child: Image.asset(
    'assets/images/logosp.png',
    width: 100,
    height: 100,
    fit: BoxFit.cover,
  ),
),
),
const Text(
  'Addictive Clothing',
  style: TextStyle(
    color: Colors.white,
    fontSize: 30.0,
    fontWeight: FontWeight.bold,
  ),
),
const SizedBox(height: 30.0),
const CircularProgressIndicator(
  valueColor: AlwaysStoppedAnimation<Color>(Colors.white),

```

➤ Login

Source code login berisi instruksi untuk mengelola proses otentikasi pengguna. Ini termasuk validasi input username dan password, pengiriman permintaan ke backend untuk memeriksa kecocokan data login, serta navigasi ke halaman utama setelah pengguna berhasil login.

```

class LoginPage extends StatefulWidget {
  const LoginPage({Key? key}) : super(key: key);

  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  bool rememberMe = false;
  bool isPasswordHidden = true;
  final TextEditingController _emailController = TextEditingController();

```

```

final TextEditingController _passwordController = TextEditingController();

Future<void> loginUser(String email, String password) async {
  var url = Uri.parse('https://websiteprojek.000webhostapp.com/api/login');
  var response = await http.post(url, body: {
    'email': email,
    'password': password,
  });

  if (response.statusCode == 200) {
    // Login berhasil
    Map<String, dynamic> userData = jsonDecode(response.body)['user'];

    // Menyimpan data pengguna ke UserManager
    UserManager().name = userData['name'];
    UserManager().email = userData['email'];
    UserManager().phone_number = userData['phone_number'];
    UserManager().address = userData['address'];

    Navigator.pushReplacement(
      context,
      MaterialPageRoute(
        builder: (context) => Home(userData: userData),
      ),
    );
  } else {
    // Login gagal
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: const Text('Login Gagal'),
          content: const Text('Email atau password salah.'),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.pop(context);
              },
              child: const Text('OK'),
            ),
          ],
        );
      },
    );
  }
}

```

➤ Register

Source code register berisi instruksi untuk mengelola proses pendaftaran pengguna baru. Ini meliputi pengumpulan informasi pengguna seperti nama, alamat email, kata sandi, dan validasi data yang diinputkan. Source code ini juga bisa berisi instruksi untuk mengirimkan data pendaftaran ke backend dan menyimpannya dalam database.

```
class RegistrationPage extends StatefulWidget {
  const RegistrationPage({Key? key}) : super(key: key);

  @override
  // ignore: library_private_types_in_public_api
  _RegistrationPageState createState() => _RegistrationPageState();
}

class _RegistrationPageState extends State<RegistrationPage> {
  bool rememberMe = false;
  bool isPasswordHidden = true;

  Future<void> _registerUser() async {
    // Mendapatkan data dari TextField
    String name = _nameController.text;
    String email = _emailController.text;
    String phoneNumber = _phoneNumberController.text;
    String address = _addressController.text;
    String password = _passwordController.text;

    var url =
Uri.parse('https://websiteprojek.000webhostapp.com/api/register');

    try {
      var response = await http.post(url, body: {
        'name': name,
        'email': email,
        'phone_number': phoneNumber,
        'address': address,
        'password': password,
      });

      if (response.statusCode == 201) {
        // Registrasi berhasil
        // ignore: use_build_context_synchronously
        showDialog(
          context: context,
          builder: (_) => AlertDialog(
            title: const Text('Registrasi Berhasil'),
            content: const Text('Akun Anda telah berhasil didaftarkan.'),
            actions: <Widget>[
```



```

        ElevatedButton(
          onPressed: () {
            Navigator.of(context).pop();
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => const
LoginPage()),
            );
          },
          child: const Text('OK'),
        ),
      ],
    ),
  );
} else {
  // Registrasi gagal
  // ignore: use_build_context_synchronously
  showDialog(
    context: context,
    builder: (_) => AlertDialog(
      title: const Text('Registrasi Gagal'),
      content: const Text(
        'Mohon cek kembali email anda dan pastikan password anda
berjumlah 8 digit.'),
      actions: <Widget>[
        ElevatedButton(
          onPressed: () {
            Navigator.of(context).pop();
          },
          child: const Text('OK'),
        ),
      ],
    ),
  );
}
} catch (e) {
  // Terjadi kesalahan saat melakukan permintaan
  showDialog(
    context: context,
    builder: (_) => AlertDialog(
      title: const Text('Terjadi Kesalahan'),
      content: const Text('Terjadi kesalahan saat melakukan permintaan.'),
      actions: <Widget>[
        ElevatedButton(
          onPressed: () {
            Navigator.of(context).pop();
          },
          child: const Text('OK'),

```

```

    ),
  ],
),
);
}
}

final TextEditingController _nameController = TextEditingController();
final TextEditingController _emailController = TextEditingController();
final TextEditingController _phoneNumberController =
TextEditingController();
final TextEditingController _addressController = TextEditingController();
final TextEditingController _passwordController = TextEditingController();

```

➤ Home

Source code home berfungsi untuk menampilkan halaman utama aplikasi. Ini mencakup instruksi untuk menampilkan menu, tombol navigasi, dan konten utama aplikasi seperti daftar produk atau promosi terbaru. Source code home juga bisa berisi logika untuk mengambil data dari backend melalui API dan menampilkan informasi dinamis kepada pengguna.

```

import 'package:flutter/material.dart';
import 'package:url_launcher/url_launcher.dart';
import 'allitem.dart';
import 'userprofile.dart';

class Home extends StatefulWidget {
  final Map<String, dynamic> userData;

  const Home({Key? key, required this.userData}) : super(key: key);

  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  late PageController _pageController;

  @override
  void initState() {
    super.initState();
    _pageController = PageController(initialPage: 0);
  }

  @override
  void dispose() {
    _pageController.dispose();
  }
}

```

```
super.dispose();  
}
```

➤ User Profile

Source code user profile berisi instruksi untuk menampilkan informasi pengguna seperti foto profil, nama, alamat, dan email akun. Ini mencakup logika untuk mengambil dan menyimpan data pengguna dari backend, Terdapat fungsi Logout pada halaman login ini.

```
class UserProfilePage extends StatelessWidget {  
  final UserManager userManager = UserManager();  
  
  UserProfilePage({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        backgroundColor: const Color(0xFF300054),  
        title: const Text('Profil Pengguna'),  
      ),  
      body: TextFormField(  
        controller: TextEditingController()..text = userManager.name,  
        readOnly: true,  
        decoration: InputDecoration(  
          labelText: 'Nama',  
          labelStyle: const TextStyle(fontSize: 14),  
          hintStyle: const TextStyle(fontSize: 14),  
          enabledBorder: OutlineInputBorder(  
            borderSide: const BorderSide(  
              color: Colors.grey,  
              width: 2,  
            ),  
            borderRadius: BorderRadius.circular(8),  
          ),  
          focusedBorder: OutlineInputBorder(  
            borderSide: const BorderSide(  
              color: Colors.transparent,  
              width: 2,  
            ),  
            borderRadius: BorderRadius.circular(8),  
          ),  
          errorBorder: OutlineInputBorder(  
            borderSide: const BorderSide(  
              color: Colors.transparent,  
              width: 2,  
            ),  
          ),  
        ),  
      ),  
    ),  
  ),  
);
```

```

        borderRadius: BorderRadius.circular(8),
      ),
      focusedErrorBorder: OutlineInputBorder(
        borderSide: const BorderSide(
          color: Colors.transparent,
          width: 2,
        ),
        borderRadius: BorderRadius.circular(8),
      ),
      filled: true,
      fillColor: Colors.white,
      contentPadding: const EdgeInsets.fromLTRB(20, 24, 0, 24),
    ),
    style: const TextStyle(fontSize: 16),
    maxLines: null,
  ),
  const SizedBox(height: 16),
  TextFormField(
    controller: TextEditingController()..text = userManager.email,
    readOnly: true,
    decoration: InputDecoration(
      labelText: 'Email',
      labelStyle: const TextStyle(fontSize: 14),
      hintStyle: const TextStyle(fontSize: 14),
      enabledBorder: OutlineInputBorder(
        borderSide: const BorderSide(
          color: Colors.grey,
          width: 2,
        ),
        borderRadius: BorderRadius.circular(8),
      ),
      focusedBorder: OutlineInputBorder(
        borderSide: const BorderSide(
          color: Colors.transparent,
          width: 2,
        ),
        borderRadius: BorderRadius.circular(8),
      ),
      errorBorder: OutlineInputBorder(
        borderSide: const BorderSide(
          color: Colors.transparent,
          width: 2,
        ),
        borderRadius: BorderRadius.circular(8),
      ),
      focusedErrorBorder: OutlineInputBorder(
        borderSide: const BorderSide(
          color: Colors.transparent,

```

```

        width: 2,
      ),
      borderRadius: BorderRadius.circular(8),
    ),
    filled: true,
    fillColor: Colors.white,
    contentPadding: const EdgeInsets.fromLTRB(20, 24, 0, 24),
  ),
  style: const TextStyle(fontSize: 16),
  maxLines: null,
),
const SizedBox(height: 16),
TextFormField(
  controller: TextEditingController()
    ..text = userManager.phone_number,
  readOnly: true,
  decoration: InputDecoration(
    labelText: 'Nomor Telepon',
    labelStyle: const TextStyle(fontSize: 14),
    hintStyle: const TextStyle(fontSize: 14),
    enabledBorder: OutlineInputBorder(
      borderSide: const BorderSide(
        color: Colors.grey,
        width: 2,
      ),
      borderRadius: BorderRadius.circular(8),
    ),
    focusedBorder: OutlineInputBorder(
      borderSide: const BorderSide(
        color: Colors.transparent,
        width: 2,
      ),
      borderRadius: BorderRadius.circular(8),
    ),
    errorBorder: OutlineInputBorder(
      borderSide: const BorderSide(
        color: Colors.transparent,
        width: 2,
      ),
      borderRadius: BorderRadius.circular(8),
    ),
    focusedErrorBorder: OutlineInputBorder(
      borderSide: const BorderSide(
        color: Colors.transparent,
        width: 2,
      ),
      borderRadius: BorderRadius.circular(8),
    ),
  ),
),

```

```

        filled: true,
        fillColor: Colors.white,
        contentPadding: const EdgeInsets.fromLTRB(20, 24, 0, 24),
    ),
    style: const TextStyle(fontSize: 16),
    maxLines: null,
),
const SizedBox(height: 16),
TextFormField(
    controller: TextEditingController()..text = userManager.address,
    readOnly: true,
    decoration: InputDecoration(
        labelText: 'Alamat',
        labelStyle: const TextStyle(fontSize: 14),
        hintStyle: const TextStyle(fontSize: 14),
        enabledBorder: OutlineInputBorder(
            borderSide: const BorderSide(
                color: Colors.grey,
                width: 2,
            ),
            borderRadius: BorderRadius.circular(8),
        ),
        focusedBorder: OutlineInputBorder(
            borderSide: const BorderSide(
                color: Colors.transparent,
                width: 2,
            ),
            borderRadius: BorderRadius.circular(8),
        ),
        errorBorder: OutlineInputBorder(
            borderSide: const BorderSide(
                color: Colors.transparent,
                width: 2,
            ),
            borderRadius: BorderRadius.circular(8),
        ),
        focusedErrorBorder: OutlineInputBorder(
            borderSide: const BorderSide(
                color: Colors.transparent,
                width: 2,
            ),
            borderRadius: BorderRadius.circular(8),
        ),
        filled: true,
        fillColor: Colors.white,
        contentPadding: const EdgeInsets.fromLTRB(20, 24, 0, 24),
    ),
    style: const TextStyle(fontSize: 16),

```



```

        {required this.name,
        required this.price,
        required this.image,
        required this.description});

factory Product.fromJson(Map<String, dynamic> json) {
  return Product(
    name: json['merek'],
    price: json['harga'],
    image: json['gambar'],
    description: json['deskripsi'],
  );
}
}

class ProductListWidget extends StatefulWidget {
  const ProductListWidget({super.key});

  @override
  // ignore: library_private_types_in_public_api
  _ProductListWidgetState createState() => _ProductListWidgetState();
}

class _ProductListWidgetState extends State<ProductListWidget> {
  List<Product> products = [];

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

  Future<void> fetchProducts() async {
    final response = await http
      .get(Uri.parse('https://websiteprojek.000webhostapp.com/api/produk'));

    if (response.statusCode == 200) {
      final data = json.decode(response.body);
      List<Product> productList = [];

      for (var item in data) {
        Product product = Product.fromJson(item);
        productList.add(product);
      }

      setState(() {
        products = productList;
      });
    }
  }
}

```



```

    } else {
      print('Failed to fetch products');
    }
  }
}

```

➤ Detail Produk

Source code detail produk berisi instruksi untuk menampilkan informasi lengkap tentang satu produk tertentu. Ini mencakup pengambilan data produk dari backend, menampilkan gambar, deskripsi, harga, dan opsi lainnya. Source code ini juga bisa mencakup instruksi untuk menambahkan produk ke keranjang belanja atau memulai proses pemesanan.

```

class DetailprodukWidget extends StatefulWidget {
  final Product item;

  const DetailprodukWidget(
    {required this.item, Key? key, required String imageUrl})
    : super(key: key);

  @override
  // ignore: library_private_types_in_public_api
  _DetailprodukWidgetState createState() => _DetailprodukWidgetState();
}

class _DetailprodukWidgetState extends State<DetailprodukWidget> {
  late PageController _pageController;
  int _pageViewCurrentIndex = 0;

  final scaffoldKey = GlobalKey<ScaffoldState>();

  @override
  void initState() {
    super.initState();
    _pageController = PageController(initialPage: _pageViewCurrentIndex);
  }

  @override
  void dispose() {
    _pageController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      key: scaffoldKey,

```

```

appBar: AppBar(
  backgroundColor: Colors.white,
  automaticallyImplyLeading: false,
  leading: InkWell(
    onTap: () async {
      Navigator.pop(context);
    },
    child: const Icon(
      Icons.arrow_back_rounded,
      color: Colors.black,
      size: 24,
    ),
  ),
  title: const Text(
    'Detail Produk',
    style: TextStyle(
      fontFamily: 'Lexend Deca',
      color: Color(0xFF151B1E),
      fontSize: 16,
      fontWeight: FontWeight.w500,
    ),
  ),
  actions: const [],
  centerTitle: true,
  elevation: 0,
),
body: Column(
  mainAxisAlignment: MainAxisAlignment.max,
  children: [
    Expanded(
      child: SingleChildScrollView(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            SizedBox(
              width: double.infinity,
              height: 400,
              child: SizedBox(
                width: double.infinity,
                height: 300,
                child: Stack(
                  children: [
                    Padding(
                      padding: const EdgeInsets.only(bottom: 40),
                      child: PageView(
                        controller: _pageController,
                        scrollDirection: Axis.horizontal,

```

```

onPageChanged: (index) {
  setState(() {
    _pageViewControllerCurrentIndex = index;
  });
},
children: [
  ClipRRect(
    borderRadius: BorderRadius.circular(8),
    child: Image.asset(
      'assets/images/produks/${widget.item.image}', // Menggunakan gambar dari API
      width: 300,
      height: 200,
      fit: BoxFit.contain,
    ),
  ),
],
),
),
Align(
  alignment: Alignment.bottomLeft,
  child: Padding(
    padding: const EdgeInsets.fromLTRB(16, 0, 0,
16),

    child: SmoothPageIndicator(
      controller: _pageController,
      count: 1,
      axisDirection: Axis.horizontal,
      onDotClicked: (index) {
        _pageController.animateToPage(
          index,
          duration: const Duration(milliseconds:
500),

          curve: Curves.ease,
        );
      },
      effect: const ExpandingDotsEffect(
        expansionFactor: 3,
        spacing: 8,
        radius: 16,
        dotWidth: 16,
        dotHeight: 8,
        dotColor: Colors.grey,
        activeDotColor: Colors.black,
        paintStyle: PaintingStyle.fill,
      ),
    ),
  ),
),
),
),
),

```



```

    ),
    Padding(
      padding: const EdgeInsets.fromLTRB(16, 20, 16, 15),
      child: SizedBox(
        width: double.infinity,
        height: 50,
        child: ElevatedButton.icon(
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => CheckoutPage(
                  imageUrl:
'assets/images/produks/${widget.item.image}',
                  productName: widget.item.name,
                  productPrice: widget.item.price,
                ),
              ),
            );
          },
          icon: const Icon(Icons.add_shopping_cart),
          label: const Text(
            'Pre Order',
            style: TextStyle(
              fontFamily: 'Poppins',
              fontSize: 18,
              fontWeight: FontWeight.w500,
            ),
          ),
          style: ElevatedButton.styleFrom(
            foregroundColor: Colors.white,
            backgroundColor: const Color(0xFF300054),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(8),

```

➤ Checkout

Source code checkout berisi instruksi untuk menampilkan halaman checkout ketika pengguna ingin melakukan pembelian produk. Ini mencakup pengambilan data produk yang dipilih, pengisian informasi pengiriman, memilih metode pembayaran, dan melibatkan komunikasi dengan backend untuk menyimpan data pesanan ke dalam database. Source code ini juga bisa berisi instruksi untuk mengirimkan bukti transfer pembayaran atau notifikasi ke admin.

```

enum PaymentMethod {
  COD,
  BRI,

```

```

    DANA,
  }

class CheckoutPage extends StatefulWidget {
  final String imageUrl;
  final String productName;
  final String productPrice;

  const CheckoutPage({
    required this.imageUrl,
    required this.productName,
    required this.productPrice,
    Key? key,
  }) : super(key: key);

  @override
  _CheckoutPageState createState() => _CheckoutPageState();
}

class _CheckoutPageState extends State<CheckoutPage> {
  String selectedSize = 'S';
  TextEditingController namaPembeliController = TextEditingController();
  TextEditingController noTelponPembeliController = TextEditingController();
  TextEditingController alamatPengirimanController = TextEditingController();
  PaymentMethod selectedPaymentMethod = PaymentMethod.COD;

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

    // Inisialisasi dan set nilai awal pada TextEditingController
    namaPembeliController = TextEditingController(text: UserManager().name);
    noTelponPembeliController =
      TextEditingController(text: UserManager().phone_number);
    alamatPengirimanController =
      TextEditingController(text: UserManager().address);
  }

  void cetakStruk() async {
    String namaPembeli = namaPembeliController.text;
    String noTelponPembeli = noTelponPembeliController.text;
    String alamatPengiriman = alamatPengirimanController.text;

    String paymentMethodText = '';
    IconData paymentMethodIcon = Icons.attach_money;

    switch (selectedPaymentMethod) {
      case PaymentMethod.COD:

```

```

        paymentMethodText = 'COD';
        paymentMethodIcon = Icons.attach_money;
        break;
    case PaymentMethod.BRI:
        paymentMethodText = 'BRI (003301133967500)';
        paymentMethodIcon = Icons.payment;
        break;
    case PaymentMethod.DANA:
        paymentMethodText = 'DANA: (085156023639)';
        paymentMethodIcon = Icons.payment;
        break;
}

const phoneNumber = '+6288231328113';
final message =
    'Halo admin, saya ingin pre-order produk dari Addictive Clothing
    nih\n'
    'berikut adalah struk pesanan saya: \n'
    '\n'
    'Nama Pembeli: $namaPembeli\n'
    'No. Telpon Pembeli: $noTelponPembeli\n'
    'Alamat Pengiriman: $alamatPengiriman\n'
    'Produk: ${widget.productName}\n'
    'Ukuran Produk: $selectedSize\n'
    'Harga Produk: ${widget.productPrice}\n'
    'Metode Pembayaran: $paymentMethodText\n';

final url =
    'https://wa.me/$phoneNumber?text=${Uri.encodeComponent(message)}';

// ignore: deprecated_member_use
launcher.launch(url);
}

void showSuccessAlert() {
    showDialog(
        context: context,
        builder: (BuildContext context) {
            return AlertDialog(
                title: const Icon(Icons.check_circle, color: Colors.green, size:
40),
                content: const Text(
                    'Pembelian sedang diproses. Mohon selesaikan transaksi dengan
                    admin dan jangan lupa kirim bukti transfernnya ya. Terimakasih!',
                    textAlign: TextAlign.center,
                ),
                actions: [
                    Container(

```

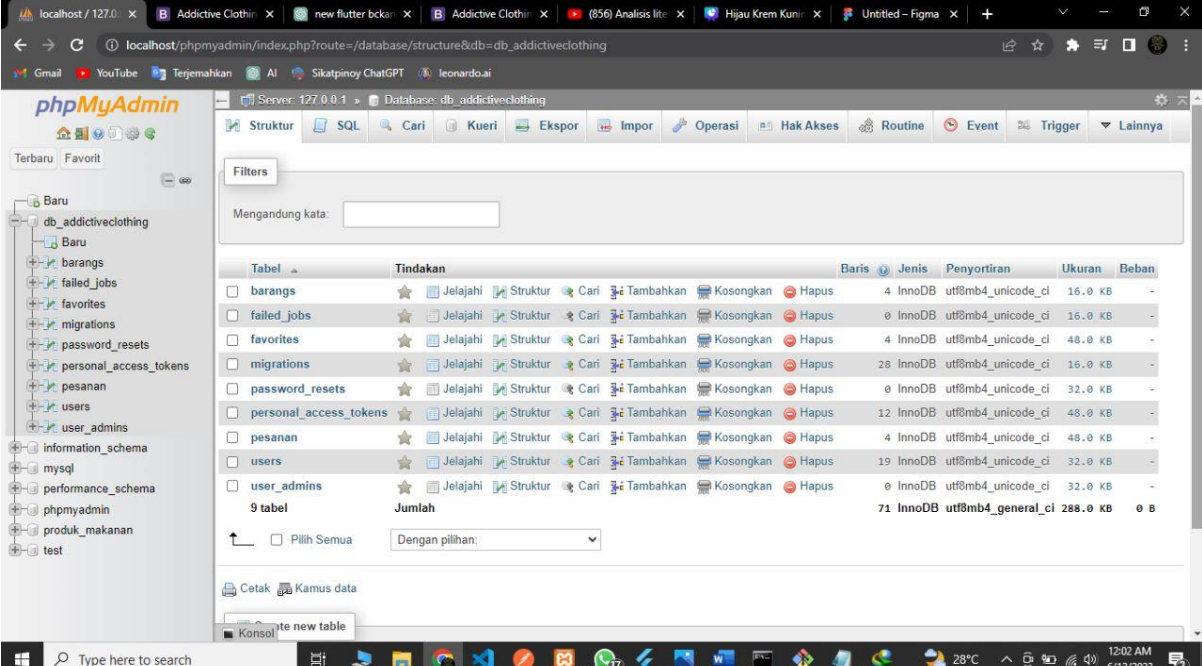
```

width: double.infinity,
alignment: Alignment.center,
child: ElevatedButton(
  onPressed: () {
    cetakStruk();
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => const Home(userData: {}),
      ),
    );
  },
  style: ElevatedButton.styleFrom(
    foregroundColor: Colors.white,
    backgroundColor: const Color(0xFF300054),
  ),
  child: const Text('Lanjutkan'),
),
),
],
);
},
);
}

```


2. Database

Pada *website* dan *mobile addictive clothing* menggunakan *database* yang sama dan didalamnya terdapat tabel-tabel seperti *barang*s, *favorites*, *migrations*, *user*, dan *pesanan* masing-masing tabel tersebut berfungsi untuk menyimpan data dan mengelolah informasi yang terkait dengan e-commerce seperti menyimpan data pengguna, menyimpan data-data penjualan seperti gambar pakaian, dan catatan pesanan.



Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
barang	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	4	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
failed_jobs	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
favorites	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	4	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
migrations	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	28	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
password_resets	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
personal_access_tokens	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	12	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
pesanan	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	4	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
user_admins	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
users	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	19	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
9 tabel	Jumlah	71	InnoDB	utf8mb4_general_ci	288.0 KB	0 B

Gambar 12. Database

3. Logo Mitra



Gambar 13. Logo Mitra

4. Foto Mitra

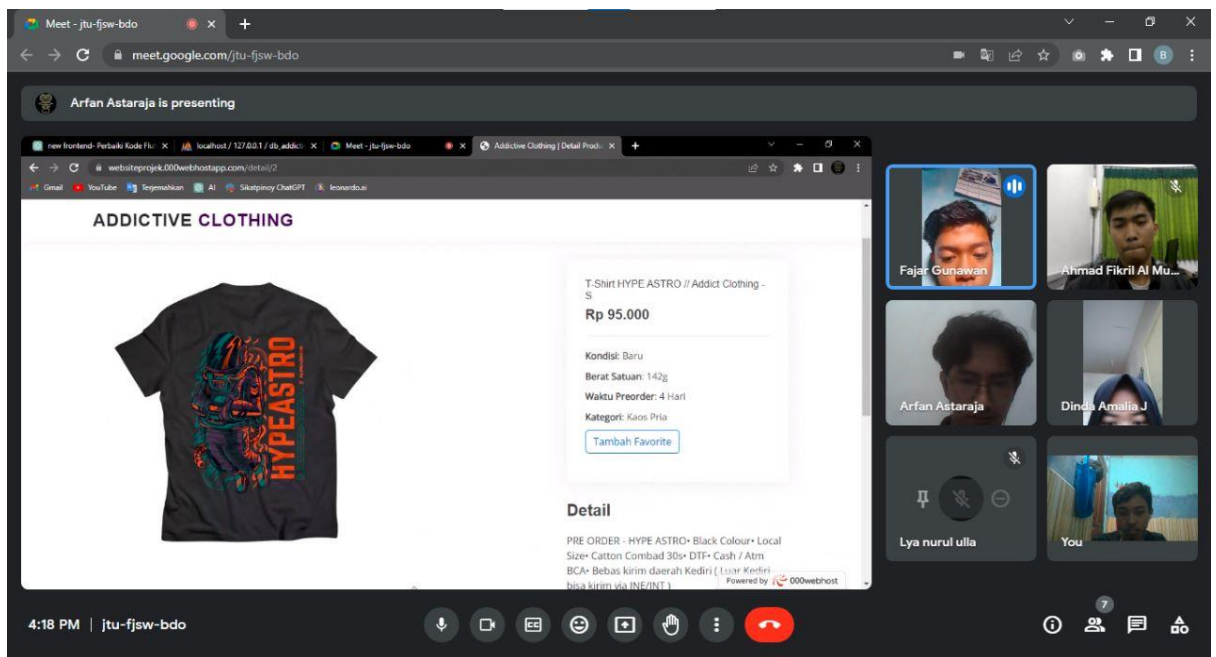


Gambar 14. Pemilik Mitra

5. Pembahasan pembuatan *website* dan aplikasi *mobile* dengan mitra



Gambar 15. Berdiskusi Pembuatan Aplikasi *Mobile*



Gambar 16. Berdiskusi Pembuatan Website