Bima Abdi: Jurnal Pengabdian Masyarakat
p-ISSN: 2797-9407, e-ISSN: 2797-9423
Volume 2, nomor 2, 2022, hal. 208-220
Doi: https://doi.org/10.53299/bajpm.v2i2.197

Pendampingan Desain Aplikasi *Food Court* untuk Pengelola Kedai dan Pramusaji Berbasis Android

Abd. Charis Fauzan*, Utrodus Said Al Baqi, Veradella Yuelisa Mafula
Universitas Nahdlatul Ulama Blitar, Blitar, Indonesia

*Coresponding Author: abdcharis@unublitar.ac.id
Dikirim: 04-08-2022; Direvisi: 07-08-2022; Diterima: 08-08-2022

Abstrak: Dewasa ini, beberapa aplikasi android yang telah dirancang untuk industri *food court* belum sepenuhnya telah menyelesaikan masalah yang terjadi pada sistem pelayanan tradisional. Oleh sebab itu, diperlukan pendampingan untuk merancang desain aplikasi *food court* berbasis android yang diperuntukkan bagi pengelola kedai dan pramusaji. Pendampingan ini berkerjasama dengan CV Ediide Infografika sebagai upaya meningkatkan kualitas pelayanan pada industri *food court.* Pendampingan desain aplikasi dilakukan dengan menggunakan beberapa *tool*, yakni *figma design tool* untuk merancang kerangka layout dasar aplikasi, membuat *visual design*, serta *flutter software development kit (SDK)* untuk membangun tampilan aplikasi yang indah. Kemudian dilakukan langkah kompilasi secara *native* untuk seluler, web, desktop, dan perangkat yang disematkan dari satu basis kode. Pendampingan desain aplikasi *food court* untuk pengelola kedai dan pramusaji berbasis android dilaksanakan dengan baik serta berhasil memenuhi kebutuhan fitur, dibuktikan dengan hasil dari *black box testing* yang telah dilakukan.

Kata Kunci: android; desain aplikasi; *food court*

Abstract: Today, several android applications that have been designed for industrial food courts have not fully resolved the problems that occur in traditional service systems. Therefore, assistance is needed to design an Android-based food court application design that is needed by shop managers and waiters. This assistance is in collaboration with CV Ediide Infografika as an effort to improve service quality in industrial food courts. Application design assistance is carried out using several tools, namely the figma design tool to design the application's basic layout framework, create visual designs, and the flutter software development kit (SDK) to build a beautiful application display. It then compiles natively for mobile, web, desktop, and embedded devices from a single codebase. Food court application design assistance for store managers and waiters based on Android is well implemented and successfully meets the feature needs, as evidenced by the results of the black box testing that has been carried out.

Keywords: android; application design; *food courts*

PENDAHULUAN

Seiring berjalannya waktu perkembangan teknologi telah membawa hal yang luar biasa yaitu merevolusi seluruh dunia industri. Industri *food court* merupakan industri yang sedang dalam proses untuk direvolusi melalui bantuan teknologi (Qamar et al., 2019). Salah satu teknologi yang bisa dimanfaatkan ialah penggunaan aplikasi android. Penggunaan aplikasi android sangat popular dibuktikan pada tahun 2015, *android* memiliki basis terinstal terbesar di semua sistem operasi seluler, pada bulan Juni 2017 *Google Play Mobile App Store* telah merilis lebih dari 3.000.000 aplikasi *android*, dan lebih dari 80 miliar aplikasi telah diunduh. Puncaknya pada tahun 2017, *Google I/O* menemukan bahwa mereka memiliki lebih dari 2 miliar pengguna

Android aktif per bulan, tidak seperti tahun sebelumnya, dengan angka sekitar 1,5 miliar pengguna aktif (Lazareska, 2017).

Pada industri *food court* tersebut ada beberapa macam jenis-jenis konsep, salah satunya yaitu konsep pesan di meja makan. Sejauh ini, sistem layanan tradisional di industri *food court* dengan konsep pesan di meja makan memiliki beberapa kendala yaitu Pengelola kedai kesulitan dalam menyimpan riwayat pesanan baru dan pesanan yang telah selesai, serta tidak adanya informasi yang tersimpan tentang stok makanan atau minuman. (Aulia et al., 2017). Kendala yang lainnya yaitu waktu pemesanan kurang cepat karena pemesanan makanan atau minuman ditulis secara manual oleh pramusaji menggunakan pulpen dan kertas (Bhargave et al., 2013), hal tersebut juga dapat menyebabkan pesanan bisa tertukar dengan konsumen lain sehingga pramusaji harus kembali ke meja konsumen dan menanyakan kembali penggantinya (Shinde et al., 2014).

Dewasa ini beberapa aplikasi android yang telah dirancang untuk industri *food court* belum sepenuhnya telah menyelesaikan masalah yang terjadi pada sistem pelayanan tradisional (Kubde et al., 2016). Oleh sebab itu maka dibutuhkan pendampingan untuk merancang desain aplikasi *food court* untuk pengelola kedai dan pramusaji berbasis android. Pendampingan ini berkerjasama dengan CV Ediide Infografika sebagai upaya meningkatkan kualitas pelayanan pada industri *food court.* Pendampingan desain aplikasi dilakukan dengan menggunakan beberapa *tool*, yakni *figma design tool* untuk merancang kerangka layout dasar aplikasi, membuat *visual design*, serta *flutter software development kit (SDK)* untuk membangun tampilan aplikasi yang indah. Kemudian dilakukan langkah kompilasi secara native untuk seluler, web, desktop, dan perangkat yang disematkan dari satu basis kode (Dagne, 2019).

KAJIAN TEORI

Food court

Food court atau pujasera (pusat jajanan serba ada) ialah kumpulan beragam kedai didalam satu area yang luas. Kedai-kedai didalam *food court* biasanya beroperasi berdampingan, dengan ruang makan komunal terpusat untuk mengakomodasi pelanggan. Sebuah food court umumnya terletak di pusat perbelanjaan atau lokasi pariwisata (Fitriandi, 2013). Tempat untuk makan didalam lokasi *food court* tidak terikat pada penggunaan eksklusif atau kepemilikan kedai mana pun, tetapi disediakan untuk pelanggan setiap restoran di atau penyewa lain di area tersebut (Onainor, 2019). Konsep-konsep pada *food court* ada beberapa macam menurut pinhome.id (Home, 2021) yaitu:

a. Konsep Cepat Saji

Konsep cepat saji mengarahkan pengunjung untuk memesan makanan atau minuman di kedai secara langsung, konsep ini lebih banyak ditemui di area yang ramai dan pengunjung yang datang memiliki waktu terbatas dengan penyajian maksimal 15 menit, contoh area *food court* dengan konsep seperti ini yaitu area perkantoran dan pusat perbelanjaan yang ramai.

b. Konsep Pesan di Meja Makan

Berbeda dengan konsep cepat saji, konsep ini dapat lebih maksimal dalam hal pelayanannya karena pengunjung bisa langsung pesan di meja makan seperti halnya restoran. Ketika makan di sini, pembeli atau konsumen hanya perlu duduk di meja

lalu pramusaji akan mendatangi dengan membawa menu. Namun, jenis produk yang ditawarkan membutuhkan waktu yang lama hingga disajikan kepada pengunjung. Maka dari itu, pujasera konsep ini sering ditemukan di mall yang dinamis.

c. Urban Food Court

Konsep ini dikenal juga urban food court yang merupakan bagian dari pujasera modern. Pada food court konvensional fungsinya pun lebih sebagai tempat makan, berbeda dengan urban food court di mana disediakan fasilitas untuk nongkrong. Artinya, urban food court adalah pujasera yang lebih modern.

Sistem Pelayanan Tradisional *Food Court*

Sistem layanan *food court* tradisional di industri memerlukan banyak dokumen dan tenaga manusia yang mungkin tidak cukup dengan permintaan yang terus meningkat untuk layanan cepat di industri. Di sisi lain, pemeliharaan manual atas informasi penting menggunakan kertas akan berisiko pada waktu-waktu tertentu (Reddy K & KGK, 2016). Untuk Sistem manajemen pesanan tradisional pada *food court* umumnya menggunakan skema swalayan, dimana pelanggan harus mengunjungi setiap kedai untuk memesan dan harus mengambil pesanan mereka sendiri. Seringkali pelanggan harus mengunjungi lebih dari tujuh hingga delapan kedai *food court* sampai mereka dapat memutuskan dari kedai makanan atau minuman mana mereka ingin memesan. Ini menimbulkan tugas besar untuk mencari kedai makanan atau minuman yang tepat sesuai dengan kepuasan pelanggan. Rata-rata langkah kaki sekitar 20.000-25.000 orang ke *food court* pada hari kerja. Jumlahnya membengkak hingga 35.000-40.000 orang, selama akhir pekan yang mempersulit pengelolaan pesanan (Kubde et al., 2016).

METODE PELAKSANAAN KEGIATAN

Metode pelaksanaan pendampingan, dilakukan dengan beberapa tahapan, antara lain:

Diskusi *Requirement* fitur aplikasi

Diskusi *requirement* fitur aplikasi dilakukan secarang langsung yaitu membahas dan meninjau penjelasan dari *client* terkait perancangan desain aplikasi *food court* untuk pengelola kedai dan pramusaji berbasis android. Diskusi tersebut dilakukan oleh penulis, pengelola CV Ediide Infografika serta *client*.

Pembuatan *information architecture* sesuai *requirement*

Berdasarkan hasil diskusi *requirement* fitur aplikasi yang telah ditentukan maka dibuat menjadi sebuah *information architecture*, Tahap-tahap dalam membuat *information architecture* ialah a) mengumpulkan hasil diskusi fitur aplikasi (berupa catatan); b) mengidentifikasi hasil diskusi fitur aplikasi; serta 3) membuat *mind map* berdasarkan hasil identifikasi dalam bentuk mind map. (dibuat dalam bentuk *mind map* menggunakan aplikasi *whimsical.com)*.

Pengumpulan referensi desain tampilan aplikasi

Dalam menentukan referensi desain tampilan aplikasi disesuaikan dengan *information architecture*. Tahapannya ialah a) mencari referensi desain tampilan aplikasi yang sesuai dengan *information architecture* didalam situs *dribble.com*; b)

hasil temuan referensi tersebut disimpan kedalam *collections* dengan fitur *save* di situs *dribble.com.*

Membuat skema warna dan tipografi

Penentuan skema warna dan tipografi ialah untuk memilih warna jenis huruf yang digunakan dalam aplikasi. Sesuai dengan hasil temuan referensi desain tampilan aplikasi, penentuan skema warna dan tipografi dapat ditujukan sebagai acuan dalam membuat aplikasi sehingga warna dan huruf yang digunakan konsisten. Langkah-langkah dalam menentukan skema warna ialah

1. Membuat sebuah *project* rancangan desain aplikasi di *figma.com.*
2. Membuat sebuah *frame* lalu membuat *rectangle object* dan diberi warna untuk uji coba warna yang sesuai dengan referensi desain tampilan aplikasi.
3. Membuat *color styles* agar memudahkan proses penyempurnaan *wireframe* ke *visual design*.

Selanjutnya untuk membuat skema tipografi tahapannya ialah:

1. Melihat beberapa *fonts* di situs *google fonts.*
2. Memilih jenis atau nama *fonts* yang akan digunakan.
3. Membuat sebuah *frame* di *project* figma yang telah dibuat.
4. Menuliskan contoh konten menggunakan *fonts* yang telah dipilih dan juga diberi ukuran *font.*
5. Tahap terakhir yaitu membuat *font style*

Membuat kerangka tampilan kasar aplikasi (*wireframe*)

Dalam membuat kerangka tampilan kasar aplikasi atau *wireframe* yang pertama ialah melihat *information architecture* yang telah dibuat terlebih dahulu dan referensi desain tampilan aplikasi sebagai acuan, lalu masuk kedalam project figma untuk membuat beberapa frame dan layout kerangka tampilan kasar fitur aplikasi pengelola kedai dan pramusaji.

Penyempurnaan *wireframe* ke *visual design*

Penyempurnaan wireframe menjadi visual design atau tampilan aplikasi secara utuh dilakukan dengan cara mengimplementasikan skema warna dan tipografi atau *color styles* dan *font style* yang telah dibuat ke masing-masing frame dan juga dibuat contoh konten untuk tampilan aplikasi.

Pengembangan aplikasi *android* dari *visual design*

Untuk mengembangkan *visual design menjadi aplikasi android* Langkah awalnya ialah menginstall beberapa *tools* yang dibutuhkan, diantaranya ialah *flutter software development kit, visual studio code, android studio & android emulator*. Kemudian berikutnya yaitu membuat sebuah *project* baru menggunakan *tools* yang telah terinstall. Langkah berikutnya ialah membuat beberapa *folder* didalam *folder* lib yang isinya ialah *controller, data, modules, routes, & shared*. *Folder c*ontroller berisi *file-file kode* terkait logika aplikasi, *folder* data isinya ialah *file-file models* atau kumpulan *object*, dan fungsi untuk komunikasi dengan sistem *backend*, *folder* modules berisi *component* tampilan aplikasi, *folder* routes berisi *file-file* kode untuk rute aplikasi atau navigasi, dan *folder shared* berisi *file-file* kode *constant* yang digunakan untuk semua *file-file* di *folder* lain. Setelah folder-folder tersebut dibuat

maka tahap berikutnya ialah menuliskan kode-kode sesuai fungsi folder yang telah dibuat.

Pengujian

Pada tahap pengujian (*testing*) aplikasi, metode yang digunakan ialah *black box testing*, pengujian *black box* dilakukan dengan cara yang relatif bertentangan dengan kebutuhan fitur aplikasi yang ada dan memastikan sistem dapat menangani semua masukan yang tidak sesuai. Oleh karena itu, pengguna hanya bisa memasukkan data yang benar ke dalam sistem.

IMPLEMENTASI KEGIATAN DAN PEMBAHASAN

Diskusi *requirement* fitur aplikasi

Langkah pertama dalam membuat aplikasi adalah menentukan fitur dan alur aplikasi yang akan dibuat. Dalam kegiatan PKL ini desain aplikasi yang dibuat ialah aplikasi foodcourt untuk pengelola kedai dan pramusaji berbasis android dengan nama bumi glagah arum. Dari diskusi tersebut menghasilkan beberapa fitur aplikasi yaitu :

a) Fitur untuk pengelola kedai, antara lain:
- *Login* aplikasi
- Melihat total pesanan, total pendapatan & statistik pendapatan bulanan.
- Membuat, merubah, menghapus menu makanan / minuman berdasarkan kategori yang telah ditentukan.
- Melihat daftar menu makanan / minuman
- Melihat daftar pesanan baru
- Melihat detail pesanan baru
- Melihat daftar pesanan selesai
- Melihat detail pesanan selesai
- Menerima pesanan baru
- Merubah foto profil, password
- *Logout* aplikasi

b) Fitur untuk pramusaji :
- *Login* aplikasi
- Melihat total pesanan siap antar
- Melihat daftar pesanan siap
- Mengubah pesanan siap menjadi pesanan tersaji
- Melihat nomor meja
- Melihat daftar pesanan berdasarkan nomor meja per tanggal
- Melihat detail pesanan berdasarkan nomor meja
- Melihat daftar menu yang tersedia
- Mencari menu makanan / minuman berdasarkan kata dan kategori
- Menambahkan menu makanan / minuman kedalam daftar konfirmasi pesanan
- Membuat catatan pesanan untuk tiap menu makanan atau minuman
- Menghapus menu makanan / minuman dari daftar konfirmasi pesanan

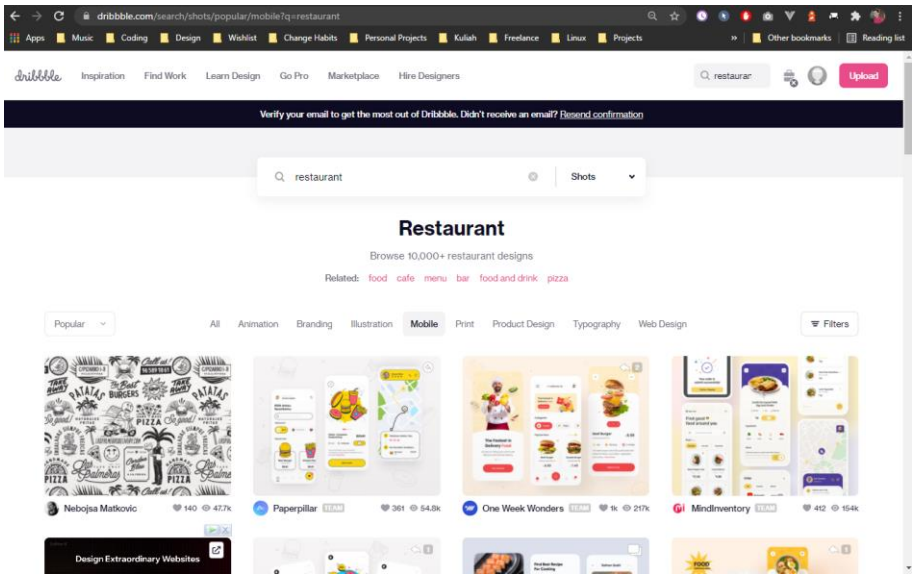
- Membuat pesanan baru dengan mengisi *form* nama pemesan, nomor meja, dan bisa merubah kuantitas menu makanan atau minuman
- Merubah foto profil, password
- *Logout* aplikasi

Pembuatan information architecture sesuai *requirement*

Fungsi information architecture ialah membantu mengidentifikasi berbagai fitur yang dibutuhkan, halaman aplikasi, navigasi serta penempatan. Dalam membuat information architecture tersebut *tools* yang digunakan ialah situs whimsical.com. Pertama masuk ke website whimsical.com kemudian login terlebih dahulu, lalu membuat project *mind map* berdasarkan fitur-fitur aplikasi untuk pengelola kedai dan pramusaji sehingga menghasilkan *information architecture*. Hasil pembuatan *information architecture* berupa *mind map* yang berisi fitur-fitur aplikasi. Tujuannya ialah sebagai acuan dalam mengembangkan aplikasi sehingga sesuai dengan kebutuhan client, selain itu juga untuk mempermudah tim dalam melaksanakan tugas masing-masing sesuai porsinya.

Mengumpulkan referensi desain tampilan aplikasi

Untuk mencari dan mengumpulkan referensi desain tampilan aplikasi kata kunci yang digunakan ialah *food court, waiter app, restaurant,* kemudian juga menyaring *(filter)* pencarian tersebut untuk layar *mobile* atau handphone saja.



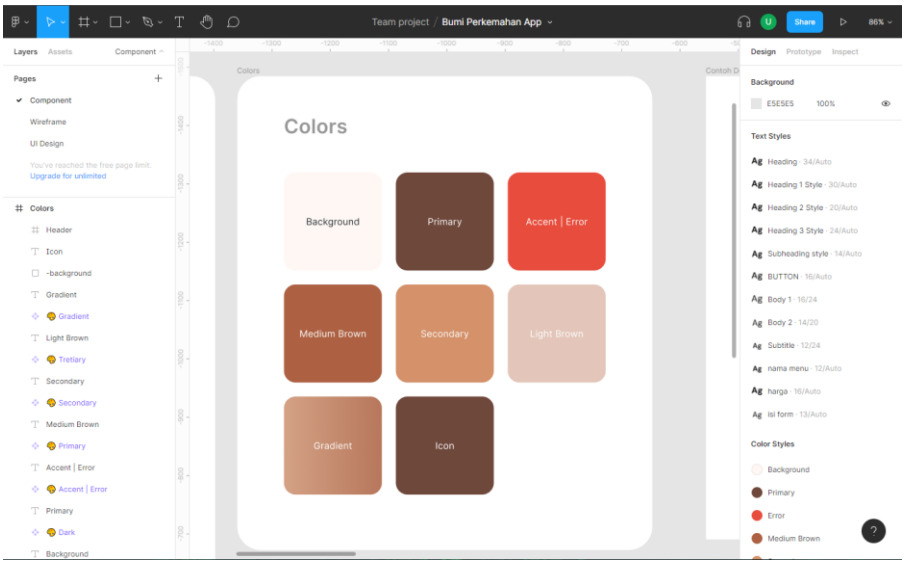
Gambar 1. Proses pencarian referensi desain aplikasi pada https://dribbble.com/

Gambar 1 menunjukan proses pencarian referensi desain aplikasi. Selanjutnya mengumpulkan beberapa contoh desain tersebut dengan cara memilih salah salah satu item dan klik tombol *save* agar tersimpan didalam koleksi referensi desain yang akan dibuat.

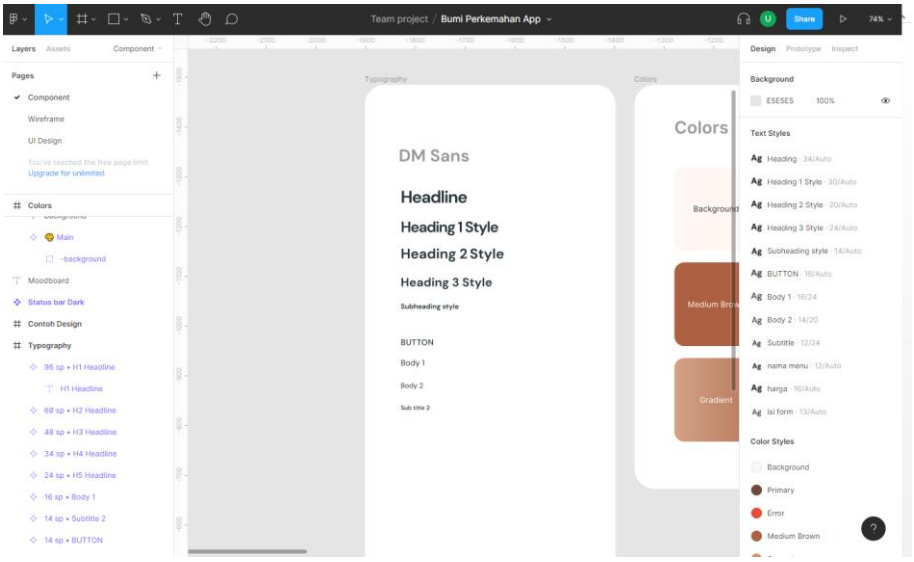
Membuat skema warna dan tipografi

Skema warna & *Font* yang digunakan dalam aplikasi baiknya konsisten agar memberikan kenyamanan untuk pengguna. Dari beberapa referensi desain tampilan aplikasi yang telah tersimpan didalam *collections* disitus *dribble.com* maka dibuatlah

menjadi skema warna dan tipografi hasilnya ditunjukkan sebagaimana Gambar 2 dan Gambar 3.



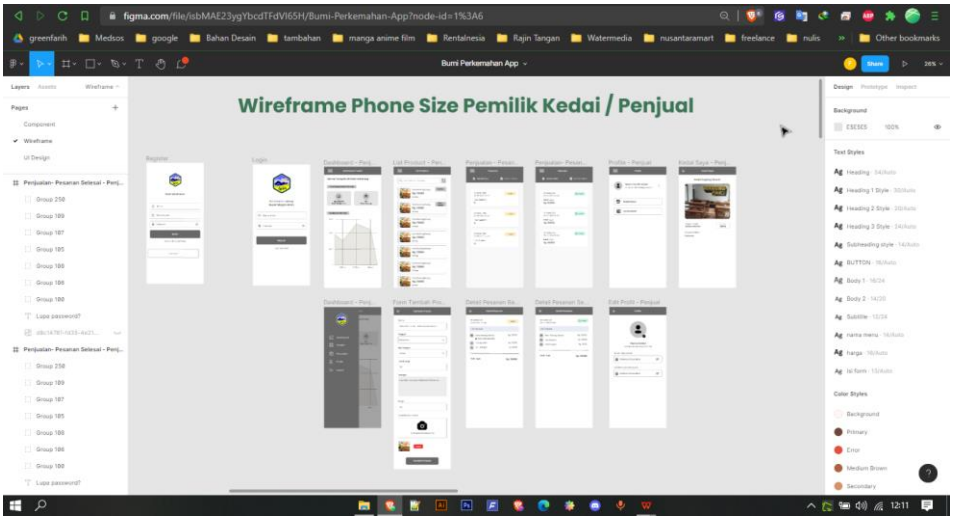
Gambar 2. Hasil pembuatan skema warna



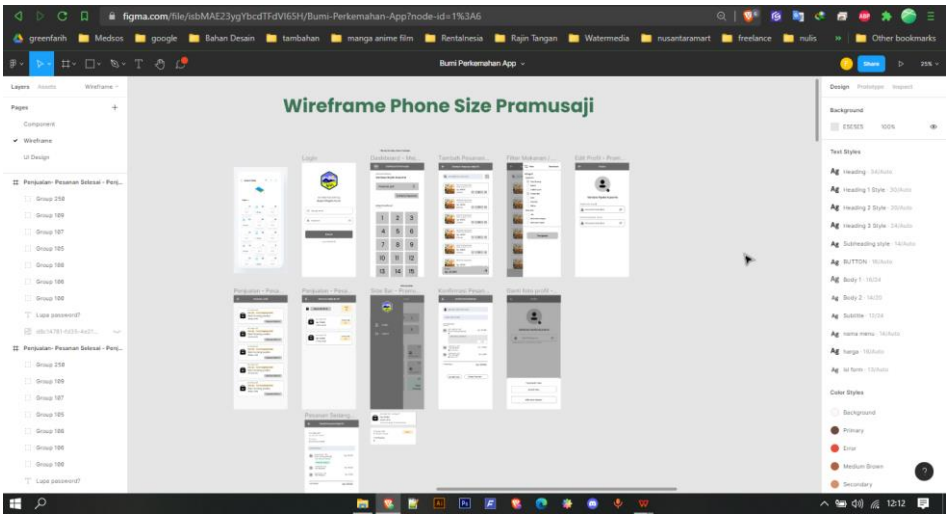
Gambar 3. Hasil pembuatan skema *fonts*

Perancangan Desain Aplikasi

Kerangka tampilan kasar aplikasi atau *wireframe* dibuat dalam bentuk tampilan per*frame* disitus *figma.com*, Pembuatan gambaran kasar ini bertujuan untuk memudahkan dalam penyesuaian terhadap requirement. Wireframe tersebut belum sampai tampilan detail desain aplikasi sehingga perubahan yang dilakukan dalam proses ini lebih mudah untuk dilakukan.



Gambar 4. Hasil *wireframe* pengelola kedai

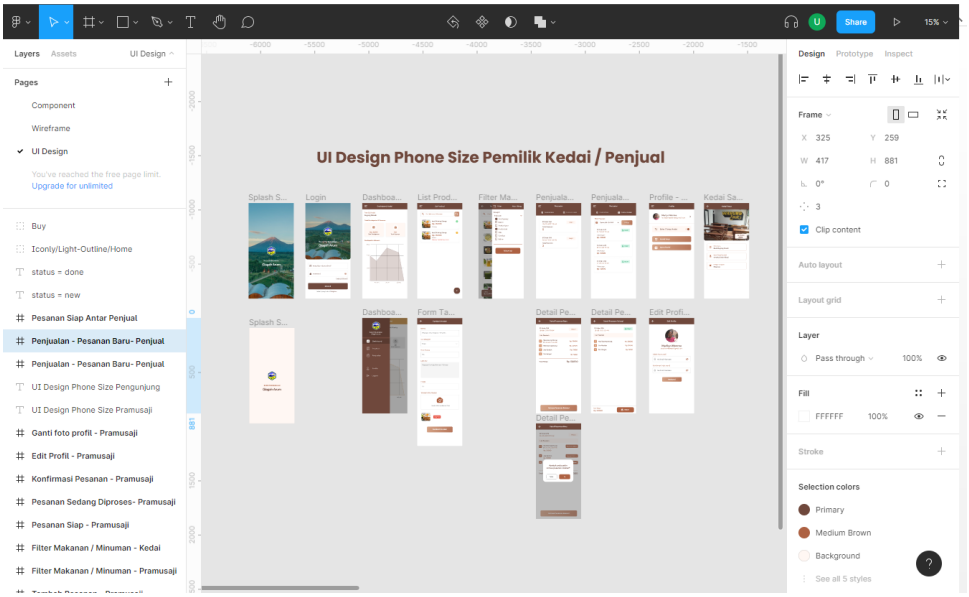


Gambar 5. Hasil *wireframe* pramusaji

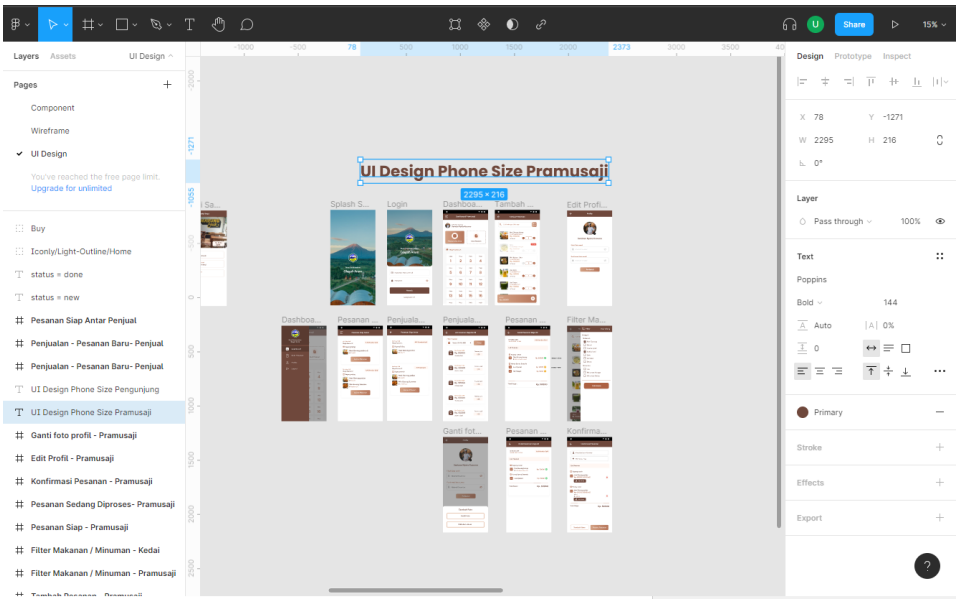
Gambar 4 dan Gambar 5 menunjukkan hasil pembuatan *wireframe* penjual dan *wireframe* pramusaji. Dalam pembuatan *wireframe* skema warna belum digunakan, sebagai gantinya hanya digunakan warna abu-abu sebagai penanda warna utama agar mempermudah saat terjadi perubahan.

Penyempurnaan wireframe ke visual design

Langkah lanjutan dari tahap sebelumnya adalah pembuatan detail desain tampilan aplikasi. *Wireframe* yang telah dibuat sebelumnya disempurnakan dengan menambahkan warna, *fonts* dan komponen tambahan lain. Tujuan dari pembuatan tampilan aplikasi ini adalah untuk mempermudah saat membuat aplikasi yang sebenarnya.



Gambar 6. Hasil pembuatan desain pengelola kedai

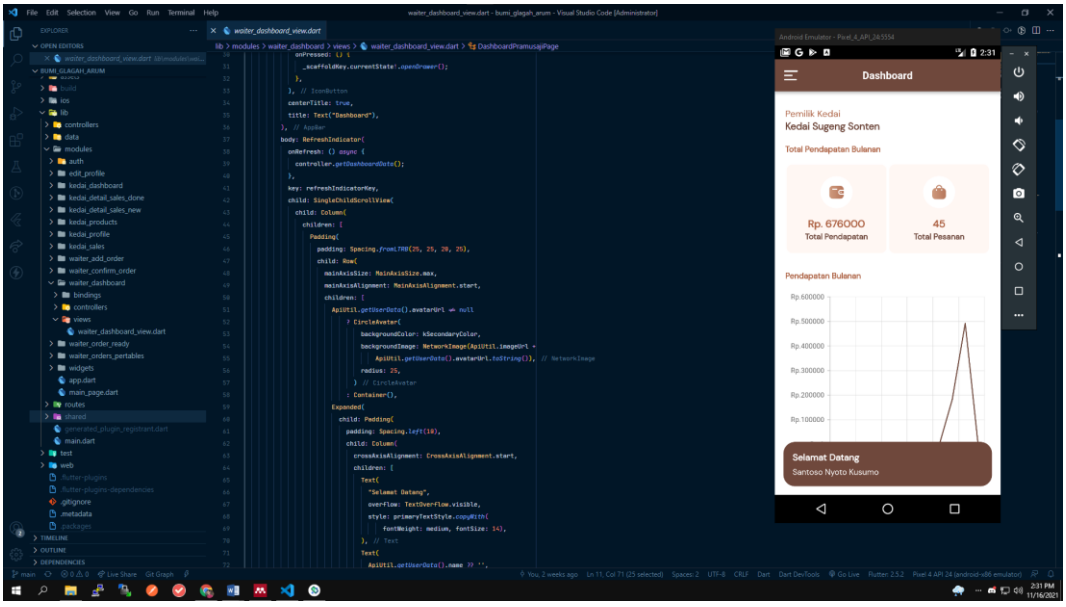


Gambar 7. Hasil pembuatan desain pramusaji

Gambar 6 dan Gambar 7 menunjukkan hasil pembuatan tampilan *desain* pengelola kedai dan pramusaji. Pada langkah ini dapat dilihat tampilan aplikasi dan penambahan detail yang menyerupai tampilan aplikasi yang sebenarnya.

Membangun Aplikasi

Setelah desain tampilan aplikasi selesai, maka tahap berikutnya ialah mengimplementasikan kedalam sebuah aplikasi android menggunakan *framework flutter*.



Gambar 8. Proses menerjemahkan komponen tampilan aplikasi kedalam bentuk kode *flutter*

Gambar 8 menunjukkan proses menerjemahkan setiap komponen tampilan aplikasi kedalam bentuk kode *flutter*. Setelah hasil rancangan desain aplikasi telah menjadi kode *flutter* langkah selanjutnya ialah mengintegrasikan tampilan tersebut dengan sistem *backend* yang telah dirancang oleh pihak CV EDIIDE Infografika. Langkah terakhir sebelum melakukan pengujian ialah *build apk* yaitu mengkompilasi kode *flutter* menjadi sebuah file *.apk*.

Pengujian

Tahap berikutnya ialah *testing* atau pengujian hasil aplikasi android, metode yang digunakan ialah metode *black box*, hasilnya tertera pada Tabel 1.

Tabel 1. Hasil *black box test*

Id *test cases*	Skenario Pengujian	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Login berbeda berdasarkan jenis pengguna Dengan cara Memasukkan email dan password Login dengan dua jenis pengguna yang berbeda yaitu pengelola kedai dan pramusaji	Email & password pengguna	Diarahkan kedalam dashboard berdasarkan jenis pengguna	Sesuai harapan	Valid
2	Menambah produk dari pengelola kedai dengan cara Memasukkan masing-masing kolom yang dibutuhkan	nama, kategori deskripsi, harga, porsi, gambar produk dan menekan tombol tambah makanan / minuman	Berhasil menambahkan produk dan *redirect* ke halaman daftar data produk	Sesuai harapan	Valid

3	Merubah produk dari pengelola kedai dengan cara Menekan salah satu produk di halaman daftar data produk dan merubah masing-masing kolom yang dibutuhkan	nama, kategori deskripsi, harga, porsi, gambar produk dan menekan tombol simpan perubahan makanan / minuman	Berhasil menambahkan produk dan *redirect* ke halaman daftar data produk	Sesuai harapan	Valid
4	Menampilkan daftar Produk kedai dengan cara Masuk ke halaman daftar produk kedai	List produk kedai	Semua produk tampil	Sesuai harapan	Valid
5	Hapus Produk Dengan cara Menekan tombol hapus pada salah satu produk	Produk	Produk yang dipilih telah terhapus	Sesuai harapan	Valid
6	Menampilkan daftar pesanan baru dengan cara Masuk ke halaman pesanan baru	List Pesanan baru	Daftar pesanan baru tampil	Sesuai harapan	Valid
7.	Menampilkan daftar pesanan selesai dengan cara Masuk ke halaman pesanan baru	List Pesanan selesai	Daftar pesanan baru tampil	Sesuai harapan	Valid
8	Buka Tutup Kedai oleh pengelola kedai dengan cara Menekan tombol *switch buka / tutup kedai*	Tombol switch buka / tutup kedai	Status kedai buka / tutup kedai berubah	Sesuai harapan	Valid
9	Menampilkan halaman dashboard sesuai jenis pengguna dengan cara Masuk ke halaman dashboard sesuai jenis pengguna (pengelola kedai / pramusaji)	List data dashboard	Semua data dashboard tampil	Sesuai harapan	Valid
10	Membuat pesanan baru oleh pramusaji dengan cara Masuk ke halaman buat, kemudian memilih produk, memasukkan nama pembeli, nomor meja dan jumlah	List makanan Tombol pesanan	Pesanan baru berhasil dibuat olelh pramusaji	Sesuai harapan	Valid

	serta catatan produk				
11	Melihat pesanan berdasarkan nomor meja oleh pramusaji dengan cara Masuk kehalaman daftar pesanan berdasarkan nomor meja oleh pramusaji	Daftar Nomor meja	Daftar pesanan berdasarkan nomor meja tampil	Sesuai harapan	Valid

KESIMPULAN

Pendampingan yang telah dilakukan bersama CV Ediide Infografika dalam kurun waktu 2 bulan, menghasilkan perancangan desain aplikasi *food court* untuk pengelola kedai dan pramusaji berbasis android. Pendampingan ini dilakukan karena sistem pelayanan tradisional pada *food court* dengan konsep pesan di meja makan dirasakan kurang efisien, sehingga diperlukan inovasi baru guna meningkatkan pelayanan. Oleh karena itu, hal yang dapat dilakukan ialah melakukan pendampingan sebuah desain aplikasi *food court* untuk pengelola kedai dan pramusaji berbasis *android*. Rancangan desain aplikasi yang telah selesai dikembangkan berhasil memenuhi kebutuhan fitur. Pendampingan desain aplikasi *food court* untuk pengelola kedai dan pramusaji berbasis android dilaksanakan dengan baik, dibuktikan dengan hasil dari *black box testing* yang telah dilakukan.

DAFTAR PUSTAKA

Aulia, R., Zakir, A., Dafitri, H., Siregar, D., & Hasdiana, H. (2017). Mechanism of Food Ordering in A Restaurant Using Android Technology. *Journal of Physics: Conference Series*, *930*(1). https://doi.org/10.1088/1742-6596/930/1/012030

Bhargave, A., Jadhav, N., Joshi, A., Oke, P., & Lahane, S. R. (2013). Digital Ordering System for Restaurant Using Android. *International Journal of Scientific and Research Publications*, *3*(4), 1–7. www.ijsrp.org

Dagne, L. (2019). Flutter for Cross-Platform App and SDK Development. *Metropolia University of Applied Sciences*, *May*. https://www.theseus.fi/bitstream/handle/10024/172866/Lukas Dagne Thesis.pdf?sequence=2&isAllowed=y

Fitriandi, A. Y. (2013). *Studi Kasus Food Court Atau Pujasera) O Leh Service Atau Pemesanan Makanan Dan Minuman (Studi Kasus : Pujasera Atau Food Court*. 62.

Home, pin. (2021). Diakses Oktober 05, 2021 dari *Apa Itu Food Court?* Pin Home. https://www.pinhome.id/kamus-istilah-properti/food-court/

Kubde, P., D'Mello, D., D'souza, S., Falcao, C., & Thomas, J. (2016). Connecting Food Courts in Mall using Mobile App. *International Journal of Computer Applications*, *139*(13), 9–11. https://doi.org/10.5120/ijca2016909508

Lazareska, L. (2017). Analysis of the Advantages and Disadvantages of Android and iOS Systems and Converting Applications from Android to iOS Platform and Vice Versa. *American Journal of Software Engineering and Applications*, *6*(5), 116. https://doi.org/10.11648/j.ajsea.20170605.11

Onainor, E. R. (2019). *Analysis of Consumers' Perception at the Food Court of Lampung Walk by Using Multidimensional Scaling Approach*. *1*, 105–112.

Qamar, T., Tariq, S., Aqeel, A., Jaffri, F., & Hussain, T. (2019). SMARTABLE-Digital Ordering System for Restaurant Using Android. *1st International Conference on Computational Sciences and Technologies*, *2019*(April), 189–193.

Reddy K, S., & KGK, C. (2016). An Online Food Court Ordering System. *Journal of Information Technology & Software Engineering*, *6*(4), 6–8. https://doi.org/10.4172/2165-7866.1000183

Shinde, R., Thakare, P., Dhomne, N., & Sarkar, S. (2014). Design and Implementation of Digital dining in Restaurants using Android. *International Journal of Advance Research in Computer Science and Management Studies*, *2*(1), 2321–7782