

PENGEMBANGAN APLIKASI E-KOST BERBASIS WEBSITE MENGUNAKAN METODE *MICROSERVICE*

Development of a Website-based E-KOST Application Using The Microservice Method

Christianto^{1)*}, M. Fauzi Isputrawan²⁾

^{1,2)}Program Studi Sistem Informasi, Universitas Bunda Mulia, Jakarta

Diterima 04 Maret 2022 / Disetujui 29 Maret 2022

ABSTRACT

The rapid development of information technology certainly supports the application of various innovations in making things, especially for boarding service providers. Currently, boarding house marketing business actors still market their rooms conventionally, and boarding house seekers still need to get counseling in the form of articles and information about boarding houses, smartphone application development using microservice architecture where each service is broken down into a database table, with the aim of being able to do it. load data that is balanced, between each service. Application development is also developed using the Agile method, this is intended to carry out more regular development of each feature, which is broken down into Adaptive Software Development. With this application, the results obtained are expected to make it easier for people to carry out more modern business activities, especially in the provision of boarding houses which will continue to develop along with the level of employment and universities that are increasingly developing in the future.

Keywords: Website, Boarding House, Agile, Adaptive Software Development

ABSTRAK

Perkembangan teknologi informasi yang semakin pesat tentunya mendukung penerapan berbagai inovasi dalam membuat sesuatu, terutama untuk penyedia layanan indekos. Saat ini, pelaku usaha bisnis pemasaran kos masih memasarkan kamarnya dengan konvensional, serta pelaku pencari kos masih perlu mendapatkan sebuah penyuluhan berupa artikel dan informasi seputar indekos, pengembangan aplikasi smartphone menggunakan arsitektur *microservice* dimana setiap *service* dipecah-pecah kedalam tabel database, bertujuan untuk dapat melakukan load data yang seimbang, antara *service* masing-masing. Pengembangan aplikasi juga dikembangkan menggunakan metode *Agile* hal ini ditujukan untuk melakukan development yang lebih teratur disetiap fitur, yang dipecah kedalam *Adaptive Software Development*. Dengan adanya aplikasi ini, hasil yang didapatkan diharapkan dapat mempermudah masyarakat dalam melakukan kegiatan bisnis yang lebih modern terutama di jasa penyediaan Indekos yang akan terus berkembang seiring dengan tingkat pekerjaan dan universitas yang semakin berkembang kedepannya.

Kata Kunci: Website, Indekos, Agile, Adaptive Software Development

*Korespondensi Penulis:

E-mail: s31170069@student.ubm.ac.id

PENDAHULUAN

Memasuki era digitalisasi kamar kos menjadi suatu usaha yang bergerak di bidang jasa yang berpotensi karena semakin tingginya tingkat urbanisasi di Indonesia (Setyoko, 2014). Rumah kos juga tersedia dalam berbagai jenis fasilitas dan harga yang diterapkan. Semakin strategis lokasi kos dan fasilitas maka semakin tinggi juga harga sewa yang ditawarkan, sehingga menyulitkan para pencari kos yang berasal dari luar daerah untuk mengetahui harga yang tepat untuk sebuah kamar kos (Sianturi *et al.*, 2018).

E-Kost merupakan sebuah platform yang mempertemukan antara pihak penyedia properti indekos dengan pihak pencari *kost*, E-Kost bertujuan untuk membantu masyarakat yang masih minim pengetahuan mengenai adanya platform untuk mereka pakai dalam mencari *kost* dan memasarkan *kost* mereka, hal ini disebutkan bahwa masyarakat masih kurang dalam pengetahuan tentang adanya aplikasi penyediaan kos.

Skema Aplikasi Utama E-Kost sendiri dibagi menjadi 2 Website yaitu Ownerkost dan KostFinder. Website Ownerkost mempunyai fitur-fitur yang dapat membantu *Owner* dari kos dalam memasarkan kamar mereka untuk ditempati atau disewakan kepada KostFinder. Website KostFinder sendiri dapat dikatakan sebagai sarana pihak pencari dalam mencari *Kost* mereka, adanya integrasi dengan pihak penyedia *payment* seperti (Midtrans), maps (Google API, locationIQ) lalu chatbot untuk mendukung adanya pertanyaan yang dapat dilontarkan dan dapat dijawab secara spontan oleh pihak penyedia chatbot.

TINJAUAN PUSTAKA

A. Sistem Informasi

Menurut O'Brian dikutip oleh Yakub (2012:17) pada buku Pengantar Sistem Informasi, sistem informasi (information system) merupakan kombinasi teratur dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, menyebarkan informasi dalam sebuah organisasi.

B. Indekos

Menurut Hartono dan Gianawati (2013) rumah *kost* memiliki fungsi yang positif, yaitu tempat sebagai rumah sementara, tempat belajar, dan tempat beristirahat. Jika dikaitkan dengan fungsi indekos ditemukan banyak fenomena pindahan indekos yang dilakukan oleh penyewa. Indekos ditulis *kost*, sebagai kependekan dari *kost*, *kost* menurut Wikipedia merupakan frasa dari bahasa Belanda yang artinya "makan di dalam", istilah yang kemudian digunakan bagi seorang yang tinggal di rumah orang lain dengan membayar 10 menurut jangka waktu tertentu, umumnya bulanan. Indekos secara sederhana didefinisikan sebagai menempati satu ruang (kamar) rumah seseorang, dengan perjanjian membayar dalam jumlah tertentu sebagai kompensasi sewa dan fasilitas lain di dalamnya, seperti makan dan perabot yang dipakai.

C. Microservice Arsitektur

Microservices adalah sebuah pendekatan untuk mengembangkan aplikasi dengan rangkaian *service-service* yang kecil, yang mana setiap *service* berjalan pada prosesnya sendiri-sendiri. Setiap *service* dapat berkomunikasi dengan mekanisme yang ringan. Tiap-tiap *service* yang dibuat harus mengenkapsulasi data dengan logika bisnis yang beroperasi pada data itu sendiri, dan hanya dapat diakses melalui *published service interface*. Tidak ada *database* yang dapat diakses secara langsung dari luar *service* dan tidak ada data yang disharing antara setiap *service*.

Karena mereka berdiri sendiri, kecil kemungkinan kegagalan sistem terjadi saat salah satu *service* mengalami kesalahan. Dengan *microservice*, kita dapat mencampur dan menggunakan beragam bahasa pemrograman, framework, dan teknologi penyimpanan database yang digunakan. (Refactory, 2017).

METODE PENELITIAN

A. Metode Pengumpulan Data

Didalam penelitian ini, peneliti menggunakan beberapa metode dalam pengumpulan data, yaitu :

- Dokumentasi adalah teknik pengumpulan data dengan cara melihat dokumen-dokumen dalam bentuk tulisan, gambar, atau data-data yang bersangkutan.
- Kuisisioner adalah teknik pengumpulan data atau informasi pada populasi yang besar dengan menggunakan sampel yang relatif lebih kecil.

B. Metode Pengembangan

Pada penelitian ini, metode pengembangan aplikasi yang digunakan adalah *agile*. Metode *agile* adalah suatu proses pengembangan perangkat lunak yang bersifat efektif dan sangat lekat untuk pengembangan jangka pendek, dan penuh akan perubahan di dalamnya tanpa mengganggu proses pengembangan yang sedang berjalan. Hal ini sesuai dengan tujuan pengembangan metode *agile* yang menekankan pada kepuasan pelanggan dan kecepatan implementasi, karena terus dituntut untuk melakukan pengembangan secara berkala dan cepat.

HASIL DAN PEMBAHASAN

Gambaran Umum Sistem

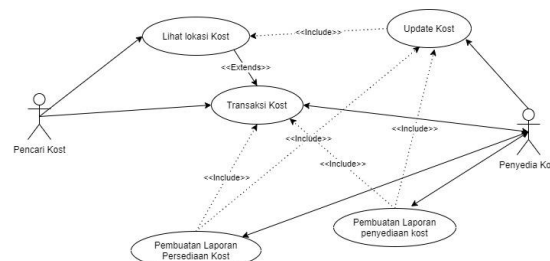
Aplikasi E-KOST berbasis *website* yang memiliki tujuan untuk membantu pemilik dan pencari *kost* dalam memasarkan dan mencari *kost*. Aplikasi E-KOST berbasis *website* memiliki fungsi secara umum untuk melakukan pemesanan *kost* bagi pencari *kost*, dan memasarkan *kost* bagi pemilik *kost*. Sehingga tujuan secara jangka panjang adalah terjadinya komunikasi dua arah antara pemilik dengan penghuni dalam melakukan hubungan timbal balik yang baik ketika menggunakan aplikasi ini.

Aplikasi E-KOST berbasis *website*. Itu berarti proses pemesanan dan pemasaran *kost* secara online, integrasi pembayaran dengan Bank Central Asia, Chat antara *user* dan pemilik *kost*, Notification broadcasting, Reporting keuangan dapat melalui media aplikasi *website* E-KOST. Hanya pencari dan pemilik *kost* yang telah diverifikasi oleh sistem yang bisa melakukan kegiatannya yang menggunakan aplikasi *website* E-KOST.

Penggunaan aplikasi yang dilakukan akan terekam masuk ke dalam sistem aplikasi yang kemudian akan divalidasi dan dikelola oleh sistem E-KOST.

Gambaran Proses Bisnis

A. Proses Bisnis Konvensional



Gambar 1. Usecase Diagram

Dari Gambar 1 dapat dilihat bahwa pemilik *kost* masih menggunakan sistem lama dengan pemasaran melalui mulut ke mulut, untuk transaksi pun masih menggunakan transaksi secara konvensional, dapat dilihat bahwa untuk mendapatkan satu bisnis proses memerlukan begitu banyak tenaga untuk mendapatkan pencari *kost*. Hal ini yang membuat penulis membuat berbagai macam fitur yang telah didiskusikan dengan pengisi kuisisioner. Pihak pencari pun masih kesulitan dalam mencari, padahal ketika mereka sudah menggunakan aplikasi, mereka hanya perlu mencari berdasarkan lokasi terdekat mereka dari tempat mereka bekerja atau berkuliah.

B. Proses Sistem Bisnis Usulan

Aplikasi *website* Pencarian *Kost* menggunakan penyebaran kuesioner terlebih dahulu yang bertujuan untuk mempermudah pembuatan aplikasi serta melakukan pembuatan Menu yang berguna untuk masyarakat yang telah melakukan pengisian kuesioner, hal ini ditujukan dengan antusias masyarakat ketika mengisi kuesioner. Pembuatan Menu dilakukan dengan pengambilan persentase di angka 50% - 100%. Pertanyaan Menu yang disediakan sudah ada dalam perencanaan, kemudian dipertanyakan dalam kuesioner.

Aplikasi *website* Pencarian *Kost* menggunakan pemecahan *website*, dimana *website* pertama berisi *website* yang

dikhususkan untuk *Website* pencari / E-kost, kemudian di lanjutkan dengan *website* lain untuk *owner kost*, dimana *owner kost* memiliki *website* tersendiri untuk memasukan *kost* yang akan disewakan kepada pihak finder melalui *website* Ownerkost.

Pertama *owner* dan E-kost harus melakukan *login* terlebih dahulu, ke masing - masing *website*, *owner login* ke Ownerkost *website*, Pencari *login* ke E-kost *website*. E-kost akan mencari *kost* jika Ownerkost sudah membuat masing-masing *kost* yang mau mereka pasarkan untuk E-kost.

Selain itu beberapa Menu Pemilik *Kost* yang telah disetujui oleh pihak pengisi kuesioner adalah :

1. Menu *Dashboard Screen*

a. Create *Kost*,

Pemilik *kost* dapat membuat *kost* yang ingin dipasarkan melalui Ownerkost *website*.

b. *Edit Kost*,

Pemilik *kost* dapat mengedit / mengupdate *kost* yang ingin dipasarkan melalui Ownerkost *website*.

c. *Delete Kost*

Pemilik *kost* dapat melakukan *delete* ketika *kost* yang dipasarkan sudah penuh / tidak jadi dipasarkan / dibuka kembali.

d. *Dashboard View*

Pemilik *kost* dapat melihat berapa banyak kamar yang tersedia, yang sudah terisi dan sudah dibuat di dalam aplikasi

2. Menu *Inbox/Message Screen*

a. Membaca *Inbox*

Pemilik *kost* dapat membaca *inbox* yang sudah mereka terima dan melakukan balasan ketika *inbox* tersebut merasa perlu dibalas.

b. *Delete Inbox*

Pemilik *kost* dapat mendelete *inbox* yang mereka rasa sudah tidak perlu untuk ditampilkan kembali

3. Menu *List Kost Screen*

a. Melihat *List Rumah Kost*

Melihat *history* dari *kost* dan kamar yang dipasarkan dan melihat apakah *kost* tersebut memiliki laporan keuangan yang sudah baik atau belum.

b. Melihat Detail Kamar

Pemilik dapat melihat detail kamar dari *kost* tersebut agar pemilik dapat lebih rinci dalam melakukan deskripsi dengan baik disetiap kamar dari rumah *kost* yang mereka miliki

c. Melihat *Financial*

Pemilik dapat melihat financial dari kamar *kost* yang mereka miliki

4. Menu *History Booking Screen*

a. Melihat *Booking* dari Pihak Pencari

Pemilik dapat melihat adanya *booking* dari pihak pencari *kost* dalam melakukan transaksi dengan pihak pemilik *kost*.

5. Profil *Screen*

Profil pemilik dapat diedit ketika pemilik merasa sudah tidak menggunakan nama / tempat / email / nomor telepon tersebut.

6. Menu *Pembayaran Screen*

Menu yang wajib ada dalam pembuatan aplikasi E-Kost adalah fitur pembayaran hal ini adalah untuk mempermudah transaksi secara konvensional digantikan dengan digital

Sedangkan untuk Menu pencari *kost* / E-kost dari pengisian dan penyebaran angket yang telah disetujui maka beberapa Menu yang akan dibuat beserta fungsinya adalah :

1. Explore *Screen*

a. Header Maps

Menu yang digunakan oleh pencari *kost* untuk mencari *kost* terdekat lainnya, tempat makan terdekat, laundry dan mini market atau market terdekat dari *kost* yang mereka tinggali.

b. Pencarian *Kost* by *Kost* Populer

Fitur yang telah disediakan oleh sistem ketika mencari *kost* berdasarkan

kota populer yang terbagi menjadi 5 kota populer yaitu Yogyakarta, Jakarta, Bandung dan Surabaya.

c. Pencarian *Kost*

Menu utama dari E-*Kost* adalah sebuah pencarian *kost*, dimana pencari *kost* dapat menemukan *kost* mereka.

d. Artikel E-*Kost*.

Menu yang dibuat untuk memberikan tutorial dalam menemukan *kost* favorite, membuat kamar menjadi nyaman dsb.

e. Detail *List Kost*

Menu untuk menampilkan detail dari masing-masing *list kost* yang ada untuk melihat *kost* tersebut apakah tersedia kamar yang diinginkan atau melihat apakah kamar dari *kost* tersebut memenuhi kriteria dari pencarian pihak pencari.

f. *Booking Kost*.

Menu yang dibuat untuk membuat agreement dengan pihak pencaari dalam melakukan *booking kost* ketika mereka tertarik untuk membayar *kost* yang akan mereka tinggali.

g. *Payment Kost*.

Menampilkan *list* dari penerimaan yang akan diajukan dari pembayaran *kost* misalnya (BCA, Mandiri, OVO, Gopay, dsb). *Payment kost* dilakukan dengan integrasi dengan midtrans.

2. Search Screen.

a. Lihat Peta.

Peta yang dapat dicari menggunakan search bar dapat menemukan *kost* dengan koordinat lokasi yang akan dikirimkan kepada map untuk melihat banyak nya *kost* didalam maps tersebut

b. Daftar *Kost*.

Daftar *kost* ditemukan ketika peta menghasilkan koordinat yang sesuai dengan alamat, atau berdekatan dengan alamat atau maps yang dicari.

3. Chat Screen

a. *Inbox*

Menu *inbox* merupakan Menu yang dibuat untuk melakukan tanya jawab antar penyewa *kost* dengan pencari *kost*, hal ini ditujukan untuk menanyakan apakah tersedianya *kost* yang mereka inginkan.

4. Booking Screen.

a. List Booking Screen

List Booking Screen adalah *list* dari daftar *kost* yang pernah di *booking* dalam kurun waktu tertentu hal ini dibuat untuk menyelaraskan data kapan masuk dan keluarnya pihak penyewa agar lebih terdata dengan baik , dan memberikan notification bahwa *kost* yang telah di*booking* siap di huni.

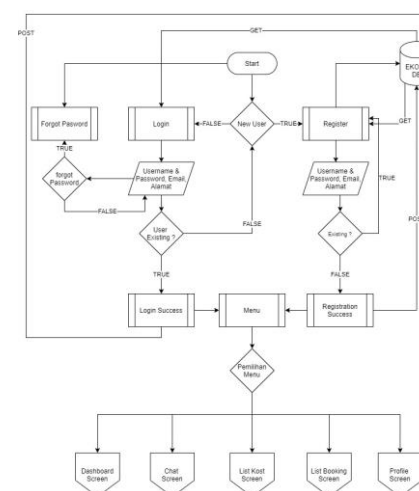
5. Profile Screen

a. Profile

Menu yang dibuat untuk melihat Profil mereka sudah sesuai atau belum disamping itu mereka dapat mengganti apakah profil kurang sesuai dengan data diri mereka.

C. Gambaran Alur Sistem Besar

a) Pemilik *Kost*

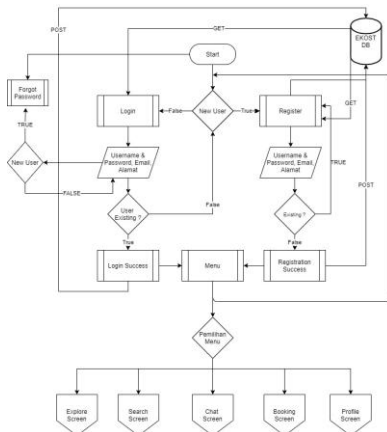


Gambar 2. Flowchart Aplikasi Milik Pemilik

Pada Gambar 2 dilihat proses untuk *user* pemilik. Pemilik harus melalui *login* terlebih dahulu sebelum mengakses menu selanjutnya sesuai dengan akses *user* pemilik. Jika *password* salah maka *user* yang bersangkutan akan tetap berada di tampilan *login*, jika

benar maka *user* pemilik bisa mengakses menu sesuai dengan menu yang telah disediakan namun jika pemilik belum bisa *login*, atau belum memiliki sebuah akun, pemilik dapat membuat akun atau dapat menggunakan *forgot password*. Untuk beberapa penjabaran menu yaitu :

b) Pencari *Kost*

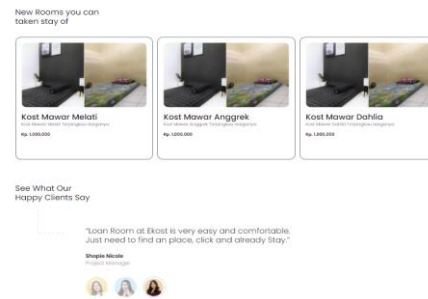
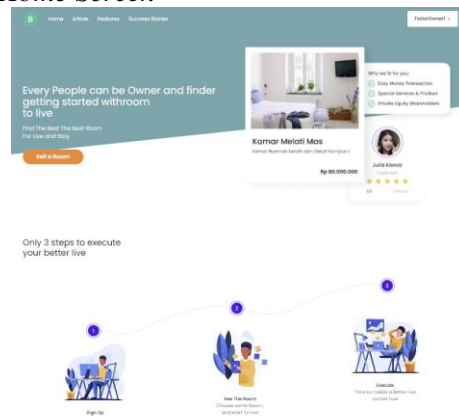


Gambar 3. Flowchart Aplikasi Milik Pencari

Pada Gambar 3, dapat dilihat proses untuk *user* Pencari. Pencari harus melalui *login* terlebih dahulu sebelum mengakses menu selanjutnya sesuai dengan akses *user* pencari. Jika *password* salah maka tetap di halaman *login*, jika benar maka *user* pencari bisa mengakses menu sesuai dengan menu yang telah disediakan. Namun jika pencari belum bisa *login*, atau belum memiliki sebuah akun, pemilik dapat membuat akun atau dapat menggunakan *forgot password*

D. Implementasi

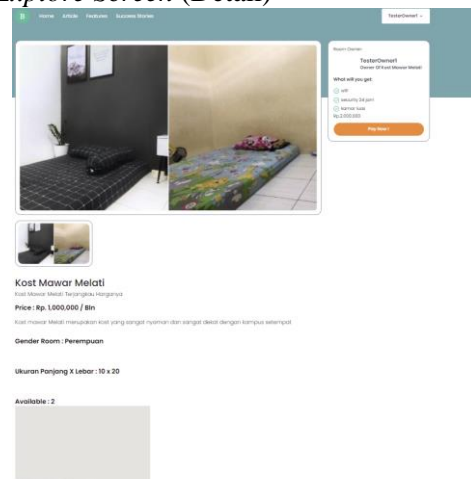
1. Home Screen



Gambar 4. Impelentasi UI Explore Screen (*Kost Finder*)

Gambar 4 merupakan sebuah UI dari implementasi dari tampilan untuk mencari *kost* dibagian Home screen agar pencari dapat mencari *kost* dengan nyaman dan baik, maka disesuaikan dengan UI yang ramah untuk *user*.

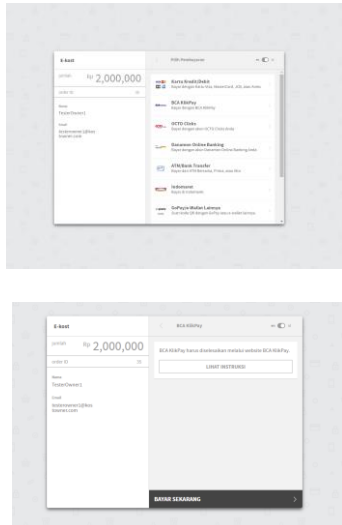
2. Explore Screen (Detail)



Gambar 5. Detail dari List Kost (*Kost Finder*)

Gambar 5 merupakan implementasi dari UI yang aktivitasnya memperlihatkan detail dari *list kost*, yang ketika masuk kedalam detail maka *user* akan dapat melakukan *booking*.

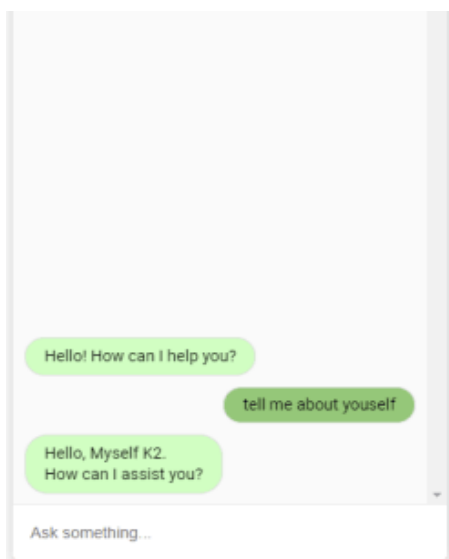
3. Payment dan Booking Detail



Gambar 6. Implementasi UI Payment dan Booking detail. (Kost Finder)

Gambar 6 merupakan implementasi sebuah UI ketika melakukan pembayaran didalam *booking* detail hal ini dilakukan ketika pihak pencari tertarik dan dapat langsung melakukan *booking* kedalam aplikasi, aplikasi yang dibangun sudah terintegrasi dengan API Client milik dari midtrans yang dapat digunakan dan dipakai menggunakan mode SANDBOX

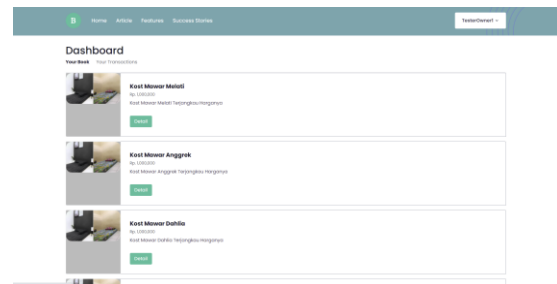
4. Chat Screen dan Detail chat Screen



Gambar 7. List Chat Screen dan Detail Chat Screen

Gambar 7 merupakan sebuah implementsai ui dari *List chat screen* dan detail chat *screen user* dapat melakukan interaksi antar sesama pemilik dan pencari, lalu ada juga untuk chat bot yang memerlukan API 3rd Party dari *snatchBot*. Untuk mempermudah komunikasi antara *user* penciri dan *user* pemilik *kost*

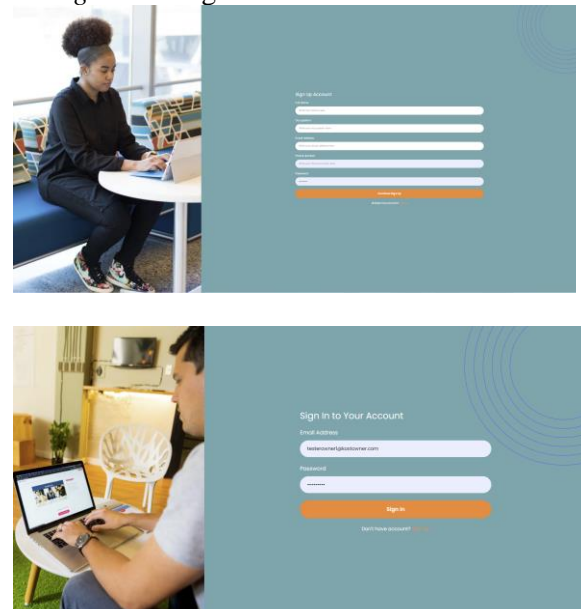
5. List Booking Screen



Gambar 8. Implementasi UI dari List Booking Screen

Gambar 8 merupakan sebuah implementasi UI dari *list booking* yang akan keluar ketika *user* telah melakukan aktivitas pembayaran.

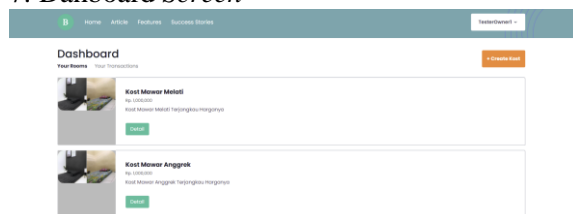
6. Login dan Register



Gambar 9. Implementasi UI dari Logtin dan Register

Gambar 9, merupakan sebuah implementasi UI dari *Login* dan register *website* E-KOST.

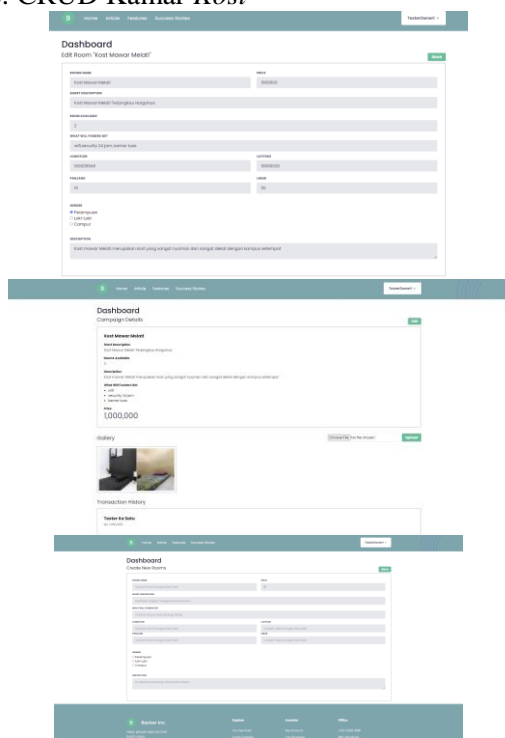
7. Dashboard Screen



Gambar 10. CRUD Pembuatan Rumah Kost

Gambar 10 merupakan implementasi UI dari pembuatan *List Kamar Kost* yang akan disewa nantinya dari pihak pencari *kost* yaitu *Kost Finder*.

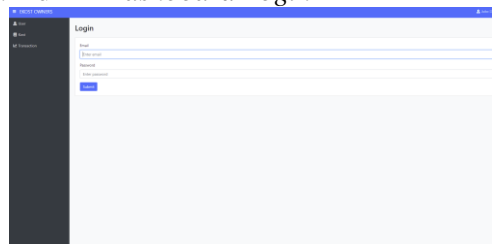
8. CRUD Kamar Kost



Gambar 11. Pembuatan Kamar Kost

Gambar 11 adalah pembuatan Kamar *Kost* yang merupakan main sistem dari *Website* E-kost hal ini ditujukan untuk memasarkan kamar dari Kriteria Rumah *Kost* pada Gambar 11.

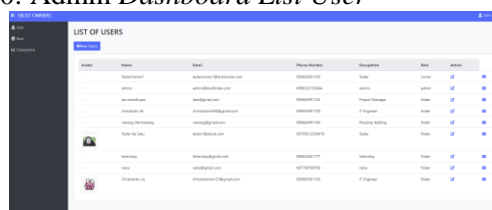
9. Admin Dashboard Login



Gambar 12. Admin Dashboard Login

Gambar 12, admin harus melakukan *login* untuk masuk dan mengolah data yang terdapat dari *website* utama dari E-KOST.

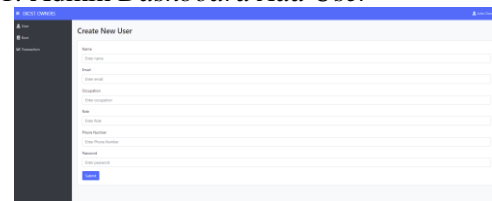
10. Admin Dashboard List User



Gambar 13. Admin Dashboard List User

Gambar 13, admin dapat melihat *list* dari *user* yang ada didalam *website* utama E-KOST.

11. Admin Dashboard Add User



Gambar 14. Admin Dashboard Add User

Gambar 14, admin dapat menambahkan *user* ketika terjadi masalah terkait *login* yang berada didalam *website* E-KOST.

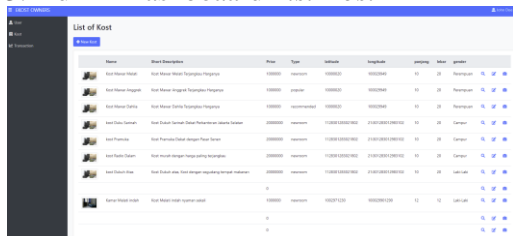
12. Admin Dashboard Edit User



Gambar 15. Admin Dashboard Edit User

Gambar 15, admin dapat melakukan *editing profile user* ketika mereka mendapatkan masalah terkait *user* yang ingin

13. Admin Dashboaard *List Kost*



Gambar 16. Admin *Dashboard* List Kost

14. Admin *Dashboard Add Kost*



Gambar 17. Admin *Dashboard Add Kost* (I)

Gambar 18. Admin *Dashboard Add Kost*
(II)

15. Admin Dashboard Edit Kost

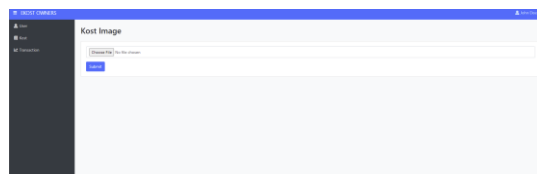


Gambar 19. Admin *Dashboard Edit Kost* (I)



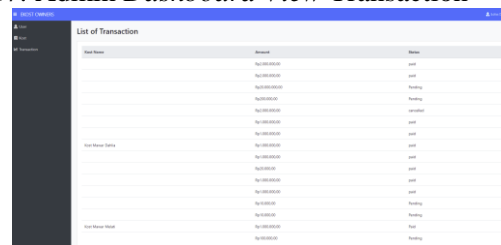
Gambar 20. Admin Dashboard Edit Kost
(II)

16. Admin *Dashboard* Add Image



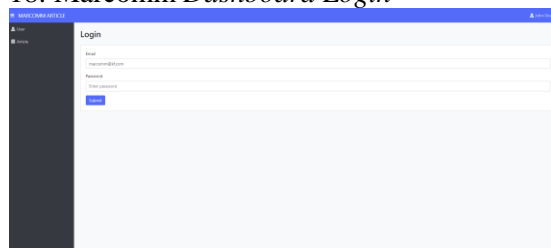
Gambar 21. Admin *Dashboard* Add Image

17. Admin *Dashboard View* Transaction



Gambar 22. Admin Dashboard Add Image

18. Marcomm Dashboard Login



Gambar 23. Marcomm *Dashboard Login*

Gambar 23. Marcomm harus *login* terlebih dahulu untuk melakukan kegiatannya

didalam sistem agar dapat menambahkan article yang berada didalam *website* E-KOST

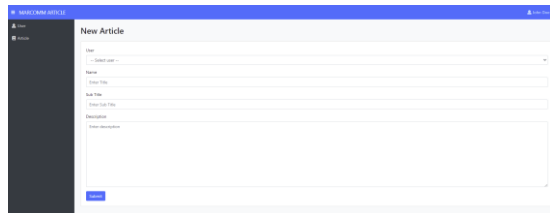
19. Marcomm List Article



Gambar 24. Marcomm Dashboard List Article

Gambar 24. Marcomm dapat melihat adanya *list* dari article yang telah dibuat didalam sistem

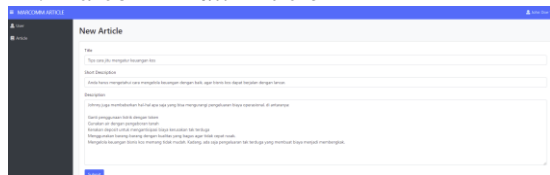
20. Marcomm Add Article



Gambar 25. Marcomm Dashboard Add Article

Gambar 25. Marcomm dapat menambahkan article sesuai dengan apa yang akan mereka buat mengenai article yang menarik untuk *user* finder dan *owner*

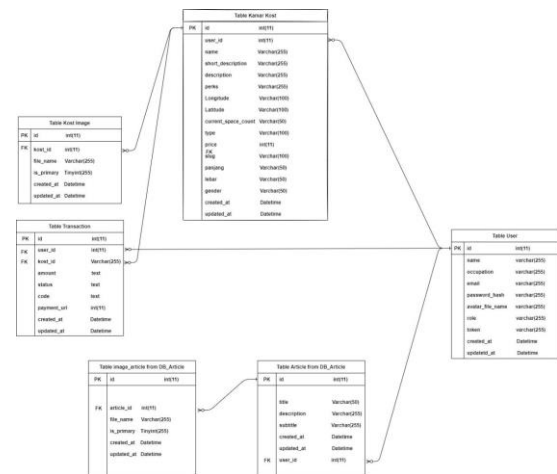
21. Marcomm Edit Article



Gambar 26. Marcomm Dashboard Edit Article

Gambar 26, Marcomm dapat melakukan update ketika mereka salah dalam membuat article yang telah mereka publish.

E. Relasi Database



Gambar 27. Relasi Tabel

Penggunaan 2 database di set untuk mendapatkan konsep *microservice* yaitu dengan adanya 2 DB (Artikel dan Core) dimaksudkan untuk mempercepat kinerja dari sistem core yang dipisahkan dengan database artikel. Karena DB Core akan mengolah *user*, *kost* dan *transaction* yang tidak perlu di campur oleh video, artikel dan foto yang berhubungan dengan konsentrasi kepada penyuluhan.

F. Adaptive Software Development Implementation

1. Speculation

Speculation adalah aktifitas *adaptive cycle planning* yaitu menggunakan informasi awal seperti misi dari klien, batasan proyek dan kebutuhan dasar untuk definisikan rangkaian software increment (produk software yang secara berkala diserahkan)

Tabel 1. Tabel Speculation

Speculation			
No	Requirement User	Ekspetasi	Realitas
1	Mencari kamar <i>kost</i> tanpa perlu keluar rumah	Adanya penyampaian <i>user</i> yang meminta penulis untuk membuat aplikasi yang membantu <i>user</i> dalam	Membuat aplikasi pencarian <i>kost</i> secara go digital, dirancang dan dilakukan secara terstruktur

		membuat aplikasi yang terintegrasi dengan sistem-sistem yang memudahkan	melalui penyebaran kuisioner
2	Tanpa melakukan pengiriman uang secara konvensional untuk melakukan transaksi	Adanya transfer antar bank yang memudahkan <i>user</i> dengan hanya melakukan nya dalam 1 aplikasi tanpa perlu berjalan menuju ATM	Melakukan integrasi dengan sistem pembayaran midtrans sebagai 3 rd Party API
3	Tidak perlu melakukan Reporting Secara Manual	Adanya pembukuan manual yang akan menyulitkan <i>user</i> dalam membuat laporan keuangan secara konvensional	Dibuatkan auto generated pdf untuk memantau keuangan <i>user</i> secara digital tanpa harus membuat laporan keuangan.
4	Tidak perlu melakukan <i>booking</i> secara manual	Adanya maps untuk mengetahui lokasi yang berdekatan dengan laundry, mart, tempat makan dsb.	Melakukan integrasi dengan api google maps, dan membuat beberapa maps checker untuk menandai laundry terdekat, tempat makan terdekat, minimart terdekat dan lokasi <i>kost</i> selain ditempat <i>user</i> tinggal
5	Tampilan Menarik dan	Adanya tampilan	Adanya melakukan

	ramah untuk <i>user</i>	menarik dan ramah akan membantu <i>user</i> dalam menggunakan aplikasi	UI/UX design sebelum melakukan implementasi kedalam tahap development
6.	Memperlihatkan Kontak dengan pemilik / PIC dari rumah dan kamar <i>kost</i>	Adanya kontak dengan berbagai penyedia kamar dan rumah <i>kost</i> untuk di tempati	Melakukan pendaftaran pemilik <i>kost</i> secara detail untuk mendapatkan hasil yang diinginkan antara <i>user</i> pemilik dan <i>user</i> pencari
7	Memasarkan tanpa harus memiliki makelar	Adanya melakukan dan memasarkan produk (rumah dan kamar <i>kost</i> secara online)	Dibuatkan <i>user</i> di list <i>kost</i> yang berada di explore screen
8.	Adanya chatting dengan pemilik <i>kost</i>	Adanya sistem chat (<i>inbox</i> , reply dan messaging) untuk pemilik dan bot untuk bertanya	Dibuatkan chatting antar pemilik dan <i>user</i> serta melakukan integrasi dengan snatch bot sebagai api 3 rd party untuk memproses bot api dalam melakukan tanya jawab secara otomatis
9	Adanya artikel untuk menambah wawasan dari <i>user</i> yang	<i>User</i> ingin untuk mendapatkan pengetahuan ketika	Dibuatkan fitur artikel untuk membantu <i>user</i> dalam

	tidak mengerti soal <i>kost</i> dan hidup mandiri	memilih <i>kost</i> dan bagaimana mencari <i>kost</i>	memilih <i>kost</i> .
--	---	---	-----------------------

2. Collaboration

Collaboration adalah aktifitas orang-orang / individu yang bermotivasi tinggi bekerja sama: saling melengkapi, rela membantu, kerja keras, trampil dibidangnya, dan komunikasikan masalah untuk hasilkan penyelesaian yang efektif

Tabel 2. Tabel Collaboration

Collaboration	
Business User	1. Penyebaran Kuisioner 2. Melakukan Analisa Kuisioner 3. Menggunakan google form 4. Menggunakan IBM SPSS sebagai alat untuk melakukan analisa kuisioner
Software Development Process	Melakukan tahapan-tahapan agile seperti : 1. Memenuhi Requirement 2. Melakukan Design 3. Melakukan Development 4. Melakukan Testing 5. Melakukan Deployment 6. Melakukan Review
Application Developer	Melakukan implementasi pengkodean dengan menggunakan teknologi terbaru dan terbaik untuk di <i>deliver</i> terhadap <i>user</i> yang melakukan pengisian kuisioner
Work	Melakukan analisa keinginan <i>user</i> yang diberikan kepada penulis ketika membuat software yang sudah jadi (seperti feedback dan review)

3. Learning

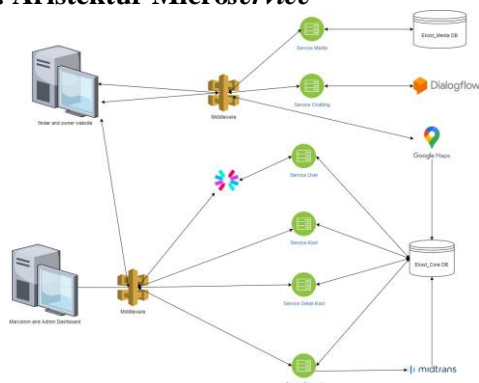
Aktivitas tim/individu sering merasa sudah tahu semua hal tentang proyek. Proses

pembelajaran proyek ini menggunakan 3 cara yaitu :

Tabel 3. Tabel Learning

Learning	
Focus User	Memberikan kuisioner feedback dan review dari apa yang sudah dikerjakan kepada <i>user</i> . (Melalui publish ke Play store dan penyebaran kuisioner) *hasil akan di <i>screen shoot</i> dan ditaruh di dalam lampiran
Formal Technique Reviews	1. Feedback yang dilakukan sudah cukup baik mengingat bahwa requirement yang dikerjakan benar dan murni pengerjaannya bersumber pada kuisioner. 2. Penggunaan library untuk mengembangkan aplikasi <i>website</i> yang digunakan dibuat untuk membantu aplikasi <i>website</i> “E-Kost” 3. Penggunaan Framework serta bahasa pemograman dibuat dan dikhususkan untuk mempermudah kondisi dan situasi antara frontend dan backend yang dipisah dari segi framework dan bahasa , agar lebih mudah untuk di maintenance
Postmortems	1. Waktu untuk mengerjakan sangat terbatas 2. Cakupan dan requirement <i>user</i> memperbesar <i>scope</i> yang ada. 3. Kinerja sudah baik namun harus lebih dipilah dari yang urgent samapai slow

SARAN



Berikut ini adalah beberapa saran perbaikan dan pengembangan aplikasi ini.

1. Perlu adanya support dari beberapa pihak seperti minimarket, tempat makan dan laundry untuk memperbesar besaran aplikasi ini , untuk menentukan rute terdekat dengan *kost* dari usaha yang bersangkutan agar pencari dapat lebih nyaman dan mendekati dengan apa yang mereka cari.
2. Perlu dikembangkan agar aplikasi ini dapat memperbesar besaran aplikasi sehingga tidak hanya *kost* namun juga apartement, hotel dan hostel.

Aplikasi dibagi menjadi 2 *Website* yaitu *Kost Finder Owner Kost*, dan *Dashboard Admin* dan *marcomm* kedua *website* perlu melewati *middleware*, lalu melakukan *generate token* untuk melakukan akses kedalam aplikasi ketika kedua aplikasi sudah melakukan *login*, untuk beberapa hal seperti *service chat* memerlukan *3rd Party API* dengan menggunakan *Diagflow*, lalu untuk pembayaran menggunakan *3rd Party API* dengan menggunakan *api client* dari *midtrans* untuk membantu pembayaran , dilanjutkan dengan *service* alamat / *service address*, menggunakan *Google Maps* yang akan berinteraksi dengan *database core* untuk menyimpan *longitude* dan *latitude*. Untuk *service media* melakukan interaksi tanpa melalui *middleware* dan langsung melalui antara *service* dan *database media*.

Effendy, F., Taufik, & Adhilaksono, B. (2019). Performance Comparison of Web Backend And Database: A Case Study Of Node.JS, Golang and MySQL, MongoDB. *Recent Advances in Computer Science and Communications*, 13. <https://doi.org/10.2174/266625581366191219104133>

Penelitian dalam tugas akhir ini memberikan beberapa kesimpulan sebagai berikut:

1. Penggunaan aplikasi dalam penelitian ini dapat mempermudah memberikan solusi atas pencarian *kost* berbasis *website*.
2. Aplikasi dibuat berdasarkan penggunaan *google maps*, arsitektur *microservice*, serta memberikan edukasi untuk masyarakat mengenai *kost* yang ada disekitar mereka.
3. Penggunaan aplikasi ini dalam pencarian rute pada aktivitas pencarian rumah *kost* memiliki fungsi utama dalam chatting, rute, pembayaran online dan edukasi.

- Gunawan, H., & Saputro, A. K. (2018). Pemanfaatan Aplikasi Mobile Untuk Mempercepat Pencarian Tempat Indekos Berbasis Android. *Jurnal Muara Sains, Teknologi, Kedokteran Dan Ilmu Kesehatan*, 1(2). <https://doi.org/10.24912/jmstik.v1i2.1454>
- Hariman, I., & Meilisa, C. (2020). Sistem Infomasi Manajemen Training Menggunakan Metode Agile Software Development The Papandayan Hotel Bandung. *ENSAINS JOURNAL*, 3(1), 60. <https://doi.org/10.31848/ensains.v3i1.371>
- Hassan, S., Ali, N., & Bahsoon, R. (2017). *Microservice Ambients: An Architectural Meta-Modelling Approach for Microservice*

- Granularity. 2017 *IEEE International Conference on Software Architecture (ICSA)*.
<https://doi.org/10.1109/icsa.2017.32>
- Nurhidayati, N., & Nur, A. M. (2021). Pemanfaatan Aplikasi Android Dalam Rancang Bangun Sistem Informasi Persebaran Indekos di Wilayah Pancor Kabupaten Lombok Timur. *Infotek : Jurnal Informatika Dan Teknologi*, 4(1), 51–62.
<https://doi.org/10.29408/jit.v4i1.2989>
- Paul, A., & Nalwaya, A. (2019). The Ecosystem: Extending React Native. *React Native for Mobile Development*, 225–232. https://doi.org/10.1007/978-1-4842-4454-8_10
- Pengertian Sistem Informasi, Ciri, Fungsi, Komponen & Menurut Para Ahli*. Artikelsiana. (2020, September 11). <https://artikelsiana.com/pengertian-sistem-informasi-ciri-fungsi/>.
- Sachdeva, S. (2016). `Scrum Methodology. *International Journal Of Engineering And Computer Science*.
<https://doi.org/10.18535/ijecs/v5i6.11>
- Wismar, L., & Ginardi, R. V. H. (2017). Rekomendasi Indekos dengan Metode Pembobotan pada Aplikasi E-Commerce CariKos Berbasis Web. *Jurnal Teknik ITS*, 6(2).
<https://doi.org/10.12962/j23373539.v6i2.23946>