# BAB III ANALISIS DAN PERANCANGAN

## Analisis

### Identifikasi Masalah

Berdasarkan saran dari penelitian yang dilakukan (Hamzah, 2020) adalah perancangan UI/UX selanjutnya disarankan agar prototipe yang dihasilkan dapat dijalankan sebagaimana mestinya menggunakan aplikasi mobile. Secara keseluruhan, meskipun hanya berupa frontend prototype dan dalam prototype tersebut bisa membuat data secara lokal, namun penting untuk diingat bahwa penyimpanan data secara lokal hanya bersifat lokal pada perangkat pengguna tersebut. Jika pengguna menggunakan aplikasi di perangkat lain atau menghapus aplikasi, data lokal tersebut tidak akan tersedia.

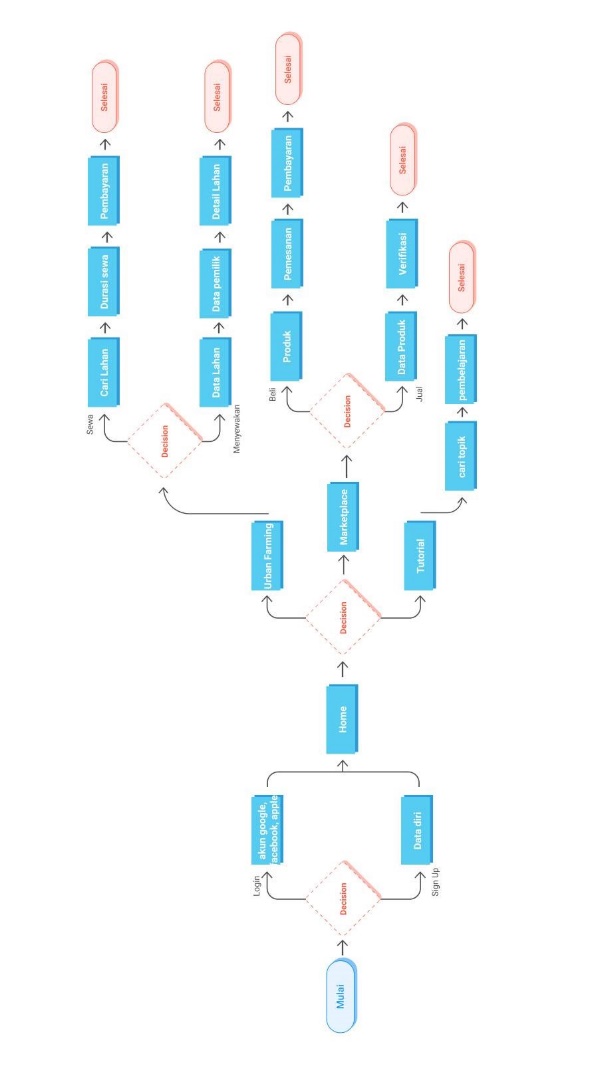
### Pemecahan Masalah

Dari identifikasi masalah diatas, maka diperlukan tujuan seperti sinkronisasi data antar perangkat atau pemulihan data sehingga backend masih diperlukan untuk menyimpan data secara terpusat dan memberikan akses yang konsisten di berbagai platform disinilah peran backend untuk aplikasi Tandur sangat penting. Backend memungkinkan untuk mengelola data, mengimplementasikan fungsionalitas interaktif, meningkatkan keamanan, meningkatkan kinerja, dan memudahkan pemeliharaan dan pengembangan. Adapun API Tandur yang akan dikembangkan menggunakan framework PHP yaitu Laravel yang mana powerful, cepat, aman, dan sederhana. Dengan terciptanya aplikasi backend Tandur yang baik, dapat memberikan kemudahan dalam penggunaan aplikasi serta pengaksesan fitur-fitur sehingga dapat meningkatkan dampak yang tinggi dari aplikasi Tandur di berbagai platform nantinya.

## Perancangan

### Perancangan Sistem

Perancangan sistem bertujuan untuk mendapatkan gambaran mengenai aplikasi yang akan dikembangkan dalam rangka pencapaian efektifitas dan efisiensi sistem. Alur flowchart dari tugas akhir (Hamzah, 2020) bisa dilihat sebagai berikut:

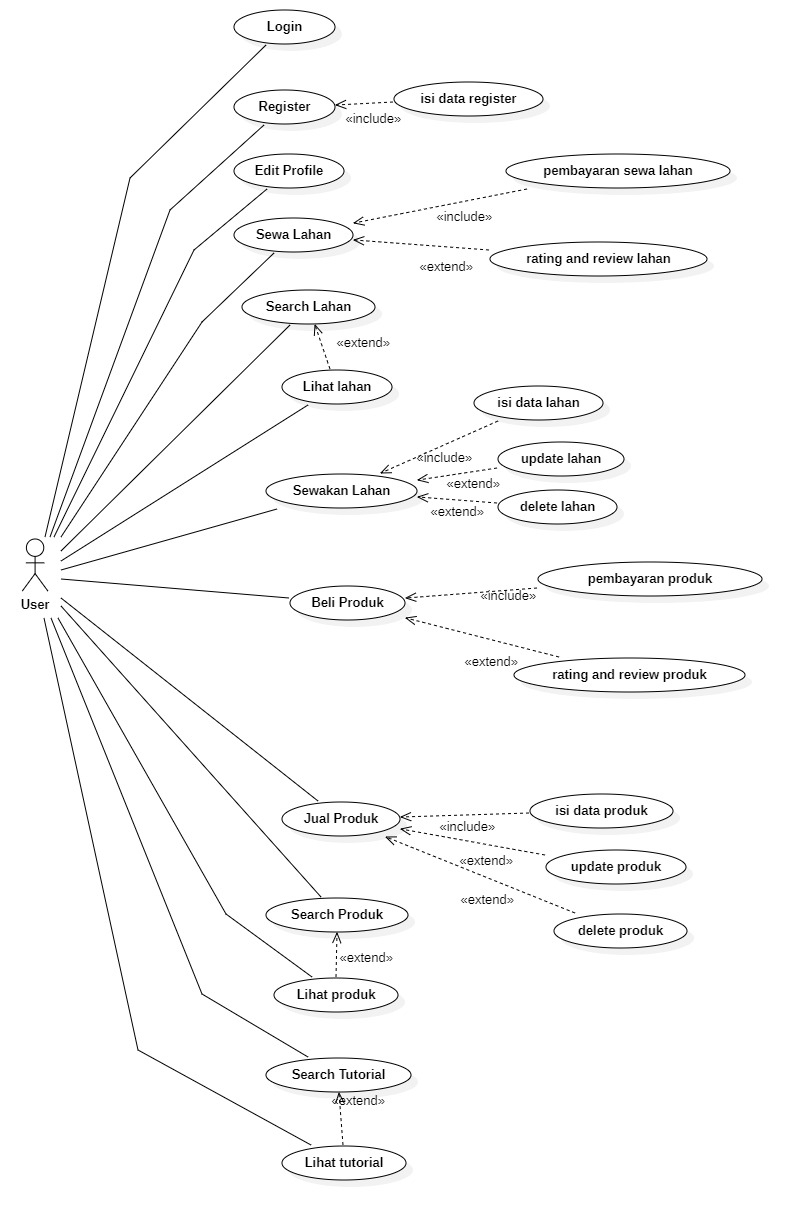


Gambar 3.1 Flowchart Tandur Azrul

Dalam penelitian ini digunakan use case diagram dan sequence diagram untuk mengetahui rancangan sistemnya. Use case diagram menggambarkan interaksi antara actor dengan sistem. Dalam use case diagram ada 2 aktor yaitu user (pemilik lahan, penyewa, penjual, pembeli) dan admin tandur. Lalu ada sequence diagram yang menggambarkan urutan pesan yang dikirim antara objek dalam suatu sistem. Dengan menggunakan flowchart Tandur sebagai referensi, penulis dapat memastikan bahwa use case dan sequence diagram yang dibuat mencerminkan alur logika dan proses yang ada dalam aplikasi Tandur dengan benar.

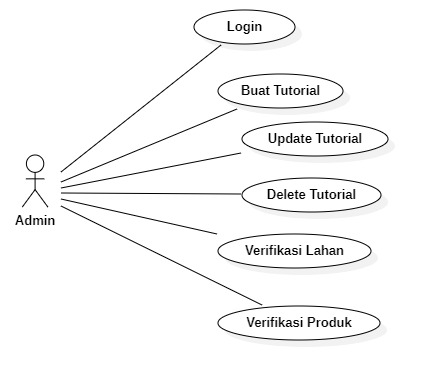
Dalam perancangan use case diagram dan sequence diagram penulis sudah menerapkan teori dasar dalam perancangan pada bab II. Dengan memahami teori UML, penulis dapat menggunakan konsep dan elemen UML untuk membentuk perancangan use case diagram dan sequence diagram yang jelas dan terstruktur. Dari teori uml tersebut penulis bisa membentuk rancangan sistem sebagai berikut :

1. Use Case Diagram User



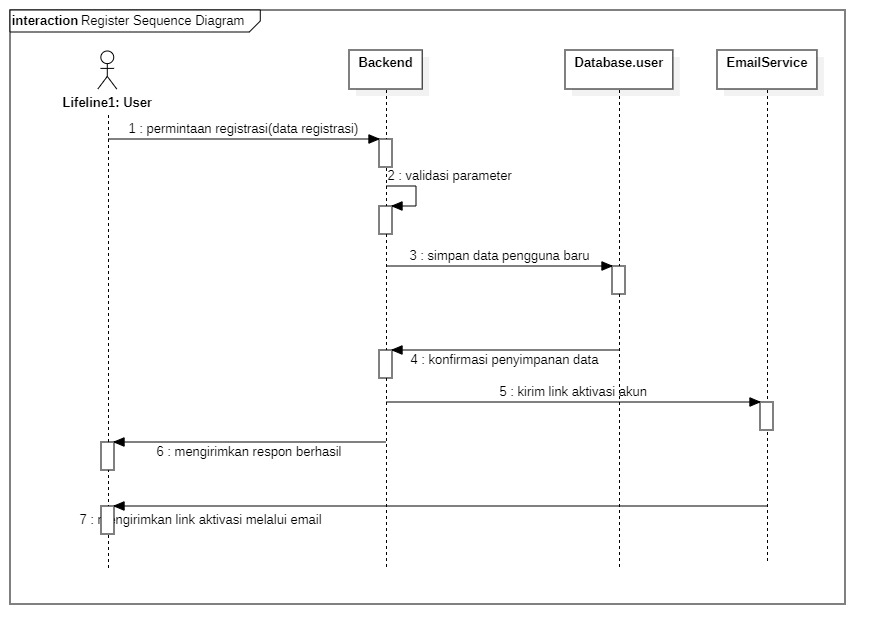
Gambar 3.2 Use Case Diagram User

1. Use Case Diagram Admin



Gambar 3.3 Use Case Diagram Admin

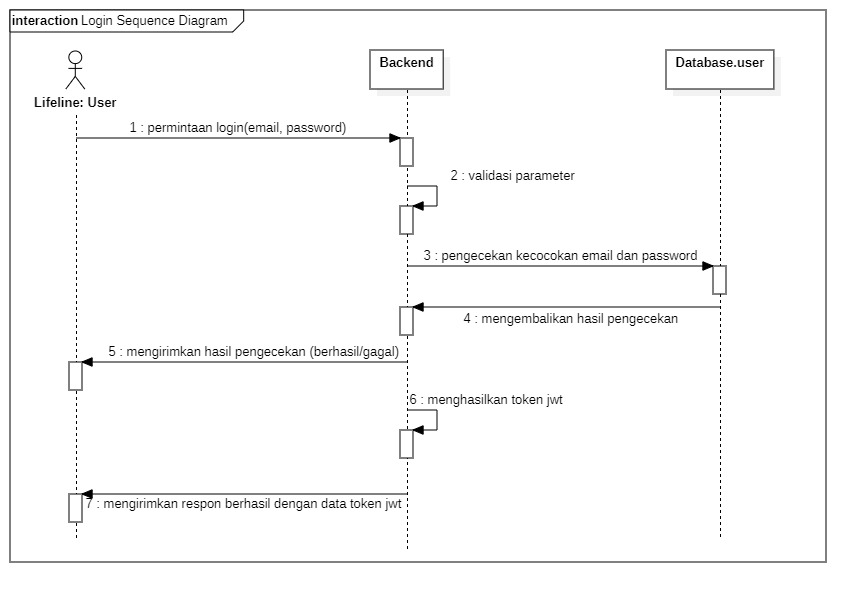
1. Sequence Diagram Register



Gambar 3.4 Sequence Diagram Register

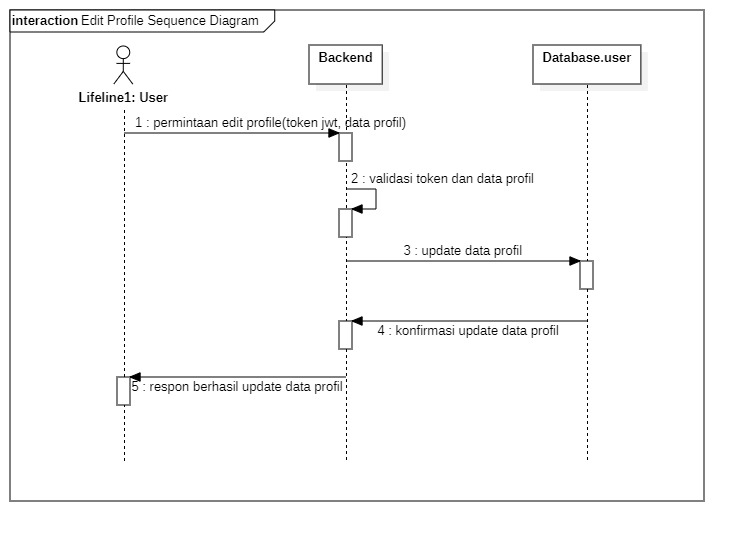
User request registrasi dan backed melakukan pengecekan parameter yang dikirim dan backend menyimpan data registrasi dan mengirimkan email kepada user yang telah registrasi

1. Sequence Diagram Login

Gambar 3.5 Sequence Diagram Login

User melakukan login dengan email dan password, lalu backend mengecek apakah user tersebut telah tersimpan di database dengan tabel user. Dan mengirimkan respon bersamaan dengan token jwt yang nantinya akan digunakan sebagai akses ke Api Tandur.

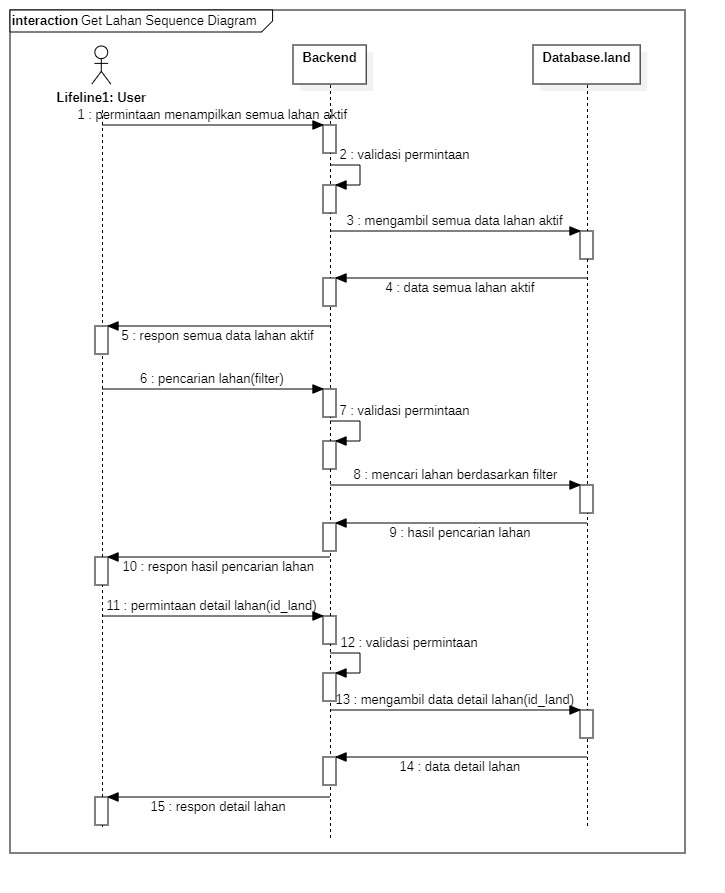
1. Sequence Diagram Edit Profil



Gambar 3.6 Sequence Diagram Edit Profil

User request edit profile dengan mengirimkan token dan data profil, selanjutnya backend validasi token dan mengubah profil pada tabel user dan mengirimkan respon.

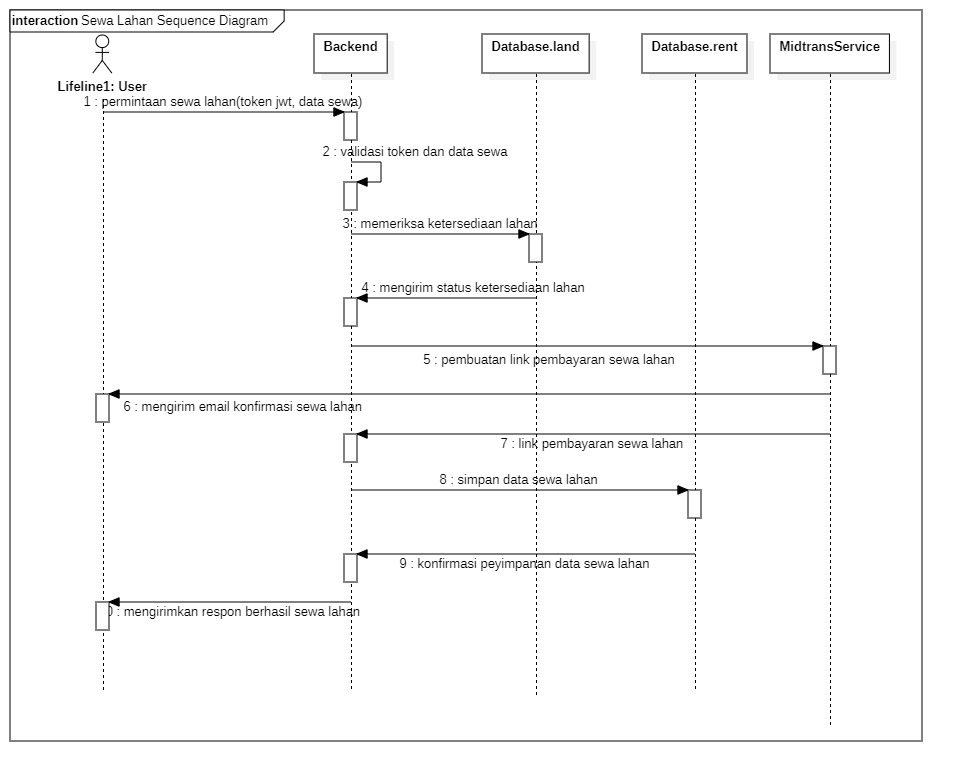
1. Sequence Diagram Lahan dan Detail Lahan



Gambar 3.7 Sequence Diagram Lahan dan Detail Lahan

User request menampilkan semua lahan dan backend melakukan pengambilan lahan aktif dari tabel land yang nantinya dikirimkan kembali dalam bentuk respon. Lalu user melakukan request untuk detail land dan backend melakukan pengambilan detail lahan berdasarkan id land dan mengirimkan kembali kepada user respon detail land.

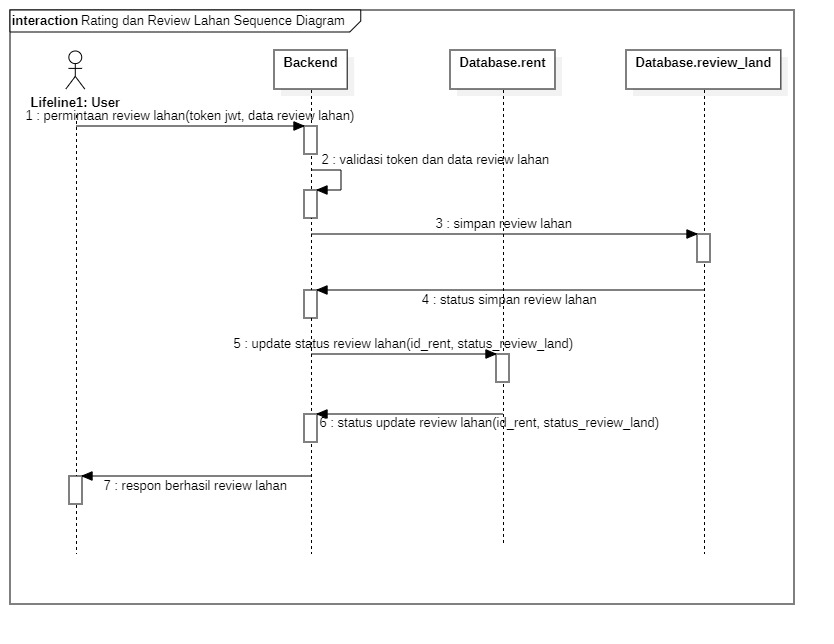
1. Sequence Diagram Sewa Lahan



Gambar 3.8 Sequence Diagram Sewa Lahan

User request sewa lahan dan backend validasi token dan data sewa yang dikirim, selanjutnya backend pengecekan lahan yang akan disewa dan mengirimkan request pembayaran ke midtrans dan midtrans memberi email konfirmasi kepada user. Setelah itu backend menyimpan data sewa dan payment yang akan diberikan kepada user dalam bentuk respon.

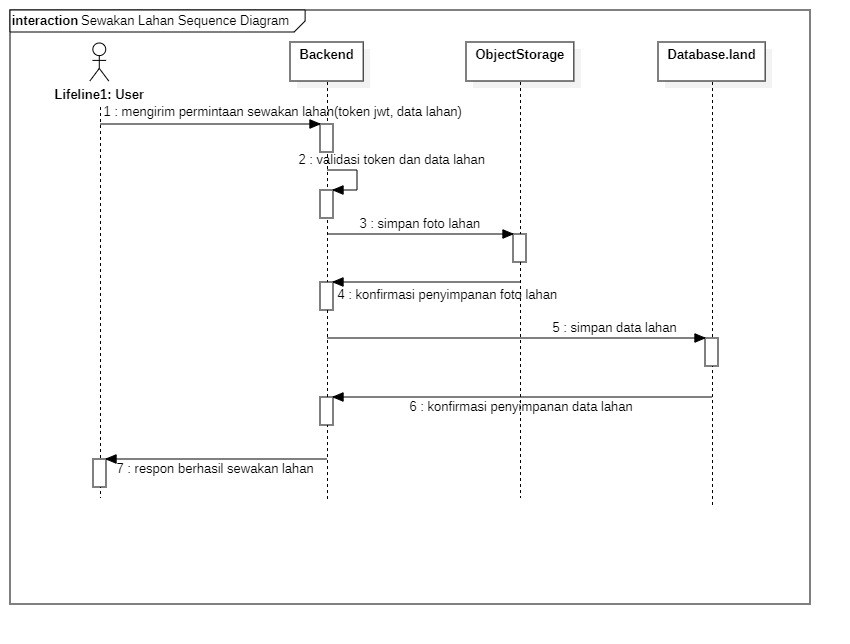
1. Sequence Diagram Rating dan Review Lahan



Gambar 3.9 Sequence Diagram Rating dan Review Lahan

User request memberikan review lahan dengan mengirim token dan data review lahan lalu backend menyimpan review lahan dan mengubah status review pada rent dan memberikan respon kepada user.

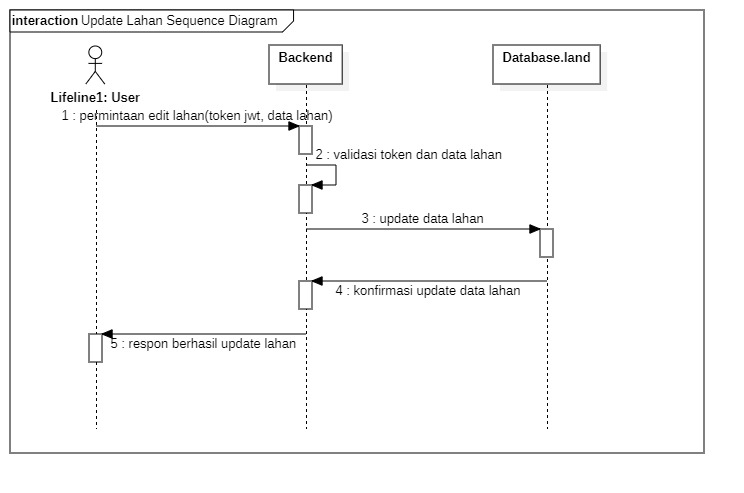
1. Sequence Diagram Sewakan Lahan



Gambar 3.10 Sequence Diagram Sewakan Lahan

User request sewakan lahan dengan token dan data lahan selanjutnya backend menyimpan gambar pada object storage dan menyimpan data lahan pada tabel land dan memberikan respon ke user.

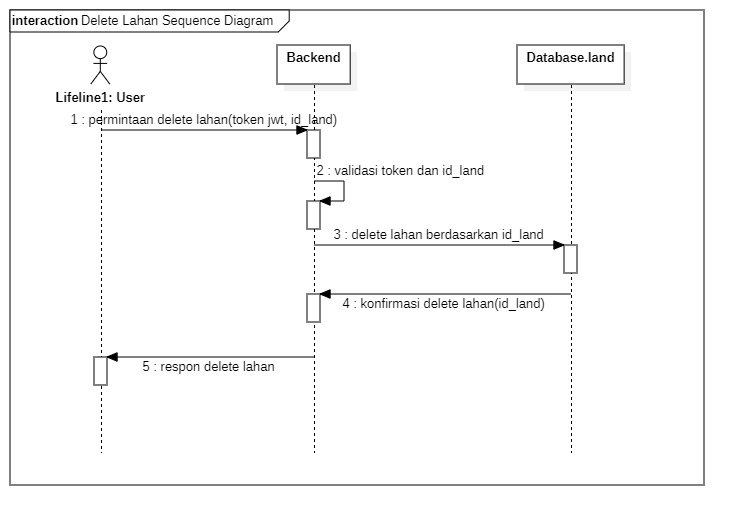
1. Sequence Diagram Update Lahan



Gambar 3.11 Sequence Diagram Update Lahan

User request mengubah lahan dengan mengirim jwt dan data lahan lalu backend validasi token dan data lahan dan mengupdate data lahan pada tabel land serta memberikan respon ke user.

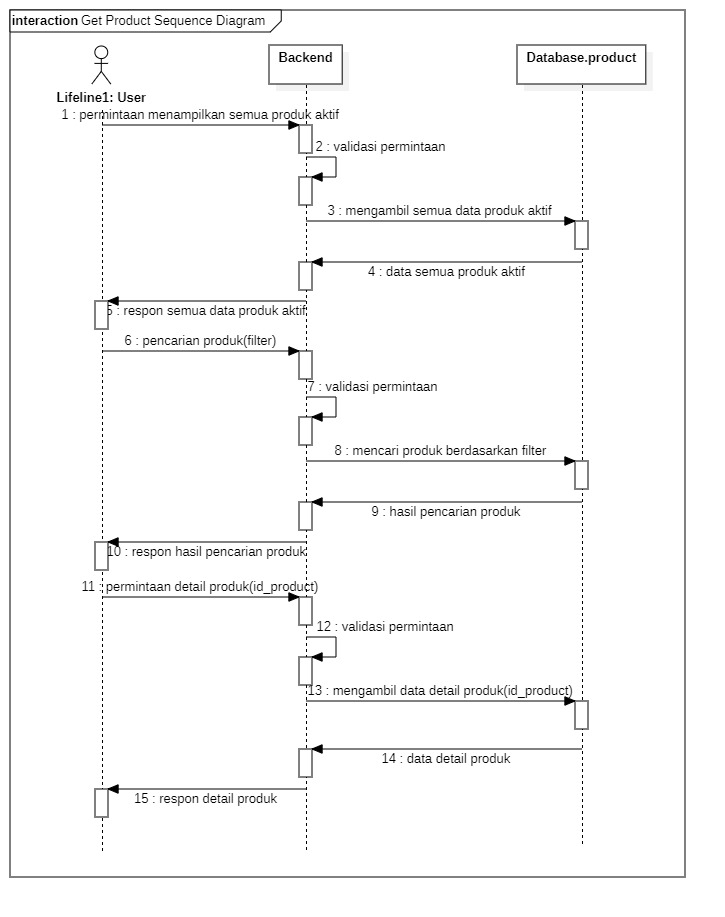
1. Sequence Diagram Delete Lahan



Gambar 3.12 Sequence Diagram Delete Lahan

User menghapus lahan dengan mengirimkan jwt dan id land lalu backend menghapus data lahan sesuai id land yang dikirim dan memberikan respon terhadap user.

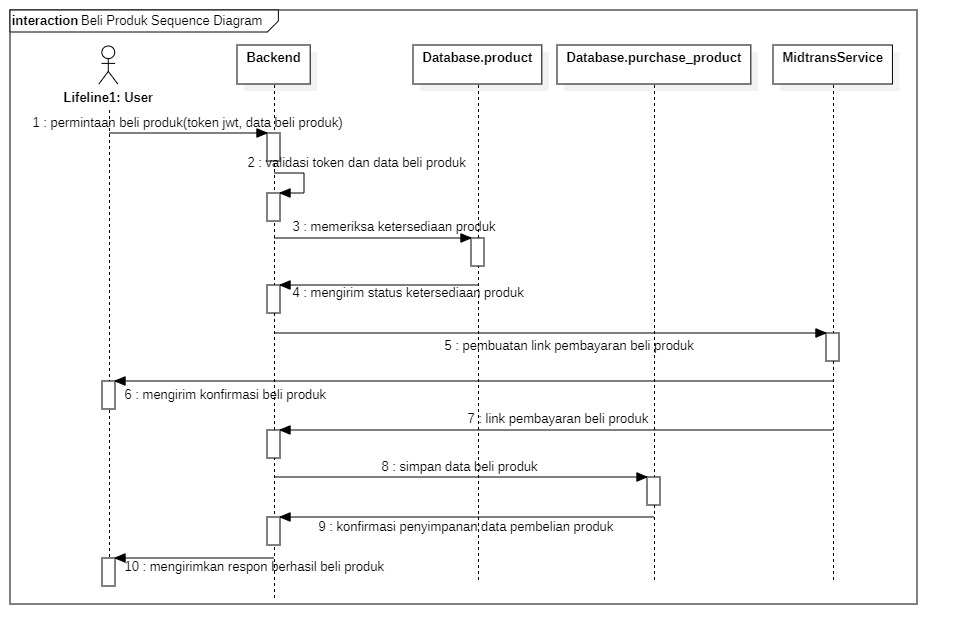
1. Sequence Diagram Produk dan Produk Detail



Gambar 3.13 Sequence Diagram Produk dan Produk Detail

User request menampilkan semua produk dan backend melakukan pengambilan produk aktif dari tabel product yang nantinya dikirimkan kembali dalam bentuk respon. Lalu user melakukan request untuk detail produk dan backend melakukan pengambilan detail produk berdasarkan id product dan mengirimkan kembali kepada user respon detail produk.

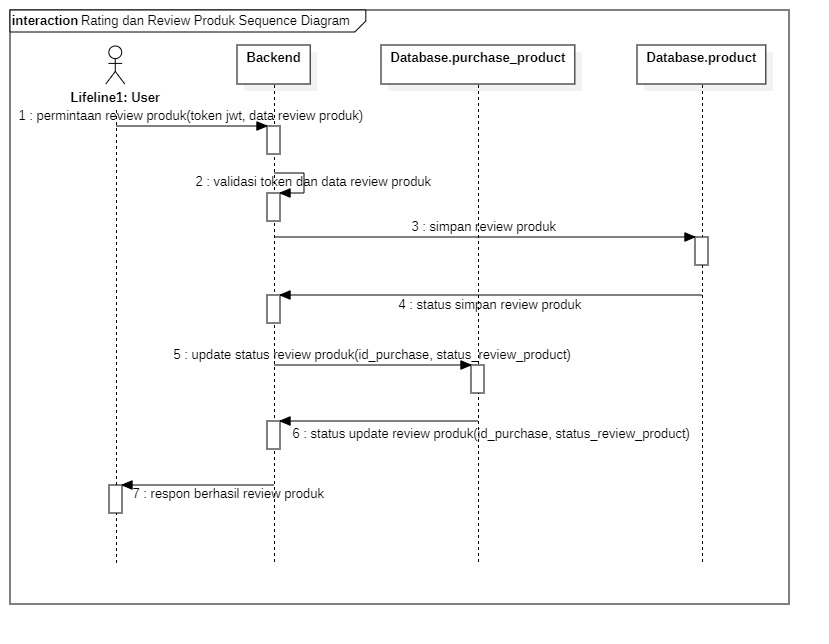
1. Sequence Diagram Beli Produk



Gambar 3.14 Sequence Diagram Beli Produk

User request beli produk dengan mengirim jwt dan produk yang dibeli selanjutnya backend validasi token dan data produk yang dikirim. Backend melakukan pengecekan lahan yang akan dibeli dan mengirimkan request pembayaran ke midtrans dan midtrans memberi email konfirmasi kepada user. Setelah itu backend menyimpan data pembelian dan payment yang akan diberikan kepada user dalam bentuk respon.

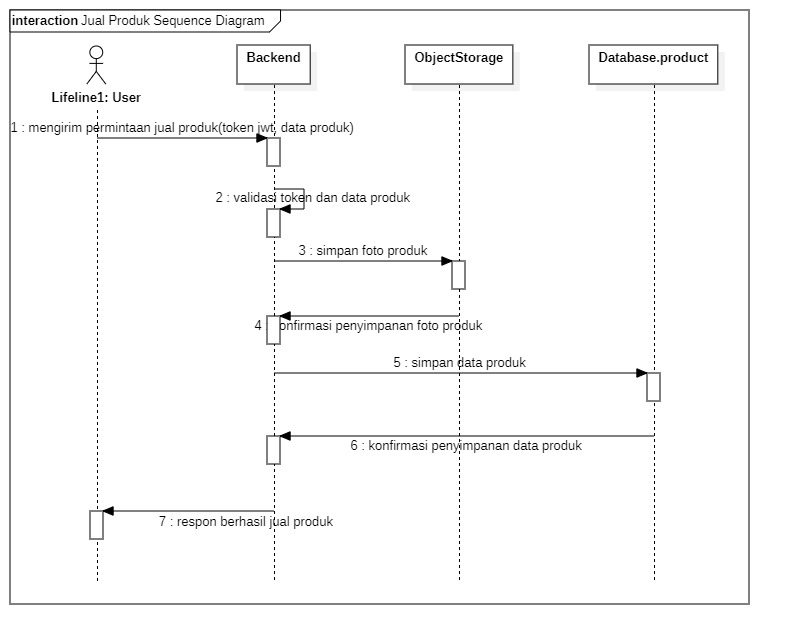
1. Sequence Diagram Rating dan Review Produk



Gambar 3.15 Sequence Diagram Rating dan Review Produk

User request memberikan review produk dengan mengirim token dan data review lahan lalu backend menyimpan review lahan dan mengubah status review pada purchase\_product dan memberikan respon kepada user.

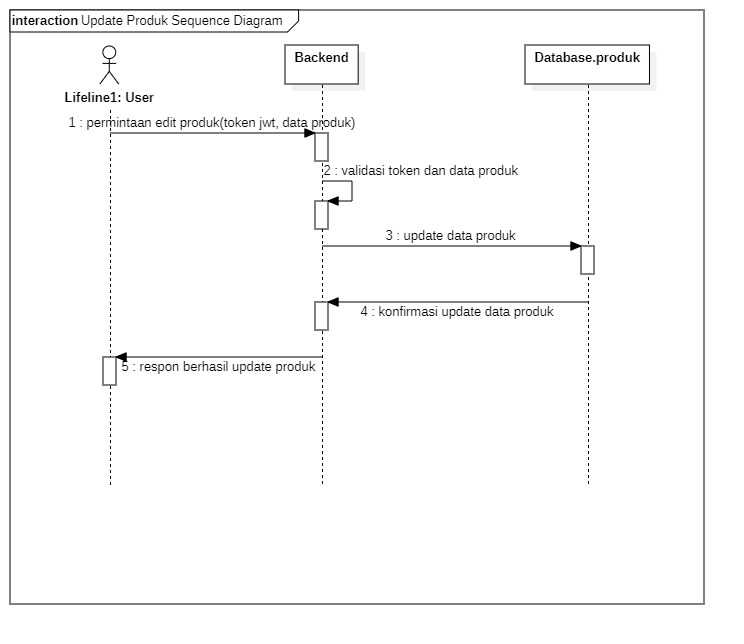
1. Sequence Diagram Jual Produk



Gambar 3.16 Sequence Diagram Jual Produk

User request menjual produk dengan token dan data produk selanjutnya backend menyimpan gambar pada object storage dan menyimpan data produk pada tabel produk dan memberikan respon ke user.

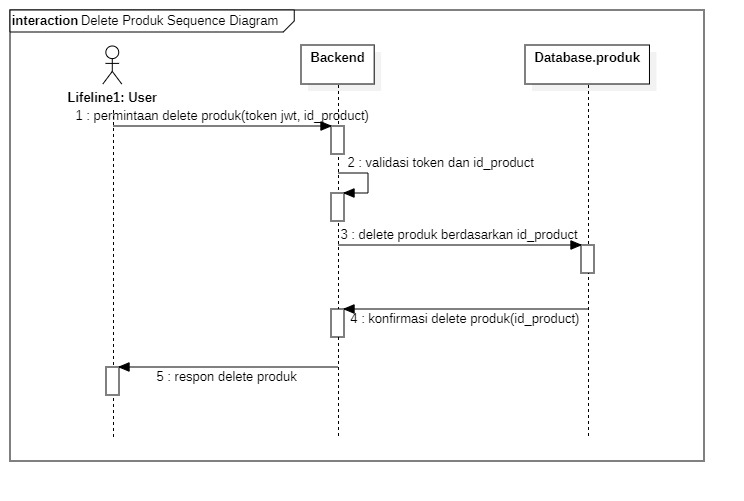
1. Sequence Diagram Update Produk



Gambar 3.17 Sequence Diagram Update Produk

User request mengubah produk dengan mengirim jwt dan data produk lalu backend validasi token dan data produk serta mengupdate data produk pada tabel produk dan memberikan respon ke user.

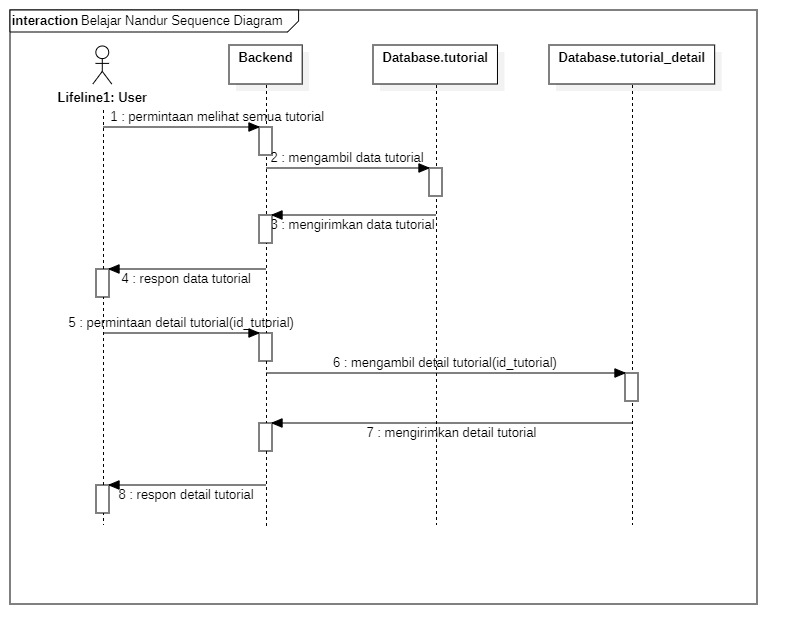
1. Sequence Diagram Delete Produk



Gambar 3.18 Sequence Diagram Delete Produk

User menghapus produk dengan mengirimkan jwt dan id product lalu backend menghapus data produk sesuai id\_product yang dikirim dan memberikan respon terhadap user.

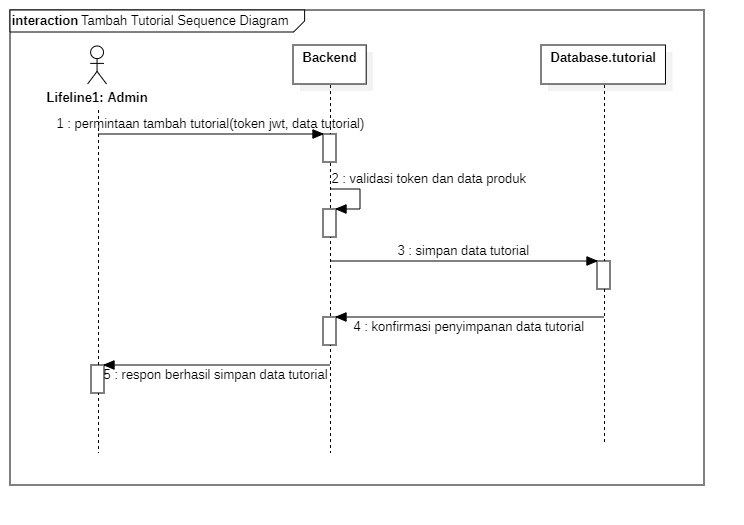
1. Sequence Diagram Belajar Nandur dan Detail



Gambar 3.19 Sequence Diagram Belajar Nandur dan Detail

User request menampilkan semua tutorial dan backend melakukan pengambilan data tutorial dari tabel tutorial yang nantinya dikirimkan kembali ke user dalam bentuk respon. Lalu user melakukan request untuk detail tutorial dan backend melakukan pengambilan detail tutorial berdasarkan id\_tutorial serta mengirimkan kembali kepada user respon detail tutorial.

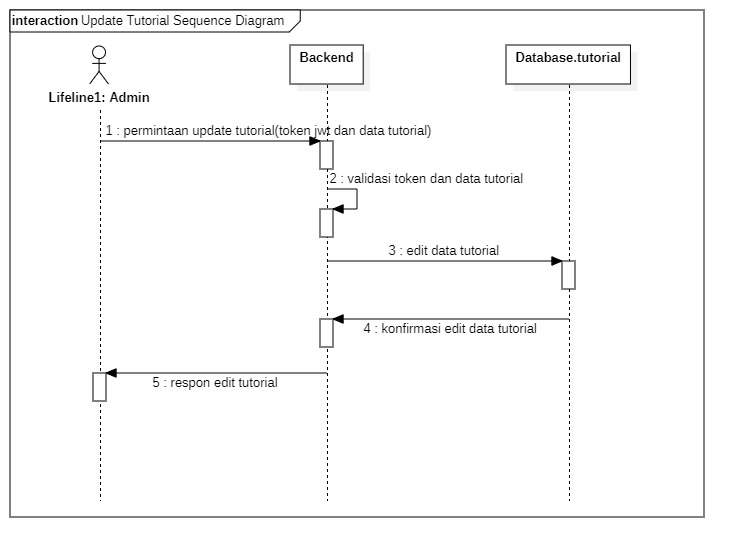
1. Sequence Diagram Tambah Tutorial



Gambar 3.20 Sequence Diagram Tambah Tutorial

Admin request tambah tutorial dengan mengirimkan token jwt dan data tutorial dan backend melakukan validasi token dan data tutorial. Selanjutnya backend menyimpan data tutorial dan memberikan respon ke admin.

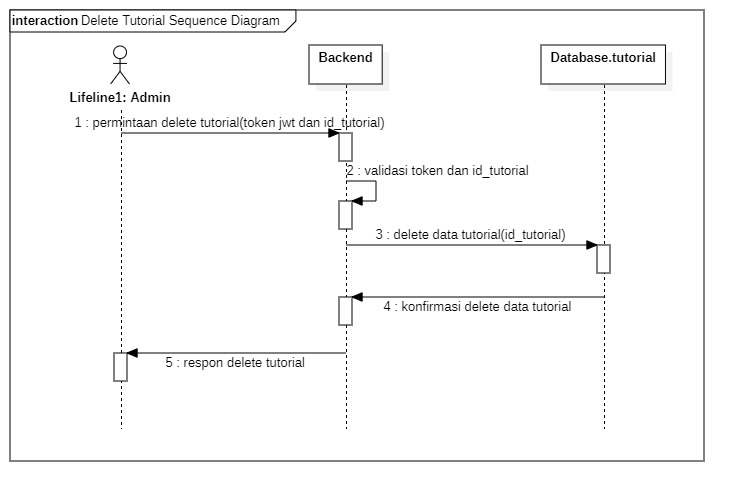
1. Sequence Diagram Update Tutorial



Gambar 3.21 Sequence Diagram Update Tutorial

Admin request update tutorial dengan mengirimkan token jwt dan data tutorial dan backend melakukan validasi token dan data tutorial. Selanjutnya backend mengubah data tutorial dan memberikan respon ke admin.

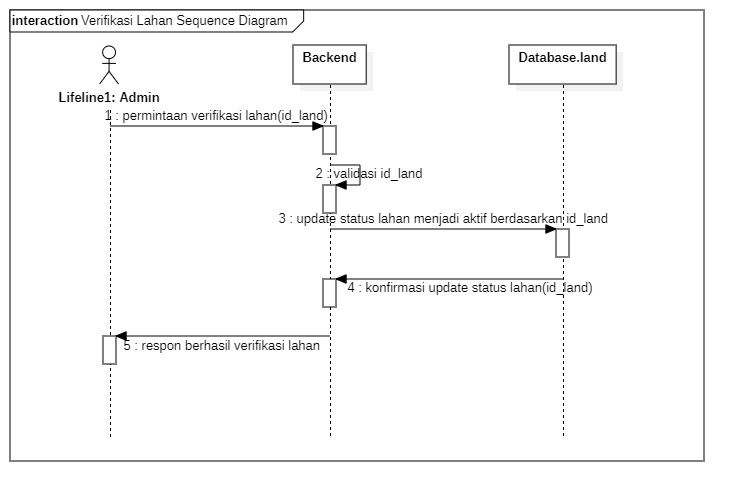
1. Sequence Diagram Delete Tutorial



Gambar 3.22 Sequence Diagram Delete Tutorial

Admin request hapus tutorial dengan mengirimkan token jwt dan id\_tutorial dan backend melakukan validasi token dan id\_tutorial. Selanjutnya backend menghapus data tutorial dan memberikan respon ke admin.

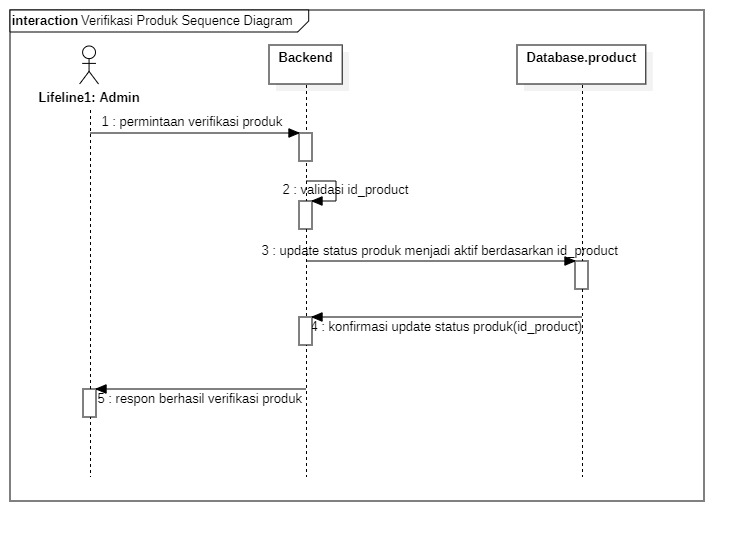
1. Sequence Diagram Verifikasi Lahan



Gambar 3.23 Sequence Diagram Verifikasi Lahan

Admin request verifikasi lahan dengan mengirimkan token jwt dan id\_land dan backend melakukan validasi token dan id\_land. Selanjutnya backend menyimpan mengubah stataus lahan menjadi aktif dan memberikan respon ke admin.

1. Sequence Diagram Verifikasi Produk



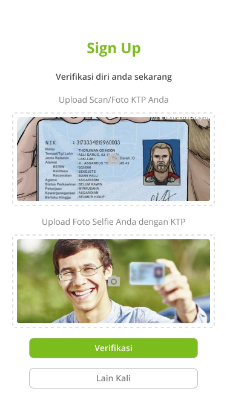
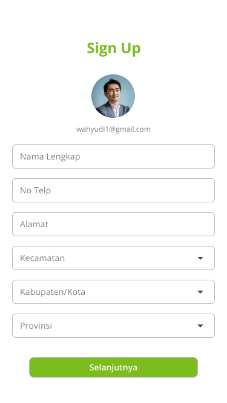
Gambar 3.24 Sequence Diagram Verifikasi Produk

Admin request verifikasi produk dengan mengirimkan token jwt dan id\_product dan backend melakukan validasi token dan id\_product. Selanjutnya backend menyimpan mengubah stataus produk menjadi aktif dan memberikan respon ke admin.

### Perancangan Data

Perancangan data merupakan komponen yang digunakan untuk mendefinisikan struktur data, yang terdiri dari Perancangan Entity Relationship Diagram (ERD) dan juga struktur file pendukung. Adapun dari setiap perancangan data yang dilakukan pada setiap tabel, mempunyai primary key dan beberapa relasi ke tabel lain atau yang disebut foreign key serta memiliki field dengan type data tertentu sesuai kebutuhan. Primary key bersifat unik dan setiap tabel mempunyai masing-masing satu. Sedangkan foreign key merupakan key unik yang nantinya akan menghubungkan dua tabel atau lebih. Selain primary key, sebuah field bisa diberi key yang unik yang sifatnya mirip seperti primary key yang disebut dengan index type unique. Biasanya digunakan digunakan untuk mencegah data yang sama kepada field yang ditentukan. Contohnya seperti data user yang mempunya email dan setiap user memiliki email yang unik masing-masing. Tabel dalam database tandur ini berdasarkan dari referensi dari mockup aplikasi yang dibuat oleh (Hamzah, 2020) yang perancangannya bisa dilihat sebagai berikut:

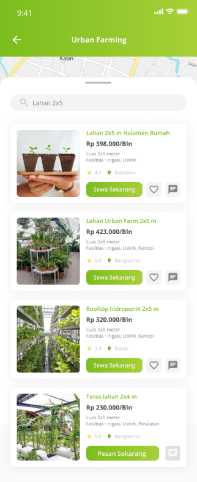
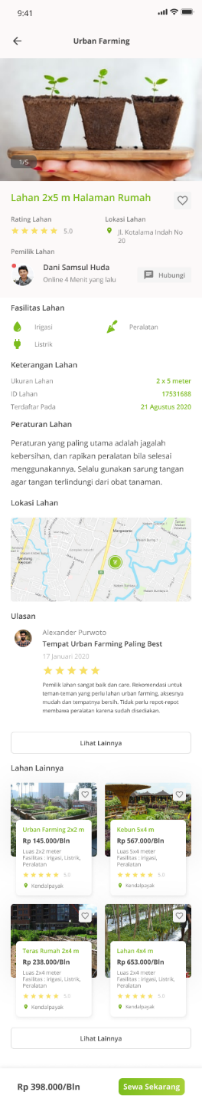
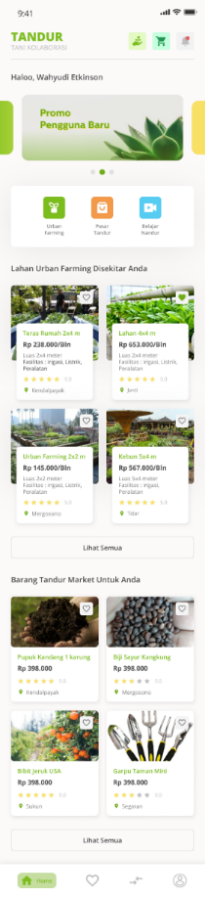
1. Login dan Register



Gambar 3.25 Mockup Login dan Register

Dalam perancangannya data, penulis mengambil beberapa data yang akan dijadikan tabel yaitu form login (email dan password) lalu form register yang berisi foto profil, nama lengkap, telpon, alamat, beserta foto ktp dan ktp selfi.

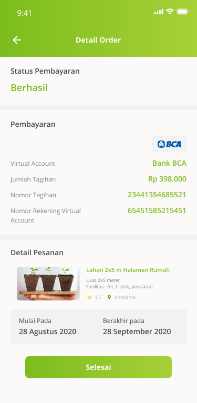
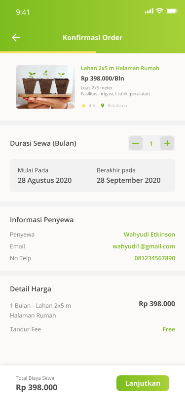
1. Urban Farming, Detail Lahan, dan Pencarian Lahan



Gambar 3.26 Mockup Urban Farming, Detail Lahan, dan Pencarian Lahan

Dari mockup tersebut bisa diambil jika akan data lahan yang tersimpan pada Tandur. Bisa dilihat pada detail lahan yang menyimpan beberapa informasi yatu nama lahan, gambar lahan, pemilik lahan, rating, alamat, fasilitas, deskripsi lahan, harga, dan pencarian lahan.

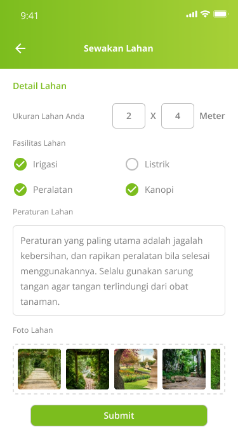
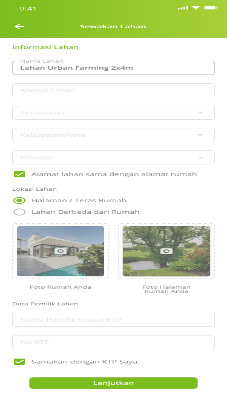
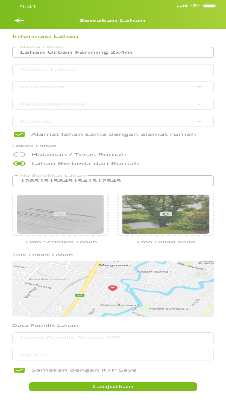
1. Sewa Lahan

*­­*

Gambar 3.27 Mockup Sewa Lahan

Dari sewa lahan bisa dilihat penulis bisa merancang untuk penyewaan lahan dari lahan mana yang akan disewa, harga, durasi sewa, tanggal sewa-akhir sewa, informasi user yang menyewa, total harga, dan detail sewa saat pembayaran telah selesai.

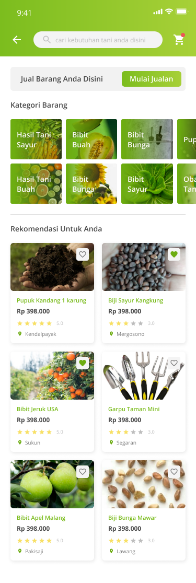
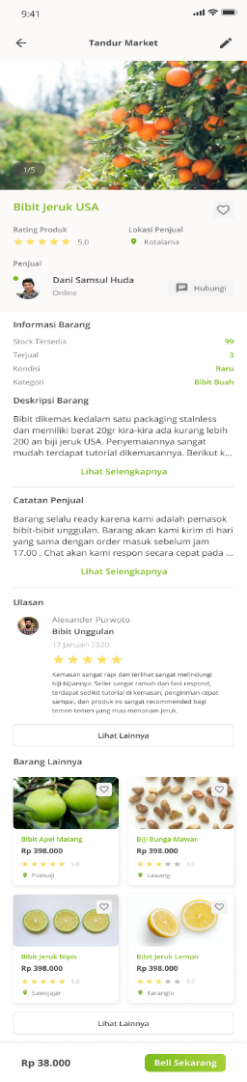
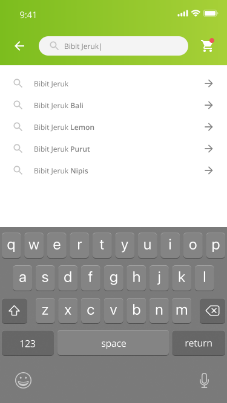
1. Sewakan Lahan

**

Gambar 3.28 Mockup Sewakan Lahan

Penulis akan merancang tabel dari form yang akan diinputkan user yaitu nama lahan, alamat, lokasi lahan, foto lahan, nama pemilik lahan, fasilitas, harga, peraturan lahan. Lalu akan ada status lahan dan menunggu verifikasi.

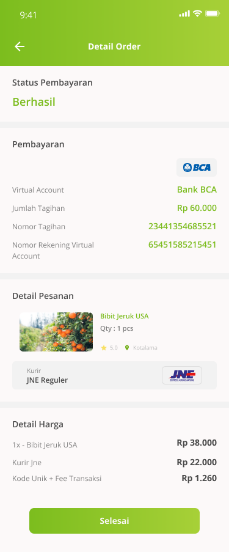
1. Tandur Market, Detail Produk, dan Pencarian Produk

Gambar 3.29 Mockup Tandur Market, Detail Produk, dan Pencarian Produk

Dari mockup penulis mengacu pada data produk Tandur. Bisa dilihat pada detail produk yang menyimpan beberapa informasi yatu nama produk, gambar lahan, pemilik produk, rating, alamat, deskripsi produk, harga, dan pencarian produk.

1. Beli Produk

**

Gambar 3.30 Mockup Beli Produk

Dari mockup beli produk bisa dilihat penulis bisa mengacu pada halaman pembelian produk dari informasi produk, harga, jumlah beli, deskripsi, penjual, total harga, dan detail pembelian serta saat pembayaran telah selesai.

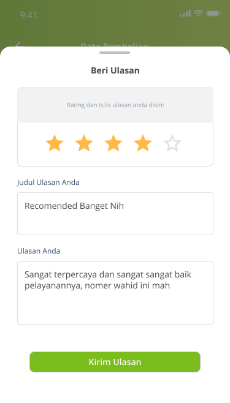
1. Jual Produk

**

Gambar 3.31 Mockup Jual Produk

Mockup tersebut menunjukkan form untuk jual produk yaitu nama produk, kategori produk, harga, stok, kondisi, deskripsi, foto, dan menunggu verifikasi.

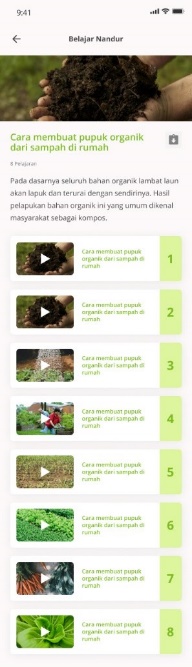
1. Beri Ulasan



Gambar 3.32 Mockup Beri Ulasan

Menunjukkan form untuk memberi ulasan dari rating, judul ulasam dan isi ulasan.

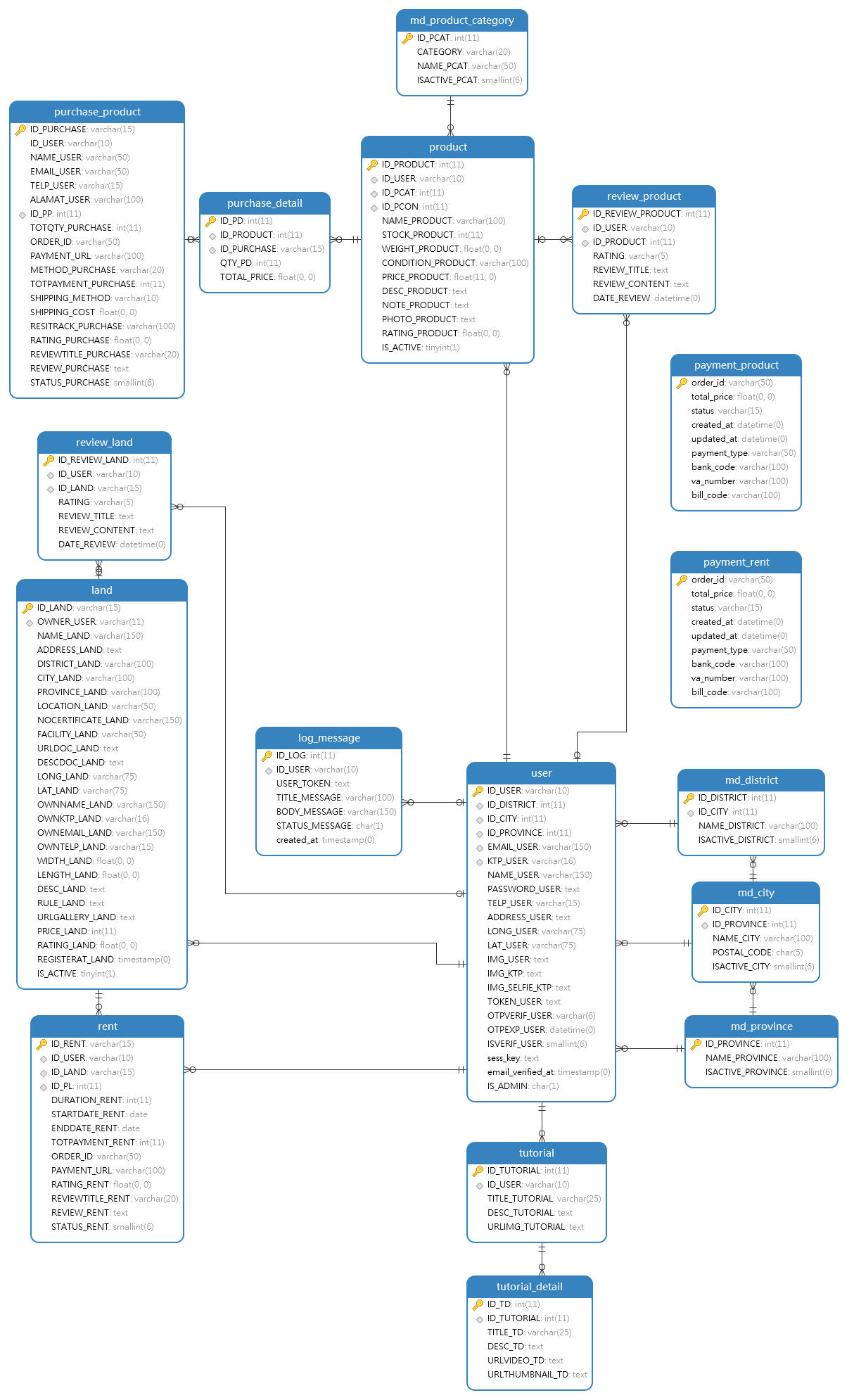
1. Belajar Nandur, Detail, dan Pencarian



Gambar 3.33 Mockup Belajar Nandur, Detail, dan Pencarian

Tutorial yang berisi thumbnail, judul topik lalu ada detail yang berisi judul dan video dari belajar nandur.

Berdasarkan mockup tandur yang disediakan, penulis dapat membuat tabel yang akan menjadi tempat penyimpanan data. Mockup tersebut memberikan panduan tentang informasi dan fitur yang perlu diwakili dalam tabel, sehingga penulis dapat mengidentifikasi atribut-atribut yang perlu dimasukkan ke dalam tabel tersebut. Berikut adalah perancangan yang dibuat:



Gambar 3.34 Design tabel Database bertandur

Dari gambar tersebut menunjukkan tabel-tabel dengan relasi atau kardinalitas untuk memodelkan atau menggambarkan hubungan antara tabel dalam database. Berikut adalah penjelasan dari gambar tersebut:

1. Tabel (Entitas): Setiap kotak mewakili sebuah tabel dalam database. Setiap tabel mewakili jenis entitas yang berbeda, misalnya, user, land, atau product.
2. Relasi: Garis yang menghubungkan dua tabel dalam menggambarkan relasi antara entitas tersebut. Relasi ini menunjukkan bagaimana tabel-tabel terkait satu sama lain dan bagaimana mereka berhubungan dalam basis data.
3. Kardinalitas: Simbol yang terletak di sepanjang garis relasi menunjukkan kardinalitas atau hubungan antara entitas. Kardinalitas menunjukkan berapa banyak entitas yang dapat terkait dengan entitas lain dalam relasi.
4. Key: Simbol yang terletak pada tabel menunjukkan key dalam setiap tabel. Simbol tersebut ada dua yaitu primary key dengan simbol kunci berwarna kuning dan foreign key dengan symbol belah ketupat

Dengan menggunakannya, Anda dapat dengan jelas melihat entitas apa yang ada dalam basis data, atribut apa yang dimiliki setiap entitas, serta bagaimana entitas tersebut saling terhubung dan berinteraksi dalam sistem. Diagram ER juga membantu dalam perancangan basis data dan mengidentifikasi kardinalitas yang sesuai untuk relasi antar entitas.

Berikut perancangan tabel untuk tandur:

1. Land

Tabel 3.1 Tabel land

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_LAND | varchar | 15 | TRUE | Primary |
| OWNER\_USER | varchar | 11 | TRUE | Foreign |
| NAME\_LAND | varchar | 150 | TRUE |  |
| ADDRESS\_LAND | text | 0 | TRUE |  |
| DISTRICT\_LAND | varchar | 100 | TRUE |  |
| CITY\_LAND | varchar | 100 | TRUE |  |
| PROVINCE\_LAND | varchar | 100 | TRUE |  |
| LOCATION\_LAND | varchar | 50 | TRUE |  |
| NOCERTIFICATE\_LAND | varchar | 150 | FALSE |  |
| FACILITY\_LAND | varchar | 50 | TRUE |  |
| URLDOC\_LAND | text | 0 | TRUE |  |
| DESCDOC\_LAND | text | 0 | TRUE |  |
| LONG\_LAND | varchar | 75 | FALSE |  |
| LAT\_LAND | varchar | 75 | FALSE |  |
| OWNNAME\_LAND | varchar | 150 | TRUE |  |
| OWNKTP\_LAND | varchar | 16 | TRUE |  |
| OWNEMAIL\_LAND | varchar | 150 | TRUE |  |
| OWNTELP\_LAND | varchar | 15 | TRUE |  |
| WIDTH\_LAND | float | 0 | TRUE |  |
| LENGTH\_LAND | float | 0 | TRUE |  |
| DESC\_LAND | text | 0 | FALSE |  |
| RULE\_LAND | text | 0 | TRUE |  |
| URLGALLERY\_LAND | text | 0 | TRUE |  |
| PRICE\_LAND | int | 11 | TRUE |  |
| RATING\_LAND | float | 0 | TRUE |  |
| REGISTERAT\_LAND | timestamp | 0 | TRUE |  |
| IS\_ACTIVE | tinyint | 1 | TRUE |  |

1. log\_message

Tabel 3.2 Tabel log\_message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_LOG | int | 11 | TRUE | Primary |
| ID\_USER | varchar | 10 | FALSE | Foreign |
| USER\_TOKEN | text | 0 | TRUE |  |
| TITLE\_MESSAGE | varchar | 100 | TRUE |  |
| BODY\_MESSAGE | varchar | 150 | TRUE |  |
| STATUS\_MESSAGE | char | 1 | FALSE |  |
| created\_at | timestamp | 0 | FALSE |  |

1. md\_city

Tabel 3.3 Tabel md\_city

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_CITY | int | 11 | TRUE | Primary |
| ID\_PROVINCE | int | 11 | TRUE | Foreign |
| NAME\_CITY | varchar | 100 | TRUE |  |
| POSTAL\_CODE | char | 5 | TRUE |  |
| ISACTIVE\_CITY | smallint | 6 | TRUE |  |

1. md\_district

Tabel 3.4 Tabel md\_district

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_DISTRICT | int | 11 | TRUE | Primary |
| ID\_CITY | int | 11 | TRUE | Foreign |
| NAME\_DISTRICT | varchar | 100 | TRUE |  |
| ISACTIVE\_DISTRICT | smallint | 6 | TRUE |  |

1. md\_product\_category

Tabel 3.5 Tabel md\_product\_category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_PCAT | int | 11 | TRUE | Primary |
| CATEGORY | varchar | 20 | TRUE |  |
| NAME\_PCAT | varchar | 50 | TRUE |  |
| ISACTIVE\_PCAT | smallint | 6 | TRUE |  |

1. md\_province

Tabel 3.6 Tabel md\_province

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_PROVINCE | int | 11 | TRUE | Primary |
| NAME\_PROVINCE | varchar | 100 | TRUE |  |
| ISACTIVE\_PROVINCE | smallint | 6 | TRUE |  |

1. payment\_product

Tabel 3.7 Tabel payment product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| order\_id | varchar | 50 | TRUE | Primary |
| total\_price | float | 0 | TRUE |  |
| status | varchar | 15 | FALSE |  |
| created\_at | datetime | 0 | FALSE |  |
| updated\_at | datetime | 0 | FALSE |  |
| payment\_type | varchar | 50 | FALSE |  |
| bank\_code | varchar | 100 | FALSE |  |
| va\_number | varchar | 100 | FALSE |  |
| bill\_code | varchar | 100 | FALSE |  |

1. payment\_rent

Tabel 3.8 Tabel payment\_rent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| order\_id | varchar | 50 | TRUE | Primary |
| total\_price | float | 0 | TRUE |  |
| status | varchar | 15 | FALSE |  |
| created\_at | datetime | 0 | FALSE |  |
| updated\_at | datetime | 0 | FALSE |  |
| payment\_type | varchar | 50 | FALSE |  |
| bank\_code | varchar | 100 | FALSE |  |
| va\_number | varchar | 100 | FALSE |  |
| bill\_code | varchar | 100 | FALSE |  |

1. product

Tabel 3.9 Tabel product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_PRODUCT | int | 11 | TRUE | Primary |
| ID\_USER | varchar | 10 | TRUE | Foreign |
| ID\_PCAT | int | 11 | TRUE | Foreign |
| ID\_PCON | int | 11 | TRUE | Foreign |
| NAME\_PRODUCT | varchar | 100 | TRUE |  |
| STOCK\_PRODUCT | int | 11 | TRUE |  |
| WEIGHT\_PRODUCT | float | 0 | FALSE |  |
| CONDITION\_PRODUCT | varchar | 100 | TRUE |  |
| PRICE\_PRODUCT | float | 11 | TRUE |  |
| DESC\_PRODUCT | text | 0 | TRUE |  |
| NOTE\_PRODUCT | text | 0 | TRUE |  |
| PHOTO\_PRODUCT | text | 0 | TRUE |  |
| RATING\_PRODUCT | float | 0 | TRUE |  |
| IS\_ACTIVE | tinyint | 1 | TRUE |  |

1. purchase\_detail

Tabel 3.10 Tabel purchase\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_PD | int | 11 | TRUE | Primary |
| ID\_PRODUCT | int | 11 | TRUE | Foreign |
| ID\_PURCHASE | varchar | 15 | TRUE | Foreign |
| QTY\_PD | int | 11 | TRUE |  |
| TOTAL\_PRICE | float | 0 | TRUE |  |

1. purchase\_product

Tabel 3.11 Tabel purchase\_product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_PURCHASE | varchar | 15 | TRUE | Primary |
| ID\_USER | varchar | 10 | TRUE | Foreign |
| NAME\_USER | varchar | 50 | FALSE |  |
| EMAIL\_USER | varchar | 50 | FALSE |  |
| TELP\_USER | varchar | 15 | FALSE |  |
| ALAMAT\_USER | varchar | 100 | FALSE |  |
| TOTQTY\_PURCHASE | int | 11 | TRUE |  |
| ORDER\_ID | varchar | 50 | FALSE |  |
| PAYMENT\_URL | varchar | 100 | FALSE |  |
| TOTPAYMENT\_PURCHASE | int | 11 | TRUE |  |
| SHIPPING\_METHOD | varchar | 10 | FALSE |  |
| SHIPPING\_COST | float | 0 | FALSE |  |
| STATUS\_REVIEW\_PRODUCT | char | 1 | FALSE |  |

1. rent

Tabel 3.12 Tabel rent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_RENT | varchar | 15 | TRUE | Primary |
| ID\_USER | varchar | 10 | TRUE | Foreign |
| ID\_LAND | varchar | 15 | TRUE | Foreign |
| DURATION\_RENT | int | 11 | TRUE |  |
| STARTDATE\_RENT | date | 0 | TRUE |  |
| ENDDATE\_RENT | date | 0 | TRUE |  |
| TOTPAYMENT\_RENT | int | 11 | TRUE |  |
| ORDER\_ID | varchar | 50 | FALSE |  |
| PAYMENT\_URL | varchar | 100 | FALSE |  |
| STATUS\_REVIEW\_LAND | char | 1 | FALSE |  |

1. review\_land

Tabel 3.13 Tabel review\_land

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_REVIEW\_LAND | int | 11 | TRUE | Primary |
| ID\_USER | varchar | 10 | FALSE | Foreign |
| ID\_LAND | varchar | 15 | FALSE | Foreign |
| RATING | varchar | 5 | FALSE |  |
| REVIEW\_TITLE | text | 0 | FALSE |  |
| REVIEW\_CONTENT | text | 0 | FALSE |  |
| DATE\_REVIEW | datetime | 0 | FALSE |  |

1. review\_product

Tabel 3.14 Tabel review\_product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_REVIEW\_PRODUCT | int | 11 | TRUE | Primary |
| ID\_USER | varchar | 10 | FALSE | Foreign |
| ID\_PRODUCT | int | 11 | FALSE | Foreign |
| RATING | varchar | 5 | FALSE |  |
| REVIEW\_TITLE | text | 0 | FALSE |  |
| REVIEW\_CONTENT | text | 0 | FALSE |  |
| DATE\_REVIEW | datetime | 0 | FALSE |  |

1. tutorial

Tabel 3.15 Tabel tutorial

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_TUTORIAL | int | 11 | TRUE | Primary |
| ID\_USER | varchar | 10 | TRUE | Foreign |
| TITLE\_TUTORIAL | varchar | 25 | TRUE |  |
| DESC\_TUTORIAL | text | 0 | TRUE |  |
| URLIMG\_TUTORIAL | text | 0 | TRUE |  |

1. tutorial\_detail

Tabel 3.16 Tabel tutorial\_detail

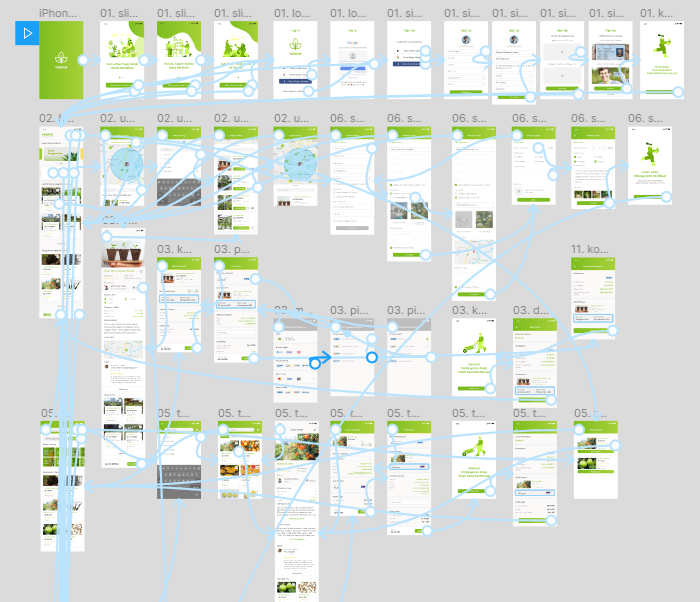
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_TD | int | 11 | TRUE | Primary |
| ID\_TUTORIAL | int | 11 | TRUE | Foreign |
| TITLE\_TD | varchar | 25 | TRUE |  |
| DESC\_TD | text | 0 | TRUE |  |
| URLVIDEO\_TD | text | 0 | TRUE |  |

1. user

Tabel 3.17 Tabel user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Not Null** | **Key** |
| ID\_USER | varchar | 10 | TRUE | Primary |
| ID\_DISTRICT | int | 11 | TRUE | Foreign |
| ID\_CITY | int | 11 | TRUE | Foreign |
| ID\_PROVINCE | int | 11 | TRUE | Foreign |
| EMAIL\_USER | varchar | 150 | TRUE | Unique |
| KTP\_USER | varchar | 16 | TRUE |  |
| NAME\_USER | varchar | 150 | TRUE |  |
| PASSWORD\_USER | text | 0 | TRUE |  |
| TELP\_USER | varchar | 15 | TRUE |  |
| ADDRESS\_USER | text | 0 | TRUE |  |
| LONG\_USER | varchar | 75 | FALSE |  |
| LAT\_USER | varchar | 75 | FALSE |  |
| IMG\_USER | text | 0 | TRUE |  |
| IMG\_KTP | text | 0 | TRUE |  |
| IMG\_SELFIE\_KTP | text | 0 | TRUE |  |
| TOKEN\_USER | text | 0 | FALSE |  |
| OTPVERIF\_USER | varchar | 6 | TRUE |  |
| OTPEXP\_USER | datetime | 0 | TRUE |  |
| ISVERIF\_USER | smallint | 6 | TRUE |  |
| sess\_key | text | 0 | FALSE |  |
| email\_verified\_at | timestamp | 0 | FALSE |  |
| IS\_ADMIN | char | 1 | FALSE |  |

### Perancangan User Interface / *Mock-up* aplikasi



Gambar 3.35Design Interface Tandur

## Rancangan Pengujian

Pada Rancangan pengujian, peneliti menggunakan Blackbox Testing. Black box testing yang hanya melihat hasil input dan output dari perangkat lunak, metode ini merupakan pengujian terhadap fungsionalitas atau kegunaan sebuah aplikasi. Black box testing cukup meninjau input dan output sistem software tersebut tanpa pengetahuan tentang internal programnya. Pada pengujian black box dilakukan hanya mengambil hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Dalam rancangan pengujian tersebut ditentukan langkah-langkah yang dapat diikuti dalam rancangan pengujian menggunakan metode Blackbox Testing:

1. Skenario

Deskripsi singkat dari situasi atau aksi yang akan diuji.

1. Kasus:

Tindakan atau langkah yang dilakukan dalam skenario pengujian.

1. Hasil yang diharapkan

Merupakan hasil yang diinginkan atau diharapkan dari skenario pengujian tersebut.

1. Hasil yang sebenarnya

Merupakan hasil yang terjadi secara aktual saat skenario pengujian dijalankan.