

EKSPANSI APLIKASI AQUA BREEDING DENGAN PENAMBAHAN FITUR INVENTARISASI UNTUK PENENTUAN HARGA DASAR PRODUK PERIKANAN BERBASIS ANDROID

Akbar Maulana Alfatih¹, Muhammad Eka Suryana², Med Irzal³

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Jakarta, Jakarta Timur, Indonesia

akbarma2002@gmail.com¹, eka-suryana@unj.ac.id², medirzal@unj.ac.id³

Abstrak— Budidaya perikanan air tawar merupakan salah satu sumber perikanan yang ada di Indonesia. Dalam berbudidaya, tentunya penting untuk mencatat indikator-indikator inventaris budidaya ikan seperti pakan ikan, suplemen ikan, aset kolam, listrik pada kolam, serta benih ikan yang berguna untuk menentukan harga jual ikan. Penelitian ini bertujuan untuk memperluas aplikasi Aqua Breeding dengan menambahkan fitur inventarisasi yang dapat digunakan untuk mencatat penggunaan inventaris serta menentukan harga jual minimum ikan yang jujur. Data pada penelitian ini diambil dari hasil diskusi bersama pembudidaya ikan air tawar JFT (J Farm Technology) dan studi literatur dengan membaca jurnal-jurnal yang terkait dengan topik penelitian. Diskusi tersebut menghasilkan suatu user requirement yang menjadi pedoman dalam membuat web service pada backend serta penerapannya pada frontend mobile. Metode pengembangan sistem ini menggunakan metode Scrum dengan jumlah Sprint sebanyak lima Sprint serta teknologi yang digunakan adalah Flask dengan bahasa Python pada backend dan Flutter dengan bahasa Dart pada frontend. Hasil akhir dari penelitian ini adalah web service berupa REST API berserta dokumentasinya dan juga penerapannya pada aplikasi berbasis Android yang telah diuji dengan metode pengujian unit testing dan User Acceptance Test (UAT).

Kata Kunci— *sistem inventarisasi, aplikasi mobile, transaksi ikan, budidaya perikanan modern, scrum.*

I. PENDAHULUAN

Perikanan adalah sumber utama pendapatan di Indonesia karena Indonesia adalah negara yang terdiri dari banyak pulau dan dikelilingi oleh laut. Banyak orang di Indonesia hidup dari usaha budidaya ikan. Namun, menangkap ikan terlalu banyak bisa menyebabkan overfishing, di mana ikan sulit berkembang biak karena terlalu banyak yang ditangkap. Akibatnya, populasi ikan menurun, dan produksi ikan berkurang. Untuk mengatasi masalah ini, diperlukan cara baru yaitu budidaya perikanan modern. Saat ini, banyak pembudidaya ikan di Indonesia masih menggunakan cara manual dalam mengelola usaha budidayanya. Tapi cara ini kurang efektif dan sulit untuk jangka panjang.

Dari permasalahan di atas, terdapat beberapa solusi seperti pada penelitian Yi-Bing Lin [1] dan tim yang membuat smart

aquarium yang bertujuan untuk meningkatkan kualitas akuarium yang bernama FishTalk. FishTalk memungkinkan sebuah sensor pada akuarium untuk menggerakkan aktuator secara real time. Kegunaan dari smart aquarium ini seperti sistem pemberian pakan otomatis dan pengendalian air dalam kolam secara otomatis.

Bing Ouyang [2] dan timnya juga membuat sebuah sistem yang dibentuk dan digunakan untuk monitoring serta decision making pada tambak perikanan, sistem ini dinamakan HAUCS (Hybrid Aerial Underwater Robotic System). Pemantauan ini dilakukan dengan memanfaatkan sistem robotik, mesin, dan operator manusia. Tujuan dibentuknya HAUCS ini adalah untuk meringankan pekerjaan manusia dari tugas yang berat, terlalu banyak biaya, dan memakan waktu dalam operasi pelaksanaan budidaya aquaculture melalui platform pemanfaatan sistem robotik.

Beberapa masalah yang dialami oleh pembudidaya ikan antara lain juga seperti mayoritas distributor memainkan timbangan, ketimpangan harga penawaran oleh distributor pada petani produsen, dan tidak ada data jelas terkait jumlah demand dan besarnya supply. Untuk mengatasi permasalahan tersebut, penelitian Andri [3] dan Gian [4] membuat sistem yang dapat melakukan tracking aktifitas budidaya untuk mengetahui data supply di tingkat pembudidaya. Selanjutnya, diperlukan sistem untuk melacak pengeluaran pembudidaya agar dapat menentukan harga jual ikan yang adil. Sistem inventaris ini akan mencatat penggunaan pasokan selama budidaya sehingga pembudidaya dapat melihat data pasokan dengan jelas. Dari sini, pembudidaya juga dapat menetapkan harga jual berdasarkan pengeluaran selama budidaya, menghindari perbedaan harga yang tidak adil saat transaksi.

Oleh karena itu, penelitian ini bertujuan untuk melengkapi sistem untuk memecahkan masalah-masalah tersebut dengan menambahkan sistem inventarisasi pada aplikasi *Aqua Breeding* ini untuk menentukan harga dasar pada produk perikanan. Dengan demikian, harga dasar tersebut dapat digunakan oleh pembudidaya ikan untuk penjualan hasil panen mereka. Selain itu, fitur inventarisasi ini juga dapat membantu para pembudidaya ikan dalam mengolah dan mengontrol kebutuhan serta pengeluaran dalam setiap musim budidaya. Berdasarkan fitur baru yang sudah dijelaskan sebelumnya, aplikasi ini

diharapkan dapat membantu para pembudidaya ikan untuk berbudidaya dalam hal penentuan harga dasar dan pengendalian kebutuhan saat budidaya berlangsung.

II. KAJIAN PUSTAKA

A. Persediaan dan Manajemen Persediaan

Pada buku Dasar-Dasar Manajemen [5], dijelaskan bahwa persediaan adalah sebuah stok aset yang dimiliki oleh perusahaan. Aset ini dapat berupa bahan mentah, bahan baku, barang jadi, barang dalam proses, hingga bahan pembantu. Persediaan merupakan aset yang berharga, karena berkaitan dengan proses produksi. Persediaan yang tidak teratur dapat menyebabkan kerugian, sehingga menerapkan manajemen persediaan dalam suatu bisnis merupakan hal yang penting.

Manajemen persediaan merupakan suatu cara untuk melakukan pengawasan, kontrol, pengelolaan terhadap persediaan yang dimiliki oleh perusahaan. Berbagai macam kegiatan yang berkaitan dengan memperoleh, menyimpan, hingga menggunakan persediaan merupakan bagian dari manajemen persediaan.

Manajemen persediaan memiliki beberapa fungsi sebagai berikut.

1. Mencegah terjadinya kekurangan persediaan.
2. Mencegah barang dari supplier tidak sesuai kebutuhan.
3. Memastikan proses produksi berjalan dengan lancar.
4. Mengantisipasi permintaan yang mendadak.
5. Menyesuaikan pembelian dengan jadwal produksi.

B. Biaya Persediaan

Penetapan biaya persediaan atau evaluasi persediaan memungkinkan perusahaan untuk memberikan nilai moneter untuk barang-barang dalam persediaan mereka. Inventaris perusahaan seringkali merupakan aset terbesarnya dan pengukuran yang tepat untuk memastikan keakuratan laporan keuangan.

Untuk menentukan biaya persediaan, diperlukan lima langkah-langkah sebagai berikut.

1. Menentukan periode waktu tertentu yang dimana perlu menemukan nilai inventaris.
2. Memastikan persediaan selalu tersedia.
3. Mengurangi resiko bahan baku yang datang terlambat.
4. Menjaga jumlah persediaan tetap stabil.
5. Mengantisipasi kemungkinan adanya perubahan dari segi penawaran ataupun permintaan.

C. Penentuan Harga Transfer

Pada buku Management Control [6], dijelaskan bahwa penentuan harga transfer atau *transfer pricing* merupakan proses harga penentuan harga yang ditetapkan dalam transaksi

penjualan dan pembelian diantara berbagai unit organisasi pada kelompok perusahaan atau instansi yang sama.

Tujuan dari penentuan harga transfer ini adalah sebagai berikut.

1. Memberikan informasi yang relevan kepada masing-masing unit usaha untuk menentukan imbal balik yang optimum antara biaya dan pendapatan perusahaan.
2. Menghasilkan keputusan yang selaras dengan cita-cita, maksudnya sistem harus dirancang sedemikian rupa sehingga keputusan yang meningkatkan laba unit usaha juga akan meningkatkan laba perusahaan.
3. Membantu pengukuran kinerja ekonomi dari unit usaha individual.
4. Sistem tersebut harus mudah dimengerti dan dikelola.

D. *Frontend* dan *Backend*

Dalam pengembangan aplikasi, terdapat dua sisi pengembangan yaitu *frontend* dan *backend*. *Frontend* merupakan bagian yang ditampilkan kepada *user* seperti contoh halaman dashboard, menu aplikasi, dan sebagainya. Oleh sebab itu, sisi *frontend* ini juga bisa dibilang sebagai *user-side* atau *client-side*.

Sementara itu, *backend* merupakan bagian yang mengurus koneksi antara server dengan *database* aplikasi. *Backend* juga bertugas untuk membuat penghubung antara aplikasi dengan server dalam bentuk *endpoint* yang nantinya akan digunakan pada sisi *frontend* untuk ditampilkan pada aplikasi. Oleh karena itu, sisi *backend* bisa dibilang sebagai *server-side*.

E. Flutter

Flutter merupakan salah satu *framework* yang digunakan dalam pengembangan aplikasi *mobile* [7]. Flutter dikembangkan dengan bahasa pemrograman Dart yang dirancang oleh Lars Bak dan Kasper Lund dan memiliki struktur berbasis *class* dan berorientasi terhadap objek. Framework ini diresmikan pada tahun 2015 oleh perkumpulan Dart *developer* summit dan merilis versi stabilnya yaitu Flutter 1.0 pada tahun 2018 di acara Flutter Live. Pengembangan *framework* ini tergolong cukup besar karena bersifat *open source* sehingga banyak komunitas dan orang-orang yang ikut mengembangkan *framework* ini.

Beberapa keunggulan dari penggunaan Flutter antara lain sebagai berikut.

1. *Multiplatform*.
2. *Open Source*.
3. Dokumentasi lengkap.

F. *Flask*

Flask merupakan *microframework* yang digunakan pada sisi backend dengan basis bahasa pemrogramannya yaitu Python [8]. Flask dirilis pada tahun 2010 dan dikembangkan oleh

Armin Ronacher, seorang python *entusiast*. Flask disebut *microframework* karena Flask tidak memerlukan alat bantu lain atau library dalam penggunaannya.

G. MongoDB

MongoDB merupakan *database* NoSQL yang dikembangkan oleh MongoDB Inc yang rilis pada tahun 2009 [9]. Database ini disebut NoSQL karena berbasis objek atau bisa disebut JSON (*JavaScript Object Notation*), berbeda dengan MySQL yang berbasis tabel dalam penggunaannya.

H. REST API

Dalam pengembangan aplikasi, untuk menghubungkan antara frontend dengan backend dapat digunakan API sebagai perantaranya. API merupakan singkatan dari Application Programming Interface yang berfungsi menerima request dan response dari sisi frontend dan backend.

REST merupakan singkatan dari Representational State Transfer. API dapat disebut sebagai REST API jika memiliki standar kriteria dari REST. Kriteria tersebut bertujuan untuk menjadi standar dalam proses komunikasi antar aplikasi dan pengguna sehingga menjadi lebih fleksibel.



GAMBAR 1. SKEMA REST API

Request client berupa CRUD (Create, Read, Update, Delete) pada HTTP berupa GET, POST, PUT, DELETE yang digunakan untuk berkomunikasi terhadap server serta response untuk client yang diterima berbentuk JSON. Metode GET berfungsi untuk mengambil data, POST dan PUT berfungsi untuk membuat dan memperbarui data, serta DELETE berfungsi untuk menghapus data pada database.

I. Scrum

Scrum merupakan metode pengembangan aplikasi yang digunakan untuk kolaborasi antar tim pengembangan [9]. Pada jurnal yang berjudul The Scrum Guide [10], dijelaskan bahwa Scrum merupakan kerangka kerja yang digunakan untuk membantu orang, tim, serta organisasi dalam menyelesaikan suatu masalah. Scrum menerapkan prinsip Agile yang dimana berfokus pada kepuasan konsumen dalam masa pengembangan.

Untuk menjalankan metode Scrum, diperlukan Scrum Master, Product Owner, serta Developer. Scrum Master bertugas sebagai pemimpin serta bertanggung jawab dalam menjalankan prinsip Scrum pada tim, Product Owner bertanggung jawab dalam membuat dan mengontrol pekerjaan tim agar sesuai dengan kebutuhan, dan Developer yang bertugas untuk menjalankan list tugas yang sudah dibuat dan disepakati.

J. Unit Testing

Unit testing merupakan suatu aktivitas dalam pengembangan aplikasi yang bertujuan untuk menguji fungsionalitas serta komponen dari aplikasi yang dikembangkan. Unit testing bertujuan untuk memastikan aplikasi dalam kondisi yang sudah sesuai dengan kebutuhan yang sudah disepakati sebelumnya. Proses dalam unit testing ini meliputi pengecekan output fungsi-fungsi yang ada pada aplikasi berdasarkan input dari tester.

K. User Acceptance Test (UAT)

Pada jurnal yang diterbitkan oleh Hareton K.N. Leung [11] tentang UAT, User Acceptance Test (UAT) adalah tahap akhir pengujian dalam pengembangan perangkat lunak. Ketika hasil pengujian memenuhi kriteria, sistem perangkat lunak dapat dirilis untuk penggunaan operasional.

III. METODE PENELITIAN

Dalam pengembangan aplikasi ini, untuk fitur penentuan harga digunakan formula sebagai berikut.

$$T = \frac{C_p + C_q + C_r + \frac{C_l}{nA} + \frac{C_a}{60 \cdot nB}}{N} \quad (1)$$

Dimana komponen dari formula tersebut dapat dijelaskan sebagai berikut.

1. T = Harga jual minimum ikan.
2. C_p = Harga total pemakaian benih selama musim berjalan.
3. C_q = Harga total pemakaian pakan selama musim berjalan.
4. C_r = Harga total pemakaian suplemen selama musim berjalan.
5. C_l = Harga tagihan listrik per bulannya
6. C_a = Harga total penggunaan aset
7. nA = Jumlah kolam aktif
8. nB = Jumlah semua kolam
9. N = Jumlah ikan yang hidup pada kolam selama musim berjalan

Formula tersebut masih merupakan hipotesa karena belum dilakukan langkah pengujian kepada pembudidaya ikan namun sudah dapat diimplementasikan pada pengembangan aplikasi.

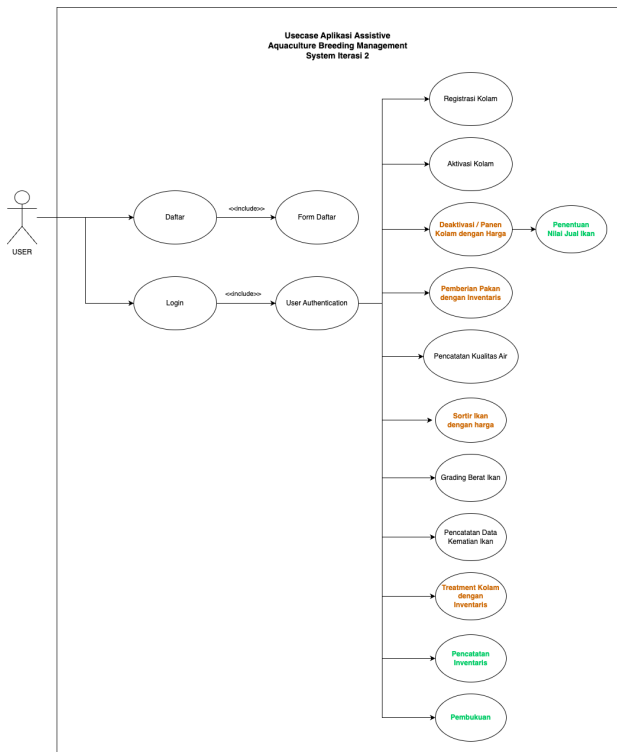
Secara garis besar, metode penelitian yang dilakukan dalam pengembangan aplikasi ini menggunakan metode Scrum yang dimana terdiri dari *Product Backlog*, *Sprint Backlog*, *Sprint*, dan *Sprint Review*.

Berikut merupakan tahapan dari penelitian ada pada ilustrasi berikut.



GAMBAR 2. TAHAPAN PENELITIAN

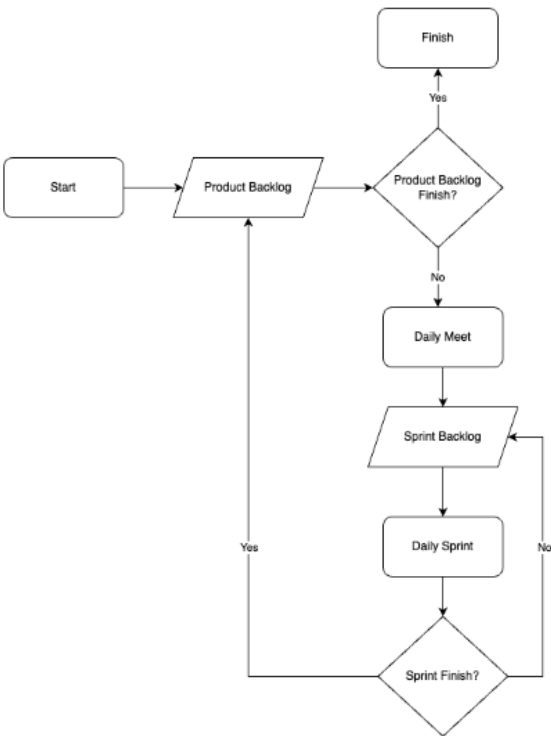
Dalam proses pengembangan aplikasi, terdapat analisa kebutuhan yang digunakan sebagai dasar pada pengembangan. Analisa kebutuhan ini dibuat berupa *use case* yang dapat dilihat pada gambar berikut.



GAMBAR 3. USE CASE APLIKASI

Dijelaskan pada use case tersebut bahwa penambahan fitur pada aplikasi ini terdapat pada fitur pencatatan inventaris, pembukuan, serta penentuan harga jual ikan.

Seperti yang dijelaskan sebelumnya, metode pengembangan disini menggunakan metode Scrum yang memiliki ilustrasi sebagai berikut.



GAMBAR 4. TAHAPAN SCRUM

Dari ilustrasi diatas, masing-masing komponen memiliki penjelasan tersendiri yang disebutkan pada poin-poin dibawah ini.

1. Product Backlog

Product backlog adalah tugas-tugas yang akan dijalankan pada penelitian dan hal yang pertama kali dilakukan sebelum memulai riset. Daftar tugas yang ada pada *product backlog* ini akan dipindahkan pada *sprint backlog* tergantung pada skala prioritas dari task itu sendiri. Berikut adalah tabel dari *product backlog* yang sudah berjalan.

TABEL 1. PRODUCT BACKLOG

No	Stories	Sprint	Status
1	Fitur Pencatatan inventaris	1, 2, 3, 4, 5	Selesai
2	Fitur Aktivasi kolam dengan inventaris	3, 4, 5	Selesai
3	Fitur Pemberian pakan yang terkoneksi dengan inventaris	4	Selesai
4	Fitur Treatment kolam yang terkoneksi dengan inventaris	5	Selesai
5	Fitur Panen termasuk harga nilai jual ikan	5	Selesai
6	Fitur Pembukuan musim budidaya	5	Selesai

2. Sprint Backlog

Sprint backlog adalah daftar tugas yang harus dijalankan selama masa *sprint* berlangsung. Tugas yang ada pada *sprint backlog* bersifat fleksibel seiring dengan berjalannya *sprint*.

3. Sprint

Progres *sprint* dilaksanakan ketika list task pada *sprint backlog* sudah disepakati bersama. Periode pengerjaan *sprint* bervariasi tergantung pada kesulitan task dari *sprint backlog* tersebut.

4. Sprint Review

Setelah *Sprint* berjalan, setiap minggunya diadakan *meet* bersama tim untuk melaksanakan *sprint review* yang bertujuan untuk melaporkan perkembangan *sprint* baik itu proses ataupun hambatan selama pengerjaan *sprint*.

5. Deploy Sistem

Ketika semua task *sprint* yang ada di *sprint backlog* selesai, maka aplikasi akan di *deploy* untuk dijalankan pengujian pada aplikasi. Pengujian aplikasi ini dilakukan dengan menggunakan *unit testing* dan *User Acceptance Test* (UAT).

Di tahap pengujian ini, unit testing dilakukan oleh tim internal developer aplikasi untuk memastikan kepastian fungsi fitur dan cara kerja fitur agar aplikasi sesuai dengan kebutuhan. Sementara UAT dilakukan agar aplikasi dapat sesuai dengan kebutuhan *user*.

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini terdapat 5 *sprint* yang masing-masing Sprintnya memiliki waktu penyelesaian yang berbeda-beda. Sebelum memulai *sprint*, dilakukan perencanaan *sprint backlog* yang diambil dari *product backlog* yang sebelumnya sudah disepakati. Adapun laporan setiap *sprint* dalam proses pengembangan sistem dapat dilihat sebagai berikut.

TABEL 2. SPRINT 1 BACKLOG

No	Stories	Task	Status
1	Fitur pencatatan inventaris	- Membuat skema database dari pencatatan inventaris	Selesai
		- Membuat integrasi skema database dengan skema database sebelumnya	Selesai
		- Membuat mockup dari fitur inventaris	Selesai

Pada tabel tersebut, terdapat list tugas yang ada pada *sprint* 1 yang dilaksanakan pada tanggal 07 Maret 2023 – 29 Maret 2023. *Sprint* tersebut mengerjakan fitur pencatatan inventaris yang sudah ada pada *product backlog*.

TABEL 3. SPRINT 2 BACKLOG

No	Stories	Task	Status
1	Fitur pencatatan inventaris	- Membuat alur UI/UX dari design aplikasi	Selesai
		- Mengupdate skema database pada inventaris	Selesai

Pada tabel tersebut, terdapat list tugas yang ada pada *sprint* 2 yang dilaksanakan pada tanggal 30 Maret 2023 – 15 April 2023. *Sprint* tersebut melanjutkan pengerjaan fitur pencatatan inventaris.

TABEL 4. SPRINT 3 BACKLOG

No	Stories	Task	Status
1	Fitur pencatatan inventaris	- Design route untuk inventaris benih (dalam bentuk RESTful API)	Selesai
		- Membuat halaman serta Integrasi RESTful API benih dengan Flutter	Selesai
2	Fitur Aktivasi kolom dengan inventaris	- Membuat tabel riwayat pemakaian benih	Selesai
		- Design route dan penerapan dengan Flutter untuk riwayat pemakaian benih (dalam bentuk RESTful API)	Selesai

Pada tabel tersebut, terdapat list tugas yang ada pada *sprint* 3 yang dilaksanakan pada tanggal 10 Mei 2023 – 21 Juni 2023. *Sprint* tersebut melanjutkan pengerjaan fitur pencatatan inventaris serta mulai mengerjakan fitur aktivasi kolom dengan inventaris.

TABEL 5. SPRINT 4 BACKLOG

No	Stories	Task	Status
1	Fitur pencatatan inventaris	- Design route dan penerapan pada Flutter untuk inventaris pakan (dalam bentuk RESTful API)	Selesai
		- Design route dan penerapan pada Flutter untuk inventaris suplemen (dalam bentuk RESTful API)	Selesai
		- Design route dan penerapan pada Flutter untuk inventaris listrik (dalam bentuk RESTful API)	Selesai

		- Design route dan penerapan pada Flutter untuk inventaris aset (dalam bentuk RESTful API)	Selesai
		- Design route dan penerapan pada Flutter untuk riwayat pemakaian pakan (dalam bentuk RESTful API)	Selesai
		- Design route dan penerapan pada Flutter untuk riwayat pemakaian suplemen (dalam bentuk RESTful API)	Selesai
2	Fitur Pemberian pakan yang terkoneksi dengan inventaris	- Integrasi data inventaris pakan pada halaman entry pakan	Selesai

Pada tabel tersebut, terdapat list tugas yang ada pada *sprint* 4 yang dilaksanakan pada tanggal 21 Juni 2023 – 12 Juli 2023. *Sprint* tersebut melanjutkan pengerjaan fitur pencatatan inventaris serta mulai mengerjakan fitur pemberian pakan yang terkoneksi dengan inventaris.

TABEL 6. *SPRINT 5 BACKLOG*

No	Stories	Task	Status
1	Fitur pencatatan inventaris	- Design route dan penerapan pada Flutter untuk merk di inventaris pakan (dalam bentuk RESTful API)	Selesai
2	Fitur Treatment kolam yang terkoneksi dengan inventaris	- Integrasi halaman Input Treatment dengan data inventaris suplemen	Selesai
3	Fitur Panen termasuk harga nilai jual ikan	- Integrasi harga jual minimum ikan pada fitur Panen	Selesai
4	Fitur Pembukuan musim budidaya	- Design route dan penerapan Flutter untuk rekapitulasi panen beserta harga jual ikan (dalam bentuk RESTful API)	Selesai

Pada tabel tersebut, terdapat list tugas yang ada pada *sprint* 5 yang dilaksanakan pada tanggal 12 Juli 2023 – 01 Agustus 2023. *Sprint* ini merupakan sprint final dari penelitian pengembangan aplikasi ini. Pada *sprint* ini dimulai dengan melanjutkan pengerjaan fitur pencatatan inventaris yang kemudian dilanjutkan dengan pengerjaan fitur treatment kolam yang terkoneksi dengan inventaris, panen termasuk harga nilai jual ikan, serta pembukuan musim budidaya.

Hasil dari sprint-sprint diatas dapat disimpulkan sebagai berikut.

1. Model *database*

```

1      class SeedInventory(db.Document):
2          id_int = db.SequenceField(required=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          fish_seed_category = db.StringField(required=True)
5          fish_type = db.StringField(required=True)
6          brand_name = db.StringField(required=True)
7          amount = db.IntField(required=True)
8          weight = db.FloatField(default=0)
9          width = db.StringField()
10         price = db.IntField(required=True)
11         total_price = db.IntField(required=True)
12         image = db.StringField(required=True)
13         created_at = db.DateTimeField(default=datetime.datetime.now)
14         updated_at = db.DateTimeField(default=datetime.datetime.now)
15

```

GAMBAR 5. MODEL *DATABASE* INVENTARIS BENIH

```

1      class SeedUsed(db.Document):
2          fish_seed_id = db.ReferenceField(SeedInventory, required=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          original_amount = db.IntField(required=True)
5          usage = db.IntField(required=True)
6          pond = db.StringField(required=True)
7          created_at = db.DateTimeField(default=datetime.datetime.now)
8          updated_at = db.DateTimeField(default=datetime.datetime.now)
9

```

GAMBAR 6. MODEL *DATABASE* RIWAYAT PEMAKAIAN BENIH

```

1      class FeedInventory(db.Document):
2          feed_name_id = db.ReferenceField(FeedName, required=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          id_int = db.SequenceField(required=True)
5          feed_category = db.StringField(required=True)
6          brand_name = db.StringField(required=True)
7          price = db.IntField(required=True)
8          amount = db.FloatField(required=True)
9          created_at = db.DateTimeField(default=datetime.datetime.now)
10         updated_at = db.DateTimeField(default=datetime.datetime.now)
11

```

GAMBAR 7. MODEL *DATABASE* INVENTARIS PAKAN

```

1      class SuplemenInventory(db.Document):
2          id_int = db.SequenceField(required=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          function = db.StringField(required=True)
5          name = db.StringField(required=True)
6          description = db.StringField(required=True)
7          price = db.IntField(required=True)
8          amount = db.FloatField(required=True)
9          type = db.StringField(required=True)
10         min_expired_period = db.IntField(required=True)
11         max_expired_period = db.IntField(required=True)
12         image = db.StringField(required=True)
13         created_at = db.DateTimeField(default=datetime.datetime.
14             now)
15         updated_at = db.DateTimeField(default=datetime.datetime.
16             now)
17

```

GAMBAR 8. MODEL *DATABASE* INVENTARIS SUPLEMEN


```

1      class ElectricInventory(db.Document):
2          id_int = db.SequenceField(required=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          type = db.StringField(required=True)
5          name = db.StringField(required=True)
6          daya = db.StringField()
7          price = db.IntField(required=True)
8          id_token = db.StringField()
9          month = db.StringField()
10         image = db.StringField(required=True)
11         created_at = db.DateTimeField(default=datetime.datetime.
now)
12         updated_at = db.DateTimeField(default=datetime.datetime.
now)
13

```

GAMBAR 9. MODEL *DATABASE* INVENTARIS LISTRIK

```

1      class AssetInventory(db.Document):
2          id_int = db.SequenceField(required=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          asset_category = db.StringField(required=True)
5          name = db.StringField(required=True)
6          description = db.StringField(required=True)
7          amount = db.IntField(required=True)
8          price = db.IntField(required=True)
9          image = db.StringField(required=True)
10         created_at = db.DateTimeField(default=datetime.datetime.
now)
11         updated_at = db.DateTimeField(default=datetime.datetime.
now)
12

```

GAMBAR 10. MODEL *DATABASE* INVENTARIS ASET

```

1      class FeedUsed(db.Document):
2          fish_feed_id = db.ReferenceField(FeedInventory, required
=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          original_amount = db.FloatField(required=True)
5          usage = db.FloatField(required=True)
6          pond = db.StringField(required=True)
7          created_at = db.DateTimeField(default=datetime.datetime.
now)
8          updated_at = db.DateTimeField(default=datetime.datetime.
now)
9

```

GAMBAR 11. MODEL *DATABASE* RIWAYAT PEMAKAIAN PAKAN

```

1      class SuplemenUsed(db.Document):
2          fish_suplemen_id = db.ReferenceField(SuplemenInventory,
required=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          original_amount = db.FloatField(required=True)
5          usage = db.FloatField(required=True)
6          pond = db.StringField(required=True)
7          created_at = db.DateTimeField(default=datetime.datetime.
now)
8          updated_at = db.DateTimeField(default=datetime.datetime.
now)
9

```

GAMBAR 12. MODEL *DATABASE* RIWAYAT PEMAKAIAN SUPLEMEN

```

1      class FeedName(db.Document):
2          id_int = db.SequenceField(required=True)
3          farm_id = db.ReferenceField(Farm, required=True)
4          type = db.StringField(required=True)
5          name = db.StringField(required=True)
6          description = db.StringField(required=True)
7          producer = db.StringField(required=True)
8          protein = db.IntField(required=True)
9          carbohydrate = db.IntField(required=True)
10         min_expired_period = db.IntField(required=True)
11         max_expired_period = db.IntField(required=True)
12         image = db.StringField(required=True)
13         created_at = db.DateTimeField(default=datetime.datetime.now)
14         updated_at = db.DateTimeField(default=datetime.datetime.now)
15

```

GAMBAR 12. MODEL *DATABASE* MERK PAKAN

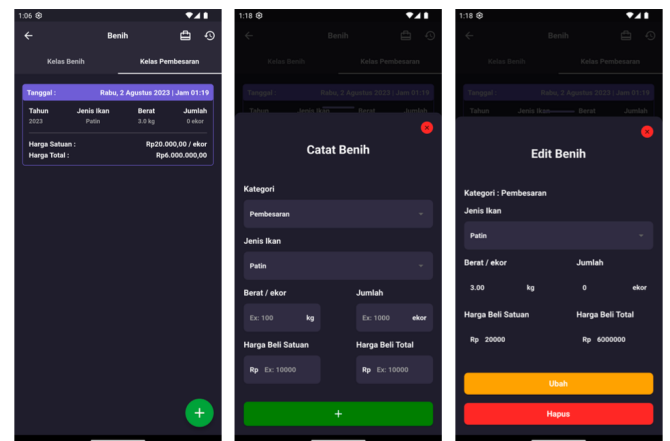
```

1      class DeactivationRecap(db.Document):
2          id_int = db.SequenceField(required=True)
3          pond_id = db.ReferenceField(Pond, required=True)
4          farm_id = db.ReferenceField(Farm, required=True)
5          fish_seed_id = db.ReferenceField(SeedInventory, required
=True)
6          fish_weight = db.FloatField(required=True)
7          fish_amount = db.IntField(required=True)
8          fish_type = db.StringField(required=True)
9          fish_category = db.StringField(required=True)
10         fish_price = db.IntField(required=True)
11         created_at = db.DateTimeField(default=datetime.datetime.
now)
12         updated_at = db.DateTimeField(default=datetime.datetime.
now)
13

```

GAMBAR 12. MODEL *DATABASE* PEMBUKUAN

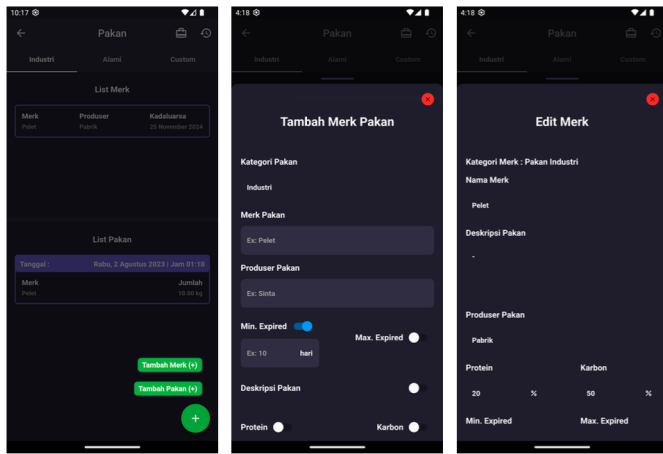
2. Tampilan *mobile*



GAMBAR 13. TAMPILAN INVENTARIS BENIH

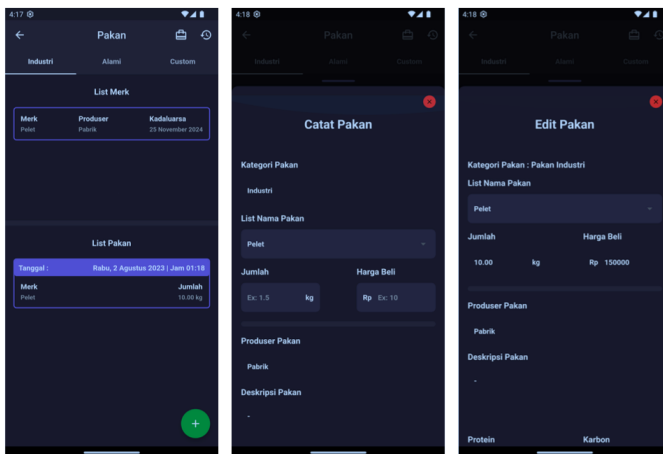
Pada tampilan ini, terdapat list dari benih yang ada pada inventaris serta *user* dapat mengakses halaman catat benih untuk menambahkan benih serta edit benih untuk mengubah data dari benih yang sudah ada.

Benih dibagi menjadi dua kelas yaitu kelas benih serta kelas pembesaran. Masing-masing kelas ini memiliki perbedaan dalam pengukuran sehingga form input berbeda di tiap kategori.



GAMBAR 14. TAMPILAN MERK PAKAN

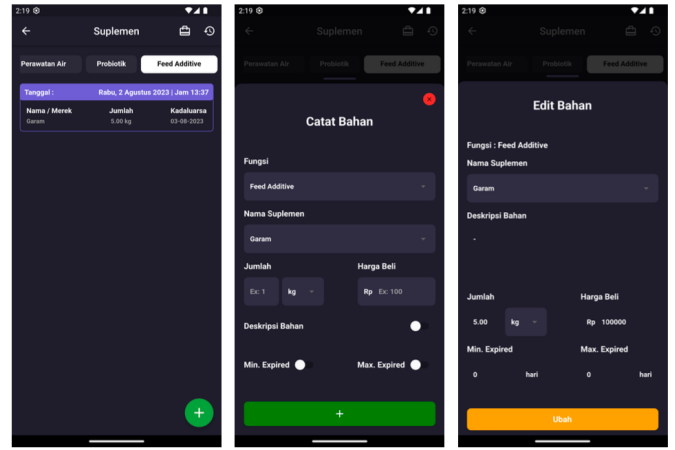
Pada halaman pakan juga terdapat opsi untuk tambah merk pakan. Terdapat form yang dapat diisi untuk menambahkan merk pakan dan list pakan juga bisa diubah melalui halaman edit merk pakan.



GAMBAR 15. TAMPILAN INVENTARIS PAKAN

Pada tampilan ini, terdapat list dari pakan yang ada pada inventaris serta *user* dapat mengakses halaman catat pakan untuk menambahkan pakan serta edit pakan untuk mengubah data dari pakan yang sudah ada.

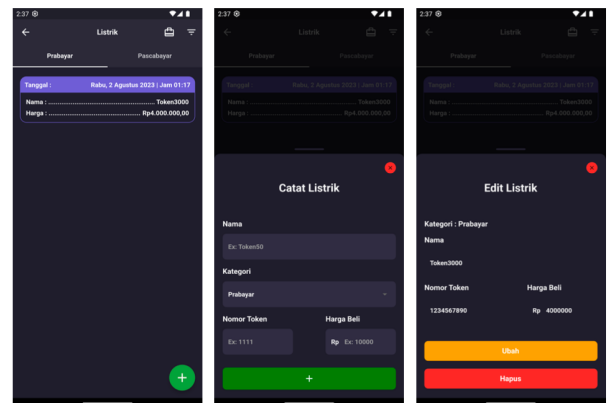
Pakan memiliki tiga kategori yaitu alami, industry, serta campuran. Untuk menambahkan pakan, wajib untuk menambahkan merk pakan terlebih dahulu agar memudahkan pendataan jika merk pakan sudah tersedia pada input pakan.



GAMBAR 16. TAMPILAN INVENTARIS SUPLEMEN

Pada tampilan ini, terdapat list dari suplemen yang ada pada inventaris serta *user* dapat mengakses halaman catat bahan untuk menambahkan suplemen serta edit bahan untuk mengubah data dari suplemen yang sudah ada.

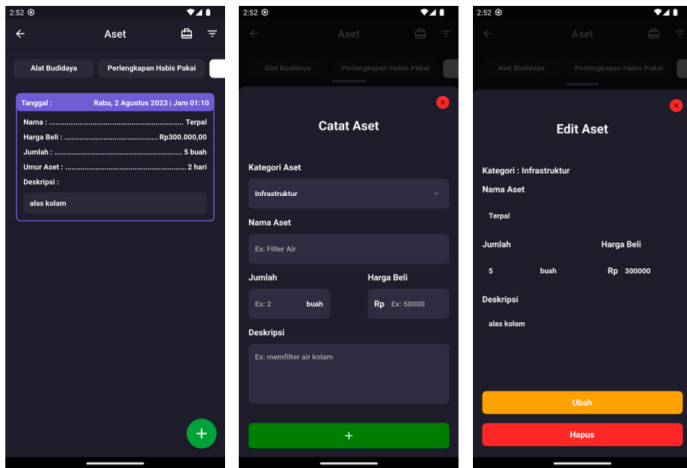
Data suplemen dibagi menjadi beberapa fungsi seperti obat, perawatan air, probiotik, serta feed additive. Masing-masing kategori tidak memiliki perbedaan dalam input.



GAMBAR 17. TAMPILAN INVENTARIS LISTRIK

Pada tampilan ini, terdapat list dari listrik yang ada pada inventaris serta *user* dapat mengakses halaman catat listrik untuk menambahkan listrik serta edit listrik untuk mengubah data dari listrik yang sudah ada.

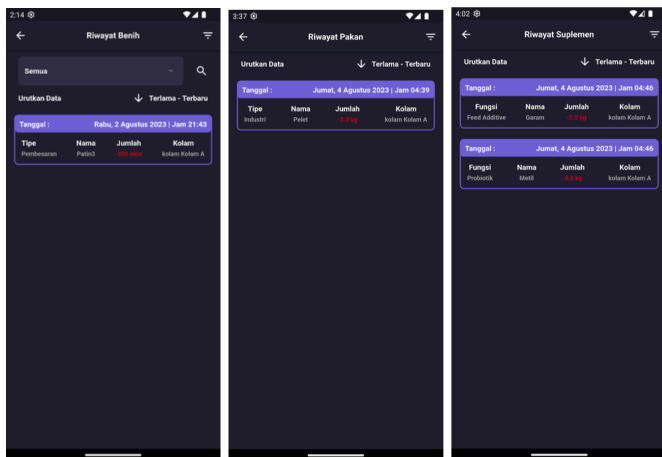
Data listrik dibagi menjadi dua kategori yaitu prabayar dan pascabayar yang masing-masing kategorinya memiliki jenis input yang berbeda.



GAMBAR 18. TAMPILAN INVENTARIS ASET

Pada tampilan ini, terdapat list dari aset yang ada pada inventaris serta *user* dapat mengakses halaman catat aset untuk menambahkan aset serta edit aset untuk mengubah data dari aset yang sudah ada.

Aset memiliki beberapa kategori seperti perlengkapan habis pakai, infrastruktur, dan lain-lain.

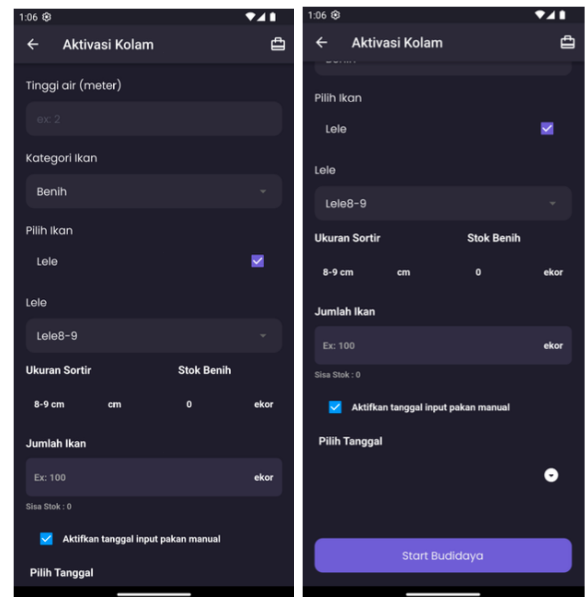


GAMBAR 19. TAMPILAN RIWAYAT PEMAKAIAN

Pada tampilan-tampilan ini, terdapat list dari pemakaian pakan, benih, dan suplemen. List ini berisi data dari nama, jumlah, serta tempat kolam yang digunakan.

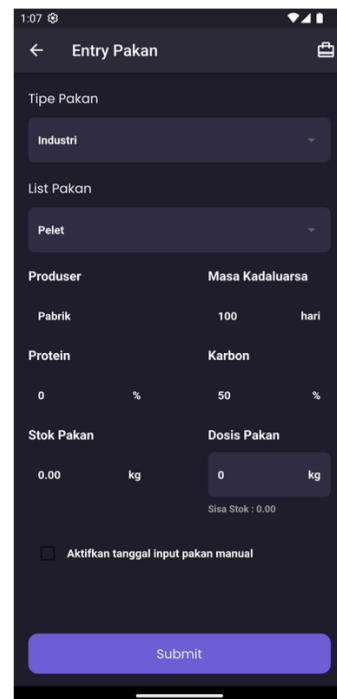
List ini juga dapat difilter berdasarkan kategori benihnya serta urutan data dari yang terbaru atau yang terlama.

Pada halaman riwayat benih, data pemakaian ini bisa di filter berdasarkan jenis dari benih. Sehingga memudahkan *user* untuk mendapatkan data yang diinginkan.



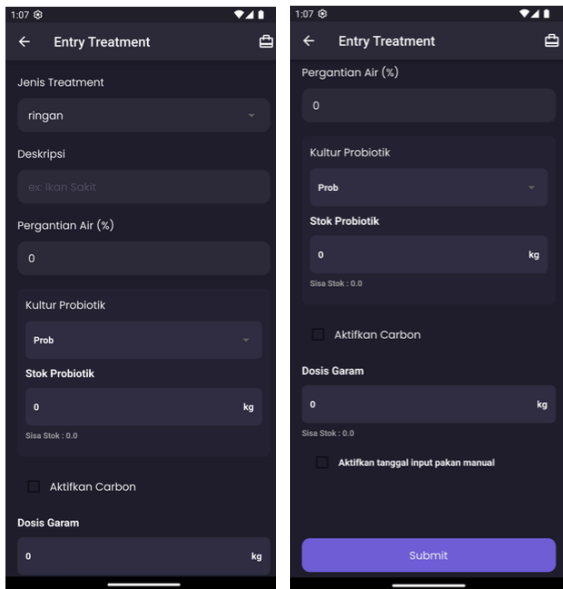
GAMBAR 20. TAMPILAN AKTIVASI KOLAM

Pada tampilan ini, halaman aktivasi kolam sudah terintegrasi dengan data dari inventaris benih. Jika ingin mengaktifkan kolam yang sudah terdaftar, pada form aktivasi kolam diharuskan untuk mencatat penggunaan ikannya.



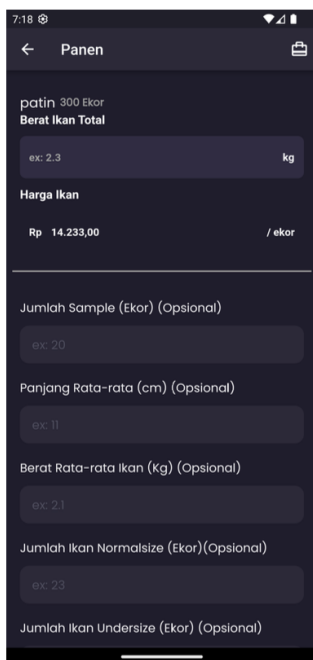
GAMBAR 21. TAMPILAN ENTRY PAKAN

Pada tampilan ini, halaman *entry* pakan sudah terintegrasi dengan data dari inventaris pakan. Jika ingin memberikan pakan, *user* dapat memilih pakan yang ingin digunakan serta memasukkan stok atau dosis yang tersedia pada inventaris pakan.



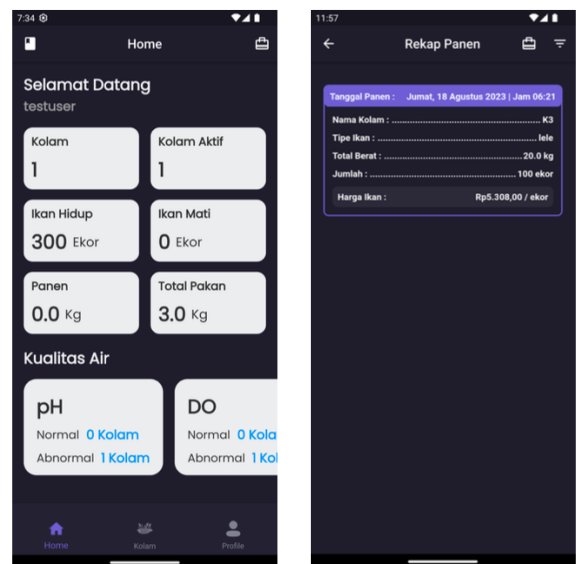
GAMBAR 22. TAMPILAN *ENTRY TREATMENT*

Pada tampilan ini, halaman *entry treatment* sudah terintegrasi dengan data dari inventaris suplemen. Pada opsi kultur prebiotik, data diambil dari inventaris suplemen kategori probiotik air. Lalu untuk karbon, data diambil dari inventaris suplemen kategori *feed additive*.



GAMBAR 23. TAMPILAN HALAMAN PANEN

Pada tampilan ini, terdapat form input yang dapat diisi jika kolam sudah siap dipanen. Selain pilihan yang opsional, terdapat data berat ikan total serta harga ikan per ekornya yang sudah dikalkulasikan berdasarkan data penggunaan inventaris yang di filter dari tanggal awal kolam dibuat hingga tanggal pada saat kolam siap panen.



GAMBAR 24. TAMPILAN REKAP PANEN

Pada tampilan ini, terdapat data dari hasil panen yang sudah dilakukan ketika panen. Di list ini terdapat informasi seperti nama kolam, tipe ikan, total berat ikan, jumlah ikan, serta harga ikan per ekornya.

TABEL 7. TABEL PENGUJIAN *UNIT TESTING*

Awal Tabel			
Skenario Pengujian	Kesesuaian		Kesimpulan
	sesuai	tidak sesuai	
Fitur inventaris pakan	✓		Diterima
Fitur inventaris suplemen	✓		Diterima
Fitur inventaris benih	✓		Diterima
Fitur inventaris listrik	✓		Diterima
Fitur inventaris aset	✓		Diterima
Akhir Tabel 4.11			

Awal Tabel			
Skenario Pengujian	Kesesuaian		Kesimpulan
	sesuai	tidak sesuai	
Aktivasi kolam dengan inventaris	✓		Diterima
Pemberian pakan dengan inventaris	✓		Diterima
Treatment kolam dengan inventaris	✓		Diterima
Panen kolam dengan harga jual minimum ikan	✓		Diterima
Pembukuan panen musim budidaya	✓		Diterima
Akhir Tabel 4.12			

Dari hasil tabel diatas, dapat disimpulkan bahwa dengan pengujian unit testing Fitur pencatatan inventaris sudah sesuai serta integarasi fitur pengembangan sebelumnya dengan fitur inventaris juga sudah sesuai. Kemudian untuk fitur panen yang memberikan *output* harga jual ikan minimum juga sudah sesuai dan diterima.

TABEL 8. TABEL PENGUJIAN *USER ACCEPTANCE TEST* (UAT)

<i>User Acceptance Test</i>					
No	<i>Acceptance Requirements</i>	Kesesuaian			
		SS	S	TS	STS
1	Fitur inventaris pakan	✓			
2	Fitur inventaris suplemen	✓			
3	Fitur inventaris benih	✓			
4	Fitur inventaris listrik	✓			
5	Fitur inventaris aset	✓			
6	Aktivasi kolam dengan inventaris	✓			
7	Pemberian pakan dengan inventaris	✓			
8	Treatment kolam dengan inventaris		✓		
9	Panen kolam dengan harga jual minimum ikan	✓			
10	Pembukuan panen musim budidaya		✓		

Dari hasil tabel diatas, dapat disimpulkan bahwa fitur-fitur dominan mendapatkan skor SS (Sangat Setuju). Untuk fitur treatment kolam dengan inventaris dan pembukuan panen musim budidaya mendapatkan skor S (Setuju) dikarenakan pada fitur treatment kolam perlu ditambahkan validasi dari masing-masing jenis inventaris suplemen agar dapat melakukan treatment dengan benar dan pada fitur pembukuan panen musim budidaya masih perlu ditambahkan informasi mengenai data panen selain data harga ikan.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil implementasi dan pengujian fitur aplikasi yang telah dirancang, maka diperoleh kesimpulan sebagai berikut:

1. Terbentuknya aplikasi *Aqua Breeding* versi kedua dengan fitur sistem inventarisasi serta penentuan harga jual minimum ikan. Adapun perancangan aplikasi ini dilakukan dengan metode Scrum dimulai dari tahap penyusunan *product backlog*, *sprint backlog*, dan dikerjakan dalam lima *sprint*.
2. Berdasarkan hasil pengujian, seluruh skenario pada *unit testing* berjalan dengan baik.

B. Saran

Adapun beberapa saran yang dapat penulis sampaikan untuk penelitian selanjutnya adalah:

1. Berdasarkan diskusi dengan *owner*, pada versi selanjutnya adalah penambahan fitur transaksi antar pembudidaya ikan yang berguna untuk penjualan ikan agar harga ikan hasil panen dapat digunakan.

DAFTAR PUSTAKA

- [1] Lin, Y.-B. (2019). Fishtalk: An iot-based mini aquarium system. *IEEE Access*
- [2] Ouyang, B. (2021). Initial development of the hybrid aerial underwater robotic system (haucs): Internet of things (iot) for aquaculture farms. *IEEE Internet of Things Journal*.
- [3] Rahmanto, A. (2022). Perancangan arsitektur aplikasi budidaya perikanan modern pada backend yang bertanggung jawab dalam melayani transaksi query webservice dengan menggunakan teknologi flask microservice. *Program Studi Ilmu Komputer Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Negeri Jakarta 2022*.
- [4] Maghriza, G. C. (2022). Perancangan frontend aplikasi pendukung teknologi perikanan modern dengan menggunakan framework flutter yang mentarget multi platform. *Program Studi Ilmu Komputer Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Negeri Jakarta 2022*.
- [5] Sim, S. (2022). *Dasar-Dasar Manajemen Keuangan (Fundamentals of Financial Management)*. Uwais Inspirasi Indonesia.
- [6] Supitriyani (2022). *Management Control*. Media Sains Indonesia.
- [7] Flutter (2017). Flutter. Diakses pada 30 Mei 2023 dari <https://flutter.dev/>.
- [8] Flask (2010). Flask. Diakses pada 30 Mei 2023 dari <https://flask.palletsprojects.com/en/2.3.x/>.
- [9] MongoDB (2009). MongoDB. Diakses pada 30 Mei 2023 dari <https://www.mongodb.com/>.
- [10] Scrum (2010). Scrum. Diakses pada 30 Mei 2023 dari <https://www.scrum.org/>.
- [11] Hareton K.N. Leung, P. W. W. (1997). A study of user acceptance tests. *Software Quality Journal volume*.