BAB I

PENDAHULUAN

A. Latar Belakang

Berkembangnya teknologi zaman sekarang membuat masyarakat ingin berbagai kemudahan dalam segala aspek pekerjaan, oleh karena itu adanya teknologi yang berkembang diharapkan membuat kemajuan dalam seluruh aspek kehidupan salah satunya memberikan manfaat bagi para petani-petani tambak budidaya udang. Ekosistem-ekosistem laut ini sangat menunjang keberlangsungan sumber daya perikanan di Indonesia (Lailiyah et al., 2018).

Udang vaname (*Litopenaeus vannamei*) merupakan salah satu produk perikanan yang diharapkan mampu menghasilkan devisa negara. Udang Vaname adalah salah satu komoditas yang sangat digemari oleh masyarakat, karena memiliki nilai ekonomi yang cukup tinggi dan banyak dibudidayakan di Indonesia. Adanya perkembangan pembudidayaan udang yang sangat pesat di semua negara dan daerah di antaranya di Indonesia di setiap pelosok-pelosok desa. Untuk menghasilkan kualitas udang, maka proses pemeliharaan harus memperhatikan aspek internal yang meliputi asal kualitas benih yang telah di tabur, air, pemberian pakan, serta pengendalian hama dan penyakit(Therapy et al., 2018).

Udang Vaname hanya dapat dibudidayakan secara tradisional, semi-intensif, dan intensif. Saat ini, teknologi yang tersedia hanya untuk pola intensif

dan semi-intensif. Pembesaran udang vaname secara intensif ditandai dengan padat tebar, modal, dan teknologi yang tinggi. Pembesaran udang dilakukan dalam beberapa tahap. Tahap pertama adalah persiapan lahan, yang mencakup pengeringan, perbaikan struktur, pengapuran, pemupukan, pemberantasan hama, dan pengisian air. Tahap kedua adalah penebaran benih, yang bergantung pada padat, jumlah, dan metode penebaran. Tahap ketiga adalah budidaya udang, yang mencakup pemeliharaan, kontrol kualitas air, pakan, dan pengendalian hama penyakit dan sampel. Tahap terakhir adalah pemrosesan udang.

Pemilihan jenis pakan, program pemberian pakan, pengkayaan pakan, pemberian pakan, waktu pemberian, kontrol anco, dan penyimpanan pakan adalah semua bagian dari manajemen pemberian pakan udang *vaname*. Untuk mempercepat laju pertumbuhan dan mempertahankan tingkat kelangsungan hidup udang *vaname*, dosis dan jenis pakan yang tepat harus diberikan. Manajemen pemberian pakan dilakukan sehingga waktu yang dibutuhkan untuk panen ukuran konsumsi udang vaname dari waktu penebaran benih di awal pemeliharaan menjadi waktu yang cepat.

Jika dilihat dari perkembangan budidaya udang vaname dari tahun ke tahun ada banyak perubahan yang terjadi dari ditemukannya teknologi untuk mengembangkan hasil budidaya udang vaname hingga ditemukannya faktor gagalnya panen dikarenakan penambak kurang mengetahui cara memelihara udang vaname. Sampai sekarang ini di Indonesia masih membudidayakan udang vaname, terkhususnya di Sulawesi Selatan di pesisir pantai Kota Pinrang banyak tambak-tambak yang membudidayakan udang *vaname* beserta ikan bandeng,

bahkan daerah pelosok hampir perbatasan Sulsel-Sulbar. Tetapi terdapat permasalahan hasil panen yang kurang maksimal bahkan mengakibatkan gagal panen yang dikarenakan banyak penambak yang tidak mengetahui informasi tentang cara pemeliharaan budidaya udang vaname. Maka dari itu penulis mengangkat judul penelitian yaitu **"APLIKASI SISTEM PANDUAN PEMBERIAN PAKAN UDANG VANAME BERBASIS ANDROID"**, sehingga membantu penambak mendapatkan informasi tanpa melakukan pencarian di internet melalui keyword. Dan semakin canggih teknologi sekarang ini di segala kalangan telah menggunakan *smart phone*, sehingga tidak menjadi alasan jika di daerah pelosok terkhususnya kota Pinrang tidak mampu mengakses informasi melalui internet karena telah terpasang menara pemancar jaringan dengan berbagai jenis seluler maupun terakses di daerah pesisir dan tersedianya *Wifi* di kantor desa dan beberapa warga memasang di rumahnya sendiri.

B. Rumusan Masalah

Berdasarkan latar belakang permasalahan tersebut permasalahan yang dapat dirumuskan adalah:

1. Bagaimana cara membuat aplikasi yang mampu membantu penambak dalam pemberian takaran pakan ideal yang diberikan pada udang vaname?
2. Bagaimana cara agar penambak mendapatkan informasi tentang budidaya udang *vaname* agar lebih kondusif dalam menggunakan aplikasi berbasis *android* pada *smart phone*-nya?

C. Batasan Masalah

Adapun batasan penelitian ini adalah jenis udang yang akan disajikan pada aplikasi berfokus pada jenis udang tambak, yaitu udang vaname yang di budidayakan di daerah kabupaten Pinrang yaitu di Kecamatan Suppa.

D. Tujuan Penelitian

Tujuan penelitian berdasarkan rumusan masalah yang telah diuraikan di atas adalah membuat aplikasi yang dapat membantu petambak mendapatkan informasi tentang panduan pemberian pakan udang *vaname* menggunakan *android* sehingga memanfaatkan *smartphone*-nya.

E. Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dari penelitian ini adalah :

1. Bagi Penulis

Penulis dapat mengaplikasikan ilmu dan teori yang telah didapatkan di bangku perkuliahan, serta dapat menambah wawasan tentang Aplikasi Panduan Pemberian Pakan Udang Vaname berbasis *Android.*

2. Bagi Pembaca

Penelitian ini diharapkan dapat memberikan informasi secara tertulis maupun sebagai referensi mengenai teknologi sistem informasi tentang takaran pakan udang vaname berbasis *android*.

3. Bagi Pengguna

Pengguna dapat mengakses informasi mengenai panduan pemberian takaran pakan udang *vaname* berbasis *android.*

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Udang Vaname

Udang vaname merupakan salah satu udang yang mempunyai nilai ekonomis dan merupakan jenis udang alternatif yang dapat dibudidayakan di Indonesia, disamping udang *windu* (*Panaeus monodon*) dan udang vaname (*Panaeus merguensis*), udang vaname tergolong mudah untuk dibudidayakan. Hal itu pula yang membuat para petambak udang di tanah air beberapa tahun terakhir banyak yang mengusahakannya. Udang vaname memiliki keunggulan yang tepat untuk kegiatan budidaya udang dalam tambak antara lain: Responsif terhadap pakan/nafsu makan yang tinggi, lebih tahan terhadap serangan penyakit, pertumbuhan lebih cepat, tingkat kelangsungan hidup tinggi, padat tebar cukup tinggi dan waktu pemeliharaan yang relatif singkat yakni sekitar 90 -100 hari per siklus.

Budidaya udang vaname dengan teknologi *intensif* mencapai padat tebar yang tinggi berkisar 100-300 ekor/m2. Tambak *intensif* adalah tambak yang dilengkapi dengan plastik mulsa yang menutupi semua bagian, pompa air, kincir air, *aerato*r, tingkat penebaran tinggi dan pakan 100% pelet. Pakan merupakan sumber nutrisi yang terdiri dari protein, lemak, karbohidrat, vitamin dan mineral yang dibutuhkan udang untuk pertumbuhan dan perkembangan secara optimal sehingga produktivitasnya bisa ditingkatkan.

Padat tebar berperan penting dalam kegiatan budidaya untuk menentukan jumlah benur yang akan ditebar dan luas tambak yang akan digunakan. Perbedaan kepadatan yang ditebar pada setiap petak tambak berpengaruh terhadap pertumbuhan dan kelangsungan hidup udang *vaname* yang dihasilkan (Musa et al., 2020).

2. Prosedur Budidaya Udang Vaname

2.1 Persiapan tambak

Perawatan tanah dasar danau dapat dilakukan beberapa hal, antara lain :

a. Evakuasi lumpur

Pengembangan harus meninggalkan sisa pengembangan sebagai slop alami dari sisa pakan, pemborosan udang harus dihilangkan dengan alasan berbahaya. Evakuasi cairan harus dapat dilakukan secara tepat dengan menggunakan alat atau atraksi dengan *siphon/alkon* air.

b. *Inversi* Tanah

Kotoran di bagian bawah danau sebaiknya dibalik dengan cara digaruk atau dicangkul untuk menghilangkan gas-gas yang ada di dalam kotoran.

c. Pengapuran

Alasan pengapuran hidroponik danau adalah untuk membunuh kotoran dan meningkatkan konsentrasi seluruh kekerasan terlihat di sekitar. Ini akan memperluas efisiensi danau hidroponik. Danau air baru dengan *alkalinitas absolut* di bawah 23 mg/l dan danau air pahit dengan *alkalinitas* di bawah 60 mg/l dan danau dengan *power hydrogen (pH)* di bawah 7 dapat diliputi oleh

pengapuran. Aturan umum untuk bagian pengapuran yang dapat diterapkan pada danau dicatat dalam tabel di bawah ini.

Tabel 2. 1 Dosis Pengapuran

Alkalinitas Total (mg/l)	***pH* Tanah**	**Kapur Pertanian *CaCO3* (Kg/Ha)**
Di Bawah 5	Di Bawah 5	3.000

Alkalinitas Total (mg/l)	***pH* Tanah**	**Kapur Pertanian *CaCO3* (Kg/Ha)**
5 – 10	5.0 – 5.4	2.500
10 – 20	5.5 – 5.9	2.000
20 – 30	6.0 – 6.4	1.500
30 – 50	6.5 – 7.0	1.000

Jika kedua data (*Alkalinitas* dan *pH* tanah) tersedia tetapi nilainya tidak sesuai dengan tabel, maka variabel yang digunakan adalah yang mempunyai dosis pengapuran paling besar. Sebagai contoh, Total alkalinitas 15 mg/l tetapi *pH* tanah5.1, dosis pengapuran yang dipakai adalah untuk *pH* 5.1 (2.*500 kg/Ha*). Di kolam air payau dengan total alkalinitas 80 *mg/l pH* tanah 5.5, dosis pengapuran yang digunakan adalah 2.000 *kg/Ha* karena *pH* (5.5).

d. Pengeringan

Pengeringan dilakukan setelah tanah diberikan kapur, biarkan hingga tanah kering dan pecah – pecah untuk membunuh bibit penyakit, Dengan pengeringan selama 2–3 minggu, sebagian besar bahan organik yang ada di tanah dasar dari siklus sebelumnya akan terurai dan senyawa anorganik 24 akan dioksidasi (Boyd dan Dippopinyo, 1994). Lamanya pengapuran tergantung tekstur tanah, *temperature* udara, kondisi angin, curah hujan dan rembesan air dari kolam sekitarnya.

e. Persiapan

Urea dapat disebarkan lebih dari 200-400 kg untuk setiap hektar tanah danau untuk memberi energi disintegrasi tanah alami, karena *nitrogen* dalam *urea* dapat mengoksidasi campuran alami, terutama untuk tanah danau yang tidak dapat mengering. Pengapuran sebaiknya tidak dilakukan pada jam aplikasi *urea* untuk mencegah *pH* tinggi. *Urea* akan dihidrolisis menjadi garam berbau. Jika *pH* lebih dari 8, sebagian garam yang berbau akan berdifusi ke udara.

2.2 Persiapan air tambak

Estimasi batas kualitas air dari hari ke hari sebaiknya dilakukan setiap hari (5.00-6.00) dan sore hari (12.00-14.00). Pada jam-jam tersebut merupakan titik dasar yang dapat menggambarkan keadaan perairan. 5.00-6.00 pagi adalah bagian bawah mutlak dari oksigen yang hancur dan kandungan *pH* dan karbon dioksida yang paling tinggi. Pukul 12.00-14.00 merupakan puncak *fotosintesis fitoplankton*, penguraian kandungan oksigen dan *pH* air. Secara keseluruhan, kualitas air yang baik dapat diperoleh dengan beberapa prosedur administrasi, termasuk mengganti air secara konsisten ketika faktor kualitas air mulai menunjukkan pembusukan, menghindari kelebihan muatan dengan melakukan pemberian makan yang sesuai, sirkulasi udara, menyedot dan membuang limbah alam di tempat pembuangan. bagian bawah danau, dan menyimpan mikroorganisme dengan ketebalan yang tinggi seperti probiotik (*bioremediasi*) seperti halnya penggunaan senyawa sintetik (*synthetic compound*).

Tabel 2. 2 *Parameter* Air

No	*Parameter*	*Kisaran Optimum*
1	Suhu	28 – 30 °C
2	Salinitas	15 -25 ppt
3	Kecerahan	30 – 40 cm
4	Oksigen terlarut	4 – 8 mg/l
5	*pH*	7 – 8,5
6	Alkalinitas / salinitas	100 – 120 mg /l

Tabel 2. 3 Kebutuhan *Aerator* / Kincir Berdasarkan *Biomassa* Udang

Kepadatan udang (ekor/m2)	Biomassa (kg)	*Aerator* (*HP*)
50	5.000	10
60	6.000	12
70	7.000	14
80	8.000	16
90	9.000	18
100	10.000	20

2.3 Penebaran benur

a. Waktu Penebaran Benih

Penebaran benih *vaname* selesai setelah petak danau disiapkan untuk pemeliharaan dengan memperhatikan kualitas air, jam pemuatan harus dilakukan pada awal hari sebelumnya pukul 08.00 atau sekitar waktu malam atau selama kondisi iklim yang tidak jelas mengingat Fakta bahwa sekitar saat itu kondisi varian suhu tidak menonjol, batas air lainnya seperti *pH*, rasa asin tidak banyak berubah. Ini tidak pernah benar-benar perasaan cemas pada benih yang akan dipasok.

b. Aklimatisasi

Aklimatisasi adalah cara menuju perubahan sesuai dengan iklim baru udang *vaname* yang akan dipindahkan ke iklim yang meninggi (danau) sehingga tidak menimbulkan tekanan yang menyebabkan lewat, sebelum benih disuplai, air danau asin dan asin sudah terjamin. benih diperiksa, suhu di danau dan suhu yang dibutuhkan. Benih ditangkap setelah itu kantong telaga di angin-anginkan di salah satu sudut telaga selama kurang lebih 30-45 menit, intinya mengubah suhu air telaga menjadi suhu yang dijaga benur. Setelah 45 menit karung benur dibuka dan air secara bertahap ditambahkan dari danau, dilakukan secara fisik menggunakan tangan atau sendok, ukuran aklimatisasi asin selesai sehingga lebih cepat, *volume* udara yang ditambahkan ke pak benur diubah (- + 1/3 volume karung benur), untuk menentukan kewajaran asinan danau dengan asinan benur yang ditangkap, dilakukan pendugaan dengan menggunakan *refraktometer*.

3. Manajemen Perhitungan Pakan Udang Vaname

Pakan berperan sangat besar dalam mencapai keberhasilan budidaya udang. Biaya pakan mencapai lebih dari 50% - 60% dari biaya total sehingga perlu adanya manajemen pemberian pakan yang baik untuk mendukung keberhasilan budidaya. Tingkat pemberian pakan dalam budidaya perairan ada dua tipe, yaitu :

1. Takaran tetap (*Blind Feeding*)

Metode *Blind Feeding* dilakukan di bulan pertama, pakan diberikan berdasarkan jumlah tebar. Sebagai contoh dalam bulan pertama (sampai umur 30

hari) pemberian pakan untuk kepadatan 100.000 ekor sebanyak 1,0 kg/hari dan pada umur 30 hari mencapai 8,5 kg/hari. Dengan cara tersebut diharapkan dapat diperoleh sintasan yang tinggi dan ukuran yang merata 2-3 g/ekor dalam 30 hari pemeliharaan.

Tabel 2.4 Pemberian pakan dalam satu bulan pertama (1-30 hari) untuk 100.000 benur vaname.

Umur (Hari)	Pakan (Kg)	Umur (Hari)	Pakan (Kg)	Umur (Hari)	Pakan (Kg)
1	1	11	3,0	21	5,0
2	1,2	12	3,2	22	5,3
3	1,4	13	3,4	23	5,6
4	1,6	14	3,6	24	5,9
5	1,8	15	3,8	25	6,2
6	2,0	16	4,0	26	6,5
7	2,2	17	4,2	27	7,0
8	2.4	18	4,4	28	7,5
9	2,6	19	4,6	29	8,0
10	2,8	20	4,8	30	8,5

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa takaran pakan udang umur 1-20 hari mengalami kenaikan sebanyak 2% setiap harinya, umur 21-26 hari mengalami kenaikan sebanyak 3% setiap harinya dan umur 27-30 mengalami kenaikan sebanyak 5% setiap harinya. Sehingga diperoleh sebuah rumus sebagai berikut :

1. JP = $\frac{\text{Jumlah tebar}}{100.000}$ + 0,2 * umur (hari) – 0,2 (untuk umur (hari) 1 - 20)

2. JP = $\frac{\text{Jumlah tebar}}{100.000}$ + 0,3 * umur (hari) – 2,3 (untuk umur (hari) 21 - 26)

3. JP = $\frac{\text{Jumlah tebar}}{100.000}$ + 0,5 * umur (hari) – 7,5 (untuk umur (hari) 27 - 30)

Keterangan:

JP = Jumlah pakan yang diberikan (kg)

2. Metode *demand feeding* (*Sampling*)

Metode *demand feeding* dilakukan mulai bulan ke-2 atau jumlah pakan harian setelah 30 hari dapat diatur dalam dua cara:

2.1 Jumlah pakan berdasarkan kebutuhan

Pemberian pakan berdasarkan kebutuhan melalui pemantauan konsumsi pakan udang dilakukan terus menerus dengan memeriksa seluruh wadah pakan (anco) yang dipasang khusus pada tempat tertentu pada tambak . Jumlah anco tergantung luas tambak.

Tabel 2.5 Jumlah anco yang dipasang berdasarkan luas tambak pada budidaya udang *vaname* secara *intensif*.

Luas tambak (ha)	Jumlah anco (unit)
0,4 - 0,5	4
0,6 - 0,7	5
0,8 - 1,0	8-10

Pemberian pakan dalam anco dilakukan setelah penebaran pakan selesai di petakan tambak. Pakan dimasukkan kedalam anco dan diturunkan secara perlahan lahan kedalam tambak agar pakan tidak terhambur kedalam air. Penyesuaian dosis pakan berdasarkan sisa pakan di anco pada waktu kontrol anco dapat di lihat

pada tabel di bawah ini, hal ini dimaksudkan untuk mengetahui kemampuan nafsu makan dan kondisi kesehatan udang sekaligus menjadi dasar pertimbangan penentuan jumlah pakan perhari.

Tabel 2.6 Penyesuaian dosis pakan berdasarkan sisa pakan di anco.

Sisa pakan di anco (%)	Skor	Penyesuaian dosis pakan(%)
0 (Habis)	0	Ditambah 5
<10	1	Tetap sama
10 – 25	2	Dikurangi 10
25 – 50	3	Dikurangi 20
>50	4	Dikurangi 40

2.2 Penyesuaian jumlah pakan berdasarkan perkiraan pakan yang ideal

Cara ini lebih jelas dengan contoh berikut, berupa perkiraan jumlah pakan harian untuk 30 – 37 hari pemeliharaan dengan padat tebar 100.000 ekor. Misalnya pada hari ke 30 mencapai sintasan 90% berat rata-rata 2 g dan pertambahan berat rata-rata(pertumbuhan) 0.15g. Proyeksi jumlah pakan harian untuk 30-37 hari berdasarkan perkiraan pertumbuhan harian berat badan rata-rata dan persentase pemberian pakan menggunakan rumus :

$$JP = JU \times S \times BB \times FR$$

Keterangan: JP = Jumlah pakan yang diberikan (kg)

JU = Jumlah udang ditebar (ekor)

S = Sintasan (%)

FR = Persentase pemberian pakan (%)

Perkiraan:

Jumlah udang ditebar : 100.000 ekor

Sintasan : 90% (dengan melihat kondisi pertumbuhan udang/target yang ingin dicapai)

Berat badan rata-rata : 2,15 g (Dilakukan dengan cara menimbang udang)

Persentase pemberian pakan : 6,80% (dapat dilihat pada tabel 2.9)

Jumlah pakan yang diberikan pada hari ke 31 : 12.600 g.

Tabel 2.7 Jumlah pakan harian selama 30 – 37 hari pemeliharaan berdasarkan perkiraan pertumbuhan harian, rata-rata berat badan, dan dosis pemberian pakan.

Pemeliharaan (hari)	Berat badan rata-rata (g)	Persentase pemberian pakan (%)	Jumlah pakan (g)
31	2,15	6,80	13.158
32	2,30	6,68	13.828
33	2,45	6,58	14.509
34	2,60	6,48	15.163
35	2,75	6,38	15.790
36	2,90	6,28	16.390
37	3,15	6,18	17.520

Persentase pemberian pakan merupakan jumlah pakan harian yang diberikan berdasarkan *biomassa* udang. Nilai persentase pemberian pakan yang digunakan sebagai pedoman pemberian pakan bukanlah merupakan suatu nilai baku, tetapi tergantung kondisi budidaya. Aplikasi persentase pemberian pakan yang tepat dalam strategi pemberian pakan dapat menghasilkan pertumbuhan optimal dan penggunaan pakan secara efisien. Nilai persentase pemberian pakan

semakin kecil apabila berat badan rata-rata semakin besar dan sebaliknya.

Tabel 2. 8 Hubungan persentase pemberian pakan dengan berat badan rata-rata udang vaname.

Berat badan rata-rata (g)	Persentase Pemberian Pakan (%)	Berat badan rata-rata (g)	Persentase Pemberian Pakan (%)
1 – 2	10	11,1 - 12	3
2,1 – 3	7	12,1 – 13	2,9
3,1 – 4	6	13,1 – 14	2,7
4,1 – 5	5,5	14,1 – 15	2,5
5,1 – 6	5	15,1 – 16	2,3
6,1 – 7	4,5	17,1 – 17	2,1
7,1 – 8	4,2	18,1 - 18	2
8,1 – 9	4	19,1 - 19	1,9
9,1 – 10	3,7	20,1 – 20	1,8
10,1 – 11	3,5	21,1 – 21	1,7

Untuk mengetahui FR (persentase pemberian pakan (%)) pengguna hanya dapat melihat berapa berat rata-rata udang kemudian mencocokkan dengan tabel sesuai dengan berat rata-rata udang tersebut maka pengguna tinggal memasukkan kedalam aplikasi vaname shrimp sehingga memperoleh jumlah pakan yang akan diberikan.

4. *Android*



Gambar 2. 1 Logo *Android*

Android adalah sebuah sistem operasi untuk perangkat lunak mobile berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc*. membeli *Android Inc* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel / smartphone. Kemudian untuk mengembangkan *Android,* dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, Htc, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia*. Pada saat perilisan perdana *Android*, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan open *source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode *android* dibawah *lisensi Apache*, sebuah *lisensi* perangkat lunak dan open platform perangkat seluler.

Didunia ini terdapat dua jenis distributor sistem operasi *android*. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services (GMS)* dan kedua adalah yang benar - benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution (OHD).*

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini *android* menjadi pesaing utama dari *Apple* pada sistem operasi Table PC. Pesatnya

pertumbuhan *Android* selain faktor yang disebutkan diatas adalah karena *android* itu sendiri adalah platform sangat lengkap baik itu sistem operasinya, aplikasi dan *Tool Developmen*, Market aplikasi *android* serta dukungan yang sangat tinggi dari komunitas *Open source* didunia, sehingga *android* terus berkembang pesat dari segi teknologi maupun dari segi jumlah device yang ada didunia (Yunus et al., 2015).

5. *Android Studio*



Gambar 2. 2 *Logo Android Studio*

Android Studio merupakan sebuah IDE *(Integrated Development Environment)* untuk mengembangkan aplikasi *Android*. Aplikasi ini diterbitkan oleh *Google* pada tanggal 16 Mei 2013 dan tersedia secara gratis dengan lisensi *Apache 2.0. Android Studio* menggantikan perangkat lunak pengembangan *Android* sebelumnya yaitu *Eclipse*. IDE (Integrated Development Environment) adalah aplikasi untuk pengembang perangkat lunak yang berisi fungsi-fungsi terintegrasi yang diperlukan untuk membangun perangkat lunak, seperti editor kode, debugger, kompiler, dan sebagainya.

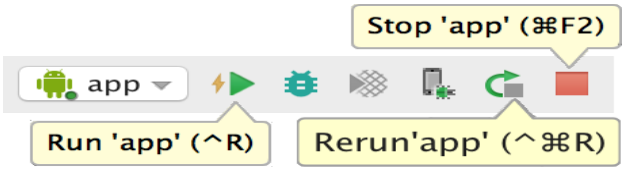
Android Studio sendiri dikembangkan berdasarkan *IntelliJ IDEA*, mirip dengan *Eclipse*, disertai dengan plugin ADT *(Android Development Tools)*. *Android studio* memiliki fitur:

1. Proyek berdasarkan *Gradle Build*.

2. Refactoring cepat dan perbaikan bug.
3. *Tools* baru bernama "*Lint*" mengklaim dapat dengan cepat memantau kecepatan, kegunaan, dan kompatibilitas aplikasi.
4. Mendukung *Proguard* dan penandatanganan aplikasi untuk keamanan.
5. Memiliki GUI aplikasi *Android* lebih mudah
6. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan.

Android Studio dipilih karena memiliki banyak fitur yang memudahkan para programmer, khususnya programmer tingkat dasar yang ingin mempelajari lebih lanjut tentang *Android*. Meski menggunakan *Android studio* memakan cukup banyak RAM pada perangkat PC, namun Android Studio memiliki sejumlah keunggulan lain, yaitu:

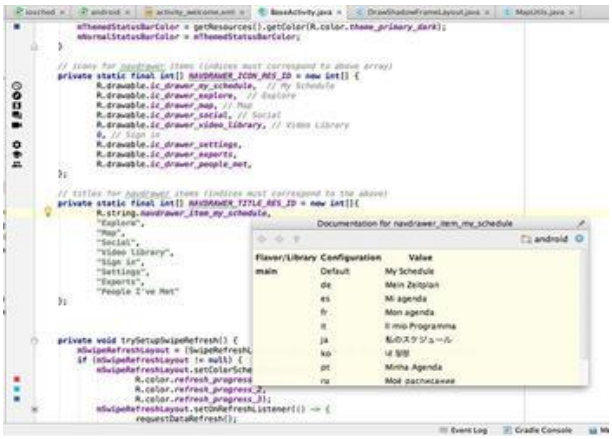
1. *Instant RUN*



Gambar 2. 3 *Instant Run*

Fitur *Instant Run* dapat memastikan program berjalan dengan cepat tanpa perlu mengkompilasi ulang aplikasi atau membuat ulang APK saat melakukan perubahan kode, sehingga proses yang dihasilkan lebih cepat.

2. *Intelligent Code Editor*



Gambar 2. 4 *Intelligent Code Editor*

Android Studio memiliki *Intelligent Code Editor* yang memudahkan analisis kode dan memberikan saran kode untuk digunakan dengan sistem *auto complete.* Saat kita mengetik kode, *Android Studio* akan secara otomatis menyarankan kelas jika kita memiliki kelas yang diinstal dan kita dapat menekan tombol TAB untuk memasukkan kode jika sesuai dengan kebutuhan. Fitur-fitur tersebut tentunya mempercepat pembuatan program sehingga membuat kinerja pembuat program menjadi lebih produktif.

3. Sistem Versi yang *Fleksibel*

Android Studio menawarkan otomatisasi versi, manajemen dependensi, dan konfigurasi versi yang dapat disesuaikan. Anda dapat mengonfigurasi proyek Anda untuk menyertakan pustaka lokal dan yang dihosting, serta menentukan varian versi yang berisi kode berbeda. Teman-teman bisa mengkonfigurasi dan menginstall library yang memudahkan teman-teman dalam membuat aplikasi *Android*. Fitur ini merupakan bagian

dari *fleksibilitas Android Studio*.

4. Dioptimalkan untuk semua perangkat *Android*

Android Studio memberi Anda tempat untuk membuat aplikasi untuk berbagai perangkat *Android,* seperti tablet *Android, Android Wear, Android TV,dan Android Auto*. Fungsi terstruktur ini memungkinkan Anda membagi proyek menjadi unit-unit fungsional yang bisa anda buat, uji,dan men-debug sesuai keinginan Anda.

5. Di desain untuk Tim

Android Studio memiliki integrasi dengan beberapa kontrol versi popular seperti *Git* dan *Subversion.* Bahkan untuk memudahkan kolaborasi, kita juga bisa menggunakan layanan *Github* langsung dari *Android Studio*. Dengan cara ini, pembaca dan tim terus bekerja secara efektif dengan proyek-proyek yang mudah diakses satu sama lain (Erni Sri Wahyuni, 2019).

6. *MySQL*



Gambar 2. 5 *Logo MySQL*

MySQL adalah perangkat lunak sistem manajemen basic data *SQL* (*database management system*) atau *DBMS*, yang bersifat *multi-threaded* dan *multi-user*. *MySQL* adalah sistem manajemen basis data relasional (*RDBMS)*, didistribusikan secara gratis di bawah *GPL* (Lisensi Publik Umum). Setiap orang

bebas menggunakan *MySQL*, tetapi tidak dapat digunakan sebagai sumber tertutup atau produk turunan komersional. *MySQL* sebenarnya merupakan turunan dari *SQL (Structured Query Language*), salah satu konsep utama dalam database lama.

SQL merupakan konsep operasi basis data, khususnya konsep operasi basis data yang digunakan untuk memilih dan memasukkan data, sehingga operasi data dapat diselesaikan dengan mudah dan otomatis. Keunggulan sistem database (*DBMS*) dapat dilihat dari cara pengoptimal memproses perintah *SQL* yang dibuat oleh pengguna dan aplikasinya. Sebagai server database, *MYSQL* dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk *query* yang dilakukan oleh single user, kecepatan *MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase* (Ramadhan & Mukhaiyar, 2020).

7. *Java*



Gambar 2. 1 *Logo Java*

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh *James Gosling* saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang

terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin atas bawah yang minimal. Aplikasi berbasis *java* umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (*JVM*). *Java* merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi *java* mampu berjalan di beberapa platform sistem operasi yang berbeda, *java* dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini *java* merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

Java adalah bahasa pemrograman yang *multi platform* dan *multidevice*. Sekali anda menuliskan sebuah program dengan menggunakan *Java*, anda dapat menjalankannya hampir di semua komputer dan perangkat lain yang support *Java*, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis *Java* ini dikompulasikan ke dalam code dan bisa dijalankan dengan *Java Virtual Machine*. Fungsionalitas dari *Java* ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya yang umum dan nonspesifik. Kelebihan *Java* yang pertama tentu saja multiplatform. *Java* dapat dijalankan dalam beberapa platform komputer dan sistem operasi yang berbeda. Hal ini sesuai dengan slogannya yang sudah dibahas sebelumnya. Yang kedua adalah OOP atau *Object Oriented Programming*.

Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam *Java*. Hal ini akan memudahkan pemrograman menjadi lebih mudah. Kelengkapan *library* semakin beragam jika ditambah dengan karya komunitas *Java*. Setiap hal pasti memiliki kelebihan dan kekurangan (Irsan, 2015).

8. UML (*Unified Modelling Language*)

(Wardhani, 2014) UML (*Unified Modelling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma beriorentasi objek.Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu :

1. Sesuatu (*things*)

   Ada empat *things* dalam *Unified Modelling Language* (UML):

   a. *Structural things,* bagian yang relatif statis dapat berupa elemen yang bersifat fisik maupun konseptual.
   b. *Behavorial things,* bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
   c. *Grouping things,* bagian yang pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem.
   d. *An notational things,* merupakan bagian yang memperjelas model UML.

Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen dalam model UML.

2. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

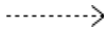
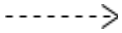
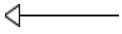
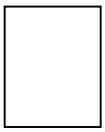
a. Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen *independent* akan mempengaruhi elemen yang bergantung padanya.
b. Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
c. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada diatasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisai sedangkan arah sebaliknya dinamakan generalisasi.
d. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.

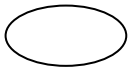
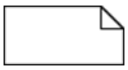
3. Diagram, *Unified Modelling Language* (UML) menyediakan sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berintraksi dengan sistem yang sedang di bangun.

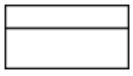
Adapun simbol-simbol *use case* diagram antara lain :

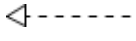
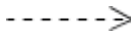
Tabel 2. 9 Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		*Actor*	Mendefinisikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case.*
2.		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
3.		*Include*	Mendefinisikan bahwa *usecase* sumber secara *eksplisi*t.
4.		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
5.		*Extend*	Menspesifikasikan bahwa *usecase* target memperluas perilaku dari *usecase* sumber pada suatu titik yang diberikan.
6.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

NO	GAMBAR	NAMA	KETERANGAN
8.		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9.		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah elemen-elemennya (sinergi).
10.		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 10 Simbol *Class* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2.		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		Class	Himpunan dari objek- objek yang berbagi atribut serta operasi yang sama.

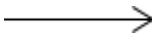
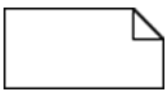
NO	GAMBAR	NAMA	KETERANGAN
4.		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5.		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 11 Simbol *Sequence* Diagram

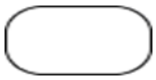
NO.	GAMBAR	NAMA	KETERANGAN
1.		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi .
2.		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

NO.	GAMBAR	NAMA	KETERANGAN
3.		Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Tabel 2. 12 Simbol *State Chart* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1.		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu yang dimiliki suatu objek.
2.		*Intial Pseudo State*	Bagaimana objek dibentuk atau diawali.
3.		*Final State*	Bagaimana objek dibentuk.
4.		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya.
5.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6.		*Node*	Elemen fisik yang eksis saat aplikasi di jalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 13 Simbol *Actifity* Diagram

NO.	GAMBAR	NAMA	KETERANGAN
1.		*Actifity*	Memperlihatkan bagaimana masing masing kelas antar muka saling berinteraksi satu sama lain.
2.		*Action*	State dari system yang mencerminkan eksekusi dari suatu aksi.
3.		*Initial Node*	Bagaimana objek dibentuk.
4.		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan.
5.		Fark node	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

B. Kajian Hasil Penelitian Terdahulu

Melihat dari beberapa judul dan tema yang berkaitan dengan penelitian yang dilakukan, diperoleh persamaan dan perbedaan dari penelitian yang akan dilakukan untuk membuktikan bahwa penulisan penelitian asli dan bukan duplikasi dari tugas akhir penelitian lain, seperti berikut ini:

1. Astrialu Kasirang, 2021 “Panduan Budidaya Udang Tambak di Kabupaten Pinrang Berbasis Android”. Dalam aplikasi ini peneliti merancang sistem informasi pembudidayaan udang tambak, yang memberikan informasi

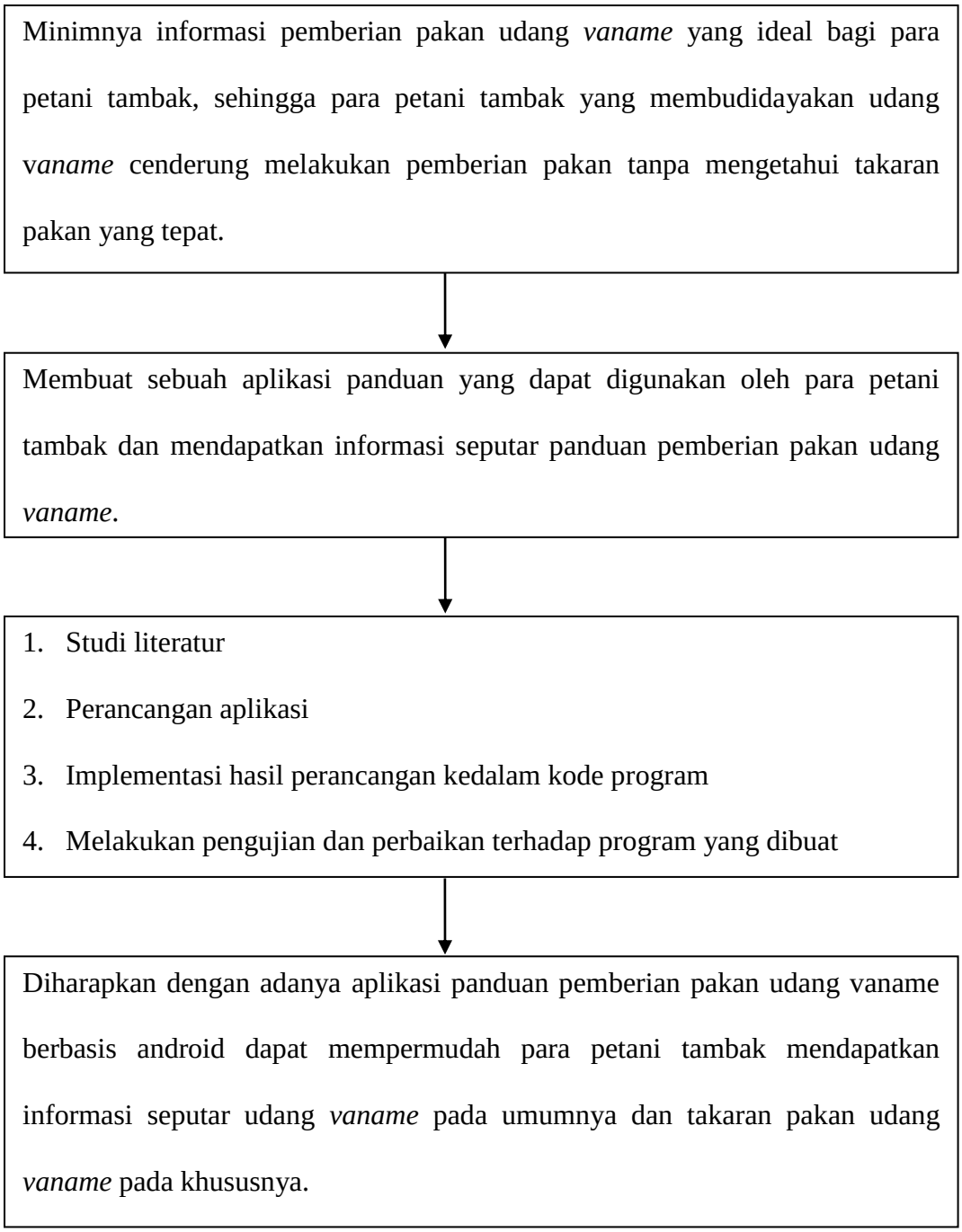
seputar budidaya udang *windu* dan udang *vaname*. Dimana di dalamnya terdapat beberapa fitur yang dapat dibaca oleh penggunanya. Sistem Informasi ini dirancang dengan menggunakan bahasa pemrograman PHP.

2. Anwar, 2018. "Perancangan Aplikasi Penentuan Takaran Pupuk Pada Tanaman". Penelitian ini mempunyai tujuan yang sama, seperti mempermudah penentuan takaran. Dalam perancangannya menggunakan bahasa pemrograman *Delphi Xe2*, dimana dapat menghasilkan informasi yang lebih cepat dibanding dengan sistem yang sedang berjalan kemudian menjadi solusi atas masalah yang dihadapi petani sehingga tercipta peningkatan produktifitas di instansi tersebut.

Dari kedua penelitian yang telah dilakukan maka dapat dilihat dari perbedaan dengan kajian terdahulu, yaitu pada penelitian pertama perbedaannya bahwa penulis berfokus pada udang *vaname*, dimana nantinya terdapat fitur hitung pakan sehingga mempermudah pengguna untuk menentukan takaran pakan yang tepat . Dan pada penelitian kedua memiliki persamaan seputar takaran, tetapi dalam hal ini penulis berfokus pada pemberian takaran pakan udang *vaname* berbasis *android.*

C. Kerangka Pikir

Untuk Memahami alur peneltian diatas, diuraikan ke dalam kerangka berpikir yang akan disajikan dalam bentuk diagram ini:



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Data yang digunakan dalam penelitian ini berupa data-data yang telah dikumpulkan melalui Penelitian Pustaka (*Library Research*). Adapun jenis data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

1. Data primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber data dan pengetahuan.

2. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi dari internet, website dan jurnal.

B. Lokasi dan Waktu Penelitian

Adapun Waktu dan Tempat Penelitian yang dilaksanakan oleh penulis yaitu ± selama 1 bulan, yang berlokasi di tambak, tepatnya di Cikuale, Desa Maritengngae, Kecamatan Suppa, Kabupaten Pinrang, Provinsi Sulawesi Selatan, Indonesia.

C. Metode Pengumpulan Data

Secara umum tahapan penelitian akan dilakukan sebagai berikut :

1.Study *Literatur*

Study *Literatur* yaitu dengan cara mengumpulkan literatur, jurnal, paper, dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

2. Observasi

Observasi yaitu dengan mengamati secara langsung proses memelihara udang *vaname*.

3. Wawancara

Wawancara yaitu cara memperoleh informasi melalui komunikasi percakapanyang dilakukan tatap muka langsung ataupun melalui telepon.

D. Tahapan Penelitian

1. Analisis data

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

2. Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta software yang digunakan selama penelitian.

3. Pengumpulan Data

Pada tahap ini peneliti melakukan observasi dengan peninjauan, pencatatan dan pengamatan langsung di tempat penelitian.

4. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang diterapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

5. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

6. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

7. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk digunakan oleh user.

E. Alat dan Bahan Penelitian

Untuk melakukan proses penelitian, maka yang harus diperlukan adalah alat dan bahan penelitian, guna mendukung kegiatan penelitian tersebut. Adapun alat dan bahan sebagai berikut:

a. Laptop *TOSHIBA* dengan spesifikasi :

i. *Processor* *: Intel Core i3*

ii. *RAM* *: 2.00 GB*

iii. *Hardisk* *: 320 GB*

iv. *LCD Monitor : 14 “*

b. **Perangkat *Android Mobile***

i. *Processor* *: OPPO A92*

ii. *RAM* *: 8GB*

iii. *LCD* *: 6,5 “*

c. *Software* yang digunakan dalam pembuatan aplikasi yaitu :

i. *Windows 10*

ii. *Java*

iii. *Android Studio*

F. Metode Pengujian

Dalam penelitian ini, digunakan 2 (dua) metode dalam pengujian datanya yaitu *blackbox testing* dan *whitebox testing*:

1. ***Blackbox testing***

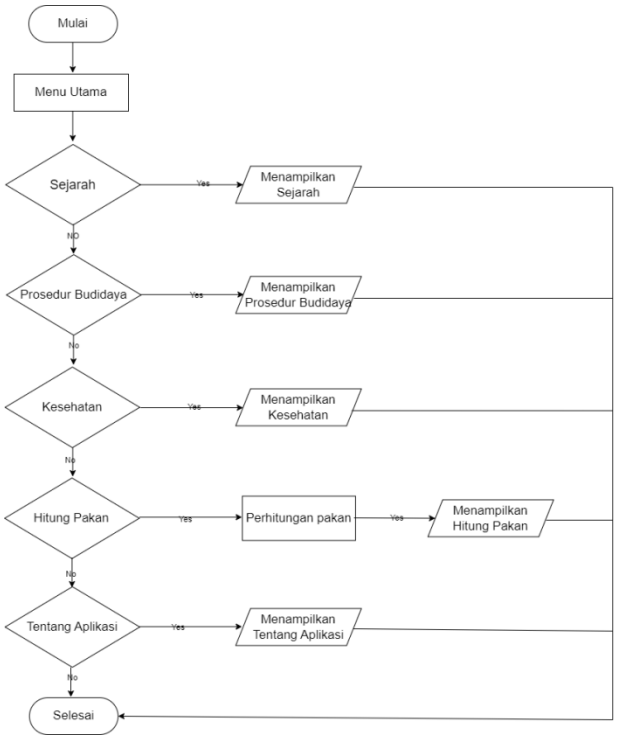
Blackbox testing terfokus pada fungsional dari program yang ada. Pada *Blackbox testing* diuji dengan cara menjalankan program kemudian diamati apakah program tersebut apakah berhasil atau tidak. *Blackbox testing* menggunakan teknik *equivalence partitions* yang merupakan pengujian berdasarkan masukan setiap menu yang terdapat pada program, setiap menu

masukan dilakukan pengujian melalui klasifikasi dan pengelompokan berdasar fungsinya.

2. ***Whitebox testing***

Whitebox testing bertujuan untuk mengetahui apakah struktur pada aplikasi yang dibuat sudah sesuai dengan ketentuan. *Whitebox testing* menitik beratkan pada pengujian dengan mengecek detail peracangan perangkat lunak. *Whitebox testing* dinilai dengan mendefiniskan semua alur dari perangkat lunak, kemudian membangun kasus yang akan digunakan dalam proses pengujian, kemudian menguji kasus tersebut untuk memperoleh hasilnya.

G. Diagram Alir

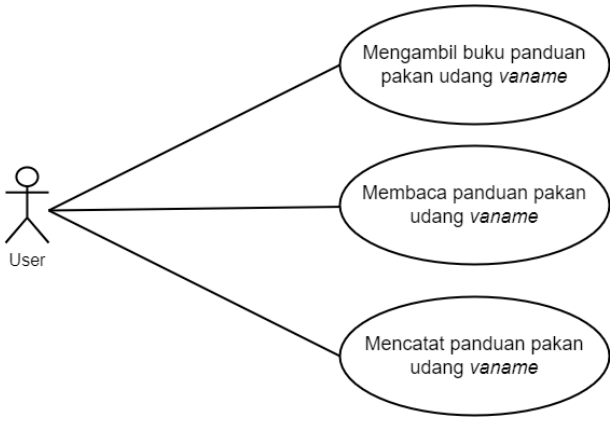


Gambar 3.1 Flowchart

Pada Gambar diatas menjelaskan mengenai alur dari Aplikasi Sistem Panduan Pemberian Pakan Udang *Vaname* Berbasis *Android* dengan alur yang sederhana. Ketika pengguna membuka aplikasi maka pengguna akan langsung ditampilkan dengan halaman utama yang menampilkan 6 opsi pilihan. Di antaranya sejarah, prosedur budidaya, kesehatan, hitung pakan, lihat budidaya dan tentang aplikasi. Dimana opsi hitung pakan terdapat perhitungan pakan sehingga mempermudah petani tambak dalam menentukan takaran pakan yang tepat. Dengan demikian, aplikasi ini memberikan pengalaman pengguna yang mudah dan langsung dalam mengakses dan mengelola informasi terkait seputar panduan pemberian pakan udang *vaname*.

H. Desain Sistem

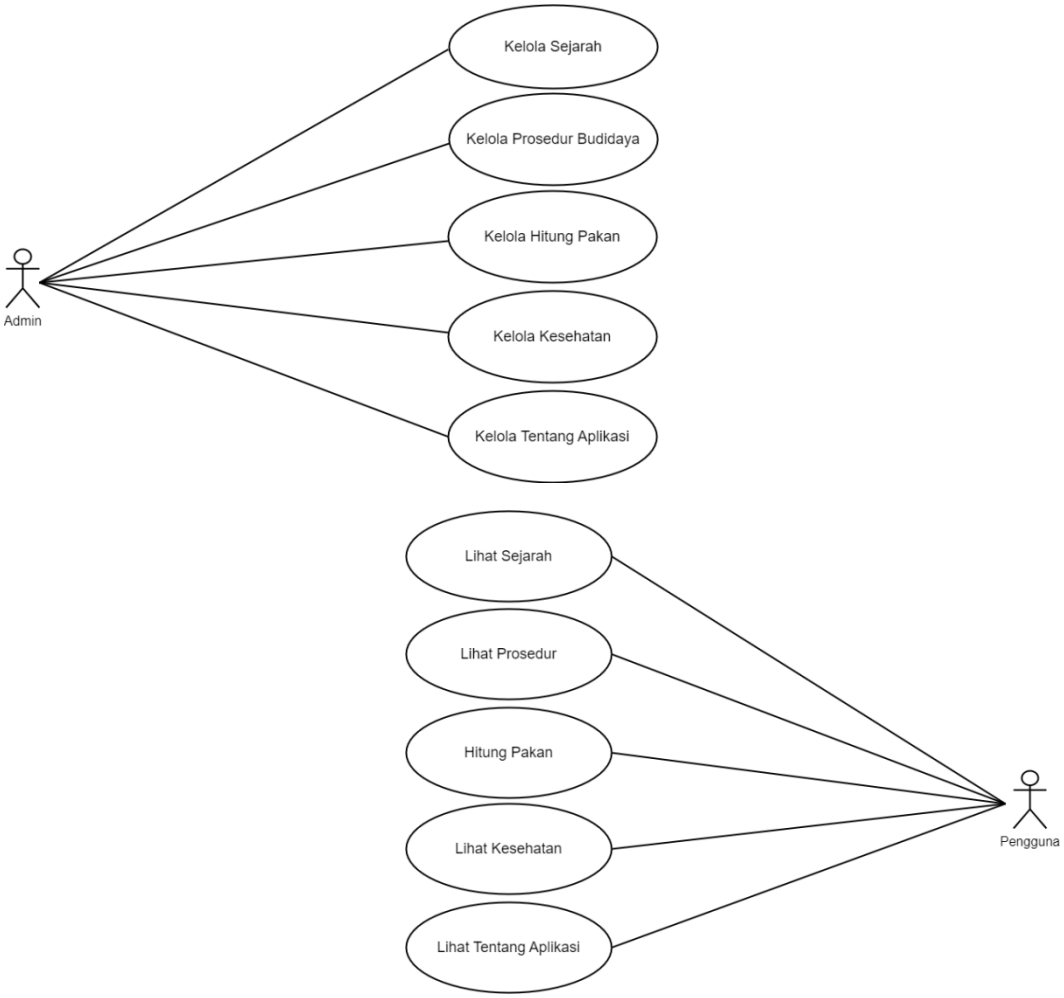
1. Desain sistem yang berjalan



Gambar 3.2 *Usecase* sistem yang berjalan

Pada gambar di atas sistem yang berjalan pada saat ini, petani tambak dapat mendapatkan seputar informasi panduan pakan udang *vaname* melalui buku/internet dan belum ada aplikasi khusus yang berisikan panduan budidaya (panduan pakan) udang *vaname*.

2. Desain sistem yang diusulkan



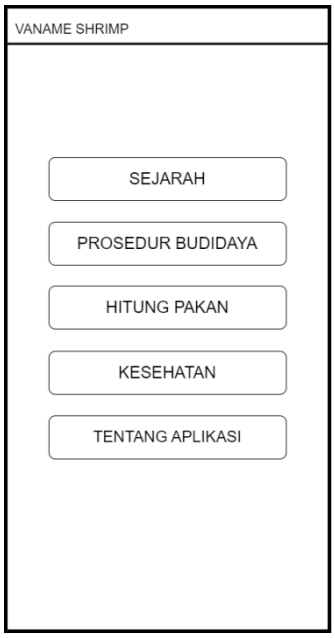
Gambar 3.3 *Use case* sistem yang diusulkan

Pada gambar di atas terdapat dua peran di dalam aplikasi. Pertama yang bertindak sebagai admin, di mana admin memiliki tanggung jawab untuk mengelola aplikasi baik itu berupa data-data materi ataupun informasi terkait *u*ser. Kedua yaitu *user, user* atau pengguna dapat mengakses aplikasi dimana nantinya terdapat beberapa fitur di dalam aplikasi yang tentunya memudahkan *user* untuk mendapatkan seputar informasii panduan budidaya (pakan) udang *vaname* menggunakan smartphone-Nya.

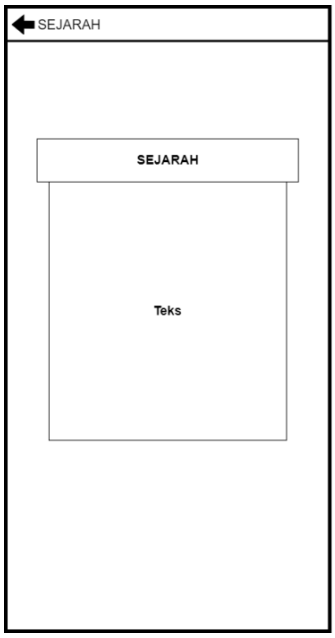
I. Rancangan *user interface* Aplikasi

Rancangan *input* dan *output* perangkat lunak di bawah ini dimaksudkan untuk menggambarkan bagaimana rancangan tampilan *UI (User Interface)* dari aplikasi yang dibuat, agar dalam *coding* pembuatan UI tidak mereka reka dalam proses pembuatannya. Pada gambar dibawah ini terdapat 5 halaman atau *page*, halaman pertama menampilkan halaman utama atau *Home* berisi 5 menu yang terdapat didalam aplikasi seperti menu sejarah berfungsi untuk menampilkan sejarah udang *vaname*, menu prosedur budidaya berfungsi untuk menampilkan informasi mengenai prosedur atau cara budidaya udang *vaname*, menu hitung pakan berfungsi untuk menentukan takaran pakan udang *vaname*, menu kesehatan berfungsi untuk menampilkan jenis penyakit udang *vaname* dan tentang aplikasi berisikan tentang nama aplikasi. Berikut rancangan tampilan atau *page* dari aplikasi:

Menu *Home* Aplikasi



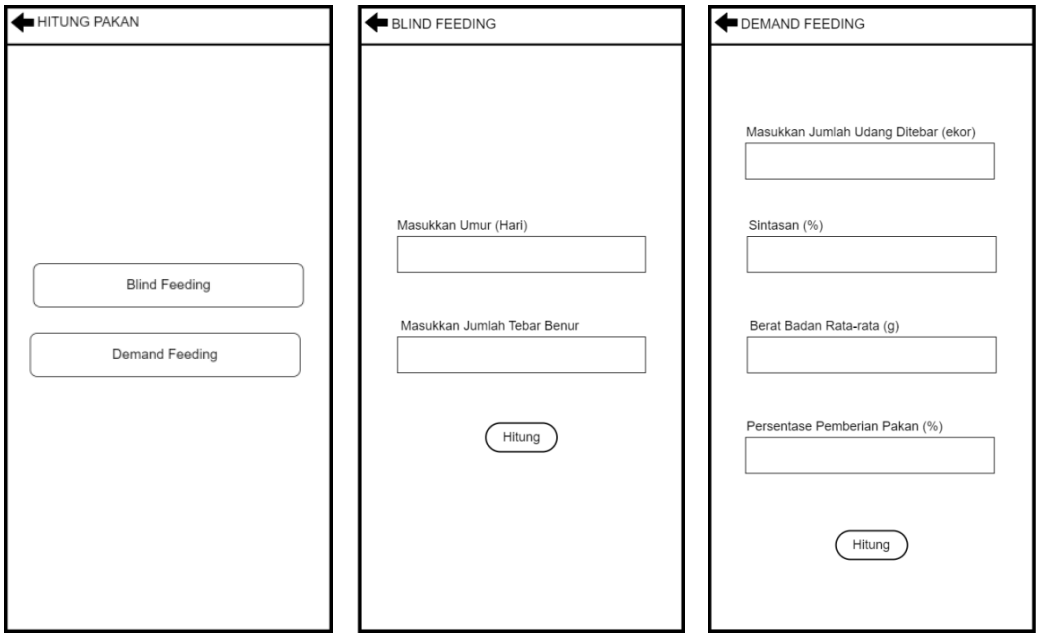
Sejarah



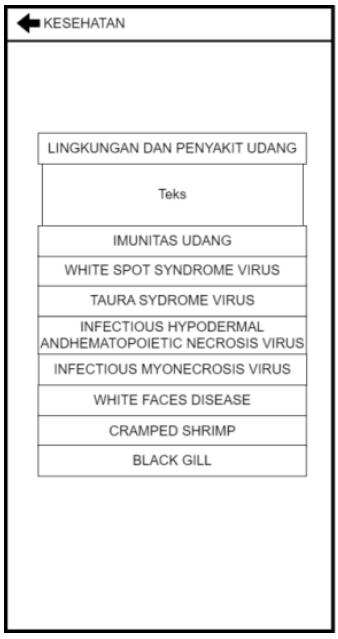
Prosedur Budidaya



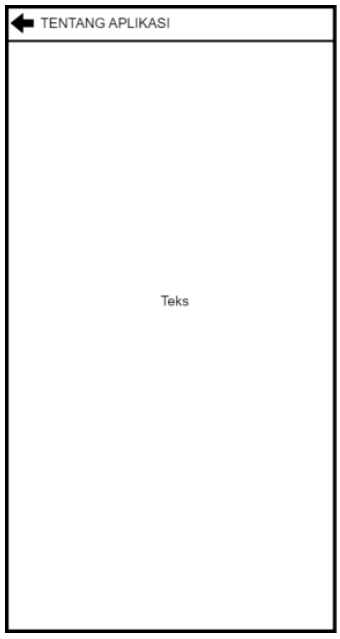
Hitung Pakan



Kesehatan



Tentang Aplikasi



Gambar 3.4 *User interface* aplikasi

BAB IV

HASIL DAN PEMBAHASAN

A. Hasil Penelitian

Berdasarkan hasil penelitian tambak pertama pada hari ke 35 dengan jumlah penebaran 80.000 benur, sintasan 90%, berat badan rerata 1,0 gram, persentase 10% maka diperoleh jumlah pakan sebanyak 7,2 kg dan data selanjutnya dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil penelitian berat badan rata-rata udang vaname

Usia (Hari)	Berat badan rata-rata Tambak 1	Jumlah Pakan	Berat badan rata-rata Tambak 2	Jumlah Pakan
35	1,0	7,2 kg	1,1	4,4 kg
36	1,1	7,9 kg	1,2	4,8 kg
37	1,2	8,6 kg	1,3	5,2 kg
38	1,3	9,3 kg	1,4	5,6 kg
39	1,4	10 kg	1,5	6 kg
40	1,5	10,8 kg	1,6	6,4 kg
41	1,6	11,5 kg	1,7	6,8 kg
42	1,7	12,2 kg	1,8	7,2 kg
43	1,8	12,9 kg	1,9	7,6 kg
44	1,9	13,6 kg	2,0	8 kg

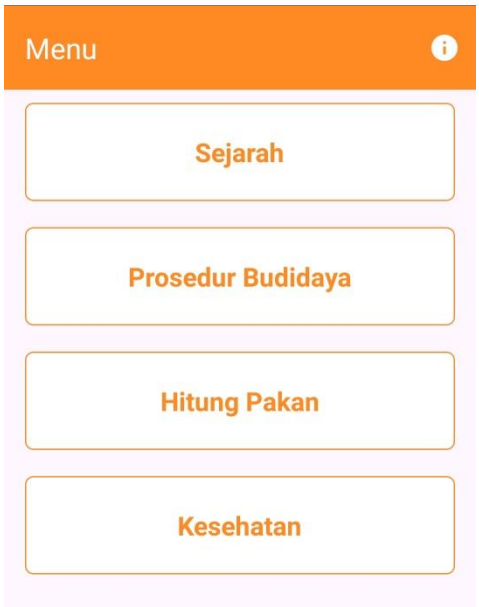
Penelitian ini dilaksanakan selama 10 hari dimana terdapat 2 tambak yang berbeda. Jumlah tebar benur tambak pertama sebanyak 80.000 ekor dan tambak kedua sebanyak 50.000 ekor yang ditebar di masing-masing tambak yang berbeda. Tambak pertama dan kedua memiliki berat badan rata-rata yang naik 0,1 gram setiap harinya dimana tambak pertama dengan berat badan rata-rata 1,65 gram yang memakan 104 kg pakan selama 10 hari, sedangkan tambak 2 dengan berat badan rata-rata 1,55 gram dengan jumlah pakan 62 kg selama 10 hari. Dari hasil penelitian tersebut maka dapat disimpulkan bahwa jumlah pakan udang vaname yang diberikan tergantung dengan jumlah tebar benur udang vaname.

Pada tahap selanjutnya, desain sistem yang telah dirancangkan diterapkan kedalambahasa pemrograman yang selanjutnya akan dilakukan pengujian sistem. Bahasa pemrograman yang digunakan untuk membuat aplikasi ini adalah Java.

1. Implementasi *Interface*

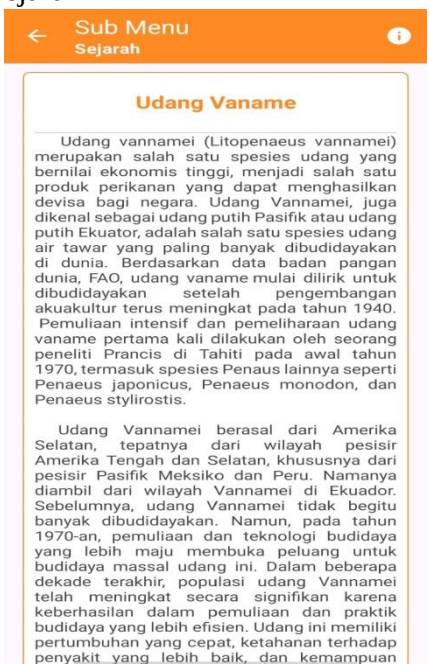
Implementasi adalah tahapan yang bertujuan mengubah hasil dari rancangan sistem menjadi bentuk nyata, dalam hal ini berupa Aplikasi Sistem Panduan Pemberian Pakan Udang Vaname Berbasis *Android* yang dapat digunakan langsung pada *smartphone*. Berikut adalah tampilan dari aplikasi tersebut :

a. Tampilan Menu Utama



Gambar 4. 1 Implementasi *Interface* Tampilan Menu Utama

b. Tampilan Menu Sejarah



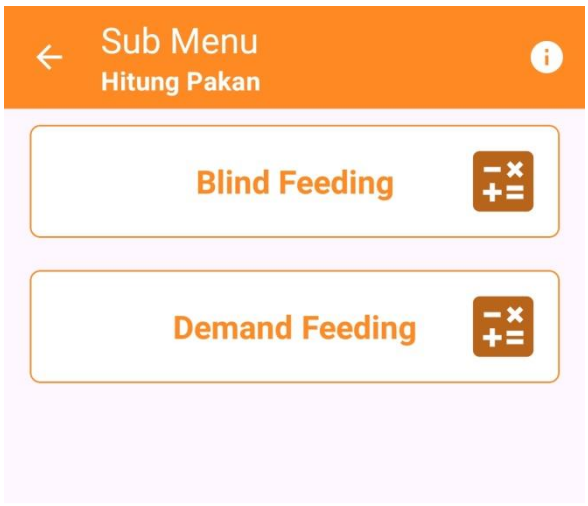
Gambar 4. 2 Implementasi *Interface* Tampilan Menu Sejarah

c. Tampilan Prosedur Budidaya



Gambar 4. 3 Implementasi *Interface* Tampilan Prosedur Budidaya

d. Tampilan Menu Hitung Pakan



Gambar 4. 4 Implementasi *Interface* Tampilan Menu Hitung Pakan

e. Tampilan Menu Kesehatan



Gambar 4. 5 Implementasi *Interface* Tampilan Menu Kesehatan

f. Tampilan Info Aplikasi



Gambar 4. 6 Implementasi *Interface* Tampilan Info Aplikasi

B. Pembahasan

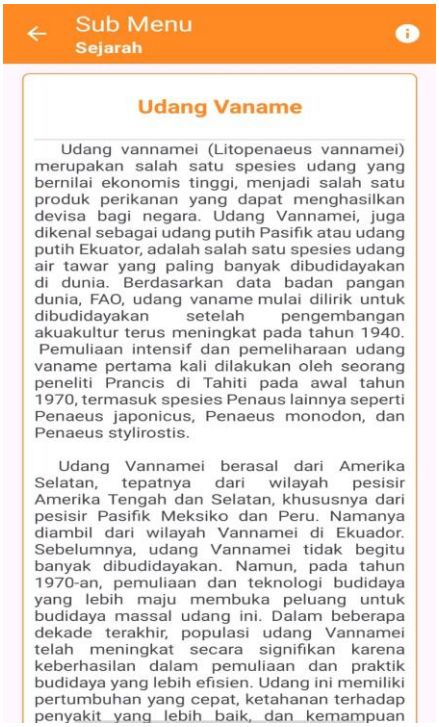
Metode pengujian perangkat lunak pada sistem ini menggunakan beberapa pengujian *BlackBox dan whitebox*, untuk menjelaskan dan membuktikan berhasilnya suatu implikasi hasil peneltian yang telah dibuat.

1. Pengujian *Black Box*

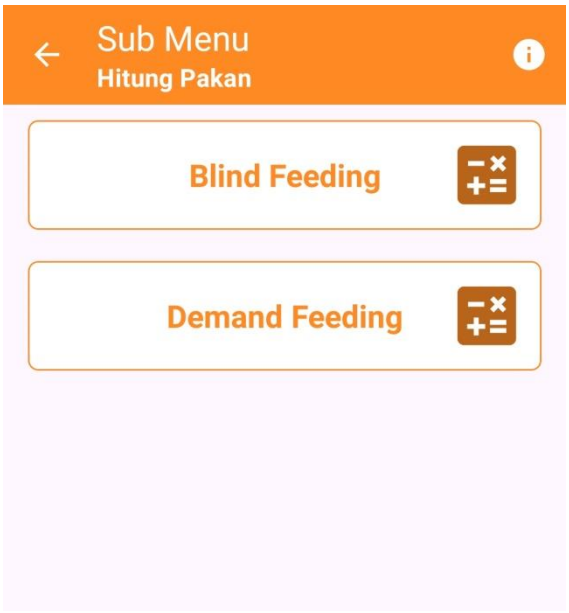
Pengujian *black box* didasarkan pada detail aplikasi, fungsi – fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna atau user pengujian ini tidak melihat dan menguji source code program. Pengujia Blackbox dapat dilihat pada tabel 4.2 di bawah ini:

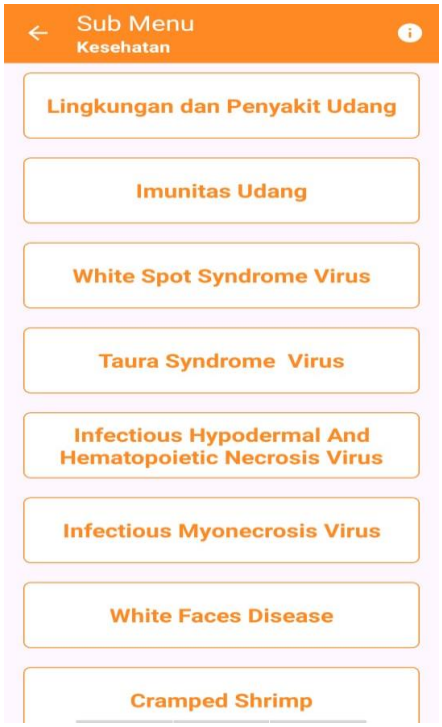
Tabel 4. 2 Pengujian *Blackbox*

NO	Test Faktor	Hasil	Keterangan
1.	Beranda (Menu Utama)	✓	Berhasil, Karena ketika aplikasi dibuka maka akan muncul tampilan menu utama aplikasi vaname shrimp.
Tangkapan Layar			
			

NO	Test Faktor	Hasil	Keterangan
2.	Menu Sejarah	✓	Berhasil, Karena ketika memilih menu sejarah maka akan menampilan teks tentang sejarah udang vaname.
Tangkapan Layar			
			

NO	Test Faktor	Hasil	Keterangan
3.	Menu Prosedur Budidaya	✓	Berhasil, Karena ketika memilih menu prosedur budidaya maka akan menampilan tentang sub menu prosedur budidaya udang vaname.
Tangkapan Layar			
			

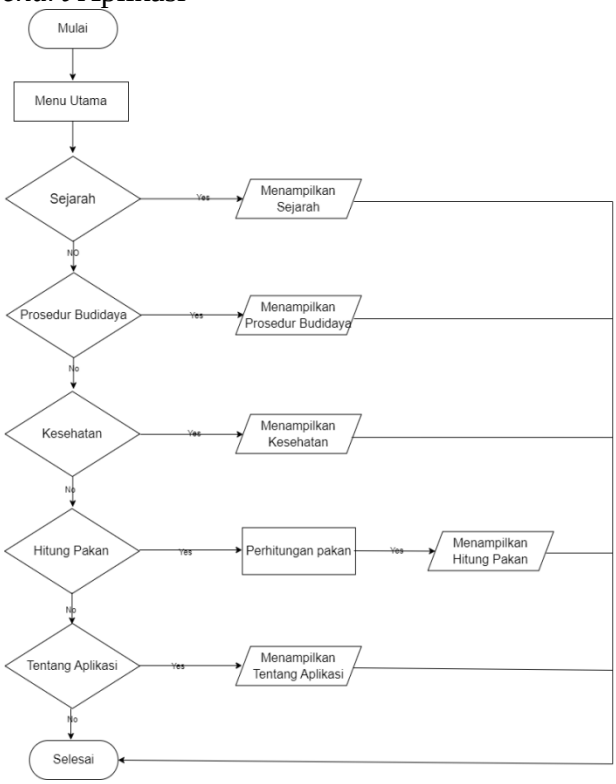
NO	Test Faktor	Hasil	Keterangan
. 4.	Menu Hitung Pakan	✓	Berhasil, Karena ketika memilih menu hitung pakan maka akan menampilan tentang sub menu hitung pakan udang vaname.
Tangkapan Layar			
			

NO	Test Faktor	Hasil	Keterangan
5.	Menu Kesehatan	✓	Berhasil, Karena ketika memilih menu kesehatan maka akan menampilan tentang sub menu kesehatan udang vaname.
Tangkapan Layar			
			

No	Test Faktor	Hasil	Keterangan
6.	Tentang Aplikasi	✓	Berhasil, Karena ketika memilih ikon tentang aplikasi maka akan menampilan tentang aplikasi vaname shrimp.
Tangkapan Layar			
			

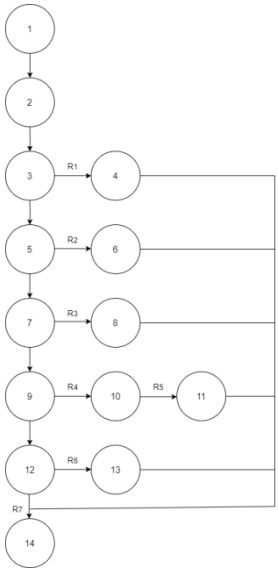
1. Pengujian *White Box*

   a. *Flowchart* Aplikasi



Gambar 4. 7 *Flowchart* Aplikasi

   b. *Flowgraph* Aplikasi



Gambar 4. 8 *Flowgraph* Aplikasi

Proses perhitungan aplikasi

Dari gambar flowgraph diatas dapat dilakukan perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexcity* V(G) dari *Edge* dan *Node* dengan rumus :

N *(node*) = 14

E (*Edge)* = 18

P (*Predikat node*) = 5

Penyelesaian : V(G) = E - N + 2

= 18 – 14 + 2

= 6

Predikat = P + 1

= 5 + 1

= 6

2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatasmemiliki *Region* = 6

3) *Independent Path* pada *Flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 5 – 7 – 9 – 12 – 14

Path 2 = 1 – 2 – 3 – 4 – 14

Path 3 = 1 – 2 – 3 – 5 – 6 – 14

Path 4 = 1 – 2 – 3 – 5 – 7 – 8 – 14

Path 5 = 1 – 2 – 3 – 5 – 7 – 9 – 10 – 11 - 14

Path 6 = 1 – 2 – 3 – 5 – 7 – 9 – 12 – 13 – 14

4) Grafik *Matrix* Aplikasi

Tabel 4. 3 Grafik *Matrix* Aplikasi

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	E-1
1		1													1 - 1 = 0
2			1												1 – 1 = 0
3				1	1										2 – 1 = 1
4														1	1 – 1 = 0
5						1	1								2 – 1 = 1
6														1	1 – 1 = 0
7								1	1						2 – 1 = 1
8														1	1 – 1 = 0
9										1		1			2 – 1 = 1
10											1				1 – 1= 0
11														1	1 – 1 = 0
12													1	1	2 – 1 = 1
13														1	1 – 1 = 0
14															
Zum (E+1)															5 + 1 = 6

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan oleh penulis dalam pembuatan Apikasi Sistem Panduan Pemberian Pakan Udang *Vaname* Berbasis *Android* yaitu :

1) Aplikasi ini dibuat mengunakan Android Studio, XAMPP, dan Visual Studio Code.
2) Tambak pertama dan kedua memiliki berat badan rata-rata yang naik 0,1 gram setiap harinya dimana tambak pertama dengan berat badan rata-rata 1,65 gram yang memakan 104 kg pakan selama 10 hari, sedangkan tambak 2 dengan berat badan rata-rata 1,55 gram dengan jumlah pakan 62 kg selama 10 hari.

B. Saran

Setelah melakukan penelitian maka penulis memberikan saran yaitu:

1) Penulis menyarankan agar pada pengembangan selanjutnya aplikasi dapat diakses tanpa menggunakan jaringan (*offline*).
2) Penulis berharap aplikasi panduan pemberian pakan udang vaname dapat dikembangkan di kemudian hari dari sisi desain tampilan dan penambahan beberapa fitur sehingga aplikasi ini menjadi lebih lengkap.

DAFTAR PUSTAKA

Erni Sri Wahyuni. (2019). *Analisis Cara Kerja Crud Dengan Menggunakan Android Studio*.

Irsan, M. (2015). *Rancang Bangun Aplikasi Mobile Notifikasi Berbasis Android Untuk Mendukung Kinerja Di Instansi Pemerintahan*. *Jurnal Penelitian Teknik_Informatika*,*1*(1),115–120. Http://Jurnal.Untan.Ac.Id/Index.Php/Justin/Article/View/9984/9752

Lailiyah, U. S., Rahardojo, S., Kristiany, M. G. E., & Mulyono, M. (2018). *Produktivitas Budidaya Udang Vaname (Litopenaeus Vannamei) Tambak Super Intensif Di Pt. Budidaya Perikanan Dewi Laut Kabupaten Garut.* Jurnal Kelautan Dan Perikanan Terapan, 1, 1–11.

Musa, M., Lusiana, E. D., Buwono, N. R., Arsad, S., & Mahmudi, M. (2020). *Efektivitas Sistem Silvofishery Dalam Pengolahan Air Di Kolam Intensif Udang Whiteleg (Litopenaeus Vannamei), Kabupaten Probolinggo, Jawa Timur, Indonesia.* Keanekaragaman Hayati, 21(10), 4695–4701. Https://Doi.Org/10.13057/Biodiv/D211031

Mansyur, A., Mangampa, M., Suwoyo, H,S., Pantjara, B., Syah, R. (2104). *Strategi Pengelolaan Pakan Pada Budidaya Udang Vaname Litopenaeus Vannamei.* Issbn: 978-979-181-806-3

Ramadhan, R. F., & Mukhaiyar, R. (2020). *Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. 1*(2), 129–134.

Supono, (2017). *Teknologi Produksi Udang Edisi Pertama.* Yogyakarta Ruko Jambusari 7 A Yogyakarta.

Therapy, C., Gordon, V., Meditation, C., Vanrullen, R., Myers, N. E., (2018). *Karakteristik Kualitas Air Dan Performa Pertumbuhan Budidaya Udang Vaname (Litopenaeus Vannamei) Pola Intensif. Jurnal Ilmiah Perikanan DanKelautan*,*21*(1),75–88. https://Doi.Org/10.1016/J.Neuron.2018.08.006

Wardhani, K. S. (2014). *Pengembangan Sistem Informasi Class Diagram Secara Digital. Lumbung Pustaka Universitas Negeri Yogyakarta*, 161. Https://Eprints.Uny.Ac.Id/20529/1/Kusumaningati Sulistya Wardhani-10520241007.Pdf

Yunus, R. M., Sujadi, H., & -, K. (2015). *Sistem Keamanan Pesan Dengan Algoritma Rivest Code 6 (Rc-6) Menggunakan Java Pada Smartphone Berbasis Android*. *J-Ensitec*, *2*(01), 1–10. Https://Doi.Org/10.31949/J-Ensitec.V2i01.45