



**RANCANG BANGUN SISTEM OPERASIONAL
BUDIDAYA TAMBAK IKAN KERAPU BERBASIS
ANDROID (STUDI KASUS: KELOMPOK TANI
BHAKTIUSAHA 2)**

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar sarjana komputer

Disusun oleh:

Nama: Ahmad Shofiudin Firdani Wafa

NIM: 145150207111084



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2019

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

Repository Universitas Brawijaya
Repository Universitas Brawijaya
Repository Universitas Brawijaya

PENGESAHAN

RANCANG BANGUN SISTEM OPERASIONAL BUDIDAYA TAMBAK IKAN KERAPU
BERBASIS ANDROID (STUDI KASUS: KELOMPOK TANI BHAKTI USAHA 2)

SKRIPSI

untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ahmad Shofiudin Firdani Wafa
NIM: 145150207111084

Skripsi ini telah diuji dan dinyatakan lulus pada
6 Maret 2019

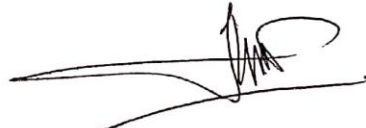
Telah diperiksa dan disetujui oleh:

Pembimbing I



Nurudin Santoso, S.T, M.T
NIP: 19740916 200012 1 001

Pembimbing II



Lutfi Fanani, S.Kom., M.T., M.Sc.
NIK. 2016078902171001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001





PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 15 Februari 2019



Ahmad Shofiudin Firdani Wafa

NIM: 145150207111084



KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya, sehingga laporan skripsi yang berjudul “Rancang Bangun Sistem Operasional Budidaya Tambak Ikan Kerapu Berbasis Android (Studi Kasus: Kelompok Tani Bhakti Usaha 2)” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Nurudin Santoso, S.T., M.T. dan Bapak Lutfi Fanani, S.Kom., M.T., M.Sc. selaku Pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis, sehingga dapat menyelesaikan skripsi ini,
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika,
3. Bapak Ir. Wafiq Ismail dan Ibu Dra. Sufarah selaku orang tua penulis, Nurvita Nuzul Fitriana Wafa S.si selaku kakak penulis dan Baharudin Ashwin Febrian Wafa selaku adik penulis, dan seluruh keluarga besar atas segala nasihat, kasing sayang, perhatian dan kesabarannya di dalam membersarkan dan mendidik penulis, serta senantiasa mendoakan dan menyemangati demi terselesainya skripsi ini,
4. Ailya Primi Apriliyana ni’sama yang dengan sabar dan senantiasa mendesak penulis sehingga dapat menyelesaikan skripsi ini.
5. Anton Firdaus yang membantu dan mengajari penulis dengan sabar, Gogot Alam Anpurnan dan Galang Perdana Putra selaku sahabat seperjuangan dalam mengerjakan skripsi.
6. Seluruh teman-teman kopma squad yang selama ini menemani dan mewarnai keseharian penulis.
7. Dan juga seluruh teman-teman informatika kelas G yang telah menemani dari awal semester.

Penulis Menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 15 Februari 2019

Penulis

Email: shofiudinwafa@gmail.com



ABSTRAK

Ahmad Shofiudin Firdani Wafa, Rancang Bangun Sistem Operasional Budidaya Tambak Ikan Kerapu Berbasis Android (Studi Kasus: Kelompok Tani Bhakti Usaha 2)

Pembimbing: Nurudin Santoso, S.T., M.T. dan Lutfi Fanani, S.Kom., M.T., M.Sc.

Budidaya perikanan adalah pemeliharaan dan pemeliharaan ikan, ataupun organisme air lain, salah satunya yaitu budidaya tambak ikan kerapu. Produksi ikan kerapu di Indonesia sendiri cukup tinggi dan juga terus mengalami peningkatan hampir tiap tahunnya. Namun terdapat permasalahan yang penulis dapatkan dari kelompok tani Bhakti Usaha 2, bahwa petani sering terjadinya kesalahan pemberian pakan yang menimbulkan tidak meratanya pertumbuhan ikan kerapu, serta kesalahan dalam perhitungan masa panen sehingga petani kurang maksimal dalam segi hasil yang didapat, hal ini dikarenakan semua proses budidaya masih dilakukan dengan manual sehingga kesalahan dari manusia (*aman error*) masih sering terjadi. Solusi yang diberikan oleh penulis adalah pembuatan sistem operasional budidaya tambak ikan kerapu berbasis android dengan fitur untuk memberikan penjadwalan pada setiap aktivitas yang akan dilakukan oleh petani tambak ikan kerapu, dari masa persiapan budidaya hingga proses panen, pencatatan detail kolam dan juga pencatatan hasil keuangan. Pada pengembangan sistem ini menggunakan metode *iterative model*. Hasil dari pengujian unit sistem didapati nilai 100% valid demikian juga dengan pengujian validasinya yang juga mendapatkan nilai 100% valid. Untuk pengujian *usability* dihasilkan nilai sangat memuaskan pada tiga aspek pengujian dan satu aspek bernilai memuaskan.

Kata Kunci: budidaya, ikan kerapu, rekayasa perangkat lunak, operasional, *mobile*, *iterative model*



ABSTRACT

Ahmad Shofiudin Firdani Wafa, Rancang Bangun Sistem Operasional Budidaya Tambak Ikan Kerapu Berbasis Android (Studi Kasus: Kelompok Tani Bhakti Usaha 2)

Pembimbing: Nurudin Santoso, S.T., M.T. dan Lutfi Fanani, S.Kom., M.T., M.Sc.

Aquaculture is the maintenance and preservation of fish or other aquatic organisms, one of them namely riparian fish grouper. Production of grouper in Indonesia itself is quite high and also continue to experience increased almost every year. But there are problems that writers get from farmers Group of Bhakti Usaha 2, that farmers often occurrence of error permemberian feed meratanya not cause growth of grouper, as well as errors in the calculation of the period of harvest so that farmers are insufficient in terms of the results obtained, this is because all the process of cultivation is still done by manual so that the fault of the humans (human error) is still often the case. The solution given by the author is making the system operational riparian android-based grouper with features to provide scheduling at each activity will be carried out by the pond grouper farmers, from the time of preparation budidaya to the process of harvesting, logging detail and also the recording of financial results. On the development of this system using the method of iterative model. Results from testing the system unit was found to be 100% valid values as well as been testing which also get value 100% valid. The resulting values of usability testing for a very satisfying on the three aspects of the test and an aspect worth satisfying.

Keywords: *aquaculture, Grouper, software engineering, operations, mobile, iterative model*



DAFTAR ISI

PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN.....	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	2
1.4.1 Manfaat Bagi Penulis.....	3
1.4.2 Manfaat Bagi Pengguna.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Kajian Pustaka.....	5
2.2 Ikan Kerapu.....	5
2.3 Kelompok Tani Tambak.....	6
2.3.1 Kelompok Tani Tambak Bhakti Usaha 2.....	6
2.4 Proses Budidaya.....	6
2.4.1 Persiapan Budidaya.....	7
2.4.2 Penebaran Bibit.....	8
2.4.3 Pembesaran Ikan.....	8
2.4.4 Panen.....	8
2.4.5 Keuangan.....	9
2.5 <i>Software Development Life Cycle (SDLC)</i>	9



BAB 5 PERANCANGAN DAN IMPLEMENTASI	42
5.1 Perancangan	42
5.1.1 Sequence Diagram	42
5.1.2 Perancangan Basis Data	44
5.1.3 Class Diagram Iterasi Pertama	45
5.1.4 Class Diagram Iterasi Kedua	46
5.1.5 Perancangan Algoritme	47
5.1.6 Perancangan Antarmuka Iterasi Pertama	49
5.1.7 Perancangan Antarmuka Iterasi Kedua	56
5.2 Implementasi	60
5.2.1 Spesifikasi Sistem	60
5.2.2 Implementasi Basis Data	62
5.2.3 Implementasi Algoritme	63
5.2.4 Implementasi Antarmuka Iterasi Pertama	67
5.2.5 Implementasi Antarmuka Iterasi Kedua	72
BAB 6 Pengujian	76
6.1 Pengujian Unit	76
6.1.1 Pengujian Unit Menambah Kegiatan Budidaya lain	76
6.1.2 Pengujian Unit Melihat Detail Keuangan	77
6.1.3 Pengujian Unit Menambah Pemasukan Uang	80
6.1.4 Analisis Pengujian Unit	82
6.2 Pengujian Validasi Iterasi Pertama	82
6.3 Pengujian Validasi Iterasi Kedua	97
6.4 Pengujian Non-Fungsional	99
6.4.1 Hasil Pengujian <i>Usability</i> Iterasi Pertama	100
6.4.2 Analisis Hasil Pengujian <i>Usability</i> Iterasi Pertama	101
6.4.3 Hasil Pengujian <i>Usability</i> Iterasi Kedua	102
6.4.4 Analisis Hasil Pengujian <i>Usability</i>	103
BAB 7 PENUTUP	105
7.1 Kesimpulan	105
7.2 Saran	106
DAFTAR PUSTAKA	107



DAFTAR TABEL

Tabel 2.1 Rincian persiapan budidaya	7
Tabel 2.2 Data Umur ikan kerapu	8
Tabel 2.3 Proses Pemeliharaan	8
Tabel 2.4 Jenis jual	9
Tabel 4.1 dentifikasi pengguna	20
Tabel 4.2 Kebutuhan fungsional iterasi pertama	21
Tabel 4.3 Kebutuhan fungsional iterasi kedua	23
Tabel 4.4 Kebutuhan non-fungsional	24
Tabel 4.5 <i>Use case</i> skenario menambah kolom	27
Tabel 4.6 <i>Use case</i> skenario melihat daftar kolom	27
Tabel 4.7 <i>Use case</i> scenario menghapus kolom	28
Tabel 4.8 <i>Use case</i> skenario melihat detail kolom	28
Tabel 4.9 <i>Use case</i> skenario mengubah detail kolom	28
Tabel 4.10 <i>Use case</i> skenario menambah persiapan budidaya	29
Tabel 4.11 <i>Use case</i> skenario melihat daftar persiapan budidaya	30
Tabel 4.12 <i>Use case</i> skenario menandai kegiatan persiapan budidaya	30
Tabel 4.13 <i>Use case</i> skenarni menghapus kegiatan persiapan budidaya	31
Tabel 4.14 <i>Use case</i> skenario mencatat penebaran bibit	31
Tabel 4.15 <i>Use case</i> skenario melihat daftar pembibitan	32
Tabel 4.16 <i>Use case</i> skenario melihat detail pembibitan	32
Tabel 4.17 <i>Use case</i> skenario mengubah detail pembibitan	33
Tabel 4.18 <i>Use case</i> skenario menambahkan kegiatan harian	33
Tabel 4.19 <i>Use case</i> skenario melihat kalender kegiatan budidaya	34
Tabel 4.20 <i>Use case</i> skenario menambah kegiatan	34
Tabel 4.21 <i>Use case</i> skenario menambah kegiatan panen	35
Tabel 4.22 <i>Use case</i> skenario melihat daftar panen	35
Tabel 4.23 <i>Use case</i> skenario melihat detail kegiatan panen	36
Tabel 4.24 <i>Use case</i> skenario menghapus detail panen	36
Tabel 4.25 <i>Use case</i> skenario menambah biaya pengeluaran	36
Tabel 4.26 <i>Use case</i> skenario menambah pemasukan uang	37



Tabel 6.15 Kasus uji mengubah detail kolam.....	85
Tabel 6.16 Kasus uji mengubah detail kolam.....	85
Tabel 6.17 Kasus uji menambah persiapan budidaya.....	86
Tabel 6.18 Kasus uji menambah persiapan budidaya.....	86
Tabel 6.19 Kasus uji melihat daftar persiapan budidaya.....	86
Tabel 6.20 Kasus uji menandai kegiatan persiapan budidaya.....	87
Tabel 6.21 Kasus uji menghapus kegiatan persiapan budidaya.....	87
Tabel 6.22 Kasus uji mencatat penebaran bibit.....	87
Tabel 6.23 Kasus uji mencatat penebaran bibit.....	88
Tabel 6.24 Kasus uji mencatat penebaran bibit.....	88
Tabel 6.25 Kasus uji melihat daftar pembibitan.....	88
Tabel 6.26 Kasus uji melihat detail pembibitan.....	89
Tabel 6.27 Kasus uji menghapus kegiatan harian.....	89
Tabel 6.28 Kasus uji menambahkan kegiatan harian.....	89
Tabel 6.29 Kasus uji melihat kalender kegiatan budidaya.....	89
Tabel 6.30 Kasus uji menambah kegiatan budidaya lain.....	90
Tabel 6.31 Kasus uji menambah kegiatan budidaya lain.....	90
Tabel 6.32 Kasus uji menambah kegiatan panen.....	90
Tabel 6.33 Kasus uji menambah kegiatan panen.....	91
Tabel 6.34 Kasus uji menambah kegiatan panen.....	91
Tabel 6.35 Kasus uji melihat daftar panen.....	91
Tabel 6.36 Kasus uji melihat detail kegiatan panen.....	91
Tabel 6.37 Kasus uji menghapus detail panen.....	92
Tabel 6.38 Kasus uji menambah biaya pengeluaran.....	92
Tabel 6.39 Kasus uji menambah biaya pengeluaran.....	92
Tabel 6.40 Kasus uji menambah biaya pengeluaran.....	93
Tabel 6.41 Kasus uji menambah pemasukan uang.....	93
Tabel 6.42 Kasus uji menambah pemasukan uang.....	93
Tabel 6.43 Kasus uji menambah pemasukan uang.....	94
Tabel 6.44 Kasus uji menambah pemasukan uang.....	94
Tabel 6.45 Kasus uji melihat daftar keuangan.....	94
Tabel 6.46 Kasus uji melihat detail keuangan.....	95



DAFTAR GAMBAR

Gambar 2.1 WBS Budidaya Tambak Ikan Kerapu	7
Gambar 2.2 Model <i>iterative</i> (Shalahuddin & Rosa, 2015)	10
Gambar 2.3 Contoh <i>use case</i> (Sommerville, 2011)	12
Gambar 2.4 Contoh <i>sequence diagram</i> (Sommerville, 2011)	12
Gambar 2.5 Contoh <i>class diagram</i> (Sommerville, 2011)	13
Gambar 2.6 Notasi <i>flow graph</i>	14
Gambar 2.7 Contoh susunan USE <i>questionnaire</i> (Lund, 2001)	15
Gambar 3.1 Diagram Alir Metode Penelitian	16
Gambar 4.1 <i>Use case diagram</i> iterasi pertama	25
Gambar 4.2 <i>Use case diagram</i> iterasi kedua	26
Gambar 5.1 <i>Sequence diagram</i> menambah kegiatan budidaya lain	42
Gambar 5.2 <i>Sequence diagram</i> melihat detail keuangan	43
Gambar 5.3 <i>Sequence diagram</i> kegiatan pemasukan uang	44
Gambar 5.4 <i>Physical data model</i>	45
Gambar 5.5 <i>Class Diagram</i> iterasi pertama	46
Gambar 5.6 <i>Class diagram</i> iterasi kedua	47
Gambar 5.7 Perancangan antarmuka menu utama sistem	50
Gambar 5.8 Perancangan antarmuka daftar kolam	50
Gambar 5.9 Perancangan antarmuka <i>form</i> tambah	51
Gambar 5.10 Perancangan antarmuka kalender kegiatan budidaya	52
Gambar 5.11 Perancangan antarmuka data pembibitan	53
Gambar 5.12 Perancangan antarmuka persiapan budidaya iterasi pertama	53
Gambar 5.13 Perancangan antarmuka keuangan	54
Gambar 5.14 Perancangan antarmuka detail keuangan	55
Gambar 5.15 Perancangan antarmuka pemasukan keuangan	55
Gambar 5.16 Perancangan antarmuka menu utama iterasi kedua	56
Gambar 5.17 Perancangan antarmuka daftar kolam iterasi kedua	57
Gambar 5.18 Perancangan antarmuka halaman <i>form</i> tambah iterasi kedua	58
Gambar 5.19 Perancangan antarmuka data pembibitan iterasi kedua	58
Gambar 5.20 Perancangan antarmuka persiapan budidaya iterasi kedua	59



Gambar 5.21 Perancangan antarmuka jadwal persiapan budidaya.....	60
Gambar 5.22 Implementasi antarmuka halaman utama.....	68
Gambar 5.23 Implementasi antarmuka halaman daftar kolom.....	68
Gambar 5.24 Implementasi antarmuka <i>form</i> tambah.....	69
Gambar 5.25 Implementasi antarmuka kalender kegiatan budidaya.....	69
Gambar 5.26 Implementasi antarmuka data pembibitan.....	70
Gambar 5.27 Implementasi antarmuka persiapan budidaya iterasi pertama.....	70
Gambar 5.28 Implementasi antarmuka keuangan.....	71
Gambar 5.29 Implementasi antarmuka detail keuangan.....	71
Gambar 5.30 Implementasi antarmuka tambah pemasukan.....	72
Gambar 5.31 Implementasi antarmuka menu utama iterasi kedua.....	72
Gambar 5.32 Implementasi antarmuka daftar kolom iterasi kedua.....	73
Gambar 5.33 Implementasi antarmuka <i>form</i> tambah iterasi kedua.....	73
Gambar 5.34 Implementasi antarmuka data pembibitan iterasi kedua.....	74
Gambar 5.35 Implementasi antarmuka persiapan budidaya iterasi kedua.....	74
Gambar 5.36 Implementasi antarmuka jadwal persiapan budidaya.....	75
Gambar 6.1 <i>Flowgraph</i> menambah kegiatan budidaya lain.....	76
Gambar 6.2 <i>Flowgraph</i> melihat detail keuangan.....	78
Gambar 6.3 <i>Flowgraph</i> menambah pemasukan keuangan.....	81



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	109
LAMPIRAN B SAMPEL HASIL PENYEBARAN KUISIONER.....	111



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Konsumsi ikan di Indonesia saat ini mengalami peningkatan tiap tahunnya, menurut data Kementerian Kelautan dan Perikanan (KKP) tahun 2018 konsumsi ikan dari tahun 2014 hingga 2017 mengalami peningkatan sebanyak 8,35 Kg/Kapita. Dengan semakin tingginya jumlah konsumsi ikan yang ada, maka harus diimbangi pula dengan produksi ikan yang besar, dari data KKP tahun 2018 juga di dapati bahwa hasil perikanan sebagian besar didapat dari hasil budidaya yang berjumlah 17,22 ton dari 23,26 ton total hasil produksi perikanan nasional pada triwulan 4 tahun 2017.

Budidaya perikanan adalah pengembang biakan dan pemeliharaan ikan, ataupun organisme air yang lain. Di Indonesia sendiri jumlah pebudidaya ikan terus mengalami peningkatan. Menurut data dari Badan Pusat Statistik (BPS) tahun 2016, jumlah petani ikan pada tahun 2016 mencapai angka 274 petani, angka ini mengalami peningkatan jika dibandingkan dengan data pada tahun 2007 yang diperoleh total 234 petani ikan. Dari data BPS salah satu budidaya yang ada di Indonesia adalah budidaya tambak ikan kerapu.

Produksi ikan kerapu di Indonesia pada tahun 2016 mencapai 15.645,24 Ton, salah satu Provinsi terbesar dalam produksi ikan kerapu adalah Jawa Timur yang menyumbang 5,768% dari total produksi, hal ini menempatkan Jawa Timur pada posisi lima besar produksi ikan kerapu terbanyak menurut data Kementerian Kelautan dan Perikanan (KKP) tahun 2016. Dari banyaknya hasil produksi ikan kerapu, salah satu diantaranya merupakan hasil dari budidaya kelompok tani Bhakti Usaha II yang ada di desa Labuhan, Kecamatan Brondong kabupaten Lamongan.

Dari hasil wawancara dengan ketua kelompok Bhakti Usaha II, didapat permasalahan yang sering didapatkan saat proses pemeliharaan adalah terjadinya kesalahan pemberian pakan yang dapat menimbulkan tidak meratanya proses pertumbuhan ikan kerapu. Selain itu juga perhitungan masa panen yang salah akan merugikan petani dikarenakan ikan kerapu yang berukuran diatas satu kilogram, akan dihitung sama dengan ukuran satu kilo gram atau dihitung per-ekor, hal ini membuat pendapatan petani ikan kerapu tidak maksimal. Sehingga petani memerlukan sebuah sistem aplikasi yang dapat mengingatkan kegiatan yang akan dilakukan, terutama dalam pengontrolan dalam memberikan pakan dan waktu ikan kerapu siap untuk dipanen. Harapannya dengan adanya sistem operasional budidaya ikan kerapu, dapat memberikan informasi yang dibutuhkan oleh petani saat proses operasional budidaya ikan kerapu, terutama dalam hal mengingat pemberian pakan dan waktu ikan kerapus siap dipanen. Sehingga dapat memberikan hasil yang maksimal bagi petani tambak ikan kerapu.

Sistem yang akan dibuat sendiri merupakan sistem aplikasi berbasis *mobile* android. Aplikasi berbasis *mobile* dirasa lebih cocok untuk digunakan dalam pembangunan sistem informasi operasional budidaya tambak ikan kerapu,



dikarenakan sudah banyaknya pengguna perangkat *mobile* di Indonesia, menurut data dari Hootsuite yang dikutip dalam bisnis.com populasi dari penggunaan perangkat *mobile* di Indonesia mencapai 177,9 juta pengguna (Sidik, 2018). *Platform* ini juga memiliki mobilitas yang baik, sehingga petani tambak ikan kerapu dapat selalu menggunakannya kapanpun dan di manapun.

Dalam penelitian ini penulis melakukan pengumpulan data serta implementasi kepada kelompok tani tambak ikan kerapu Bhakti Usaha II Kelompok tani ikan ini merupakan sebuah perkumpulan dari petani yang ada di desa Labuhan kecamatan Brondong kabupaten Lamongan, yang saling bekerjasama dalam pembudidayaan ikan kerapu.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah diuraikan pada latar belakang, maka didapatkan rumusan masalah sebagai berikut:

1. Bagaimana analisis kebutuhan sistem informasi operasional budidaya tambak ikan kerapu?
2. Bagaimana hasil perancangan sistem informasi operasional budidaya tambak ikan kerapu?
3. Bagaimana implementasi sistem informasi operasional budidaya tambak ikan kerapu?
4. Bagaimana hasil pengujian sistem informasi operasional budidaya tambak ikan kerapu?

1.3 Tujuan

Berdasarkan dari rumusan masalah yang telah dipaparkan, maka tujuan yang dimaksudkan dalam penelitian ini adalah:

1. Mengidentifikasi kebutuhan fungsional dan kebutuhan non fungsional dari sistem informasi operasional budidaya tambak ikan kerapu.
2. Merancang sistem informasi operasional budidaya tambak ikan kerapu menggunakan pemodelan berorientasi objek.
3. Mengimplementasikan sistem informasi operasional budidaya tambak ikan kerapu menggunakan sistem berbasis *mobile*.
4. Menguji sistem informasi operasional budidaya tambak ikan kerapu sesuai dengan kebutuhan fungsional dan non fungsional.

1.4 Manfaat

Manfaat yang dapat diambil dari penelitian ini dapat dinilai dari beberapa sisi, yaitu manfaat bagi penulis dan bagi pengguna sistem operasional budidaya ikan kerapu ini.



1.4.1 Manfaat Bagi Penulis

Penulis dapat menerapkan ilmu yang telah didapatkan dari Laboratorium Rekayasa Perangkat Lunak Fakultas Ilmu Komputer Universitas Brawijaya dalam proses pembangunan Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK).

1.4.2 Manfaat Bagi Pengguna

Sistem ini dapat memudahkan petani ikan kerapu untuk mengurangi kesalahan yang dilakukan dalam proses budidaya ikan kerapu, sistem dapat memberikan informasi bagi petani untuk mengetahui kegiatan operasional yang harus dilakukan. Bagi pemula dapat digunakan sebagai acuan dalam proses operasional budidaya ikan kerapu.

1.5 Batasan Masalah

Batasan masalah dalam perancangan sistem operasional budidaya ikan kerapu ini adalah sebagai berikut:

1. Sistem yang dirancang berbentuk *prototype* berbasis *mobile* android versi 4.4 (KitKat) hingga android versi 7.0 (Nougat).
2. Observasi dilakukan hanya dilakukan pada lingkup kelompok tani Bhakti Usaha II
3. Data dan Parameter yang ada dalam sistem hanya dari kelompok tani Bhakti Usaha II.
4. Model *software Development Life Cycle* (SDLC) yang digunakan adalah *Iterative*

1.6 Sistematika Pembahasan

Sistematika pembahasan penelitian ditunjukkan untuk memberikan gambaran dari uraian penyusunan skripsi secara garis besar yang meliputi beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Menguraikan mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, serta sistematika yang digunakan dalam penyusunan skripsi ini.

BAB II LANDASAN KEPUSTAKAAN

Menguraikan kajian pustaka, dasar-dasar teori yang melandasi penulisan dan penelitian dari skripsi pembangunan sistem operasional budidaya ikan kerapu.

BAB III METODOLOGI

Menguraikan metodologi penelitian pengerjaan skripsi ini. Bab ini terdiri dari beberapa sub bab antara lain studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian, dan pengambilan kesimpulan.



BAB IV ANALISIS KEBUTUHAN

Membahas analisis kebutuhan yang direpresentasikan dengan menggunakan *use case* dan *use case scenario*.

BAB V PERANCANGAN DAN IMPLEMENTASI

Membahas tentang perancangan sistem berdasarkan analisis kebutuhan dan akan dirancang menggunakan perancangan basis data, diagram *sequence*, dan *class diagram*. Bab ini juga membahas implementasi sistem berupa bahasa pemrograman yang digunakan, metode yang digunakan, serta prosedur-prosedur yang dilakukan pada pembuatan sistem.

BAB VI PENGUJIAN

Memuat tentang pengujian dan analisis terhadap sistem yang dibangun. Pengujian dan analisis akan dijelaskan dalam bentuk gambar-gambar dari proses pengujian dan analisis sistem.

BAB VII PENUTUP

Memuat kesimpulan yang didasarkan atas pengujian dan analisis yang dilakukan dalam proses penelitian, serta saran-saran dari hasil yang diperoleh yang diharapkan dapat bermanfaat dalam pengembangan selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi tinjauan pustaka yang terdiri dari kajian pustaka dan dasar teori yang digunakan penulis untuk menyusun skripsi. Kajian pustaka membahas penelitian yang telah ada yang akan diuraikan berdasarkan studi kasus yang berkaitan dengan penelitian yang dilakukan oleh penulis. Dasar teori digunakan untuk membahas teori pendukung yang digunakan untuk membangun sistem.

2.1 Kajian Pustaka

Kajian pustaka yang digunakan dalam penelitian ini membahas tentang penelitian sebelumnya mengenai pembangunan sistem panduan budidaya.

Pada penelitian sebelumnya yang dilakukan oleh Aditya Sheli Cahya Utama (2015) yang berjudul “Rancang Bangun Aplikasi *Mobile* Panduan Budidaya Tanaman Cabai Organik Secara Mandiri”. Pada penelitian tersebut penulis ingin membangun sebuah sistem informasi berbasis *mobile* dan web (*hybrid*). Program ini meliputi langkah-langkah budidaya tanaman cabai dari persemaian, penanaman, cara perawatan, dan pengendalian hama dengan pestisida alami. Sistem panduan budidaya tanaman cabai organik ini dapat memudahkan petani dalam melakukan budidaya cabai secara organik serta untuk mendapatkan hasil yang optimal. Perbedaan dengan penelitian yang sedang dibuat adalah sistem yang akan dibuat tidak hanya menjadi petunjuk bagi petani ikan kerapu juga dapat menjadi pengingat kegiatan.

Sedangkan pada penelitian yang dilakukan oleh Faisal Rahman (2016) berjudul “Aplikasi Peternakan Ayam Broiler Berbasis Android”. Dalam penelitian ini penulis membangun sebuah aplikasi yang dapat menjadwalkan jurnal dan SOP (*Standart Operating Sistem*) harian, aplikasi peternakan ayam boiler juga mampu menampilkan data stok yang ada di gudang. Sehingga aplikasi ini dapat membantu dalam pengelolaan peternakan ayam broiler. Perbedaan penelitian ini dengan penelitian yang penulis buat adalah dari jadwal yang disediakan, penulis menyediakan penjadwalan kegiatan budidaya tambak ikan kerapu.

2.2 Ikan Kerapu

Ikan kerapu (*Epinephelus* spp) merupakan jenis ikan yang hidup secara berkelompok, yang memangsa ikan dan krustase yang merupakan salah satu komoditas perikanan yang memiliki peluang baik di pasar domestik maupun pasar internasional, serta memiliki nilai jual yang cukup tinggi (Langkosono, 2007 dan Triana, 2010 disitasi dalam Paruntu, 2015). Ikan kerapu yang memiliki harga jual yang tinggi membuat banyak nelayan yang menangkap ikan ini. Semakin banyaknya permintaan pasar membuat nelayan tidak bisa memenuhi seluruh kebutuhan pasar, yang membuat budidaya ikan kerapu semakin berkembang saat ini. Pada umumnya budidaya ikan dilakukan dengan menggunakan kerambah jaring apung (KJA) dan dengan menggunakan tambak. Ikan kerapu memiliki masa pertumbuhan yang berbeda-beda tergantung dari jenis ikan kerapu tersebut.



2.3 Kelompok Tani Tambak

Menurut Peraturan Menteri Pertanian (Permentan) No. 82 tahun 2013 tentang pedoman pembinaan kelompok tani menyebutkan bahwa, kelompok tani adalah kumpulan dari petani/peternak/pekebun yang terbentuk karena memiliki kesamaan kepentingan, kesamaan kondisi lingkungan, kesamaan komoditas, dan keakraban untuk mengembangkan dan meningkatkan usaha anggota. Dari pengertian tersebut dapat ditarik kesimpulan bahwa kelompok tani tambak adalah kumpulan petani tambak yang dibentuk berdasarkan kepentingan yang sama dan tujuan untuk mengembangkan usaha usaha tani tambak anggotanya.

Kelompok tani sudah mulai berkembang secara kuantitas pada tahun 1993 yang tercatat terdapat 250.000 kelompok tani, sedangkan pada tahun 2010 tercatat 279.523 kelompok tani. Perkembangan kelompok tani tidak lepas dari peranan Permentan No.82 tahun 2013 tentang pembinaan kelompok tani yang mengatur bahwa pembinaan kelompok tani diarahkan pada peningkatan peran serta petani, penerapan sistem agribisnis, dan anggota masyarakat pedesaan lainnya, dengan menumbuhkan kerjasama antar petani dan pihak terkait lainnya untuk pengembangan usaha tani. Pembinaan kelompok tani juga diharapkan dapat membantu untuk menggali potensi, memudahkan dan memecahkan masalah anggota usaha tani secara lebih efektif.

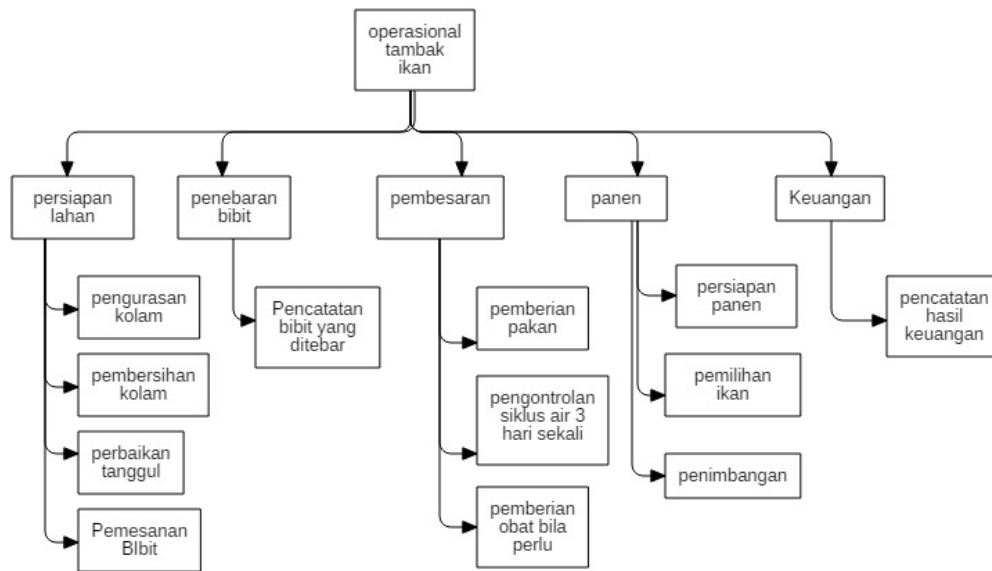
2.3.1 Kelompok Tani Tambak Bhakti Usaha 2

Untuk meningkatkan produktivitas kegiatan usaha budidaya ikan di desa Labuhan kecamatan Brondong kabupaten Lamongan, maka dibentuk kelompok tani tambak Bhakti Usaha. Pembentukan kelompok tani tambak didasari oleh kondisi lingkungan yang mendukung, banyaknya petani tambak di desa Labuhan.

Perkembangan selanjutnya pada tanggal 4 November 1993 kelompok tani Bhakti Usaha dipecah menjadi dua oleh kepala desa Labuhan pada tanggal sehingga terbentuk kelompok tani Bhakti Usaha 2 sebagai kelompok tani kelas pemula. Pada saat awal terbentuk kelompok tani Bhakti Usaha 2 tercatat memiliki jumlah anggota sebanyak 20 orang. Dengan ini penulis ingin membantu kelompok tani tambak Bhakti Usaha 2 untuk memudahkan pengelolaan operasional petani ikan kerapu.

2.4 Proses Budidaya

Proses budidaya ikan kerapu dapat dilakukan dengan dua jenis tempat budidaya, yaitu menggunakan kerambah jaring apung (KJA) dan menggunakan tambak. Kelompok tani Bhakti Usaha 2 menggunakan tambak sebagai tempat budidaya ikan kerapu. Menurut hasil wawancara, proses budidaya ikan kerapu dengan menggunakan tambak dapat dilihat pada diagram *Work Breakdown Sistem* (WBS) yang ditunjukkan oleh Gambar 2.1



Gambar 2.1 WBS Budidaya Tambak Ikan Kerapu

2.4.1 Persiapan Budidaya

Pada tahap persiapan budidaya petani melakukan pengurusan kolam terlebih dahulu untuk melihat kondisi kolam, membersihkan kolam dari sisa-sisa budidaya sebelumnya dan juga kotoran yang ada pada kolam. Selain itu persiapan yang dilakukan juga memeriksa kerusakan tanggul dari kolam yang akan digunakan untuk budidaya ikan kerapu. Saat kolam sudah pada kondisi siap digunakan petani melakukan pemesanan atau pembelian bibit ikan kerapu. Data rincian persiapan budidaya dapat di lihat pada Tabel 2.1.

Tabel 2.1 Rincian persiapan budidaya

No	Kegiatan	Keterangan
1	Pengurusan Kolam	Dilakukan saat ikan sudah selesai dipanen
2	Perbaikan tanggul	Dilakukan saat tanggul kolam mengalami kerusakan
3	Pembersihan kolam	Kolam dibersihkan dari kotoran sisa budidaya sebelumnya dan membersihkan hama yang ada di kolam
4	Pemesanan bibit	Bibit di pesan dengan perbandingan 1 ikan dalam 1,5 m ² ukuran kolam



2.4.2 Penebaran Bibit

Pada tahap ini petani ikan kerapu melakukan pengecekan kepada bibit ikan kerapu apakah bibit layak atau tidak untuk dibesarkan, setelah itu air dari kantong bibit akan dicampurkan dengan air dari kolam untuk melakukan penyesuaian air terhadap bibit ikan kerapu. Selanjutnya bibit ikan kerapu akan ditabur ke dalam kolam, berikut umur ikan kerapu yang ditunjukkan pada tabel 2.2.

Tabel 2.2 Data Umur ikan kerapu

No.	Kegiatan	Ukuran Ikan 400-500 gram
1	Lama Pemeliharaan Ikan Kerapu Cantang (bulan)	7-8
2	Lama Pemeliharaan Ikan Kerapu Lumpur (bulan)	8-9

2.4.3 Pembesaran Ikan

Pada proses pembesaran ikan kerapu, petani ikan kerapu melakukan pemberian pakan dua kali sehari saat ikan kerapu berumur dibawah dua bulan, setelah ikan kerapu berumur lebih dari dua bulan maka pemberian pakan berubah menjadi satu kali sehari. Selain itu petani ikan kerapu melakukan pengontrolan siklus air sekali dalam tiga hari, pada saat ikan teridentifikasi penyakit maka petani melakukan pemberian obat/vitamin untuk mencegah penyakit menyebar kepada ikan yang lain.

Tabel 2.3 Proses Pemeliharaan

No	Kegiatan	Keterangan
1	Pemberian pakan	1-2 bulan = 2 kali pakan
		≥ 3 bulan = 1 kali pakan
		1 hektar = 30 kg pakan
2	Pemberian obat	Saat ikan teridentifikasi penyakit
3	Kontrol siklus air	2-3 hari sekali

2.4.4 Panen

Pada proses panen petani akan melakukan penangkapan total ikan kerapu, selanjutnya ikan akan dipilah menurut kualifikasi perhitungan yaitu ikan kerapu yang masuk kriteria perhitungan perkilo gram dan yang masuk perhitungan perekor, setelah itu petani melakukan penimbangan ikan kerapu yang sudah dipanen. Pemilahan jenis penjualan dapat dilihat pada Tabel 2.4.

Tabel 2.4 Jenis jual

Jenis jual	ukuran
Kiloan	≥ 500 gram
Ekoran	> 1kg

2.4.5 Keuangan

Petani akan menghitung nota modal dan biaya yang dikeluarkan saat proses budidaya, perhitungan ini dilakukan untuk mengetahui keuntungan yang didapat dari hasil penjualan ikan kerapu kepada pengepul.

2.5 *Software Development Life Cycle (SDLC)*

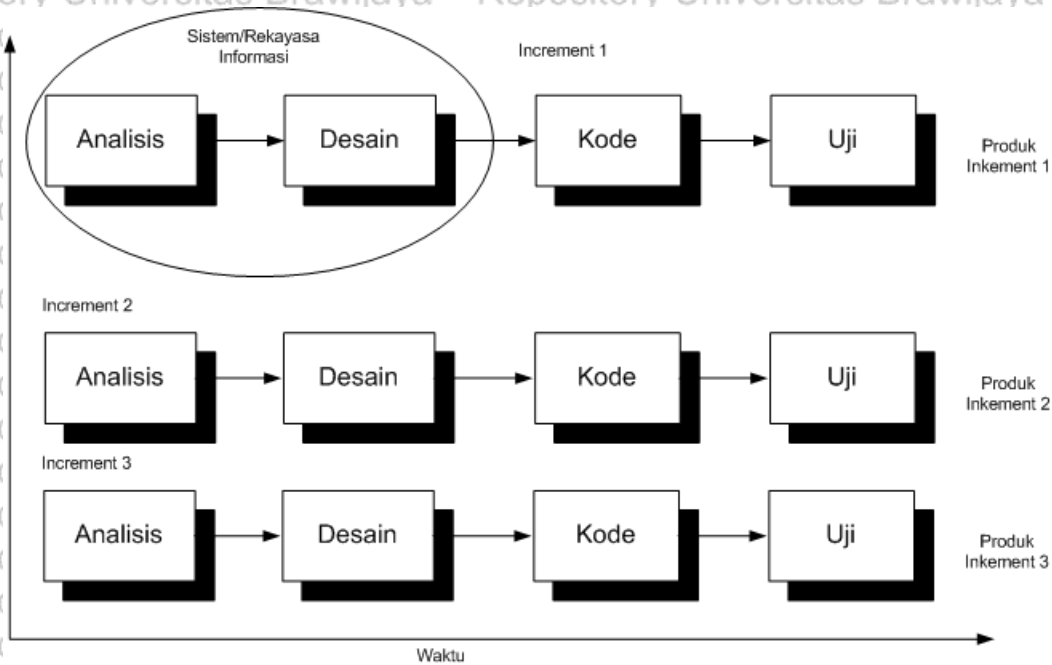
Software Development Life Cycle (SDLC) adalah metodologi yang digunakan untuk pengembangan sistem perangkat lunak. Terdapat beberapa jenis proses pengembangan perangkat lunak, proses tersebut harus memenuhi empat aktivitas sebagai dasar perangkat lunak, yaitu: (Sommerville, 2011).

1. *Software specification*, merupakan proses pendefinisian fungsionalitas sistem dan batasan dari perangkat lunak.
2. *Software desing and implementation*, merupakan proses menghasilkan perangkat lunak sesuai dengan spesifikasi sistem.
3. *Software validation*, merupakan proses validasi untuk memastikan perangkat lunak sudah sesuai dengan kebutuhan.
4. *Software evolution*, proses pengembangan perangkat lunak dengan memperbaiki sesuai perubahan kebutuhan dari aktor.

Terdapat beberapa model dalam SDLC, salah satunya adalah model *iteratif* yang akan diterapkan ke dalam penelitian ini.

2.5.1 Model *Iterative*

Model *iteratif* merupakan gabungan dari model *prototype* dan model *waterfall*, model perancangan ini lebih cocok digunakan pada saat perputaran staf yang tinggi (Shalahuddin & Rosa, 2015). Pada model *iterative* ini dikatakan penggabungan model *waterfall* dan *prototype* dikarenakan proses ini bekerja dengan menggunakan alur pengembangan dari proses analisis hingga uji secara berurutan selayaknya model *waterfall*, setelah proses uji dilakukan maka sistem dapat kembali ke proses analisis bila hasil diskusi dengan *stackholder* saat sistem didistribusikan terdapat penambahan atau perubahan, proses ini selayaknya seperti proses dari model *prototype*. Pada Gambar 2.2 merupakan representasi dari model *iteratif*.



Gambar 2.2 Model *iterative* (Shalahuddin & Rosa, 2015)

Tahapan-tahapan utama pada model perancangan *iterative* yang menggambarkan kegiatan pengembangan yang mendasar:

1. Analisis
Proses analisis merupakan proses untuk memahami sifat dari sistem yang akan dibuat, mengetahui kebutuhan dari sistem, serta fungsi dan antarmuka dari sistem yang dibangun.
2. Desain
Proses desain ini merupakan sebuah proses penerjemahan persyaratan ke dalam representasi *software* yang dapat dinilai kualitasnya sebelum memulai proses pengkodean sistem. Desain juga didokumentasikan dan menjadi sebuah bagian dari konfigurasi *software*.
3. Kode
Proses kode merupakan proses penerjemahan desain program ke dalam bahasa komputer, yaitu dengan cara melakukan penulisan kode program.
4. Uji
Setelah kode program selesai dibuat, maka pengujian program dimulai. Proses ini berfokus pada internal logis dari perangkat lunak, memastikan semua *statements* telah diuji, dan juga semua fungsionalitas sesuai dengan kebutuhan dan keluaran yang dihasilkan sesuai dengan masukan yang ada.

2.6 Android

Android merupakan salah satu sistem operasi yang dikembangkan oleh Google dengan menggunakan Linux sebagai dasar pengembangannya, sistem operasi ini dirancang untuk bergerak seperti *smartphone* dan computer tablet. Sistem operasi ini dirilis secara resmi oleh Google pada tahun 2007, bersamaan



dengan didirikannya OHA (*Open Handset Alliance*), yang merupakan konsorsium dari berbagai perusahaan, tercatat 34 perusahaan besar terlibat dalam pengembangan sistem operasi ini. Saat ini sebagian besar *vendor smartphone* sudah menggunakan android, hal ini didasari oleh sistem operasi android yang bersifat *open source* sehingga bebas untuk dipakai oleh *vendor* manapun (Safaat, 2012)

Perkembangan android diawali dengan dirilisnya Android beta yang dirilis pada November 2007, yang merupakan versi android pertama yang dikomersialkan, setahun berselang tepatnya pada September 2008 android 1.0 dirilis. Sejak tahun April 2009 penamaan dari versi pengembangan Android didasari oleh nama-nama makanan ringan. Masing-masing versi diurutkan sesuai urutan alphabet. Saat ini versi android sudah memasuki Android Oreo (8.0).

2.7 SQLite

SQLite adalah salah satu *software embedded* yang sangat populer, penggunaan *memory* yang sangat sedikit dan kecepatan yang sangat cepat, semua versi dari android dapat menggunakan SQLite Karena di android termasuk sebagai aplikasi yang *runtime* (Safaat, 2012).

Untuk membuka dan membuat *database* adalah dengan menggunakan *libraries* SQLite, yang menyediakan tiga metode yaitu:

- Construction, menyediakan representasi versi *database* serta skema *database* yang digunakan.
- *onCreate()*, menyediakan SQLiteDatabase *object* yang digunakan dalam inialisasi data dan referensi tabel.
- *onUpgrade()*, menyediakan fasilitas konversi *database* dari versi *database* yang lama ke *database* versi baru.

Dari kemudahan dan kelebihan yang ada dalam pembuatan *database*, maka pengguna menggunakan SQLite sebagai *software* pembuatan *database*.

2.8 Unified Modeling Language (UML)

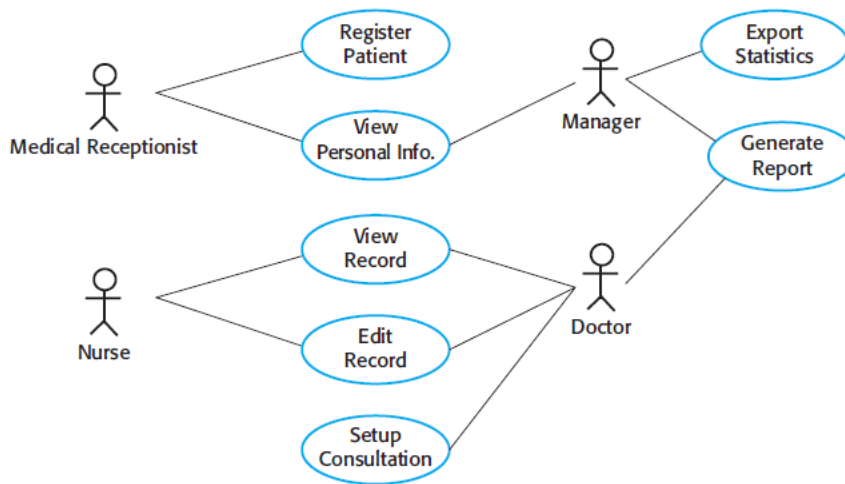
Unified Modeling Language (UML) adalah sekumpulan pemodelan konvensi untuk menentukan atau menggambarkan suatu sistem perangkat lunak yang berkaitan dengan objek (Whitten, et al., 2004). Dalam merancang suatu perangkat lunak biasanya menggunakan beberapa komponen dalam UML.

2.8.1 Use Case Diagram

Use case diagram sekarang telah menjadi sebuah fitur dasar dari bahasa pemodelan sistem perangkat lunak. Pada bentuk yang paling sederhana, *use case* digunakan untuk mengidentifikasi aktor yang terlibat interaksi dan menamai interaksinya, kemudian dilengkapi dengan informasi tambahan yang digunakan untuk menggambarkan interaksi dengan sistem. informasi tambahan tersebut dapat berupa deskripsi tekstual atau model grafis seperti urutan UML



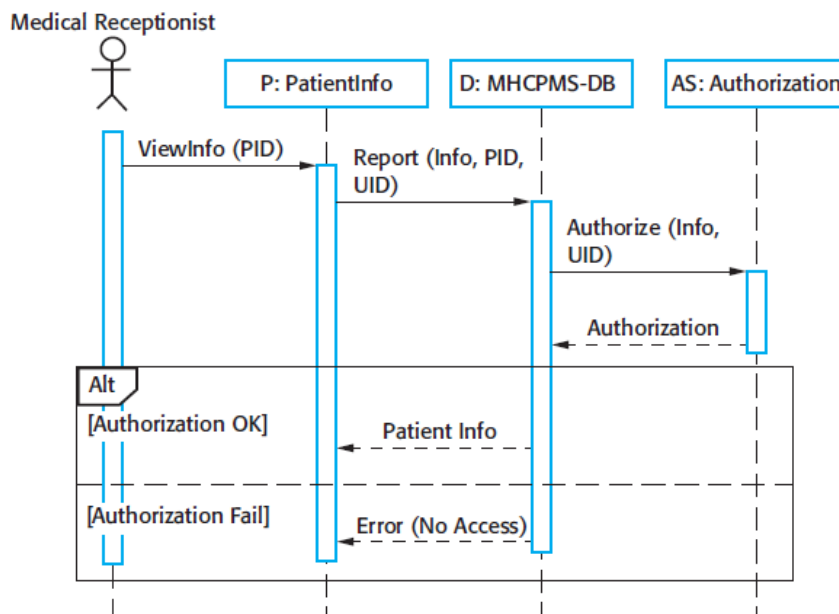
(Sommerville, 2011). Pada Gambar 2.3 merupakan contoh dari penerapan *use case diagram*.



Gambar 2.3 Contoh *use case* (Sommerville, 2011)

2.8.2 Sequence Diagram

Sequence diagram dalam UML digunakan untuk memodelkan interaksi antar aktor dengan objek dalam suatu sistem dan interaksi antara objek itu sendiri (Sommerville, 2011). Pada Gambar 2.4 merupakan contoh dari penggunaan *Sequence* diagram yang menggambarkan notasi-notasi dasar.

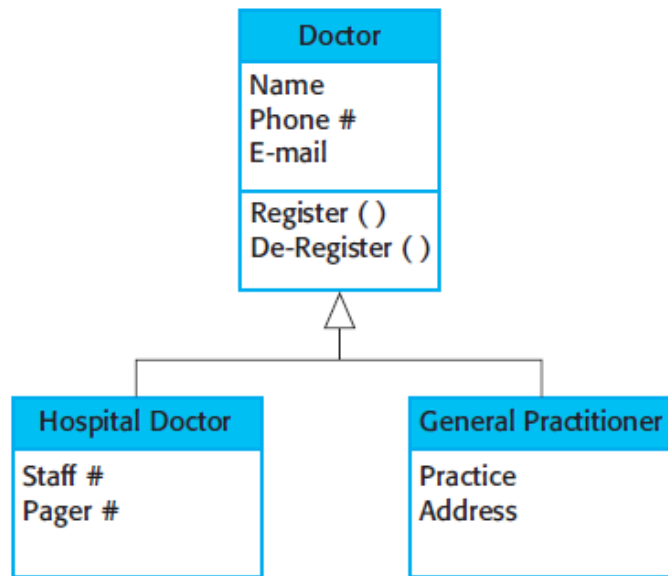


Gambar 2.4 Contoh *sequence* diagram (Sommerville, 2011)



2.8.3 Class Diagram

Class diagram digunakan saat mengembangkan sistem dengan menggunakan model pengembangan sistem berorientasi objek yang berguna untuk menunjukkan kelas di dalam pembuatan sistem dan asosiasi antar kelas (Sommerville, 2011). Pada Gambar 2.5 merupakan contoh dari *class* diagram yang menggambarkan asosiasi antar kelas.



Gambar 2.5 Contoh *class* diagram (Sommerville, 2011)

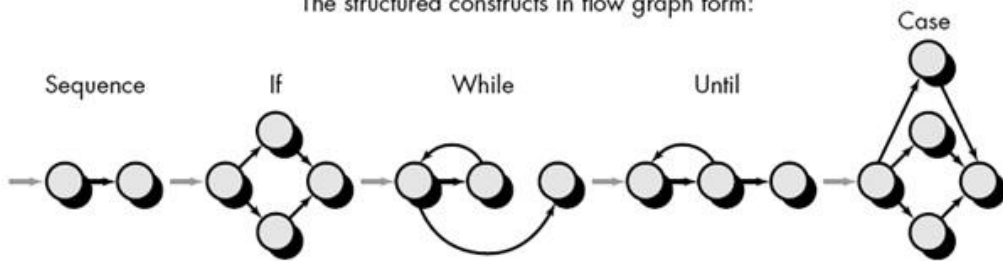
2.9 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan tahapan yang dilakukan untuk memastikan bahwa sistem dapat memenuhi semua persyaratan perangkat lunak. Pengujian perangkat lunak memerlukan sebuah perancangan kasus uji untuk dapat menemukan kesalahan dalam waktu yang singkat. Terdapat dua jenis pengujian yang digunakan oleh penulis yang diuraikan sebagai berikut.

2.9.1 White-Box Testing

White-box testing merupakan sebuah metode pengujian yang berfokus pada isi dari sistem perangkat lunak (Rizky, 2011). Pengujian ini hanya bisa dilakukan saat perangkat lunak sudah dinyatakan selesai dan telah melalui tahap analisis awal. Pada penelitian ini penulis menggunakan pengujian *basis path testing*, pengujian jenis ini menggunakan *cyclomatic complexity* yang bertujuan untuk mengetahui aliran logika yang digambarkan menggunakan *flow graph* yang nantinya akan menghasilkan jalur independen (Pressman, 2010). Berikut ini adalah notasi dari *flow graph*

The structured constructs in flow graph form:



Gambar 2.6 Notasi flow graph (Pressman, 2010)

2.9.2 Black-Box Testing

Black-box testing merupakan sebuah tipe pengujian perangkat lunak yang kinerja internalnya tidak diketahui (Rizky, 2011). Pengujian ini memandang sebuah perangkat lunak layaknya sebuah kotak hitam yang tidak dapat dilihat dari dalam sistem (kode program), namun cukup dilakukan pengujian pada bagian luar sistem. Teknik pengujian *black-box* yang digunakan oleh penulis adalah *Scenario based testing* dan *equivalence partitioning*. *Scenario based testing* adalah suatu pengujian yang memiliki acuan bagaimana aktivitas *user* terhadap sistem, tetapi tidak dengan aktivitas sistem tersebut, sedangkan *equivalence partitioning* didasari oleh evaluasi dari kelas ekuivalensi terhadap kondisi masukan yang akan menggambarkan sekumpulan keadaan yang valid atau tidak valid dalam beberapa kelas (Pressman, 2010). Kondisi dari masukan berupa nilai numerik, kumpulan nilai, pusran nilai yang berhubungan atau juga kondisi boolean.

2.9.3 Pengujian Usability

usability merupakan kebutuhan non-fungsional yang mengacu terhadap bagaimana pengguna sistem dapat menggunakan dan mempelajari sistem untuk memperoleh tujuan dan seberapa puas pengguna terhadap penggunaan sistem (Dumas & Radish, 1999). Konteks penggunaan terdiri dari pengguna, tugas dan peralatan yang mencakup *hardware* dan *software*. Cara pengujian dalam *usability* yaitu dengan cara menyebarkan kuesioner kepada beberapa pengguna untuk mendapatkan sampel perhitungan kepuasan dari pengguna sistem.

2.10 USE Questionnaire

Usefulness, Satisfaction, and Ease of use (USE) questionnaire merupakan salah satu metode kuesioner yang dapat digunakan dalam mengukur *usability* dari suatu sistem, USE dapat mencakup tiga aspek dalam mengukur *usability* yaitu efisiensi, efektivitas dan kepuasan (Lund, 2001). Menurut Arnold Lund (2001) juga dari banyaknya penelitian yang telah dilakukan menunjukkan bahwa *usability* menunjukkan bahwa adanya keterkaitan antara *Ease of Use* dan *usefulness*. Sehingga peningkatan pada *Ease of Use* maka akan diikuti dengan peningkatan pada *usefulness* dan begitupun sebaliknya.

Pada *USE questionnaire* pertanyaan akan dibagi ke dalam empat aspek yaitu:



a. *Usefulness* (kegunaan)

Pada aspek ini digunakan untuk mengukur sejauh mana sistem perangkat lunak dapat memecahkan masalah yang dihadapi oleh *user*.

b. *Ease of Use* (kemudahan penggunaan)

Pada aspek ini merupakan pengukuran seberapa mudah *user* dalam menggunakan sistem perangkat lunak.

c. *Ease of Learning* (kemudahan belajar)

Pada aspek ini digunakan untuk mengukur seberapa cepat *user* dapat mempelajari sistem.

d. *Satisfaction* (kepuasan)

Pada aspek ini berhubungan dengan penilaian persepsi, perasaan dan tanggapan *user* terhadap sistem perangkat lunak yang dibuat.

Berikut merupakan beberapa contoh pertanyaan dari *USE questionnaire* yang dapat digunakan, yang ditunjukkan pada Gambar 2.7.

USEFULNESS

1. It helps me be more effective.
2. It helps me be more productive.
3. It is useful.
4. It gives me more control over the activities in my life.
5. It makes the things I want to accomplish easier to get done.
6. It saves me time when I use it.
7. It meets my needs.
8. It does everything I would expect it to do.

EASE OF USE

9. It is easy to use.
10. It is simple to use.
11. It is user friendly.
12. It requires the fewest steps possible to accomplish what I want to do with it.
13. It is flexible.
14. Using it is effortless.
15. I can use it without written instructions.
16. I don't notice any inconsistencies as I use it.
17. Both occasional and regular users would like it.
18. I can recover from mistakes quickly and easily.
19. I can use it successfully every time.

EASE OF LEARNING

20. I learned to use it quickly.
21. I easily remember how to use it.
22. It is easy to learn to use it.
23. I quickly became skillful with it.

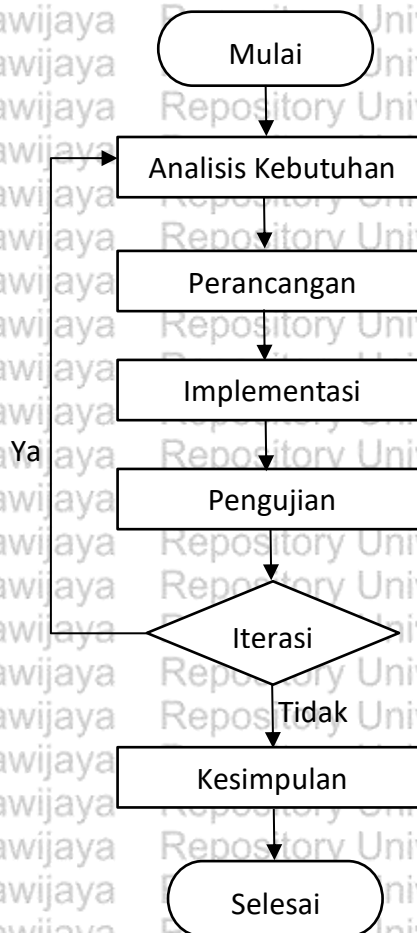
SATISFACTION

24. I am satisfied with it.
25. I would recommend it to a friend.
26. It is fun to use.
27. It works the way I want it to work.
28. It is wonderful.
29. I feel I need to have it.
30. It is pleasant to use.

Gambar 2.7 Contoh susunan *USE questionnaire* (Lund, 2001)

BAB 3 METODOLOGI

Tipe penelitian yang digunakan dalam penelitian ini adalah implementasi–pembangunan, yaitu tipe penelitian yang hasil akhirnya akan dihasilkan sebuah produk perangkat lunak dengan menggunakan prinsip rekayasa perangkat lunak. Pada bab ini menjelaskan langkah-langkah yang akan dilakukan untuk pembangunan Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK). Pengambilan dasar teori menggunakan metode studi literature dan wawancara kepada pihak yang berada di dalam lingkup studi kasus. Data tersebut diperoleh dari studi literatur dan hasil observasi yang dilakukan oleh penulis. Penjelasan melalui langkah-langkah yang akan dilakukan selama pengerjaan skripsi, dimulai dari studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian, dan penulisan laporan. Langkah-langkah tersebut diilustrasikan pada Gambar 3.1 diagram alir di bawah ini.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Studi literatur merupakan metode untuk mendapatkan dasar teori dalam membangun sebuah sistem. Studi literatur berisikan landasan dari teori yang terkait dengan konsep penelitian. Hal ini diperlukan untuk pengetahuan dasar



guna membangun sebuah sistem supaya terpenuhi dengan baik. Studi literatur yang digunakan oleh penulis dalam membangun sistem operasional budidaya ikan kerapu ini didapat dari buku, skripsi, artikel ilmiah, dan juga informasi yang sersedia di internet baik berupa artikel maupun proyek. Studi literatur yang digunakan meliputi:

1. Proses kerja dalam budidaya tambak ikan kerapu
2. *Software Development Life cycle* (SDLC) model iteratif
3. *Unified modeling language* (UML)
4. Pengujian perangkat lunak

3.2 Analisis Kebutuhan

Analisis kebutuhan adalah metode yang digunakan untuk mendapatkan seluruh kebutuhan yang diperlukan untuk membangun sistem. Analisis kebutuhan dapat dilakukan dengan mengidentifikasi kebutuhan yang diperlukan dalam perangkat lunak yang akan dibuat serta siapa saja yang akan terlibat di dalamnya.

Analisis juga dapat digunakan untuk mengetahui kondisi lapangan saat ini, sehingga dapat diketahui implementasi dari perangkat lunak yang akan dibuat. Kebutuhan-kebutuhan yang telah didapatkan akan dikategorikan bersarkan kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan sistem didapat dengan cara melakukan wawancara dengan pihak yang terlibat dalam budidaya tambak ikan kerapu, proses wawancara tersebut menggunakan metode wawancara tidak terstruktur dengan narasumber ketua kelompok tani Bhakti Usaha 2. kebutuhan yang telah didapat nantinya akan dimodelkan ke dalam *use Case diagram* yang berguna untuk mengetahui interaksi dari pengguna dengan sistem, setelah itu akan dilakukan pembuatan *use case* skenario yang berguna untuk menjelaskan cara kerja sistem berdasarkan *use case diagram* yang telah dibuat. Proses analisis kebutuhan dari Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) meliputi:

1. Elisitasi Kebutuhan

Pada tahap ini akan dilakukan wawancara untuk mengetahui kebutuhan sistem yang akan dibangun. Wawancara ini akan dilakukan kepada ketua kelompok tani Bhakti Usaha II, Bapak Wafiq Ismail.

2. Deskripsi Sistem

Pada tahap ini akan berisi penjelasan mengenai sistem yang akan dibuat dan bagaimana implementasinya. Sistem akan dibuat seperti apa dan gambaran umum dari fungsi yang akan ada dalam sistem.

3. Identifikasi aktor

Pada tahap ini bertujuan untuk mengetahui aktor yang terlibat dan akan berinteraksi dengan sistem.

4. Spesifikasi kebutuhan

Pada tahap ini kebutuhan yang telah didapatkan akan di jelaskan secara spesifik, dan akan di kategorikan ke dalam kebutuhan fungsional dan kebutuhan non-fungsional.



5. Pemodelan kebutuhan sistem

Pada tahap ini semua proses analisis yang telah dilakukan akan dimodelkan ke dalam *use case diagram*, yang nantinya akan dimodelkan lagi ke dalam *use case* skenario untuk mengetahui cara kerja dari kebutuhan fungsional sistem.

3.3 Perancangan

Perancangan dilakukan setelah proses analisis selesai dilakukan, perancangan sistem yang akan digunakan dalam pembuatan Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) sebagai berikut:

1. Perancangan arsitektur

Pada tahap ini perancangan akan dijelaskan dengan menggunakan *Unified Modeling Language* (UML) yang berupa *sequence diagram* dan *class diagram*. Pada penelitian ini *sequence diagram* hanya akan diambil tiga sampel dari perangkat lunak. Sedangkan untuk *class diagram* akan menjelaskan gambaran dari relasi kelas dalam sistem.

2. Perancangan struktur

Pada tahap ini akan dibuat sebuah perancangan basis data yang nantinya akan digunakan sebagai dasar dari pembuatan *database* sistem.

3. Perancangan algoritme

Pada tahap ini akan dilakukan perancangan algoritme yang akan digunakan ke dalam pengkodean sistem. dalam penelitian ini akan dicantumkan tiga sampel algoritme yang akan digunakan.

4. Perancangan antar muka

Pada tahap ini akan dibuat perancangan dari tampilan sistem, yang bertujuan untuk memberikan gambaran dari antarmuka yang akan diterapkan ke dalam sistem.

3.4 Implementasi

Implementasi dilakukan berdasarkan dari rancangan sistem yang telah dibuat. Implementasi sistem nantinya akan dibuat dengan menggunakan bahasa pemrograman *mobile android*, yaitu bahasa JAVA untuk pengkodean programnya, sedangkan untuk antarmuka akan menggunakan bahasa XML. Model pengembangan sistem yang akan digunakan adalah model intensif, sedangkan untuk implementasi basis datanya menggunakan SQLite. Pada sistem ini akan diimplementasikan kepada android versi 4.4 hingga android versi 7.0.

3.5 Pengujian

Pengujian dilakukan untuk mengetahui apakah sistem sudah memenuhi kebutuhan yang telah didefinisikan sebelumnya, pengujian juga digunakan untuk mengukur kelayakan sistem sebelum digunakan. Pada penelitian ini pengujian dilakukan dengan menggunakan metode *black-box testing*, *white-box testing* dan pengujian *usability*. Pengujian *usability* ini digunakan untuk mengetahui apakah sistem mudah digunakan oleh pengguna.



Black-box testing digunakan untuk menguji kebutuhan fungsional dari sistem, pada pengujian ini akan digunakan pengujian validasi yang nantinya akan menguji semua fitur-fitur yang ada pada sistem. sedangkan *white-box testing* akan difokuskan kepada pengujian unit yang akan digunakan untuk menguji komponen tertentu dengan cara memastikan bahwa algoritme dari suatu komponen dapat bekerja dengan baik. Tahapan pengujian ini dilakukan dengan membuat *flow graph*, yang nantinya akan menjadi acuan untuk menentukan *cyclimatic complexity* serta jalur independen yang ada pada fungsi terkait.

Sedangkan pengujian *usability* digunakan untuk mengetahui bagaimana tanggapan pengguna saat menggunakan sistem. Pengujian ini dilakukan dengan cara memberikan *prototype* dari sistem yang telah selesai dibuat kepada pengguna, setelah itu pengguna akan diberikan kuesioner yang berfungsi untuk memberikan nilai terhadap kemudahan dan kesesuaian sistem yang telah dibuat terhadap pengguna.

3.6 Iterasi

Pada proses ini akan dilakukan setelah proses pengujian dilakukan. Bila *stackholder* menginginkan perubahan atau penambahan fungsi pada sistem, maka proses pengembangan akan kembali ke proses analisis sistem yang nantinya. Proses ini akan dinamakan proses iterasi kedua dan seterusnya tergantung dari berapa kali iterasi telah dilakukan. Bila *stackholder* tidak menginginkan perubahan atau penambahan fungsi maka sistem dapat dinyatakan selesai atau *final*

3.7 Kesimpulan dan Saran

Pengambilan kesimpulan akan diambil berdasarkan dari hasil analisis, perancangan, implementasi dan pengujian. Kesimpulan yang dibuat akan menjawab permasalahan dari penelitian. Hasil penelitian nantinya akan diambil berdasarkan keseluruhan dari proses pembuatan sistem informasi budidaya tambak ikan kerapu untuk menunjukkan apakah permasalahan yang ada dapat terjawab dan tujuan dari penelitian tercapai. Dengan adanya kesimpulan makan nantinya dapat memberikan saran untuk pengembangan sistem atau perancangan perangkat lunak selanjutnya.



BAB 4 REKAYASA KEBUTUHAN

4.1 Deskripsi Sistem

Sistem ini adalah sebuah sistem informasi operasional budidaya tambak ikan kerapu yang bertujuan sebagai sistem pembantu dalam operasional pembudidayaan tambak ikan kerapu bagi para petani ikan kerapu. Pada sistem ini petani dapat mengetahui kegiatan perencanaan budidaya, mencatat data penebaran bibit, mengetahui informasi kegiatan operasional, merencanakan kegiatan panen dalam satu siklus pembesaran ikan kerapu, serta mengetahui hasil keuangan dari budidaya.

Pada sistem ini hanya memiliki satu aktor yaitu petani ikan kerapu, petani dapat memasukkan perencanaan budidaya dan mendapatkan informasi persiapan yang harus dilakukan. Petani dapat memasukkan data bibit ikan kerapu yang ditebar, mengetahui kegiatan operasional yang harus dikerjakan, mengetahui kapan panen dapat dilakukan dan hasil keuntungan dari budidaya ikan kerapu.

Sistem ini dibuat dengan menggunakan *platform* android dan SQLite sebagai pengolahan data aplikasi. Dengan menggunakan teknologi *mobile smartphone* android, aplikasi ini dapat dijalankan pada semua lingkungan *device smartphone* android.

4.2 Identifikasi Aktor

Pada tahap ini dapat diidentifikasi aktor yang dapat berinteraksi dengan aplikasi. Pada Tabel 4.1 menunjukkan pengguna aplikasi dan penjelasannya.

Tabel 4.1 dentifikasi pengguna

Aktor	Deskripsi
Pengguna	Pengguna adalah petani ikan kerapu yang dapat menggunakan aplikasi untuk melakukan budidaya ikan kerapu, menjadwalkan kegiatan pembelian bibit, memasukkan data bibit ikan, mengetahui kegiatan operasional budidaya, mengetahui kapan panen bisa dilakukan, dan mengetahui hasil dari keuntungan budidaya ikan kerapu.

4.3 Spesifikasi Kebutuhan

Berdasarkan pada hasil identifikasi sistem dan pengguna, didapatkan beberapa kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan *fungsional* merupakan kebutuhan yang dapat dilakukan oleh pengguna, sedangkan kebutuhan non-fungsional merupakan kebutuhan yang secara tidak langsung harus ada dalam sistem.



4.3.1 Kebutuhan Fungsional Iterasi Pertama

Tabel 4.2 Kebutuhan fungsional iterasi pertama

No	Kode	Nama fungsi	Deskripsi
1	SIOBTIK_F_01	Menambah Kolam	Pengguna dapat menambahkan kolam yang akan digunakan untuk budidaya ikan kerapu.
2	SIOBTIK_F_02	Melihat Daftar Kolam	Pengguna dapat melihat daftar kolam yang sudah ditambahkan ke dalam sistem.
3	SIOBTIK_F_03	Menghapus Kolam	Pengguna dapat menghapus kolam yang sudah tidak digunakan dalam budidaya ikan kerapu.
4	SIOBTIK_F_04	Melihat Detail Kolam	Pengguna dapat melihat detail informasi kolam yang telah dimasukkan ke dalam sistem.
5	SIOBTIK_F_05	Mengubah Detail Kolam	Pengguna dapat mengubah detail informasi kolam yang digunakan dalam budidaya ikan kerapu.
6	SIOBTIK_F_06	Menambah Persiapan Budidaya	Pengguna dapat menambahkan persiapan budidaya ikan kerapu.
7	SIOBTIK_F_07	Melihat Daftar Persiapan Budidaya	Pengguna dapat melihat daftar centang yang dapat digunakan untuk memantau proses persiapan budidaya ikan kerapu.
8	SIOBTIK_F_08	Menandai Kegiatan Persiapan Budidaya	Pengguna dapat menandai ceklis kegiatan persiapan budidaya yang telah dilakukan.
9	SIOBTIK_F_09	Menghapus Kegiatan Persiapan Budidaya	Sistem akan menghapus kegiatan persiapan budidaya secara otomatis saat pengguna sudah memasukkan data tebar bibit pada kolam yang sama.
10	SIOBTIK_F_10	Mencatat Penebaran Bibit	Pengguna dapat mencatat bibit yang ditebar sebagai acuan proses operasional budidaya ikan kerapu.
11	SIOBTIK_F_11	Melihat Daftar Pembibitan	Pengguna dapat melihat daftar pencatatan penebaran bibit.



Tabel 4.2 Kebutuhan fungsional iterasi pertama (lanjutan)

No	Kode	Nama fungsi	Deskripsi
12	SIObTIK_F_12	Melihat Detail Pembibitan	Pengguna dapat melihat detail informasi penebaran bibit.
13	SIObTIK_F_13	Menghapus Kegiatan Harian	Sistem akan menghapus semua kegiatan harian secara otomatis saat kolam yang sama saat pengguna memasukkan kegiatan persiapan budidaya.
14	SIObTIK_F_14	Menambahkan Kegiatan harian	Sistem dapat menambahkan kegiatan harian secara otomatis ke kalender kegiatan berdasarkan data penebaran bibit.
15	SIObTIK_F_15	Melihat Kalender Kegiatan Budidaya	Pengguna dapat melihat kalender kegiatan selama proses budidaya ikan kerapu.
16	SIObTIK_F_16	Menambah Kegiatan Budidaya Lain	Pengguna dapat menambahkan kegiatan operasional sesuai kebutuhan.
17	SIObTIK_F_17	Menambah Kegiatan Panen	Pengguna dapat menambahkan kegiatan yang akan dilakukan saat panen ikan kerapu.
18	SIObTIK_F_18	Melihat Daftar Panen	Pengguna dapat melihat daftar panen dari budidaya ikan kerapu.
19	SIObTIK_F_19	Melihat Detail Kegiatan Panen	Pengguna dapat melihat daftar kegiatan panen yang akan dilakukan pada sebuah kolam.
20	SIObTIK_F_20	Menghapus Detail Panen	Pengguna dapat menghapus daftar kegiatan panen pada kolam yang dipilih.
21	SIObTIK_F_21	Menambah Biaya Pengeluaran	Pengguna dapat menambahkan biaya pengeluaran saat proses budidaya ikan kerapu.
22	SIObTIK_F_22	Menambah Pemasukan Uang	Pengguna dapat menambahkan pemasukan uang yang diperoleh saat budiday ikan kerapu.
23	SIObTIK_F_23	Melihat Daftar Keuangan	Pengguna dapat melihat daftar keuangan dari budidaya ikan kerapu.

**Tabel 4.2** Kebutuhan fungsional iterasi pertama (lanjutan)

No	Kode	Nama fungsi	Deskripsi
24	SIOBTIK_F_24	Melihat Detail Keuangan	Pengguna dapat melihat detail dari keuangan yang ada di daftar.
25	SIOBTIK_F_25	Menghapus Keuangan	Pengguna dapat menghapus hasil keuangan yang diinginkan.
26	SIOBTIK_F_26	Pemberitahuan Kegiatan	Pengguna mendapatkan pemberitahuan kegiatan dari sistem, untuk setiap kegiatan.

4.3.2 Kebutuhan Fungsional Iterasi Kedua

Dari proses iterasi pertama yang telah dilakukan, maka diperoleh bahwa sistem ditambahkan fungsional pada iterasi kedua, fungsionalitas tersebut yaitu: menambah jadwal persiapan budidaya, melihat jadwal persiapan budidaya, mengubah jadwal persiapan budidaya, menghapus jadwal persiapan budidaya. Kebutuhan fungsional Iterasi Kedua dapat dilihat pada Tabel 4.3.

Tabel 4.3 Kebutuhan fungsional iterasi kedua

No	Kode	Nama fungsi	Deskripsi
1	SIOBTIK_F_27	Menambah Jadwal Persiapan Budidaya	Pengguna dapat menambahkan jadwal kegiatan persiapan budidaya yang akan dilakukan.
2	SIOBTIK_F_28	Melihat Jadwal Persiapan Budidaya	Pengguna dapat melihat daftar kegiatan jadwal persiapan budidaya yang akan dilakukan.
3	SIOBTIK_F_29	Menghapus Jadwal Persiapan Budidaya	Sistem akan menghapus secara otomatis kegiatan persiapan budidaya saat pengguna sudah memasukkan data tebar bibit pada kolam yang sama.

4.3.3 Kebutuhan Non-Fungsional

Kebutuhan non fungsional didapat dari hasil analisis kebutuhan. Kebutuhan non fungsional tidak secara langsung terkait dengan fitur yang ada pada sistem. Tabel 4.4 merupakan daftar kebutuhan non fungsional dari Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK).



Tabel 4.4 Kebutuhan non-fungsional

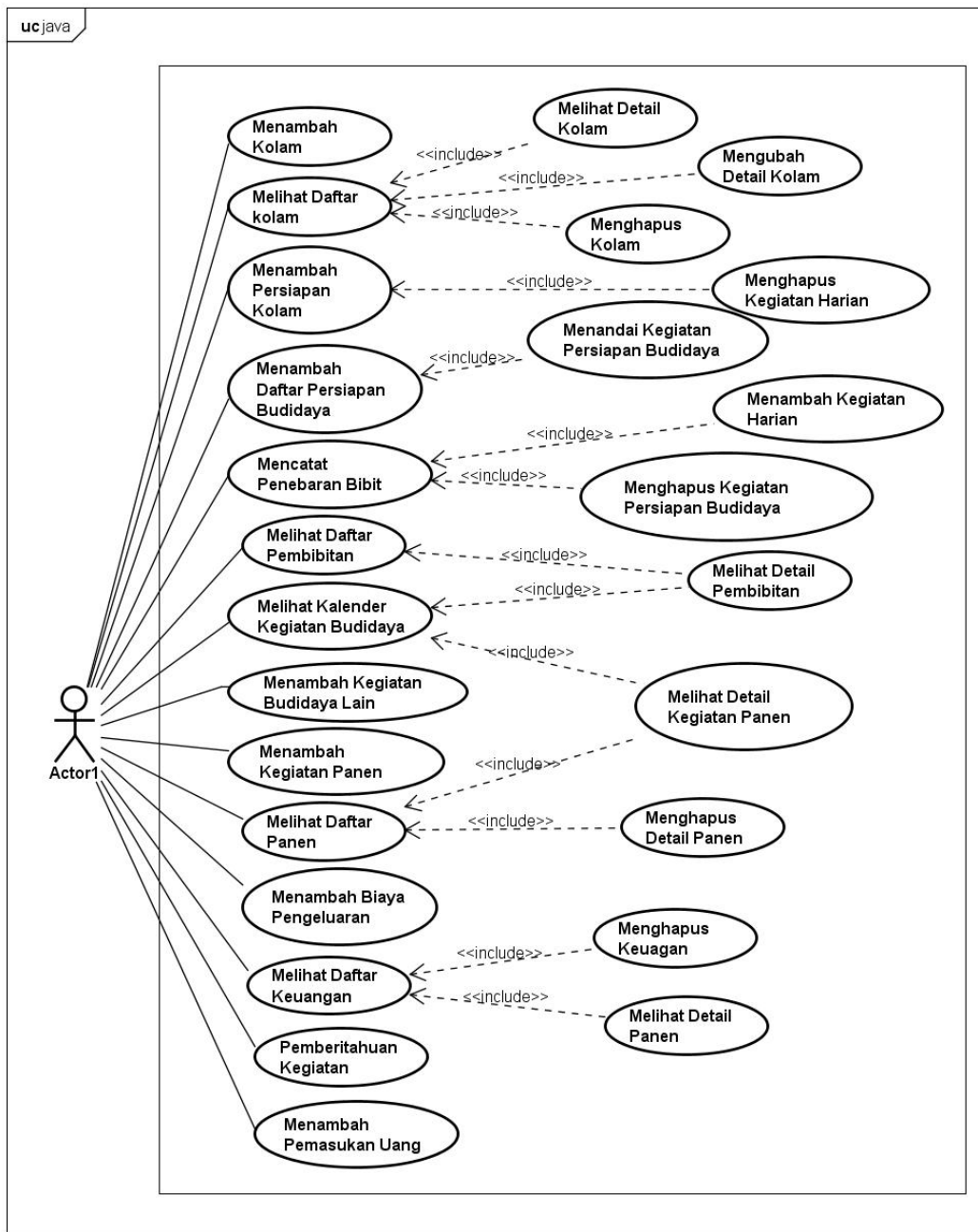
No	Kode	Nama Fungsi	Deskripsi
1	SOBIK_NF_01	<i>Usability</i>	Aplikasi dapat dengan mudah digunakan oleh pengguna melalui penyebaran <i>USE questionnaire</i> , variabel yang digunakan yaitu kemudahan pengguna aplikasi dan pencapaian kegunaan.

4.4 Pemodelan Kebutuhan Sistem

Pada proses pemodelan kebutuhan berguna untuk memudahkan dalam proses pembacaan kebutuhan pada saat proses perancangan. Pada pemodelan kebutuhan akan digambarkan menggunakan *use case diagram* dan akan dijelaskan secara lebih detail dengan menggunakan *use case* skenario.

4.4.1 Use case Diagram Iterasi Pertama

Use case diagram berguna untuk mengetahui interaksi antara aktor dengan sistem. Identifikasi aktor yang dapat berinteraksi dengan sistem didasari dari hasil analisis identifikasi aktor. Berdasarkan dari tabel fungsionalitas iterasi pertama, maka *use case diagram* iterasi pertama dapat digambarkan untuk mengetahui keterlibatan aktor ke dalam sistem. *Use case diagram* dari Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) iterasi pertama dapat dilihat pada Gambar 4.1.



Gambar 4.1 Use case diagram iterasi pertama

4.4.2 Use Case Diagram Iterasi Kedua

Berdasarkan dari tabel fungsionalitas iterasi pertama dan iterasi kedua, maka use case diagram pada iterasi kedua dapat digambarkan untuk mengetahui keterlibatan aktor ke dalam sistem. Use case diagram dari Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) iterasi kedua dapat dilihat pada Gambar 4.2.



dari masing-masing *use case* diagram iterasi pertama Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK).

1. *Use Case* Skenario Menambah Kolam

Tabel 4.5 *Use case* skenario menambah kolam

Nama Fungsional	Menambah Kolam
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor saat menambahkan kolam.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor berada di halaman menu kolam.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol tambah. 2. Sistem menampilkan halaman tambah kolam yang berisi <i>form</i> untuk memasukkan nama kolam, luas kolam, posisi kolam. 3. Aktor memasukkan nama kolam dan menekan tombol "simpan". 4. Sistem menyimpan data kolam yang telah diisi oleh aktor.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan pesan "Gagal Disimpan", bila data tidak berhasil disimpan. 2. Salah satu data pada <i>textbox</i> kosong maka akan terdapat sebuah pesan pada <i>textbox</i> berupa "Data Masih Kosong".
<i>Post-Condition</i>	Sistem berhasil menambahkan data kolam.

2. *Use Case* Skenario Melihat Daftar Kolam

Tabel 4.6 *Use case* skenario melihat daftar kolam

Nama Fungsional	Tambah Kolam
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor saat melihat daftar kolam.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor berada di halaman menu awal sistem.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu kolam. 2. Sistem menampilkan menu kolam, halaman ini menampilkan daftar nama-nama kolam yang sudah ada.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem berhasil menampilkan daftar kolam.



3. Use Case Skenario Menghapus Kolum

Tabel 4.7 Use case scenario menghapus kolum

Nama Fungsional	Menghapus Kolum
Deskripsi	Use case ini mendeskripsikan proses aktor saat menghapus kolum.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu kolum.
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol hapus yang ada pada setiap daftar kolum yang ada. 2. Sistem menampilkan pesan konfigurasi “yakin ingin menghapus?”. 3. Aktor memilih tombol “Yes”. 4. Sistem menghapus kolum yang dipilih oleh aktor.
Alternative Flow	1. Aktor memilih tombol “No” pada pesan konfigurasi yang muncul maka proses menghapus kolum akan dibatalkan.
Post-Condition	Sistem menghapus nama kolum dari <i>database</i> .

4. Use Case Skenario Melihat Detail Kolum

Tabel 4.8 Use case skenario melihat detail kolum

Nama Fungsional	Melihat Detail Kolum
Deskripsi	Use case ini mendeskripsikan proses melihat detail kolum.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu kolum.
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan nama kolum yang diinginkan. 2. Sistem menampilkan data detail kolum.
Alternative Flow	-
Post-Condition	Sistem berhasil menampilkan data detail kolum.

5. Use Case Skenario Mengubah Detail Kolum

Tabel 4.9 Use case skenario mengubah detail kolum

Nama Fungsional	Mengubah Detail Kolum
Deskripsi	Use case ini mendeskripsikan proses penyuntingan kolum.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu kolum.

Tabel 4.8 *Use case* skenario mengubah detail kolam (lanjutan)

<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>edit</i> pada nama kolam yang diinginkan. 2. Sistem menampilkan menu edit kolam, halaman ini berisi <i>form</i> penyuntingan kolam. 3. Aktor mengisi <i>form</i> dan menekan tombol "simpan". 4. Sistem menyimpan data ke dalam <i>database</i> dan menampilkan pesan "Berhasil Disimpan", bila data berhasil disimpan.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan pesan "Gagal Disimpan", bila data tidak berhasil disimpan. 2. Salah satu data pada <i>textbox</i> kosong maka akan terdapat sebuah pesan pada <i>textbox</i> berupa "Data Masih Kosong".
<i>Post-Condition</i>	Sistem berhasil menambahkan hasil penyuntingan ke dalam <i>database</i> .

6. *Use Case* Skenario Menambah Persiapan BudidayaTabel 4.10 *Use case* skenario menambah persiapan budidaya

Nama Fungsional	Menambah Persiapan Budidaya
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penambahan Persiapan budidaya.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu persiapan budidaya.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol tambah persiapan budidaya. 2. Sistem menampilkan menu tambah persiapan budidaya, yang berisi <i>form</i> untuk memilih kolam. 3. Aktor kolom dan menekan tombol "simpan". 4. Sistem menyimpan data ke dalam <i>database</i> dan menampilkan daftar kegiatan persiapan budidaya.
<i>Alternative Flow</i>	1. Sistem menampilkan pesan "perencanaan tidak berhasil ditambah", bila data tidak berhasil disimpan.
<i>Post-Condition</i>	Sistem berhasil menambahkan daftar persiapan budidaya dan menampilkan pesan "perencanaan berhasil ditambah".



7. Use Case Skenario Melihat Daftar Persiapan Budidaya

Tabel 4.11 Use case skenario melihat daftar persiapan budidaya

Nama Fungsional	Melihat Daftar Persiapan Budidaya
Deskripsi	Use case ini mendeskripsikan proses aktor dalam melihat daftar persiapan budidaya.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam halaman menu awal sistem.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu persiapan budidaya. 2. Sistem menampilkan menu persiapan budidaya, halaman ini menampilkan daftar centang persiapan budidaya.
Alternative Flow	
Post-Condition	Sistem berhasil menampilkan daftar centang kegiatan persiapan budidaya.

8. Use Case Skenario Menandai Kegiatan Persiapan Budidaya

Tabel 4.12 Use case skenario menandai kegiatan persiapan budidaya

Nama Fungsional	Menandai Kegiatan Persiapan Budidaya
Deskripsi	Use case ini mendeskripsikan proses aktor dalam menandai kegiatan persiapan budidaya yang telah dilakukan.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam halaman persiapan budidaya dan telah menambahkan persiapan budidaya.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih <i>checklist</i> kegiatan. 2. Sistem menambah simbol penanda pada <i>ceklis</i> kegiatan yang dipilih.
Alternative Flow	-
Post-Condition	Sistem berhasil menambah simbol penanda pada <i>ceklis</i> kegiatan yang dipilih.



9. Use Case Skenario Menghapus Kegiatan Persiapan Budidaya

Tabel 4.13 Use case skenarni menghapus kegiatan persiapan budidaya

Nama Fungsional	Menghapus Kegiatan Persiapan Budidaya
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor saat menghapus kegiatan persiapan budidaya.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu tambah pembibitan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menambahkan data pembibitan. 2. Sistem menghapus data persiapan budidaya secara otomatis pada kolom yang sama, bila pengguna sudah memasukkan data pembibitan.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem berhasil menghapus data kegiatan persiapan budidaya pada kolom yang sama dari <i>database</i> .

10. Use Case Skenario Mencatat Penebaran Bibit

Tabel 4.14 Use case skenario mencatat penebaran bibit

Nama Fungsional	Mencatat Penebaran Bibit
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor dalam memasukkan data penebaran bibit.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam halaman kalender kegiatan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu tebar bibit. 2. Sistem menampilkan menu tebar bibit, halaman ini menampilkan <i>form</i> untuk memasukkan data penebaran bibit. 3. Aktor mengisi <i>form</i> dan menekan tombol simpan. 4. Sistem menyimpan data tebar bibit ke dalam <i>database</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Sistem nemanpilkan pesan “penebaran tidak berhasil ditambah” dan data tidak berhasil disimpan. 2. Salah satu data pada <i>textbox</i> kosong maka akan terdapat sebuah pesan pada <i>textbox</i> berupa “Data Masih Kosong”.
<i>Post-Condition</i>	Sistem menampilkan pesan “penebaran berhasil ditambah” dan berhasil menambahkan data penebaran bibit ke dalam <i>database</i> .



11. Use Case Skenario Melihat Daftar Pembibitan

Tabel 4.15 Use case skenario melihat daftar pembibitan

Nama Fungsional	Melihat Daftar Pembibitan
Deskripsi	Use case ini mendeskripsikan proses melihat daftar pembibitan.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu utama sistem.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu data pembibitan. 2. Sistem menampilkan daftar pembibitan.
Alternative Flow	-
Post-Condition	Sistem berhasil menampilkan daftar pembibitan.

12. Use Case Skenario Melihat Detail Pembibitan

Tabel 4.16 Use case skenario melihat detail pembibitan

Nama Fungsional	Melihat Detail Pembibitan
Deskripsi	Use case ini mendeskripsikan proses melihat detail pembibitan.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu data pembibitan.
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol detail pada daftar pembibitan yang diinginkan. 2. Sistem menampilkan detail pembibitan.
Alternative Flow	-
Post-Condition	Sistem berhasil menampilkan detail pembibitan.



13. Use Case Skenario Menghapus Kegiatan Harian

Tabel 4.17 Use case skenario mengubah detail pembibitan

Nama Fungsional	Menghapus Kegiatan Harian
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penghapusan data kegiatan harian
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu tambah persiapan budidaya
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menambahkan kegiatan persiapan budidaya. 2. Sistem menghapus semua kegiatan harian secara otomatis yang ada pada kolam yang sama, bila pengguna berhasil menyimpan kegiatan persiapan budidaya.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem berhasil menghapus semua kegiatan harian dan data pembibitan yang ada pada kolam yang sama dari <i>database</i> .

14. Use Case Skenario Menambahkan Kegiatan Harian

Tabel 4.18 Use case skenario menambahkan kegiatan harian

Nama Fungsional	Menambahkan Kegiatan Harian
Deskripsi	<i>Use case</i> ini mendeskripsikan proses sistem saat menambahkan kegiatan harian secara otomatis ke dalam kalender kegiatan.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah memasukkan ke dalam menu tambah penebaran bibit.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menambah data tebar bibit. 2. Sistem menambahkan kegiatan harian secara otomatis sesuai dengan data penebaran bibit dan menampilkan ke dalam kalender kegiatan.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem berhasil menampilkan kegiatan harian sesuai data penebaran bibit.



15. Use Case Skenario Melihat Kalender Kegiatan Budidaya

Tabel 4.19 Use case skenario melihat kalender kegiatan budidaya

Nama Fungsional	Melihat Kegiatan Budidaya
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor dalam melihat kegiatan harian budidaya ikan kerapu.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam kegiatan budidaya dan telah memasukkan data penebaran bibit.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih tanggal yang ingin dilihat. 2. Sistem menampilkan menu kegiatan budidaya dan menampilkan kegiatan harian pada tanggal yang dipilih sesuai dengan data penebaran bibit.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem berhasil menampilkan daftar kegiatan harian sesuai data penebaran bibit.

16. Use Case Skenario Menambah Kegiatan Budidaya Lain

Tabel 4.20 Use case skenario menambah kegiatan

Nama Fungsional	Menambah Kegiatan
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor dalam menambahkan kegiatan yang diinginkan untuk proses budidaya.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu kegiatan budidaya.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah kegiatan. 2. Sistem menampilkan menu tambah kegiatan yang berisi <i>form</i> penambahan kegiatan. 3. Aktor mengisi <i>form</i> dan memilih tombol simpan. 4. Sistem menambahkan data kegiatan ke dalam <i>database</i> dan menampilkan ke dalam kalender kegiatan.
<i>Alternative Flow</i>	1. Salah satu data pada <i>textbox</i> kosong maka akan terdapat sebuah pesan pada <i>textbox</i> berupa "Data Masih Kosong".
<i>Post-Condition</i>	Sistem berhasil menambahkan kegiatan ke dalam kalender harian.



17. Use Case Skenario Menambah Kegiatan Panen

Tabel 4.21 Use case skenario menambah kegiatan panen

Nama Fungsional	Menambah Kegiatan Panen
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor dalam menambahkan kegiatan pada saat panen.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu panen.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah kegiatan panen. 2. Sistem menampilkan menu tambah kegiatan panen yang berisi <i>form</i> penambahan kegiatan panen. 3. Aktor mengisi <i>form</i> dan memilih tombol simpan. 4. Sistem menambahkan data kegiatan ke dalam <i>database</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan pesan “tidak berhasil ditambah”, bila data tidak berhasil disimpan. 2. Salah satu data pada <i>textbox</i> kosong maka akan terdapat sebuah pesan pada <i>textbox</i> berupa “Data Masih Kosong”.
<i>Post-Condition</i>	Sistem berhasil menambahkan kegiatan panen dan menampilkan pesan “berhasil ditambah”.

18. Use Case Skenario Melihat Daftar Panen

Tabel 4.22 Use case skenario melihat daftar panen

Nama Fungsional	Melihat Daftar Panen
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor dalam melihat daftar panen
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu utama sistem
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu tanggal panen 2. Sistem menampilkan menu tanggal panen yang berisi daftar panen.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem berhasil menampilkan daftar panen.



19. Use Case Skenario Melihat Detail Kegiatan Panen

Tabel 4.23 Use case skenario melihat detail kegiatan panen

Nama Fungsional	Melihat Detail Kegiatan Panen
Deskripsi	<i>Use case</i> ini mendeskripsikan proses melihat detail kegiatan panen.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu panen.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol detail pada daftar panen yang diinginkan. 2. Sistem menampilkan daftar kegiatan panen.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem berhasil menampilkan daftar kegiatan panen.

20. Use Case Skenario Menghapus Detail Panen

Tabel 4.24 Use case skenario menghapus detail panen

Nama Fungsional	Menghapus Detail Panen
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor saat menghapus detail panen.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu panen.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol hapus yang ada pada setiap daftar panen yang ada. 2. Sistem menampilkan pesan konfigurasi "yakin ingin menghapus?". 3. Aktor memilih tombol "Ya". 4. Sistem menghapus kolom yang dipilih oleh aktor.
<i>Alternative Flow</i>	1. Aktor memilih tombol "Tidak" pada pesan konfigurasi yang muncul maka proses menghapus data kegiatan panen akan dibatalkan.
<i>Post-Condition</i>	Sistem menghapus data kegiatan panen dari <i>database</i> .

21. Use Case Skenario Menambah Biaya Pengeluaran

Tabel 4.25 Use case skenario menambah biaya pengeluaran

Nama Fungsional	Menambah Pengeluaran Biaya
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor dalam menambahkan biaya pengeluaran.



Tabel 4.25 Use case skenario menambah biaya pengeluaran (lanjutan)

Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu hasil
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah pengeluaran. 2. Sistem menampilkan menu tambah pengeluaran yang berisi <i>form</i> biaya pengeluaran. 3. Aktor mengisi <i>form</i> dan memilih tombol simpan. 4. Sistem menambahkan data pengeluaran biaya ke dalam <i>database</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Sistem nemanpilkkan pesan “tidak berhasil ditambah”, bila data tidak berhasil disimpan. 2. Salah satu data pada <i>textbox</i> kosong maka akan terdapat sebuah pesan pada <i>textbox</i> berupa “Data Masih Kosong”.
<i>Post-Condition</i>	Sistem berhasil menambahkan biaya pengeluaran dan menampilkan pesan “berhasil ditambah”.

22. Use Case Skenario Menambah Pemasukan Uang

Tabel 4.26 Use case skenario menambah pemasukan uang

Nama Fungsional	Menambah Pemasukan Uang
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor dalam menambahkan Pemasukan Uang.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu hasil.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah pemasukan. 2. Sistem menampilkan menu tambah pemasukan yang berisi <i>form</i> pemasukan uang. 3. Aktor mengisi <i>form</i> banyak ekor dan banyak kiloan, aktor memilih tombol simpan. 4. Sistem menambahkan banyak kiloan dan banyak ekor ke dalam <i>database</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Hanya menambahkan data banyak kiloan ke dalam <i>database</i>, bila aktor hanya mengisi data kiloan. 2. Hanya menambahkan data banyak ekor ke dalam <i>database</i>, bila aktor hanya mengisi data ekor. 3. Salah satu data pada <i>textbox</i> kosong maka akan terdapat sebuah pesan pada <i>textbox</i> berupa “Data Masih Kosong”.
<i>Post-Condition</i>	Sistem berhasil menambahkan banyak kiloan dan banyak ekor ke dalam <i>database</i> .



23. Use Case Skenario Melihat Daftar Keuangan

Tabel 4.27 Use case skenario melihat daftar keuangan

Nama Fungsional	Melihat Daftar Keuangan
Deskripsi	Use case ini mendeskripsikan proses aktor dalam melihat daftar keuangan.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu utama sistem.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu keuangan. 2. Sistem menampilkan menu keuangan dan menampilkan daftar keuangan.
Alternative Flow	-
Post-Condition	Sistem berhasil menampilkan daftar keuangan.

24. Use Case Skenario Melihat Detail Keuangan

Tabel 4.28 Use case skenario melihat detail keuangan

Nama Fungsional	Melihat Detail Keuangan
Deskripsi	Use case ini mendeskripsikan proses aktor dalam melihat detail keuangan.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu keuangan.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih nama kolam di daftar keuangan yang diinginkan. 2. Sistem menampilkan menu detail keuangan berupa daftar pengeluaran, daftar pemasukan, jumlah selisih keuntungan dan status "untung".
Alternative Flow	<ol style="list-style-type: none"> 1. Sistem menampilkan menu detail keuangan berupa daftar pemasukan, jumlah selisih dan status "Untung" bila data pengeluaran kosong. 2. Sistem menampilkan menu detail keuangan berupa daftar pengeluaran, jumlah selisih dan status "Rugi" bila data pemasukan kosong. 3. Sistem menampilkan menu detail keuangan berupa daftar pemasukan, daftar pengeluaran, jumlah selisih dan status "Netral" bila selisih pengeluaran dan pemasukan sama.



Tabel 4.28 Use case skenario melihat detail keuangan (lanjutan)

<i>Alternative Flow</i>	1. Sistem menampilkan menu detail keuangan berupa daftar pemasukan, daftar pengeluaran, jumlah selisih dan status “Rugi” bila selisih pengeluaran lebih besar dari pemasukan.
<i>Post-Condition</i>	Sistem menampilkan menu detail keuangan berupa daftar pengeluaran, daftar pemasukan, jumlah hasil keuntungan dan status “untung”.

25. Use Case Skenario Menghapus Keuangan

Tabel 4.29 Use case skenario menghapus keuangan

Nama Fungsional	Menghapus Keuangan
Deskripsi	Use case ini mendeskripsikan proses aktor saat menghapus keuangan.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam menu hasil.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol hapus yang ada pada setiap daftar keuangan yang ada. 2. Sistem menampilkan pesan konfigurasi “yakin ingin menghapus?”. 3. Aktor memilih tombol “Yes”. 4. Sistem menghapus detail hasil keuangan yang dipilih oleh aktor.
<i>Alternative Flow</i>	1. Aktor memilih tombol “No” pada pesan konfigurasi yang muncul maka proses menghapus keuangan akan dibatalkan.
<i>Post-Condition</i>	Sistem menghapus keuangan dari <i>database</i> .

26. Use Case Skenario Pemberitahuan Kegiatan

Tabel 4.30 Use case skenario pemberitahuan kegiatan

Nama Fungsional	Pemberitahuan Kegiatan
Deskripsi	Use case ini mendeskripsikan proses aktor saat mendapatkan pemberitahuan kegiatan dari sistem.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah menambahkan catatan pembibitan.

Tabel 4.30 *Use case* skenario pemberitahuan kegiatan (lanjutan)

<i>Main Flow</i>	1. Sistem memberikan pemberitahuan kegiatan budidaya tambak ikan kerapu.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem berhasil memberikan pemberitahuan kegiatan budidaya tambak ikan kerapu.

4.4.4 *Use case* skenario Iterasi Kedua

Use case skenario iterasi kedua merupakan proses penjabaran dari setiap *use case* yang ada pada Gambar 4.2. Pada *use case* diagram iterasi kedua terdapat penambahan *use case*, berikut merupakan *use case* skenario dari *use case* diagram iterasi kedua.

1. *Use Case* Skenario Menambah Jadwal Persiapan Budidaya

Tabel 4.31 *Use case* skenario menambah jadwal persiapan budidaya

Nama Fungsional	Menambah Jadwal Persiapan Budidaya
Deskripsi	<i>Use case</i> ini mendeskripsikan proses aktor dalam menambahkan jadwal kegiatan persiapan budidaya pada saat panen.
Aktor	Pengguna
<i>Pre-Condition</i>	Aktor telah masuk ke dalam Persiapan Budiaya.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah jadwal persiapan budidaya. 2. Sistem menampilkan menu tambah jadwal persiapan budidaya yang berisi <i>form</i> penambahan kegiatan panen. 3. Aktor mengisi <i>form</i> dan memilih tombol simpan. 4. Sistem menambahkan data kegiatan ke dalam <i>database</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Sistem nemanpalkan pesan “tidak berhasil ditambah”, bila data tidak berhasil disimpan. 2. Salah satu data pada <i>textbox</i> kosong maka akan terdapat sebuah pesan pada <i>textbox</i> berupa “Data Masih Kosong”.
<i>Post-Condition</i>	Sistem berhasil menyimpan data ke dalam <i>database</i> dan menampilkan pesan “berhasi ditambah”.



2. Use Case Skenario Melihat Jadwal Persiapan Budidaya

Tabel 4.32 Use case skenario melihat jadwal persiapan budidaya

Nama Fungsional	Melihat Jadwal Persiapan Budidaya
Deskripsi	Use case ini mendeskripsikan proses aktor dalam melihat jadwal persiapan budidaya.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu persiapan budidaya.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih jadwal persiapan budidaya. 2. Sistem menampilkan daftar kegiatan persiapan budidaya.
Alternative Flow	-
Post-Condition	Sistem berhasil menampilkan daftar jadwal persiapan budidaya.

3. Use Case Skenario Menghapus Jadwal Persiapan Budidaya

Tabel 4.33 Use case skenario menghapus jadwal persiapan budidaya

Nama Fungsional	Menghapus Jadwal Persiapan Budidaya
Deskripsi	Use case ini mendeskripsikan proses sistem saat menghapus kegiatan persiapan budidaya.
Aktor	Pengguna
Pre-Condition	Aktor telah masuk ke dalam menu tambah pembibitan.
Main Flow	<ol style="list-style-type: none"> 1. Aktor menambahkan data pembibitan. 2. Sistem menghapus data persiapan budidaya secara otomatis pada kolom yang sama, bila pengguna berhasil menambah data pembibitan.
Alternative Flow	-
Post-Condition	Sistem berhasil menghapus data kegiatan persiapan budidaya pada kolom yang sama dari database.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

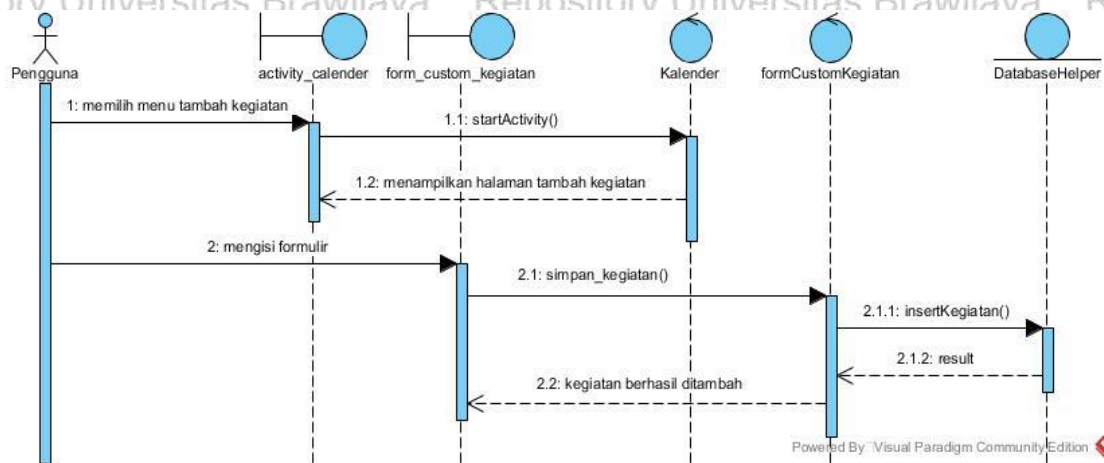
5.1 Perancangan

Perancangan sistem adalah tahapan yang cukup penting, dimana perancangan merupakan sebuah acuan implementasi dalam pengembangan sistem. Dalam pengembangan sistem ini, penulis membuat beberapa representasi dalam perancangan berupa *sequence diagram*, perancangan basis data, *class diagram*, perancangan algoritma, dan perancangan antarmuka.

5.1.1 Sequence Diagram

Pada tahap ini penulis mengambil tiga sampel *Use case* yang akan direpresentasikan ke dalam *sequence diagram*. Tiga sampel *use case* yang digunakan adalah *Use case* menambah kegiatan, melihat kegiatan budidaya dan menandai kegiatan persiapan budidaya. *Sequence diagram* digunakan sebagai penunjuk rangkaian interaksi yang ada pada sistem.

5.1.1.1 Menambah Kegiatan Budidaya Lain

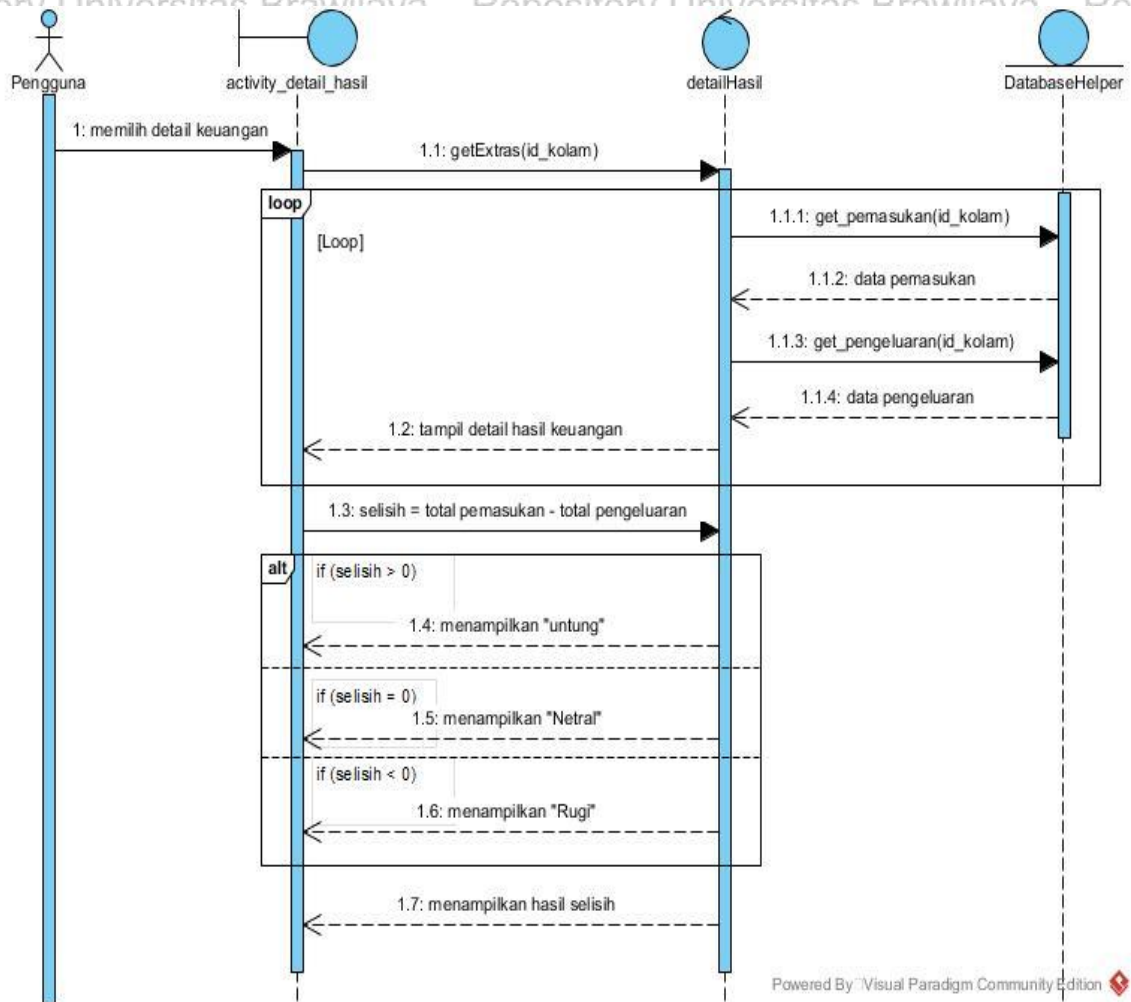


Gambar 5.1 Sequence diagram menambah kegiatan budidaya lain

Pada gambar 5.1 menunjukkan *sequence diagram* dari *use case* menambah kegiatan. Pengguna memilih menu tambah kegiatan, maka *class* Kalender akan menjalankan *event* `startActivity()` untuk menampilkan menu tambah kegiatan. Setelah itu pengguna mengisi formulir penambahan kegiatan dan menekan tombol simpan, maka *class* `formCustomKegiatan` akan menjalankan fungsi `simpan_kegiatan()`, data akan diteruskan ke dalam `DatabaseHelper` dengan menggunakan fungsi `insertKegiatan()`.



5.1.1.2 Melihat Detail Keuangan



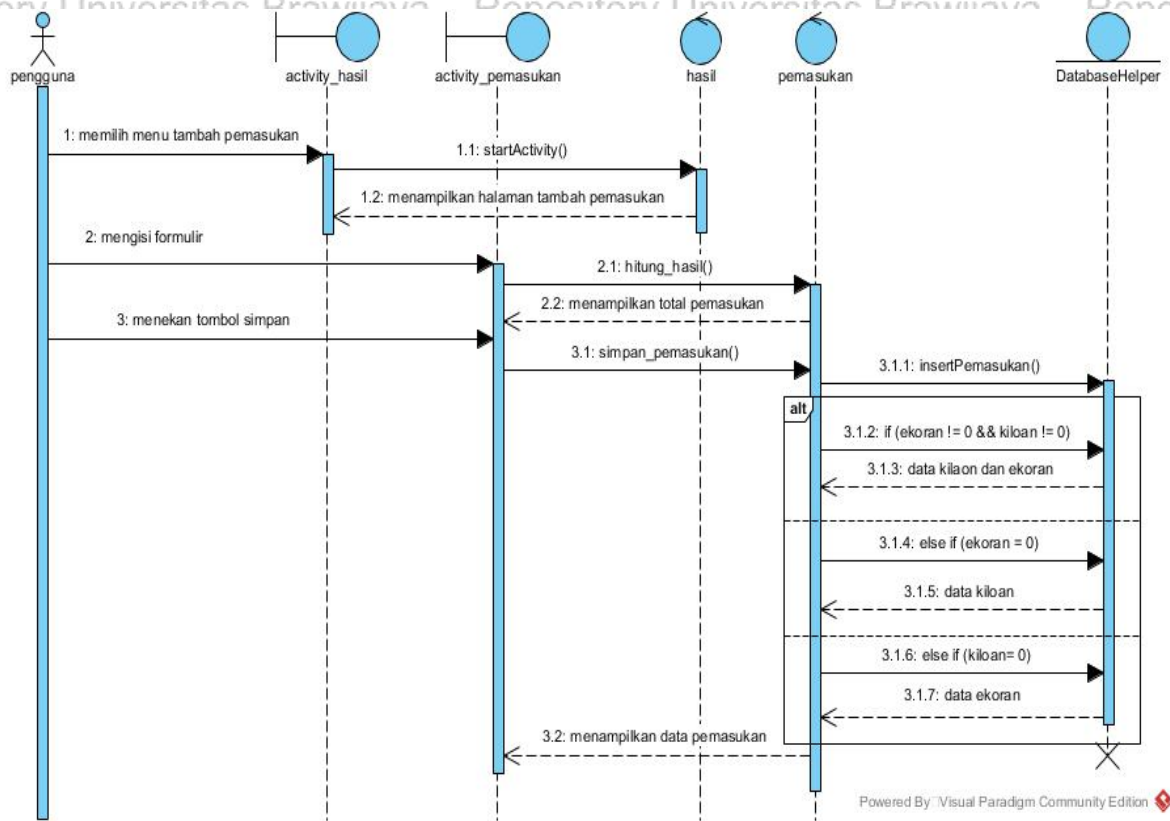
Powered By Visual Paradigm Community Edition

Gambar 5.2 Sequence diagram melihat detail keuangan

Pada gambar 5.2 menunjukkan *sequence diagram* dari *use case* melihat detail keuangan. Pada kegiatan ini pengguna memilih daftar kolom yang ingin dilihat detail keuangannya, maka *class* detailHasil akan menjalankan fungsi `getExtras(id_kolam)`. Maka akan diteruskan ke DatabaseHelper untuk mengambil data pemasukan dan data pengeluaran, dengan menggunakan fungsi `get_pemasukan` dan `get_pengeluaran`. Setelah data diambil maka akan dicari selisih dari hasil perhitungan pemasukan dan pengeluaran. Dari hasil tersebut akan ditampilkan hasil perhitungan selisih dan keterangan untung atau rugi.



5.1.1.3 Menambah Pemasukan Uang



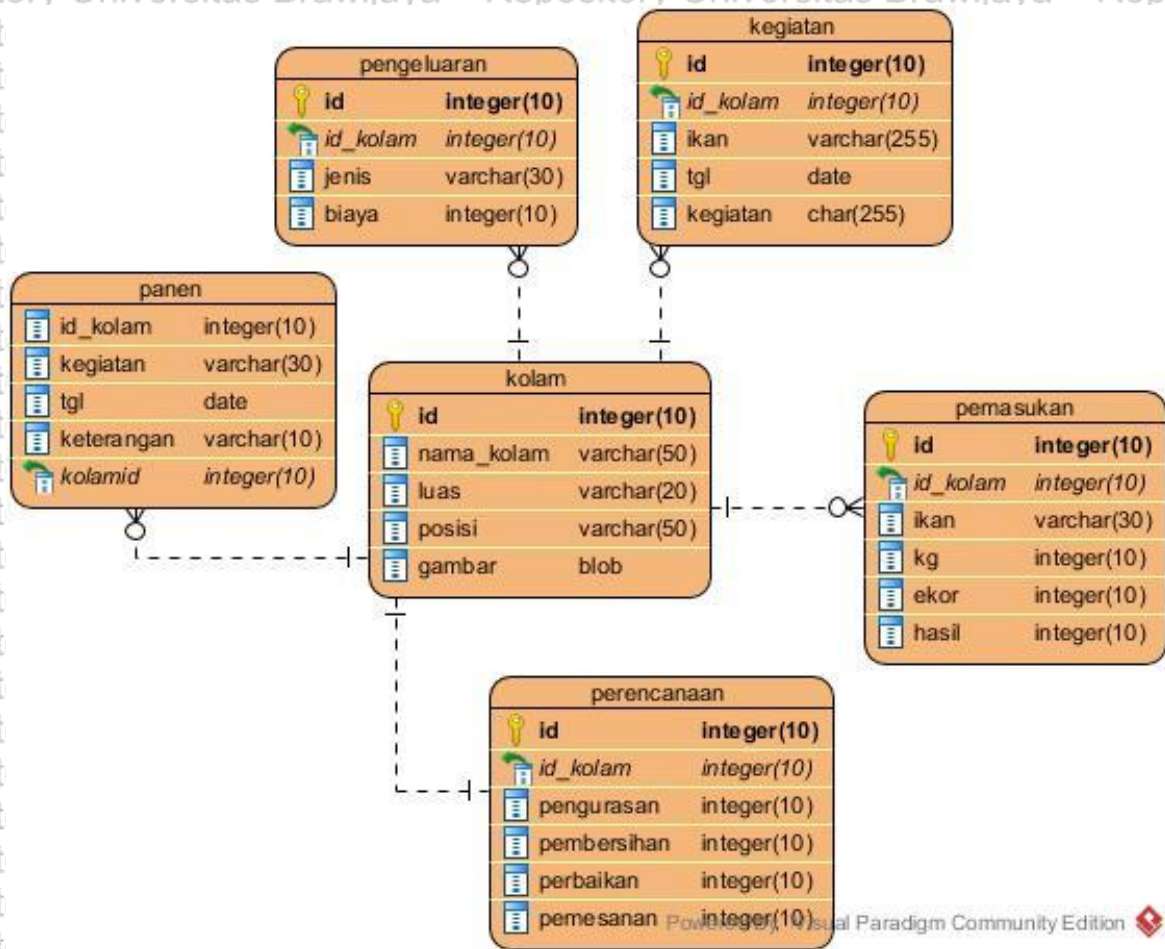
Powered By Visual Paradigm Community Edition

Gambar 5.3 Sequence diagram kegiatan pemasukan uang

Pada gambar 5.3 menunjukkan *sequence diagram* dari *use case* menambah pemasukan uang. Pengguna memilih menu tambah pemasukan, maka *class* hasil akan menjalankan *event* `startActivity()` untuk menjalankan *class* pemasukan. Maka akan muncul layout *activity_pemasukan* yang berisi formulir untuk menambah pemasukan, setelah pengguna mengisi formulir pemasukan maka *class* pemasukan akan menjalankan fungsi `hitung_hasil` dan menampilkan total pemasukan. Pada saat pengguna telah menekan tombol simpan, maka *class* pemasukan akan menjalankan fungsi `simpan_pemasukan`, maka sistem akan memasukkan data penghasilannya dan menampilkannya.

5.1.2 Perancangan Basis Data

Perancangan basis data bertujuan untuk memodelkan relasi dari entitas sistem. perancangan ini dilakukan dengan menggunakan *physical data model*. pemodelan ini dapat merepresentasikan rancangan basis data dan dapat mendefinisikan semua struktur dari tabel basis data, rancangan *physical data model* pada Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) ditunjukkan dengan Gambar 5.4.

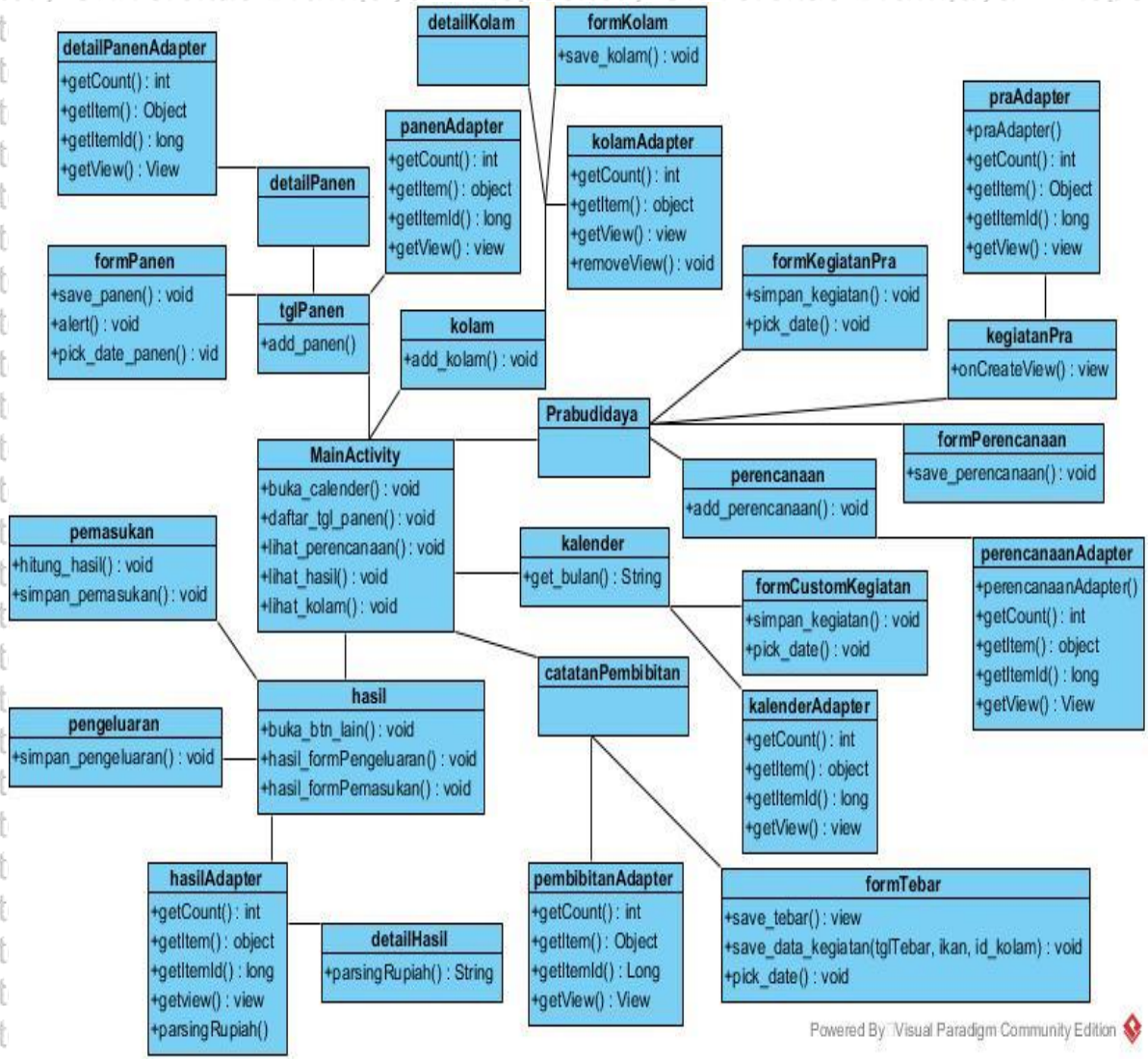


Gambar 5.4 Physical data model

5.1.3 Class Diagram Iterasi Pertama

Class diagram bertujuan untuk mendefinisikan struktur dari class dan hubungan antar class yang digunakan pada pembangunan sistem. pada Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) struktur kelas terdiri dari, MainActivity yang merupakan class untuk menu utama pada sistem, class kalender memiliki turunan berupa class kalenderAdapter, formTebar dan formCustomKegiatan. Sedangkan pada class kolam memiliki class turunan kolamAdapter dan formKolam.

Class panenAdapter merupakan turunan class dari tglPanen, adapun class perencanaan memiliki dua turunan class, yaitu class formPerencanaan dan class perencanaanAdapter. Pada Class hasil memiliki turunan class pengeluaran, pemasukan dan hasilAdapter, untuk class hasilAdapter sendiri juga memiliki turunan class detailHasil. Pada Gambar 5.5 merupakan rancangan dari class diagram iterasi pertama dari sistem.



Powered By Visual Paradigm Community Edition

Gambar 5.6 Class diagram iterasi kedua

5.1.5 Perancangan Algoritme

Pada bagian perancangan algoritme ini, akan menjelaskan perancangan algoritme sistem informasi operasional budidaya tambak ikan kerapu. Perancangan algoritme ini dirancang dengan berdasar pada class diagram yang telah dibuat. Perancangan algoritma ini bertujuan untuk menjabarkan algoritme dari fungsi yang ada di dalam class, pada perancangan algoritme ini dicantumkan tiga algoritme berbeda yang ada dalam sistem.

5.1.5.1 Algoritme Menambah Kegiatan Budidaya Lain

Pada Tabel 5.1 merupakan perancangan algoritme untuk menambahkan kegiatan tambahan dari pengguna sistem. algoritme ini akan diterapkan pada fungsi simpan_kegiatan(), fungsi ini terdapat pada class formCustomKegiatan.



Tabel 5.1 Algoritme menambah kegiatan budidaya lain

No.	Algoritme
1	Mulai
2	Menerima masukan tanggal, kegiatan, kolam, keterangan
3	Memasukkan nilai tgl, Keg, kolam, jet ke dalam database
4	IF (data tersimpan)
5	Mencetak pesan "Berhasil Ditambah"
6	ELSE
7	Mencetak pesan "Tidak Berhasil Ditambah"
8	END IF
9	Selesai

5.1.5.2 Algoritme Melihat Detail Keuangan

Pada Tabel 5.2 merupakan perancangan algoritme untuk melihat detail keuangan. algoritme ini akan diterapkan pada *method* onCreate(), fungsi ini terdapat pada *class* detailHasil.

Tabel 5.2 Algoritme melihat kegiatan budidaya

No.	Algoritme
1	Mulai
2	Ambil id_kolam
3	Ambil pemasukan
4	Ambil pengeluaran
5	FOREACH (pemasukan)
6	Tampil detail pemasukan
7	Tampil Total pemasukan
8	END FOREACH
9	FOREACH (pengeluaran)
10	Tampil detail pengeluaran
11	Tampil total pengeluaran
12	END FOREACH
13	Selisih = total pemasukan - total pengeluaran
14	IF (selisih < 0)
15	Tampil "Untung"
16	ELSE IF (selisih = 0)
17	Tampil "Netral"
18	ELSE
19	Tampil "Rugi"
20	END IF
21	Tampil selisih
22	Selesai

5.1.5.3 Algoritme Menambah Penghasilan Uang

Pada Tabel 5.3 merupakan perancangan algoritme untuk menambah pemasukan uang. algoritme ini akan diterapkan pada fungsi *simpan_pemasukan()*, fungsi ini terdapat pada *class* pemasukan.



Tabel 5.3 Perancangan algoritme menambah penghasilan

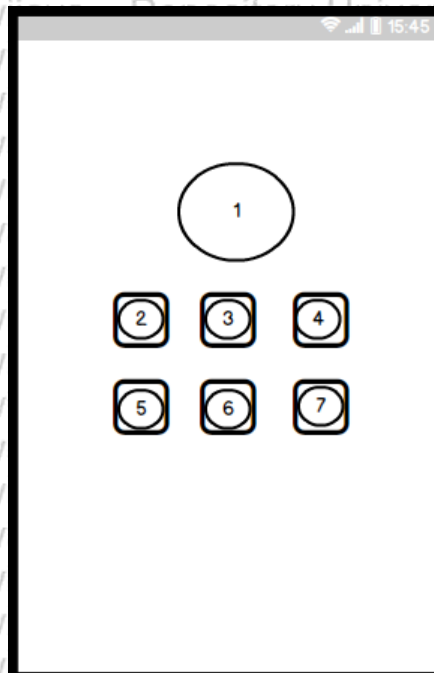
No.	Algoritme
1	Mulai
2	Mengambil data masukan pengguna
3	Hitung total = hasil_kiloan + hasil_ekoran
4	Simpan penghasilan
6	Ambil banyak_ekor
7	Ambil banyakKG
8	IF (banyak_ekor != 0 dan banyakKg != 0)
9	Simpan pemasukan hasil_kiloan dan hasil_ekoran
10	ELSE IF (banyak_ekor = 0)
11	Simpan pemasukan hasil_kiloan
12	ELSE IF (banyakKG = 0)
13	Simpan pemasukan hasil_ekoran
14	END IF
15	Menampilkan detail pemasukan
16	selesai

5.1.6 Perancangan Antarmuka Iterasi Pertama

Pada perancangan antarmuka ini menjelaskan tampilan yang akan diterapkan ke dalam Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK). perancangan antarmuka ini akan dijelaskan menggunakan gambar sebagai berikut.

5.1.6.1 Halaman Menu Utama Sistem

Halaman ini merupakan halaman yang akan ditampilkan saat sistem pertama kali dijalankan. Gambar 5.7 merupakan rancangan dari antarmuka menu utama sistem.





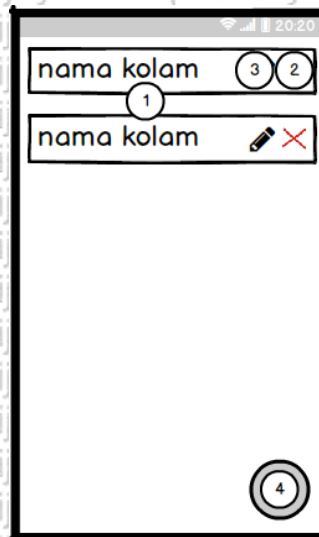
Gambar 5.7 Perancangan antarmuka menu utama sistem

Keterangan rancangan antarmuka :

1. Angka 1 merupakan logo dari Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) berupa gambar.
2. Angka 2 merupakan tombol daftar kolam, tombol ini digunakan untuk menuju menu daftar kolam.
3. Angka 3 merupakan tombol data pembibitan, tombol ini digunakan untuk menuju menu data pencatatan penebaran bibit. Angka 4 merupakan tombol tanggal panen, yang digunakan untuk menuju ke menu tanggal panen.
4. Angka 5 merupakan tombol kegiatan budidaya, tombol ini digunakan untuk menuju menu kalender kegiatan budidaya.
5. Angka 6 merupakan tombol persiapan budidaya, tombol ini digunakan untuk menuju menu persiapan budidaya.
6. Angka 7 merupakan tombol keuangan, yang digunakan untuk menuju ke dalam menu keuangan.

5.1.6.2 Halaman Daftar Kolam

Halaman ini merupakan halaman yang akan ditampilkan saat pengguna menekan tombol kegiatan budidaya pada menu utama. Gambar 5.8 merupakan rancangan dari antarmuka menu daftar kolam.



Gambar 5.8 Perancangan antarmuka daftar kolam

Keterangan :

1. Angka 1 merupakan daftar nama kolam.
2. Angka 2 merupakan tombol untuk menghapus kolam yang diinginkan.
3. Angka 3 merupakan tombol *edit*, yang digunakan untuk mengubah nama kolam yang diinginkan.
4. Angka 4 merupakan tombol untuk menuju menu tambah kolam.



5.1.6.3 Halaman *Form* Tambah

Halaman ini merupakan halaman yang akan ditampilkan pada saat pengguna menekan tombol tambah kolom, tambah kegiatan budidaya, tambah pembibitan, tambah persiapan budidaya dan pengeluaran. Gambar 5.9 merupakan rancangan dari antarmuka menu *form* tambah

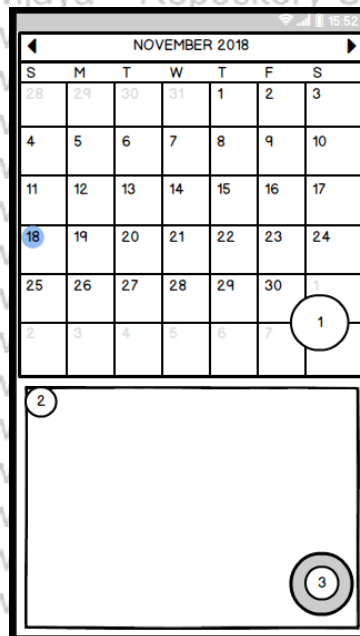
Gambar 5.9 Perancangan antarmuka *form* tambah

Keterangan :

1. Angka 1 merupakan *form* input.
2. Angka 2 merupakan tombol untuk menyimpan nama kolam.
3. Angka 3 merupakan nama dari halaman.

5.1.6.4 Halaman Kalender Kegiatan Budidaya

Halaman ini merupakan halaman yang akan ditampilkan setelah pengguna menekan tombol kegiatan budidaya yang ada di dalam menu utama sistem. Gambar 5.10 merupakan rancangan dari antarmuka menu kegiatan budidaya.



Gambar 5.10 Perancangan antarmuka kalender kegiatan budidaya

Keterangan :

1. Angka 1 merupakan tampilan kalender untuk melihat kegiatan harian budidaya.
2. Angka 2 merupakan bagian yang berisi daftar kegiatan budidaya.
3. Angka 3 merupakan tombol untuk menambahkan kegiatan.

5.1.6.5 Halaman Data Pembibitan

Halaman ini merupakan halaman yang akan ditampilkan setelah pengguna menekan tombol tambah data pembibitan yang ada di halaman utama sistem. Gambar 5.11 merupakan rancangan dari antarmuka menu data pembibitan.

20-01-2018	2	Nama Kolam
Jenis Ikan		
Ukuran Bibit		
Banyak Bibit	1	
20-01-2018		Nama Kolam
20-01-2018		Nama Kolam



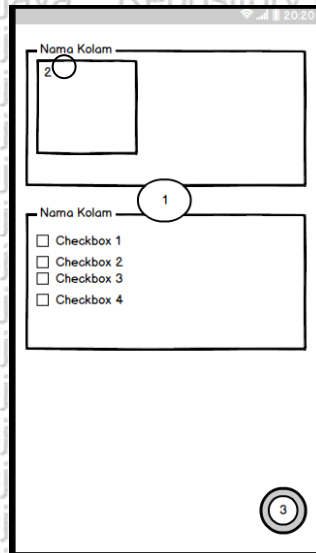
Gambar 5.11 Perancangan antarmuka data pembibitan

Keterangan :

1. Angka 1 merupakan *list* daftar data pembibitan.
2. Angka 2 merupakan rincian data pembibitan yang akan ditambahkan bila salah satu *list* data pembibitan dipilih. Rincian tersebut terdiri dari tanggal pembibitan yang berada di pojok kiri atas, nama kolam yang berada di pojok kanan atas, sedangkan untuk rincian data pembibitan yang lain berada di bawah tanggal pembibitan.

5.1.6.6 Halaman Persiapan Budidaya Iterasi Pertama

Halaman ini merupakan halaman yang akan ditampilkan setelah pengguna menekan tombol persiapan budidaya yang ada di halaman utama sistem. Gambar 5.12 merupakan rancangan dari antarmuka halaman persiapan budidaya.



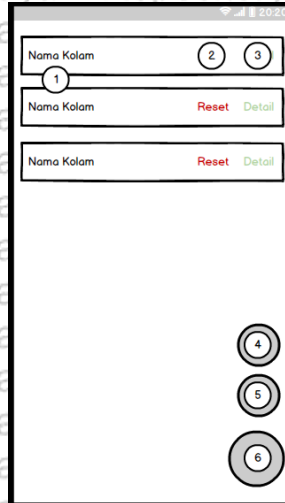
Gambar 5.12 Perancangan antarmuka persiapan budidaya iterasi pertama

Keterangan :

1. Angka 1 merupakan bagian antarmuka yang berisi daftar persiapan budidaya.
2. Angka 2 merupakan daftar *checkbox* kegiatan persiapan budidaya.
3. Angka 3 merupakan tombol tambah persiapan budidaya yang bertujuan untuk masuk ke dalam halaman tambah persiapan budidaya.

5.1.6.7 Halaman Keuangan

Halaman ini merupakan halaman yang akan ditampilkan setelah pengguna menekan tombol keuangan yang ada pada halaman utama sistem. halaman ini juga digunakan pada halaman daftar panen. Gambar 5.13 merupakan rancangan dari antarmuka dari halaman keuangan.



Gambar 5.13 Perancangan antarmuka keuangan

Keterangan :

1. Angka 1 adalah tampilan dari daftar keuangan.
2. Angka 2 merupakan tombol hapus yang digunakan untuk menghapus data keuangan yang dipilih.
3. Angka 3 adalah tombol detail yang digunakan untuk masuk ke dalam halaman detail keuangan yang dipilih.
4. Angka 4 merupakan tombol tambah pemasukan keuangan yang digunakan untuk berpindah menuju halaman tambah pemasukan.
5. Angka 5 merupakan tombol tambah pengeluaran keunagn yang digunakan untuk berpindah menuju halaman tambah pengeluaran.
6. Angka 6 merupakan tombol untuk memunculkan tombol tambah pengeluaran dan tombol pemasukan.

5.1.6.8 Halaman Detail Keuangan

Halaman ini merupakan halaman yang akan ditampilkan setelah pengguna menekan tombol detail yang ada di halaman keuangan. Gambar 5.14 merupakan rancangan dari antarmuka halaman detail keuangan.



Pengeluaran		
1	Keterai	Rp. 200
2	Keterai	Rp. 200
Pemasukan		
1	Jenis Ikan 00	Rp. 000
2	Jenis Ikan 00	Rp. 000
3 Untung		Rp. 100000

Gambar 5.14 Perancangan antarmuka detail keuangan

Keterangan :

1. Angka 1 merupakan nama kolom.
2. Angka 2 merupakan detail dari pemasukan dan pengeluaran keuangan.
3. Angka 3 merupakan total perhitungan keuangan dan status untung atau rugi.

5.1.6.9 Halaman Tambah Pemasukan Keuangan

Halaman ini merupakan halaman yang akan ditampilkan setelah pengguna menekan tombol tambah pemasukan keuangan yang ada pada halaman keuangan. Gambar 5.15 merupakan rancangan dari antarmuka halaman tambah pemasukan keuangan.

Nama Kolam		
Nama Kolam		
Kiluan :		
Jenis	Banyak	Hasil
Jenis ikan	<input type="text"/>	<input type="text"/>
Ekor :		
Jenis	Banyak	Hasil
Jenis ikan	<input type="text"/>	<input type="text"/>
Total		00000
<input type="text"/>		2

Gambar 5.15 Perancangan antarmuka pemasukan keuangan

Keterangan :

1. Angka 1 merupakan formulir untuk memasukkan detail pemasukan keuangan.
2. Angka 2 merupakan tombol simpan.



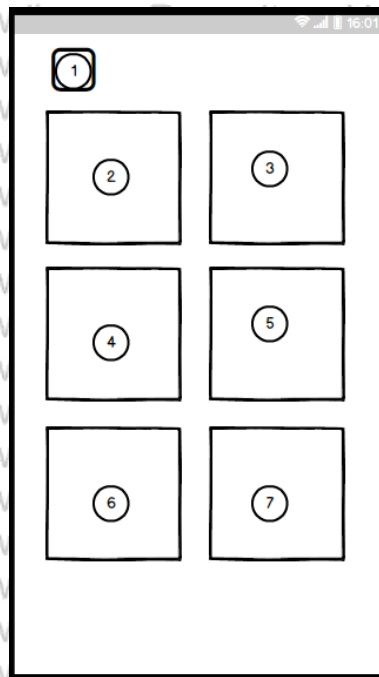
- Angka 3 merupakan jumlah perhitungan pemasukan keuangan.

5.1.7 Perancangan Antarmuka Iterasi Kedua

Pada perancangan antarmuka iterasi kedua ini didapat perubahan perancangan tampilan antarmuka, perubahan ini didasari dari hasil evaluasi dari proses iterasi pertama yang telah dilakukan untuk memperbaiki antarmuka sistem.

5.1.7.1 Halaman Menu Utama Sistem Iterasi Kedua

Halaman ini merupakan halaman utama sistem, pada halaman ini terdapat perubahan pada logo dan tombol menu sistem. Gambar 5.16 merupakan rancangan dari antarmuka halaman utama iterasi kedua.



Gambar 5.16 Perancangan antarmuka menu utama iterasi kedua

Keterangan rancangan antarmuka :

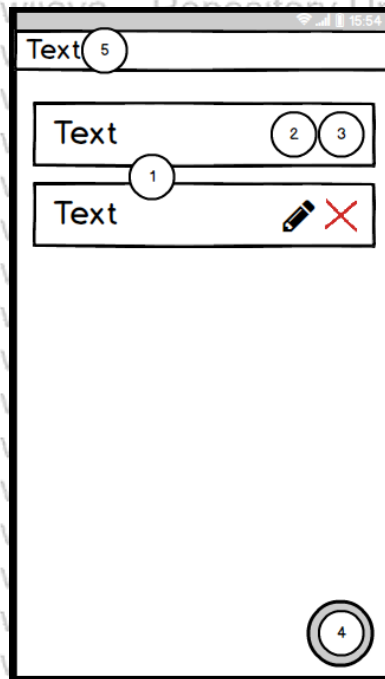
- Angka 1 merupakan logo dari Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) berupa gambar.
- Angka 2 merupakan tombol daftar kolam, tombol ini digunakan untuk menuju menu daftar kolam.
- Angka 3 merupakan tombol persiapan budidaya, tombol ini digunakan untuk menuju menu persiapan budidaya.
- Angka 4 merupakan tombol data pembibitan, tombol ini digunakan untuk menuju menu data pencatatan penebaran bibit.
- Angka 5 merupakan tombol kegiatan budidaya, tombol ini digunakan untuk menuju menu kalender kegiatan budidaya.
- Angka 6 merupakan tombol kegiatan tanggal panen, yang digunakan untuk menuju ke menu tanggal panen.



- Angka 7 merupakan tombol keuangan, yang digunakan untuk menuju ke dalam menu keuangan.

5.1.7.2 Halaman Daftar Kolam Iterasi Kedua

Halaman ini merupakan halaman yang akan ditampilkan setelah pengguna menekan tombol kolam. Pada halaman ini terdapat perubahan pada judul halaman. Gambar 5.17 merupakan rancangan dari antarmuka halaman daftar kolam iterasi kedua.



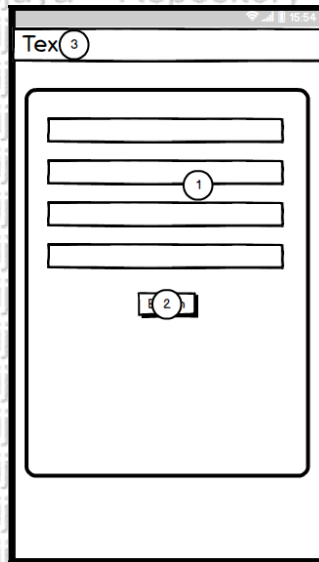
Gambar 5.17 Perancangan antarmuka daftar kolam iterasi kedua

Keterangan :

- Angka 1 merupakan daftar nama kolam.
- Angka 2 merupakan tombol untuk menghapus kolam yang diinginkan.
- Angka 3 merupakan tombol *edit*, yang digunakan untuk mengubah nama kolam yang diinginkan.
- Angka 4 merupakan tombol untuk menuju menu tambah kolam.
- Angka 5 merupakan judul dari halaman menu.

5.1.7.3 Halaman *Form* Tambah iterasi kedua

Pada halaman ini merupakan perancangan antarmuka iterasi kedua dari semua halaman *form* yang ada pada sistem. Gambar 5.18 merupakan perancangan antarmuka iterasi kedua dari halaman *form* Tambah.



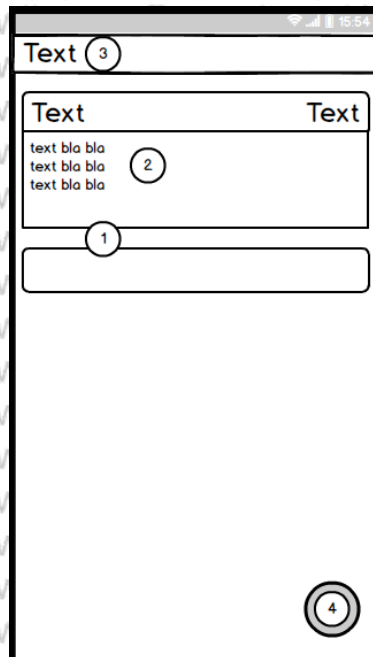
Gambar 5.18 Perancangan antarmuka halaman *form* tambah iterasi kedua

Keterangan :

1. Angka 1 merupakan *form* input.
2. Angka 2 merupakan tombol untuk menyimpan nama kolom.
3. Angka 3 merupakan nama dari halaman.

5.1.7.4 Halaman Data Pembibitan Iterasi Kedua

Halaman ini merupakan halaman yang akan dimunculkan bila pengguna menekan tombol pembibitan pada halaman utama. Pada Gambar 5.19 merupakan perancangan antarmuka dari halaman data pembibitan iterasi kedua.

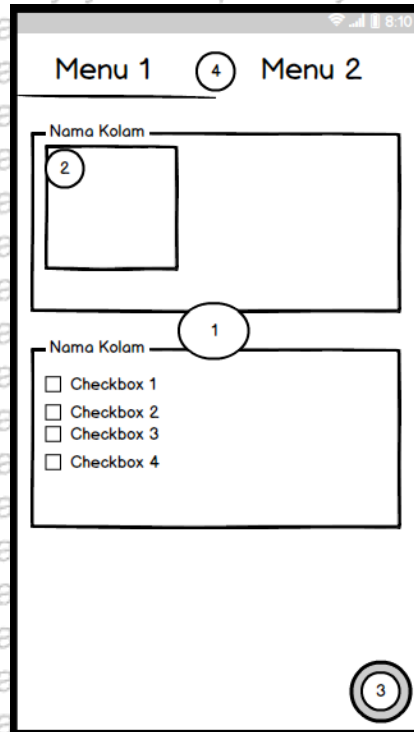


Gambar 5.19 Perancangan antarmuka data pembibitan iterasi kedua



5.1.7.5 Halaman Persiapan Budidaya Iterasi Kedua

Halaman ini merupakan halaman yang akan ditampilkan setelah pengguna menekan tombol persiapan budidaya yang ada di halaman utama sistem, pada halaman ini merupakan perubahan dari halaman persiapan budidaya iterasi kedua. Gambar 5.20 merupakan rancangan dari antarmuka halaman persiapan budidaya.



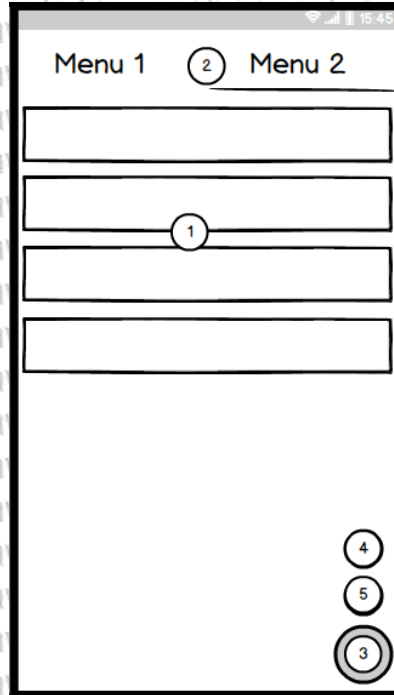
Gambar 5.20 Perancangan antarmuka persiapan budidaya iterasi kedua

Keterangan:

1. Angka 1 merupakan bagian antarmuka yang berisi daftar persiapan budidaya.
2. Angka 2 merupakan daftar *checkbox* kegiatan persiapan budidaya.
3. Angka 3 merupakan tombol tambah persiapan budidaya yang bertujuan untuk masuk ke dalam halaman tambah persiapan budidaya.
4. Angka 4 merupakan *option menu*.

5.1.7.6 Halaman Jadwal Persiapan Budidaya

Halaman ini akan ditampilkan pada saat pengguna masuk ke dalam menu persiapan budidaya dan menekan *submenu* kedua. Pada menu ini merupakan penambahan antarmuka iterasi kedua. Rancangan antarmuka dapat dilihat pada Gambar 5.21.



Gambar 5.21 Perancangan antarmuka jadwal persiapan budidaya

Keterangan:

1. Angka 1 merupakan bagian antarmuka yang berisi daftar jadwal persiapan budidaya.
2. Angka 2 merupakan tombol untuk berpindah ke *submenu* yang lain
3. Angka 3 merupakan tombol untuk menampilkan tombol tambah kegiatan.
4. Angka 4 dan 5 merupakan tombol untuk menambah kegiatan persiapan budidaya.

5.2 Implementasi

Pada sub bab implementasi akan membahas bagaimana Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK) diimplementasikan berdasarkan metodologi dan perancangan. Implementasi ini akan menjabarkan spesifikasi sistem, implementasi algoritme dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Dalam mengimplementasikan sistem yang telah dirancang, dibutuhkan perangkat keras dan juga perangkat lunak. Spesifikasi perangkat keras dan perangkat lunak sendiri bertujuan untuk mengetahui spesifikasi standar dalam pembuatan aplikasi.

5.2.1.1 Spesifikasi Perangkat Keras

Dalam pengembangan sistem ini diimplementasikan dengan menggunakan perangkat berupa komputer dan juga *smartphone*. Perangkat komputer digunakan untuk pembuatan kode program, sedangkan perangkat *smartphone*



digunakan sebagai media pengujian sistem yang telah dibuat. Pada Tabel 5.4 menunjukkan spesifikasi perangkat keras komputer.

Tabel 5.4 Spesifikasi perangkat keras komputer

Nama Komponen	Spesifikasi
Model	Asus GL551VW
Processor	Intel Core i7-6700HQ 2.6GHz
Memory	8GB RAM, 1TB HDD
Display	Nvidia GeForce GTX 960M 4GB

Tabel 5.5 menunjukkan spesifikasi perangkat keras *smartphone* yang digunakan.

Tabel 5.5 Spesifikasi perangkat keras *smartphone*

Nama Komponen	Spesifikasi
Model	ASUS_X00TD
Processor	Qualcomm SDM636 Snapdragon 636, Octa-core 1.8 GHz Kryo 260
Memory	4GB RAM
Display	6 inci, 2.160 x 1.080 piksel, aspect ratio 18:9

5.2.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan oleh penulis dalam pengembangan sistem. Tabel 5.6 merupakan spesifikasi perangkat lunak pada komputer.

Tabel 5.6 Spesifikasi perangkat lunak komputer

Nama Komponen	Spesifikasi
Operating System	Windows 10 64-bit
Programming Language	Java
Programming Environment	Android Studio
Database Management System	SQLite

Pada Tabel 5.7 merupakan spesifikasi perangkat lunak pada *Smartphone*.

Tabel 5.7 spesifikasi perangkat lunak *Smartphone*

Nama Komponen	Spesifikasi
Operating System	Android Oreo 8.1



5.2.2 Implementasi Basis Data

Pada implementasi basis data akan menerapkan perancangan basis data yang telah dibuat sebelumnya. Seperti yang ditunjukkan pada Tabel 5.8.

Tabel 5.8 Implementasi basis data Iterasi Pertama

No.	Data Definition Language
1	<code>public static final String db_name ="budidaya";</code>
2	<code>public static final String tbl_kegiatan="kegiatan";</code>
3	<code>public static final String tbl_perencanaan="perencanaan";</code>
4	<code>public static final String tbl_pemasukan="pemasukan";</code>
5	<code>public static final String tbl_pengeluaran="pengeluaran";</code>
6	<code>public static final String tbl_kolam="kolam";</code>
7	<code>public static final String tbl_panen="panen";</code>
8	
9	<code>public void onCreate(SQLiteDatabase db) {</code>
10	<code>db.execSQL("create table "+tbl_kegiatan+" (id INTEGER</code>
11	<code>PRIMARY KEY AUTOINCREMENT, id_kolam integer, ikan text,</code>
12	<code>tgl DATE, kegiatan TEXT)");</code>
13	
14	<code>db.execSQL("create table "+tbl_kolam+" (id INTEGER</code>
15	<code>PRIMARY KEY AUTOINCREMENT, nama_kolam varchar(50), luas</code>
16	<code>varchar(30), posisi varchar(50), gambar blob)");</code>
17	
18	<code>db.execSQL("create table "+tbl_pemasukan+" (id INTEGER</code>
19	<code>PRIMARY KEY AUTOINCREMENT, id_kolam integer, ikan</code>
20	<code>varchar(30), kg integer, ekor integer, hasil integer)");</code>
21	
22	<code>db.execSQL("create table "+tbl_pengeluaran+" (id</code>
23	<code>INTEGER PRIMARY KEY AUTOINCREMENT, id_kolam integer, jenis</code>
24	<code>varchar(30), biaya integer)");</code>
25	
26	<code>db.execSQL("create table "+tbl_perencanaan+" (id</code>
27	<code>INTEGER PRIMARY KEY AUTOINCREMENT, id_kolam integer,</code>
28	<code>pengurusan integer(10), pembersihan integer(10), perbaikan</code>
29	<code>integer(10), pemesanan integer(10))");</code>
30	
31	<code>db.execSQL("create table "+tbl_panen+" (id INTEGER</code>
32	<code>PRIMARY KEY AUTOINCREMENT, id_kolam integer, kegiatan</code>
33	<code>varchar(30), tgl DATE, keterangan TEXT)");</code>
34	<code>}</code>

Penjelasan:

Pada baris 1 merupakan *syntax* untuk deklarasi pembuatan *database*.

Pada baris 2-7 merupakan deklarasi variabel untuk nama tabel.

Pada baris 10-12 merupakan deklarasi pembuatan tabel kegiatan.

Pada baris 14-16 merupakan deklarasi pembuatan tabel kolam.

Pada baris 18-20 merupakan deklarasi pembuatan tabel pemasukan.

Pada baris 22-24 merupakan deklarasi pembuatan tabel pengeluaran.

Pada baris 26-29 merupakan deklarasi pembuatan tabel perencanaan.

Pada baris 31-33 merupakan deklarasi pembuatan tabel panen.



5.2.3 Implementasi Algoritme

Pada implementasi algoritme, perancangan algoritme yang telah dibuat akan diubah menjadi kode program yang dimengerti oleh komputer. Kode program ini dibuat dengan menggunakan bahasa java. Berikut merupakan tiga sampel dari kode program.

5.2.3.1 Implementasi Algoritme Menambah Kegiatan Budidaya Lain

Tabel 5.9 Merupakan implementasi kode program dari algoritme menambah kegiatan budidaya lain. Implementasi algoritme tersebut terdapat pada *class* formCustomKegiatan, algoritme yang diterapkan terdapat pada *method* onCreate() yang digunakan untuk mengambil semua masukan dari pengguna, sedangkan pada *method* simpanKegiatan() digunakan untuk menyimpan data dari pengguna ke dalam *database* dengan menggunakan fungsi insertKegiatan().

Tabel 5.9 Implementasi algoritme menambah kegiatan budidaya lain

No.	Kode Program
1	public class formCustomKegiatan extends AppCompatActivity
2	{
3	private EditText tgl, ket;
4	Spinner kolam, keg;
5	private SimpleDateFormat dateFormat;
6	private Calendar choosen_date= Calendar.getInstance();
7	
8	@Override
9	protected void onCreate(Bundle savedInstanceState) {
10	super.onCreate(savedInstanceState);
11	setContentView(R.layout.form_custom_kegiatan);
12	tgl =(EditText) findViewById(R.id.tgl);
13	keg =(Spinner) findViewById(R.id.kegiatan);
14	
15	ArrayAdapter<CharSequence>adapter=ArrayAdapter.createFromR
16	esource(this,R.array.Kegiatan,
17	android.R.layout.simple_spinner_item);
18	
19	adapter.setDropDownViewResource(android.R.layout.simple_sp
20	inner_dropdown_item);
21	if (keg!=null){
22	keg.setAdapter(adapter);
23	}
24	kolam=(Spinner)
25	findViewById(R.id.customKeg_kolam);
26	ket=(EditText) findViewById(R.id.keterangan);
27	tgl.setKeyListener(null);
28	dateFormat=new SimpleDateFormat("dd-MM-yyyy");
29	ArrayAdapter<dataKolam> dataAdapter = new
30	ArrayAdapter<dataKolam>(this, android.R.layout.simple_spinn
31	er_item, new DatabaseHelper(this).get_kolam());



Tabel 5.9 Implementasi algoritme menambah kegiatan budaya lain (lanjutan)

No.	Kode Program
32	
33	<code>dataAdapter.setDropDownViewResource (android.R.layout.simpl</code>
34	<code>e_spinner_dropdown_item);</code>
35	<code>kolam.setAdapter(dataAdapter);</code>
36	
37	<code>}</code>
38	<code>public void simpan_kegiatan(View view) {</code>
39	<code>SimpleDateFormat dateFormat=new</code>
40	<code>SimpleDateFormat("yyyy-MM-dd");</code>
50	<code>String kegiatan =</code>
51	<code>keg.getSelectedItem().toString();</code>
52	<code>String keterangan = ket.getText().toString();</code>
53	<code>String</code>
54	<code>picked_tgl=dateFormat.format(choose_date.getTime());</code>
55	<code>dataKolam</code>
56	<code>selected_kolam=(dataKolam)kolam.getSelectedItem();</code>
57	<code>int pilihan_kolam= selected_kolam.getId();</code>
58	<code>boolean insert =new</code>
59	<code>DatabaseHelper(this).insertKegiatan(kegiatan,pilihan_kolam</code>
60	<code>,picked_tgl,keterangan);</code>
61	<code>if (insert) {</code>
62	<code>System.out.println("kegiatan berhasil</code>
63	<code>ditambah");</code>
64	<code>} else{</code>
65	<code>System.out.println("kegiatan tidak berhasil</code>
66	<code>ditambah");</code>
67	<code>}</code>
68	<code>/*Intent intent = new Intent(this,</code>
69	<code>Kalender.class);</code>
67	<code>startActivity(intent);*/</code>
68	<code>finish();</code>
69	<code>}</code>
70	<code>}</code>

Penjelasan :

Pada baris 3-6 merupakan deklarasi variabel.

Pada baris 12, 13, 24-26 merupakan deklarasi pemanggilan id dari layout.

Pada baris 15-23, 29-35 merupakan pemanggilan nilai yang akan dimasukkan ke dalam *spinner*.

Pada baris 27-28 merupakan pengambilan tanggal.

Pada baris 38-68 merupakan fungsi untuk menyimpan masukan dari pengguna ke dalam *database*.

Pada baris 61-63 merupakan kondisi bila kegiatan berhasil ditambah.

Pada baris 63-66 merupakan kondisi bila kondisi pertama tidak memenuhi.



5.2.3.2 Implementasi Algoritme Melihat Kegiatan Budidaya

Pada Tabel 5.10 merupakan penerapan dari perancangan algoritme melihat kegiatan budidaya. Algoritme ini diimplementasikan ke dalam *class* kegiatanAdapter, lebih tepatnya terdapat pada *method* getView().

Tabel 5.10 Implementasi algoritme melihat kegiatan budidaya

No.	Kode Program
1	public View getView(int i, View view, ViewGroup
2	viewGroup) {
3	View v = view;
4	if (v == null) {
5	LayoutInflater vi;
6	vi = LayoutInflater.from(activity);
7	v = vi.inflate(R.layout.item_kegiatan, null);
8	}
9	
10	Object p = getItem(i);
11	if (p != null) {
12	kegiatan = (TextView)
13	v.findViewById(R.id.kegiatan);
14	nama_kolam = (TextView)
15	v.findViewById(R.id.kolam);
16	ket = (TextView) v.findViewById(R.id.ket);
17	nama_kolam.setText(data.get(i).getKolam());
18	ket.setText(data.get(i).getKet());
19	kegiatan.setText(data.get(i).getKegiatan());
20	}
21	return v;
22	}

Keterangan:

Pada baris 3-7 merupakan deklarasi untuk memanggil layout item_kegiatan.

Pada baris 10 adalah deklarasi variabel p sama dengan *item* yang ditampilkan.

Pada baris 11-13 merupakan kondisi jika variabel p atau masukan dari pengguna tidak kosong maka data kegiatan akan ditampilkan.

5.2.3.3 Implementasi Algoritme Menambah Penghasilan Uang

Pada Tabel 5.11 merupakan implementasi algoritme menambah penghasilan uang, algoritme tersebut diimplementasikan ke dalam *class* pemasukan. Pada *class* ini terdapat dua *method* yang digunakan untuk implementasi algoritme, *method* pertama adalah hitung_hasil() yang digunakan untuk menghitung jumlah pendapatan yang dimasukkan oleh pengguna. Sedangkan *method* kedua adalah simpan_pemasukan(), pada *method* ini berfungsi untuk melakukan penyimpanan data yang telah diperoleh ke dalam *database*.



Tabel 5.11 Implementasi algoritme menambah penghasilan uang

No.	Kode Program
1	public void hitung_hasil(){
2	
3	long hasil_kiloan=0;
4	long hasil_ekoran=0;
5	if (!hasilKiloan.getText().toString().equals("")){
6	hasil_kiloan=Long.valueOf(hasilKiloan.getText().toString())
7	};
8	}
9	if (!hasilEkor.getText().toString().equals("")){
10	hasil_ekoran=Long.valueOf(hasilEkor.getText().toString());
11	}
12	
13	long total = hasil_ekoran+hasil_kiloan;
14	jumlah.setText(String.valueOf(total));
15	}
16	public void simpan_pemasukan(View view){
17	String pilihan_ikanEkor=
18	ikanEkor.getSelectedItem().toString();
19	String pilihan_ikanKilo=
20	ikanKiloan.getSelectedItem().toString();
21	dataKolam
22	selected_kolam=(dataKolam)namaKolam.getSelectedItem();
23	int pilihan_kolam=selected_kolam.getId();
24	int banyakKg=0;
25	int banyak_ekor=0;
26	long hasil_kiloan=0, hasil_ekoran=0;
27	if (!banyakKiloan.getText().toString().equals("")){
28	
29	banyakKg=Integer.valueOf(banyakKiloan.getText().toString())
30	};
31	hasil_kiloan=Long.valueOf(hasilKiloan.getText().toString())
32	};
33	}
34	if (!banyakEkor.getText().toString().equals("")){
35	
36	banyak_ekor=Integer.valueOf(banyakEkor.getText().toString()
37));
38	}
39	hasil_ekoran=Long.valueOf(hasilEkor.getText().toString());
40	}
41	}
42	if (banyak_ekor!=0&& banyakKg!=0){
43	new
44	DatabaseHelper(this).insertPemasukan(pilihan_kolam,
45	pilihan_ikanKilo, banyakKg,0,hasil_kiloan);
46	new
47	DatabaseHelper(this).insertPemasukan(pilihan_kolam,
48	pilihan_ikanEkor, 0,banyak_ekor,hasil_ekoran);
49	}else if (banyak_ekor==0){
50	new
51	DatabaseHelper(this).insertPemasukan(pilihan_kolam,
52	pilihan_ikanKilo, banyakKg,0,hasil_kiloan);
53	}else if (banyakKg==0){
54	
55	}
56	}
57	
58	
59	
60	
61	
62	

Tabel 5.11 Implementasi algoritme menambah penghasilan uang (lanjutan)

No.	Kode Program
63	new
64	DatabaseHelper(this).insertPemasukan(pilihan_kolam,
65	pilihan_ikanEkor, 0,banyak_ekor,hasil_ekoran);
66	}
67	}

Penjelasan:

Pada baris 3-4 merupakan deklarasi variabel.

Pada baris 5-7 merupakan kondisi jika hasilKiloan tidak kosong maka nilai hasil kiloan diambil.

Pada baris 9-11 merupakan kondisi bila hasilEkor tidak kosong maka nilai hasil ekor akan diambil

Pada baris 12-13 merupakan perhitungan total hasil.

Pada baris 17-26 merupakan deklarasi variabel untuk mengambil masukan sebelumnya.

Pada baris 27-33 merupakan kondisi bila banyak kiloan tidak kosong, maka sistem akan mengambil jumlah ikan perkilogram dan hasil pendapatan ikan perkilogram.

Pada baris 34-40 merupakan kondisi bila banyak ekor tidak kosong, maka sistem akan mengambil jumlah ikan perekor dan hasil pendapatan ikan perekor.

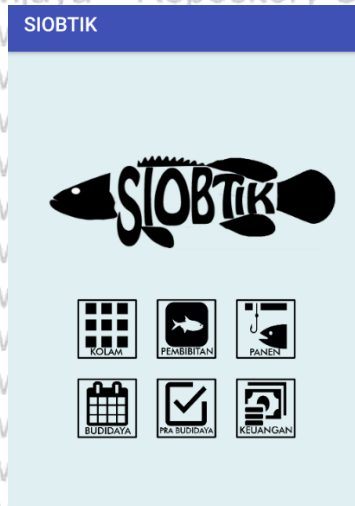
Pada baris 51-66 merupakan kumpulan kondisi untuk menyimpan pemasukan dengan kondisi ikan perkilo dan ikan perekor memiliki nilai atau tidak.

5.2.4 Implementasi Antarmuka Iterasi Pertama

Pada implementasi antarmuka merupakan penerapan dari perancangan antarmuka yang telah dibuat sebelumnya, antarmuka juga digunakan sebagai sarana interaksi langsung antara pengguna dan sistem. berikut merupakan implementasi antarmuka Sistem Operasional Budidaya Tambak Ikan Kerapu (SIOBTIK).

5.2.4.1 Implementasi Antarmuka Halaman Menu Utama Sistem

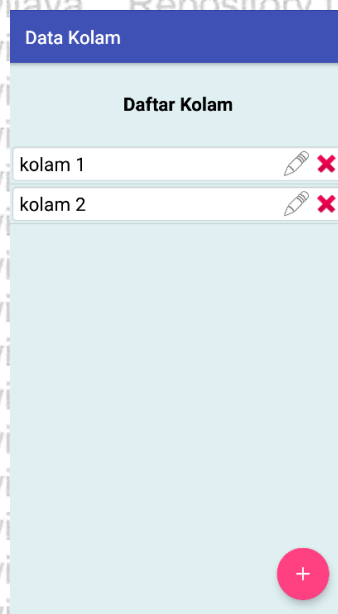
Antarmuka halaman menu utama sistem akan ditampilkan saat sistem pertama kali dijalankan. Pada Gambar 5.22 merupakan implementasi antarmuka dari halaman menu utama sistem.



Gambar 5.22 Implementasi antarmuka halaman utama

5.2.4.2 Implementasi Antarmuka Halaman Daftar Kolam

Antarmuka halaman daftar kolam akan ditampilkan saat pengguna menekan menu kolam. Pada Gambar 5.23 merupakan implementasi antarmuka dari halaman menu utama sistem.



Gambar 5.23 Implementasi antarmuka halaman daftar kolam

5.2.4.3 Implementasi Antarmuka Halaman Form Tambah

Antarmuka halaman *form* tambah akan ditampilkan pada semua menu *form* atau *edit* yang ada pada sistem. Pada Gambar 5.24 merupakan implementasi antarmuka dari halaman menu *form* tambah.



SIOBTIK

Masukkan Kegiatan

Tanggal

Nama Kolam:

Nama Kegiatan:

Keterangan

SIMPAN

Gambar 5.24 Implementasi antarmuka form tambah

5.2.4.4 Implementasi Antarmuka Kalender Kegiatan Budidaya

Antarmuka halaman kalender kegiatan budidaya akan ditampilkan saat pengguna menekan menu budidaya. Pada Gambar 5.25 merupakan implementasi antarmuka dari halaman menu kalender kegiatan budidaya.

SIOBTIK

<
November 2018
>

M	S	S	R	K	J	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

18 November 2018

kolam 1 Persiapan Kolam

kuli bisa mulai perbaikan

kolam 2 Kerapu Cantang

Tebar Bibit Kerapu Cantang ukuran 30 cm sebanyak 2000 ekor

kolam 2 Kerapu Cantang

Gambar 5.25 Implementasi antarmuka kalender kegiatan budidaya

5.2.4.5 Implementasi Antarmuka Data Pembibitan

Antarmuka halaman data pembibitan akan ditampilkan saat pengguna menekan menu pembibitan. Pada Gambar 5.26 merupakan implementasi antarmuka dari halaman data pembibitan.



SIOBTIK

Daftar Pembibitan

^ kolom 2

18-Nov-2018

Kerapu Cantang

Tebar Bibit Kerapu Cantang ukuran 30
cm sebanyak 2000 ekor

Gambar 5.26 Implementasi antarmuka data pembibitan

5.2.4.6 Implementasi Antarmuka Persiapan Budidaya Iterasi Pertama

Antarmuka halaman persiapan budidaya akan ditampilkan saat pengguna menekan menu persiapan budidaya. Pada Gambar 5.27 merupakan implementasi antarmuka dari halaman persiapan budidaya.

prabudidaya

kolam 1

Persiapan Kolam:

- Pengurasan Kolam
- Pembersihan Kolam
- Perbaikan Tanggul

Persiapan Bibit:

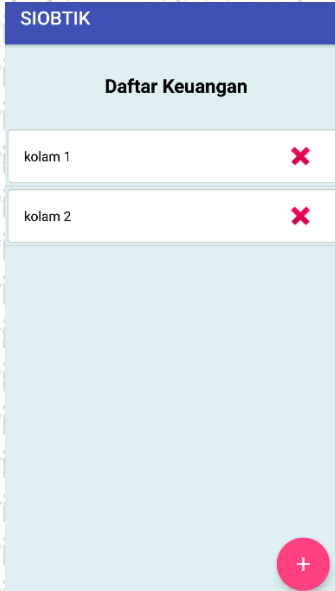
- Pemesanan Bibit



Gambar 5.27 Implementasi antarmuka persiapan budidaya iterasi pertama

5.2.4.7 Implementasi Antarmuka Halaman Keuangan

Antarmuka halaman keuangan akan ditampilkan saat pengguna menekan menu keuangan. Pada Gambar 5.28 merupakan implementasi antarmuka dari halaman keuangan.



Gambar 5.28 Implementasi antarmuka keuangan

5.2.4.8 Implementasi Antarmuka Detail Keuangan

Antarmuka halaman detail keuangan akan ditampilkan saat pengguna menekan nama kolam pada menu keuangan. Pada Gambar 5.29 merupakan implementasi antarmuka dari halaman detail keuangan.

SIOBTIK			
kolam 1			
Pengeluaran:			
1	pembelian bibit		Rp. 300.000
Pemasukan:			
1	Kerapu Cantang	300 Kg	Rp. 2.000.000
2	Kerapu Lumpur	30 Ekor	Rp. 1.000.000
Untung			Rp. 2.700.000

Gambar 5.29 Implementasi antarmuka detail keuangan

5.2.4.9 Implementasi Antarmuka Tambah Pemasukan Keuangan

Antarmuka halaman tambah pemasukan keuangan akan ditampilkan saat pengguna memilih tombol tambah pemasukan pada menu keuangan. Pada Gambar 5.30 merupakan implementasi antarmuka dari halaman tambah pemasukan keuangan.



Gambar 5.30 Implementasi antarmuka tambah pemasukan

5.2.5 Implementasi Antarmuka Iterasi Kedua

Pada implementasi antarmuka iterasi kedua ini merupakan implementasi dari perancangan antarmuka iterasi kedua, yang mendapatkan perubahan pada beberapa menu sistem.

5.2.5.1 Implementasi Antarmuka Menu Utama Iterasi kedua

Antarmuka halaman menu utama akan ditampilkan pada saat sistem pertama kali dijalankan. Pada Gambar 5.31 merupakan implementasi dari halaman menu utama iterasi kedua.



Gambar 5.31 Implementasi antarmuka menu utama iterasi kedua

5.2.5.2 Implementasi Antarmuka Daftar Kolam Iterasi Kedua

Antarmuka halaman daftar kolam merupakan halaman yang akan ditampilkan bila pengguna menekan menu kolam, tampilan antarmuka ini juga akan digunakan pada menu daftar yang lain dengan sedikit perubahan. Gambar



5.32 merupakan implementasi antarmuka dari halaman daftar kolam iterasi kedua.



Gambar 5.32 Implementasi antarmuka daftar kolam iterasi kedua

5.2.5.3 Implementasi Antarmuka Form Tambah Iterasi Kedua

Antarmuka halaman *form* Tambah akan ditampilkan pada semua menu tambah pada sistem. Gambar 5.33 merupakan implementasi antarmuka dari halaman *form* tambah iterasi kedua.

Gambar 5.33 Implementasi antarmuka form tambah iterasi kedua



5.2.5.4 Implementasi Antarmuka Data Pembibitan Iterasi Kedua

Antarmuka halaman data pembibitan iterasi kedua ini akan ditampilkan pada saat pengguna menekan tombol pembibitan pada menu utama. Gambar 5.34 merupakan implementasi antarmuka dari halaman data pembibitan iterasi kedua.



Gambar 5.34 Implementasi antarmuka data pembibitan iterasi kedua

5.2.5.5 Implementasi Antarmuka Persiapan Budidaya Iterasi Kedua

Antarmuka halaman persiapan budidaya iterasi kedua akan ditampilkan saat pengguna menekan menu persiapan budidaya dan memilih submenu pertama. Pada Gambar 5.27 merupakan implementasi antarmuka dari halaman persiapan budidaya iterasi kedua.



Gambar 5.35 Implementasi antarmuka persiapan budidaya iterasi kedua



5.2.5.6 Implementasi Antarmuka Jadwal Persiapan Budidaya

Antarmuka halaman jadwal persiapan budidaya akan ditampilkan saat pengguna menekan menu persiapan budidaya dan memilih *submenu* kedua. Pada Gambar 5.28 merupakan implementasi antarmuka dari halaman jadwal persiapan budidaya.



Gambar 5.36 Implementasi antarmuka jadwal persiapan budidaya



BAB 6 PENGUJIAN

Pada bab ini akan membahas tentang pengujian dari sistem operasional budidaya tambak ikan kerapu. Pengujian ini bertujuan untuk menemukan kesalahan yang ada pada sistem, serta untuk memastikan bahwa sistem telah sesuai dengan kebutuhan.

6.1 Pengujian Unit

Pengujian *white box* digunakan untuk menguji detail dari perancangan sistem, yang akan dimulai dengan struktur dari desain sistem secara procedural dan nantinya akan dipecah menjadi beberapa kasus uji. Algoritme dari sistem akan disusun menggunakan *flow graph* yang bertujuan untuk mengetahui jumlah dari *cyclomatic complexity* dan jalur independennya.

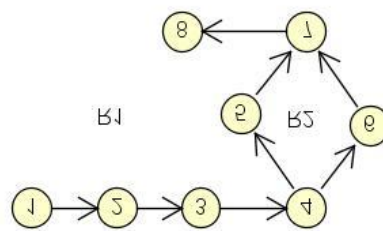
6.1.1 Pengujian Unit Menambah Kegiatan Budidaya lain

Pada Tabel 6.1 merupakan algoritme menambah kegiatan budidaya lain, yang digunakan untuk mengetahui jumlah *node flow graph*.

Tabel 6.1 Pengujian unit menambah kegiatan budidaya lain

No.	Algoritme	Node
1	Mulai	1
2	Menerima masukan tanggal, kegiatan, kolam, keterangan	2
3	Memasukkan nilai tgl, Keg, kolam, jet ke dalam <i>database</i>	3
4	IF (data tersimpan)	4
5	Mencetak pesan "kegiatan berhasil ditambah"	5
6	ELSE	6
7	Mencetak pesan "kegiatan tidak berhasil ditambah"	7
8	END IF	8
9	Selesai	8

Berdasarkan dari algoritme pada Tabel 6.1, maka diperoleh bentuk *flow graph* yang ditunjukkan oleh Gambar 6.1.



Gambar 6.1 *Flowgraph* menambah kegiatan budidaya lain

Berdasarkan dari jumlah *flowgraph* yang diperoleh, maka dapat ditentukan dua jalur independen serta jumlah kompleksitas, seperti berikut:

Jalur 1: 1-2-3-4-6



Jalur 2: 1-2-3-5-6

Perhitungan *cyclomatic complexity*:

$$V(G) = \text{Jumlah Region} = 2$$

$$V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

$$V(G) = P + 1 = 1 + 1 = 2$$

Berdasarkan dari jalur yang diperoleh, maka terbentuk kasus uji. Pada Tabel 6.2 merupakan pemaparan kasus uji algoritme fungsi menambah kegiatan budidaya lain

Tabel 6.2 Hasil pengujian unit menambah kegiatan budidaya lain

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	Data kegiatan berhasil disimpan ke dalam <i>database</i>	Sistem menampilkan pemberitahuan "kegiatan berhasil ditambah".	Sistem menampilkan pemberitahuan "kegiatan berhasil ditambah".	valid
2	Data kegiatan tidak berhasil disimpan ke dalam <i>database</i>	Sistem menampilkan pemberitahuan "kegiatan tidak berhasil ditambah".	Sistem menampilkan pemberitahuan "kegiatan tidak berhasil ditambah".	valid

6.1.2 Pengujian Unit Melihat Detail Keuangan

Pada Tabel 6.3 merupakan algoritme melihat detail kegiatan, yang digunakan untuk mengetahui jumlah *node flow graph*.

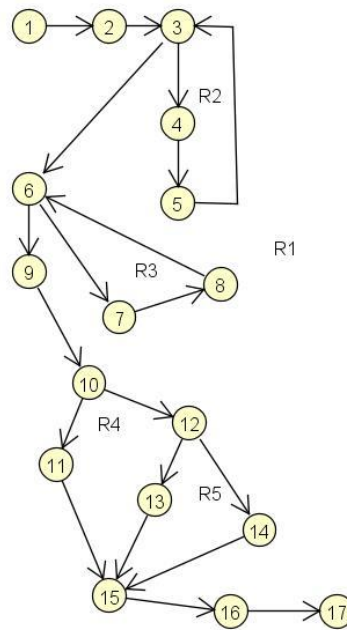
Tabel 6.3 Pengujian unit melihat detail keuangan

No.	Algoritme	Node
1	Mulai	1
2	Ambil id_kolam	2
3	Ambil pemasukan	
4	Ambil pengeluaran	
5	FOREACH (pemasukan)	3
6	Tampil detail pemasukan	4
7	Tampil Total pemasukan	
8	END FOREACH	5
9	FOREACH (pengeluaran)	6
10	Tampil detail pengeluaran	7
11	Tampil total pengeluaran	
13	END FOREACH	8
13	Selisih = total pemasukan – total pengeluaran	9
14	IF (selisih < 0)	10

Tabel 6.3 Pengujian unit melihat detail keuangan (lanjutan)

No.	Algoritme	Node
15	Tampil "Untung"	11
16	ELSE IF (selisih = 0)	12
17	Tampil "Netral"	13
18	ELSE Tampil "Rugi"	14
19	END IF	15
20	Tampil selisih	16
21	Selesai	17

Berdasarkan dari algoritme pada Tabel 6.3, maka diperoleh bentuk *flow graph* yang ditunjukkan oleh Gambar 6.2.



Gambar 6.2 *Flowgraph* melihat detail keuangan

Berdasarkan dari jumlah *flowgraph* yang diperoleh, maka dapat ditentukan dua jalur independen serta jumlah kompleksitas, seperti berikut:

Jalur 1: 1-2-3-6-9-10-12-13-16-17-18

Jalur 2: 1-2-3-4-5-3-6-9-10-11-16-17-18

Jalur 3: 1-2-3-6-7-8-6-9-10-12-14-15-16-17-18

Jalur 4: 1-2-3-4-5-3-6-7-8-6-10-11-16-17-18

Jalur 5: 1-2-3-4-5-3-6-7-8-6-10-12-13-16-17-18

Jalur 6: 1-2-3-4-5-3-6-7-8-6-10-12-14-15-16-17-18

Perhitungan *cyclomatic complexity*:



$$V(G) = \text{Jumlah Region} = 6$$

$$V(G) = E - N + 2 = 20 - 17 + 2 = 2$$

$$V(G) = P + 1 = 1 + 1 = 2$$

Berdasarkan dari jalur yang diperoleh, maka terbentuk kasus uji. Pada Tabel 6.4 merupakan pemaparan kasus uji algoritme fungsi melihat detail keuangan.

Tabel 6.4 Hasil pengujian unit melihat detail keuangan

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	Data detail pemasukan tidak sama dengan null, data pengeluaran sama dengan null.	Sistem akan menampilkan detail pemasukan dan detail pengeluaran kosong. Sistem akan menampilkan keterangan "Untung" dan hasil selisih.	Sistem akan menampilkan detail pemasukan dan detail pengeluaran kosong. Sistem akan menampilkan keterangan "untung" dan hasil selisih.	valid
2	Data detail pemasukan sama dengan null, data pengeluaran tidak sama dengan null.	Sistem akan menampilkan detail pengeluaran dan detail pemasukan kosong. Sistem akan menampilkan keterangan "Rugi" dan hasil selisih.	Sistem akan menampilkan detail pengeluaran dan detail pemasukan kosong. Sistem akan menampilkan keterangan "Rugi" dan hasil selisih.	valid
3	Data hasil pemasukan > data hasil pengeluaran	Sistem akan menampilkan detail pemasukan dan detail pengeluaran. Sistem akan menampilkan keterangan "untung" dan hasil selisih.	Sistem akan menampilkan detail pemasukan dan detail pengeluaran. Sistem akan menampilkan keterangan "untung" dan hasil selisih.	valid



Tabel 6.4 Hasil pengujian unit melihat detail keuangan (lanjutan)

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
5	Data pemasukan < data pengeluaran	hasil pemasukan detail pengeluaran. Sistem akan menampilkan keterangan "Rugi" dan hasil selisih.	akan detail pemasukan dan pengeluaran. Sistem akan menampilkan keterangan "Rugi" dan hasil selisih.	valid

6.1.3 Pengujian Unit Menambah Pemasukan Uang

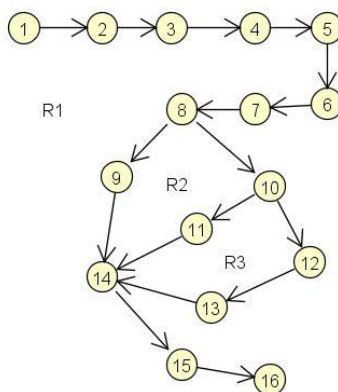
Pada Tabel 6.5 merupakan algoritme menambah pemasukan keuangan, yang digunakan untuk mengetahui jumlah *node flow graph*.

Tabel 6.5 Pengujian unit menambah pemasukan keuangan

No.	Algoritme	Node
1	Mulai	1
2	Mengambil data masukan pengguna	2
3	Hitung total = hasil_kiloan + hasil_ekoran	3
4	Simpan penghasilan	4
6	Ambil banyak_ekor	6
7	Ambil banyakKg	7
8	IF (banyak_ekor != 0 dan banyakKg != 0)	8
9	Simpan pemasukan hasil_kiloan dan hasil_ekoran	9
10	ELSE IF (banyak_ekor = 0)	10
11	Simpan pemasukan hasil_kiloan	11
12	ELSE IF (banyakKG = 0)	12
13	Simpan pemasukan hasil_ekoran	13
14	END IF	14
15	Menampilkan detail pemasukan	15
16	Selesai	16



Berdasarkan dari algoritme pada Tabel 6.3, maka diperoleh bentuk *flow graph* yang ditunjukkan oleh Gambar 6.2.



Gambar 6.3 Flowgraph menambah pemasukan keuangan

Berdasarkan dari jumlah *flowgraph* yang diperoleh, maka dapat ditentukan dua jalur independen serta jumlah kompleksitas, seperti berikut:

Jalur 1: 1-2-3-4-5-6-7-8-9-14-15-16

Jalur 2: 1-2-3-4-5-6-7-8-10-11-14-15-16

Jalur 3: 1-2-3-4-5-6-7-8-10-12-13-14-15-16

Perhitungan *cyclomatic complexity*:

$$V(G) = \text{Jumlah Region} = 3$$

$$V(G) = E - N + 2 = 17 - 16 + 2 = 3$$

$$V(G) = P + 1 = 2 + 1 = 3$$

Berdasarkan dari jalur yang diperoleh, maka terbentuk kasus uji. Pada Tabel 6.6 merupakan pemaparan kasus uji algoritme fungsi menambah penghasilan keuangan.

Tabel 6.6 Hasil pengujian unit menambah penghasilan keuangan

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	Banyak_ekor tidak = 0 dan banyakKG tidak = 0	Sistem berhasil menyimpan hasil_kiloan dan hasil_ekoran ke dalam <i>database</i> .	Sistem berhasil menyimpan hasil_kiloan dan hasil_ekoran ke dalam <i>database</i> .	valid
2	Banyak_ekor = 0	Sistem hanya menyimpan hasil_kiloan ke dalam <i>database</i> .	Sistem hanya menyimpan hasil_kiloan ke dalam <i>database</i> .	valid

Tabel 6.6 Hasil pengujian unit menambah penghasilan keuangan (lanjutan)

Jalur	Prosedur Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
30	Banyak_kiloan =	Sistem hanya menyimpan hasil_ekoran ke dalam <i>database</i> .	Sistem hanya menyimpan hasil_ekoran ke dalam <i>database</i> .	valid

6.1.4 Analisis Pengujian Unit

Pada pengujian unit dengan menggunakan pengujian *white box* dengan menggunakan metode *basis path testing*. Dari tiga sampel yang diuji dihasilkan dua *cyclomatic complexity* pada kasus uji menambah kegiatan budidaya lain, enam *cyclomatic complexity* pada kasus uji melihat detail keuangan, sedangkan pada kasus uji menambah pemasukan uang terdapat tiga *cyclomatic complexity*. Dari hasil pengujian ini semua kasus uji mendapatkan hasil valid, sehingga dapat disimpulkan bahwa sistem berjalan sesuai dengan analisis sistem.

6.2 Pengujian Validasi Iterasi Pertama

Pengujian validasi dilakukan pada semua kebutuhan pada sistem dengan menggunakan metode pengujian *black box*. Pengujian ini akan dilakukan dengan cara melihat apakah sistem sudah berjalan dengan benar dan tidak terdapat *error* pada sistem. Pengujian validasi dilakukan untuk mengetahui sistem sudah berjalan sesuai dengan kebutuhan fungsional yang telah dibuat.

Tabel 6.7 Kasus uji menambah kolam

Nama Kasus Uji	Kasus Uji Menambah Kolam
Kode Kebutuhan	SI0BTIK_F_01
Prosedur	1. Aktor menekan tombol tambah. 2. Aktor memasukkan nama kolam, luas kolam, posisi kolam dan gambar kolam. 3. Aktor menekan tombol "simpan".
Hasil yang diharapkan	Sistem menyimpan data kolam ke dalam <i>database</i> dan menampilkan pesan "Berhasil Disimpan".
Hasil	Sistem menyimpan data kolam ke dalam <i>database</i> dan menampilkan pesan "Berhasil Disimpan".
Status	Valid



Tabel 6.8 Kasus uji menambah kolom
Alternatif 1: Data tidak tersimpan ke dalam *database*

Nama Kasus Uji	Kasus Uji Menambah Kolam
Kode Kebutuhan	SIOBTIK_F_01
Prosedur	1. Aktor menekan tombol tambah. 2. Aktor memasukkan nama kolom, luas kolom, posisi kolom dan gambar kolom. 3. Aktor menekan tombol "simpan".
Hasil yang diharapkan	Sistem gagal menyimpan data kolom ke dalam <i>database</i> dan menampilkan pesan "Gagal Disimpan"
Hasil	Sistem menyimpan data kolom ke dalam <i>database</i> dan menampilkan pesan "Berhasil Disimpan", bila data berhasil disimpan.
Status	Valid

Tabel 6.9 Kasus uji menambah kolom
Alternatif 2: Salah satu data kosong

Nama Kasus Uji	Kasus Uji menambah kolom
Kode Kebutuhan	SIOBTIK_F_01
Prosedur	1. Aktor memilih tombol tambah. 2. Aktor memasukkan nama kolom, luas kolom dan gambar kolom pada formulir. 3. Aktor memilih tombol "simpan".
Hasil yang diharapkan	Sistem akan menampilkan pesan pada <i>textbox</i> yang kosong berupa "data masih kosong".
Hasil	Sistem menampilkan pesan pada <i>textbox</i> yang kosong berupa "data masih kosong".
Status	Valid

Tabel 6.10 Kasus uji melihat daftar kolom

Nama Kasus Uji	Kasus Uji Melihat Daftar Kolam
Kode Kebutuhan	SIOBTIK_F_02
Prosedur	1. Aktor menekan menu kolom.
Hasil yang diharapkan	Sistem menampilkan daftar nama kolom.
Hasil	Sistem menampilkan daftar nama kolom.
Status	Valid

**Tabel 6.11 Kasus uji menghapus kolom**

Nama Kasus Uji	Kasus Uji Menghapus Kolam
Kode Kebutuhan	SIOBTIK_F_03
Prosedur	1. Aktor menekan tombol hapus pada salah daftar kolom. 2. Aktor memilih tombol "yes".
Hasil yang diharapkan	Sistem menghapus data kolom dari <i>database</i> .
Hasil	Sistem menghapus data kolom dari <i>database</i> .
Status	Valid

Tabel 6.12 Kasus uji menghapus kolom**Alternatif 1: aktor memilih "no"**

Nama Kasus Uji	Kasus Uji Menghapus Kolam
Kode Kebutuhan	SIOBTIK_F_03
Prosedur	1. Aktor menekan tombol hapus pada salah daftar kolom. 2. Aktor memilih tombol "yes".
Hasil yang diharapkan	Sistem tidak jadi menghapus data kolom dari <i>database</i> .
Hasil	Sistem tidak jadi menghapus data kolom dari <i>database</i> .
Status	Valid

Tabel 6.13 kasus uji melihat detail kolom

Nama Kasus Uji	Kasus Uji Melihat Detail Kolam
Kode Kebutuhan	SIOBTIK_F_04
Prosedur	1. Aktor menekan tombol nama kolom yang diinginkan
Hasil yang diharapkan	Sistem menampilkan nama kolom, luas kolom, posisi kolom dan gambar kolom.
Hasil	Sistem menampilkan nama kolom, luas kolom, posisi kolom dan gambar kolom.
Status	Valid

Tabel 6.14 Kasus uji mengubah detail kolom

Nama Kasus Uji	Kasus Uji Mengubah Detail Kolam
Kode Kebutuhan	SIOBTIK_F_05
Prosedur	1. Aktor menekan tombol <i>edit</i> pada nama kolom yang diinginkan. 2. Aktor mengubah detail kolom. 3. Aktor menekan tombol "simpan".



Tabel 6.14 Kasus uji mengubah detail kolom (lanjutan)

Hasil yang diharapkan	Sistem menyimpan data kolom ke dalam <i>database</i> dan menampilkan pesan “Berhasil Disimpan”.
Hasil	Sistem menyimpan data kolom ke dalam <i>database</i> dan menampilkan pesan “Berhasil Disimpan”.
Status	Valid

Tabel 6.15 Kasus uji mengubah detail kolom
Alternatif 1: data tidak berhasil disimpan

Nama Kasus Uji	Kasus Uji Mengubah Detail Kolam
Kode Kebutuhan	SIObTIK_F_05
Prosedur	1. Aktor menekan tombol <i>edit</i> pada nama kolom yang diinginkan. 2. Aktor mengubah detail kolom. 3. Aktor menekan tombol “simpan”.
Hasil yang diharapkan	Sistem gagal menyimpan data kolom ke dalam <i>database</i> dan menampilkan pesan “Gagal Disimpan”.
Hasil	Sistem menyimpan data kolom ke dalam <i>database</i> dan menampilkan pesan “Gagal Disimpan”.
Status	Valid

Tabel 6.16 Kasus uji mengubah detail kolom
Alternatif 2: salah satu data kosong

Nama Kasus Uji	Kasus Uji megubah detail kolom
Kode Kebutuhan	SIObTIK_F_05
Prosedur	1. Aktor menekan tombol <i>edit</i> pada nama kolom yang diinginkan. 2. Aktor memasukkan nama kolom, luas kolom dan gambar kolom pada formulir. 3. Aktor memilih tombol “Simpan”.
Hasil yang diharapkan	Sistem akan menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Hasil	Sistem menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Status	Valid

**Tabel 6.17 Kasus uji menambah persiapan budidaya**

Nama Kasus Uji	Kasus Uji Menambah Persiapan Budidaya
Kode Kebutuhan	SIObTIK_F_06
Prosedur	1. Aktor menekan tombol tambah persiapan budidaya. 2. Aktor memilih kolam. 3. Aktor menekan tombol “simpan”.
Hasil yang diharapkan	Sistem berhasil menambahkan daftar persiapan budidaya dan menampilkan pesan “perencanaan berhasil ditambah”.
Hasil	Sistem berhasil menambahkan daftar persiapan budidaya dan menampilkan pesan “perencanaan berhasil ditambah”.
Status	Valid

Tabel 6.18 Kasus uji menambah persiapan budidaya**Alternatif 1: persiapan budidaya tidak berhasil ditambah**

Nama Kasus Uji	Kasus Uji Menambah Persiapan Budidaya
Kode Kebutuhan	SIObTIK_F_06
Prosedur	1. Aktor menekan tombol tambah persiapan budidaya. 2. Aktor memilih kolam. 3. Aktor menekan tombol “simpan”.
Hasil yang diharapkan	Sistem tidak berhasil menambahkan daftar persiapan budidaya dan menampilkan pesan “perencanaan tidak berhasil ditambah”.
Hasil	Sistem berhasil menambahkan daftar persiapan budidaya dan menampilkan pesan “perencanaan tidak berhasil ditambah”.
Status	Valid

Tabel 6.19 Kasus uji melihat daftar persiapan budidaya

Nama Kasus Uji	Kasus Uji Melihat Daftar Persiapan Budidaya
Kode Kebutuhan	SIObTIK_F_07
Prosedur	1. Aktor memilih menu persiapan budidaya.
Hasil yang diharapkan	Sistem menampilkan daftar centang kegiatan persiapan budidaya.
Hasil	Sistem menampilkan daftar centang kegiatan persiapan budidaya.
Status	Valid



Tabel 6.20 Kasus uji menandai kegiatan persiapan budidaya

Nama Kasus Uji	Kasus Uji Menandai Kegiatan Persiapan Budidaya
Kode Kebutuhan	SIOBTIK_F_08
Prosedur	1. Aktor memilih <i>checklist</i> .
Hasil yang diharapkan	Sistem berhasil menambah simbol penanda pada <i>ceklist</i> kegiatan yang dipilih.
Hasil	Sistem berhasil menambah simbol penanda pada <i>ceklist</i> kegiatan yang dipilih.
Status	Valid

Tabel 6.21 Kasus uji menghapus kegiatan persiapan budidaya

Nama Kasus Uji	Kasus Uji Menghapus Kegiatan Persiapan Budidaya
Kode Kebutuhan	SIOBTIK_F_09
Prosedur	1. Aktor menambahkan data pembibitan.
Hasil yang diharapkan	Sistem berhasil menghapus data kegiatan persiapan budidaya pada kolam yang sama dari <i>database</i> .
Hasil	Sistem berhasil menghapus data kegiatan persiapan budidaya pada kolam yang sama dari <i>database</i> .
Status	Valid

Tabel 6.22 Kasus uji mencatat penebaran bibit

Nama Kasus Uji	Kasus Uji Mencatat Penebaran Bibit
Kode Kebutuhan	SIOBTIK_F_10
Prosedur	1. Aktor memilih menu tebar bibit. 2. Aktor memilih nama kolam, memilih tanggal tebar, memilih jenis ikan kerapu, mengisi banyak bibit, mengisi ukuran bibit. 3. Aktor menekan tombol "save"
Hasil yang diharapkan	Sistem menampilkan pesan "penebaran berhasil ditambah dan menambahkan data penebaran bibit ke dalam <i>database</i> ."
Hasil	Sistem menampilkan pesan "penebaran berhasil ditambah", bila berhasil menambahkan data penebaran bibit ke dalam <i>database</i> .
Status	Valid



**Tabel 6.23 Kasus uji mencatat penebaran bibit
Alternatif 1: data tidak berhasil ditambah**

Nama Kasus Uji	Kasus Uji mencatat penebaran bibit
Kode Kebutuhan	SIOBTIK_F_10
Prosedur	1. Aktor memilih menu tebar bibit. 2. Aktor memilih nama kolam, memilih tanggal tebar, memilih jenis ikan kerapu, mengisi banyak bibit, mengisi ukuran bibit. 3. Aktor menekan tombol "save"
Hasil yang diharapkan	Sistem gagal menyimpan data kolam ke dalam <i>database</i> dan menampilkan pesan "Gagal Disimpan".
Hasil	Sistem menyimpan data kolam ke dalam <i>database</i> dan menampilkan pesan "Gagal Disimpan".
Status	Valid

**Tabel 6.24 Kasus uji mencatat penebaran bibit
Alternatif 2: salah satu data kosong**

Nama Kasus Uji	Kasus Uji mencatat penebaran bibit
Kode Kebutuhan	SIOBTIK_F_10
Prosedur	1. Aktor memilih menu tebar bibit. 2. Aktor memilih nama kolam, memilih tanggal tebar, memilih jenis ikan kerapu, mengisi banyak bibit, mengisi ukuran bibit. 3. Aktor menekan tombol "save"
Hasil yang diharapkan	Sistem akan menampilkan pesan pada <i>textbox</i> yang kosong berupa "data masih kosong".
Hasil	Sistem menampilkan pesan pada <i>textbox</i> yang kosong berupa "data masih kosong".
Status	Valid

Tabel 6.25 Kasus uji melihat daftar pembibitan

Nama Kasus Uji	Kasus Uji Melihat Daftar Pembibitan
Kode Kebutuhan	SIOBTIK_F_11
Prosedur	1. Aktor memilih menu pembibitan
Hasil yang diharapkan	Sistem menampilkan daftar tebar bibit.
Hasil	Sistem menampilkan daftar tebar bibit.
Status	Valid



Tabel 6.26 Kasus uji melihat detail pembibitan

Nama Kasus Uji	Kasus Uji Melihat Detail Pembibitan
Kode Kebutuhan	SIOBTIK_F_12
Prosedur	1. Aktor menekan daftar kolam yang diinginkan.
Hasil yang diharapkan	Sistem menampilkan jenis ikan kerapu dan detail bibit yang ditebar.
Hasil	Sistem menampilkan jenis ikan kerapu dan detail bibit yang ditebar.
Status	Valid

Tabel 6.27 Kasus uji menghapus kegiatan harian

Nama Kasus Uji	Kasus Uji Menghapus Kegiatan Harian
Kode Kebutuhan	SIOBTIK_F_13
Prosedur	1. Aktor menambah kegiatan persiapan budidaya.
Hasil yang diharapkan	Sistem menghapus semua kegiatan harian dan data pembibitan pada kolam yang sama pada <i>database</i> .
Hasil	Sistem menghapus semua kegiatan harian dan data pembibitan pada kolam yang sama pada <i>database</i> .
Status	Valid

Tabel 6.28 Kasus uji menambahkan kegiatan harian

Nama Kasus Uji	Kasus Uji Menambahkan Kegiatan Harian
Kode Kebutuhan	SIOBTIK_F_14
Prosedur	1. Aktor menambah data tebar bibit.
Hasil yang diharapkan	Sistem menambahkan dan menampilkan kegiatan harian sesuai data penebaran bibit.
Hasil	Sistem menambahkan dan menampilkan kegiatan harian sesuai data penebaran bibit.
Status	Valid

Tabel 6.29 Kasus uji melihat kalender kegiatan budidaya

Nama Kasus Uji	Kasus Uji Melihat Kalender Kegiatan Budidaya
Kode Kebutuhan	SIOBTIK_F_15
Prosedur	1. Aktor memilih tanggal yang ingin dilihat.
Hasil yang diharapkan	Sistem menampilkan daftar kegiatan harian sesuai data penebaran bibit.
Hasil	Sistem menampilkan daftar kegiatan harian sesuai data penebaran bibit.
Status	Valid



Tabel 6.30 Kasus uji menambah kegiatan budidaya lain

Nama Kasus Uji	Kasus Uji Menambah Kegiatan Budidaya Lain
Kode Kebutuhan	SIOBTIK_F_16
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah kegiatan. 2. Aktor memilih nama kolam dan kegiatan, mengisi keterangan kegiatan. 3. Aktor menekan tombol “simpan”
Hasil yang diharapkan	Sistem menambahkan kegiatan ke dalam kalender harian.
Hasil	Sistem menambahkan kegiatan ke dalam kalender harian.
Status	Valid

Tabel 6.31 Kasus uji menambah kegiatan budidaya lain
Alternatif 2: Salah satu data kosong

Nama Kasus Uji	Kasus Uji menambah kegiatan budidaya lain
Kode Kebutuhan	SIOBTIK_F_16
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah kegiatan. 2. Aktor memilih nama kolam dan mengisi keterangan kegiatan. 3. Aktor menekan tombol “simpan”
Hasil yang diharapkan	Sistem akan menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Hasil	Sistem menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Status	Valid

Tabel 6.32 Kasus uji menambah kegiatan panen

Nama Kasus Uji	Kasus Uji Menambah Kegiatan Panen
Kode Kebutuhan	SIOBTIK_F_17
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah kegiatan panen. 2. Aktor memilih nama kolam, mengisi tanggal, kegiatan dan keterangan. 3. Aktor menekan tombol “simpan”
Hasil yang diharapkan	Sistem menambahkan kegiatan panen dan menampilkan pesan “berhasil ditambah”.
Hasil	Sistem menambahkan kegiatan panen dan menampilkan pesan “berhasil ditambah”.
Status	Valid



Tabel 6.33 Kasus uji menambah kegiatan panen
Alternatif 1: bila data kegiatan panen tidak berhasil ditambah

Nama Kasus Uji	Kasus Uji Menambah Kegiatan Panen
Kode Kebutuhan	SIOBTIK_F_17
Prosedur	1. Aktor memilih menu tambah kegiatan panen. 2. Aktor memilih nama kolam, mengisi tanggal, kegiatan dan keterangan. 3. Aktor menekan tombol “simpan”
Hasil yang diharapkan	Sistem tidak menambahkan kegiatan panen dan akan menampilkan pesan “tidak berhasil ditambah”.
Hasil	Sistem menambahkan kegiatan panen dan menampilkan pesan “tidak berhasil ditambah”.
Status	Valid

Tabel 6.34 Kasus uji menambah kegiatan panen
Alternatif 2: Salah satu data kosong

Nama Kasus Uji	Kasus Uji menambah kegiatan panen
Kode Kebutuhan	SIOBTIK_F_17
Prosedur	1. Aktor memilih menu tambah kegiatan panen. 2. Aktor memilih nama kolam, mengisi tanggal dan keterangan. 3. Aktor menekan tombol “simpan”
Hasil yang diharapkan	Sistem akan menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Hasil	Sistem menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Status	Valid

Tabel 6.35 Kasus uji melihat daftar panen

Nama Kasus Uji	Kasus Uji Melihat Daftar Panen
Kode Kebutuhan	SIOBTIK_F_18
Prosedur	1. Aktor memilih menu tanggal Panen
Hasil yang diharapkan	Sistem menampilkan daftar panen.
Hasil	Sistem menampilkan daftar panen.
Status	Valid

Tabel 6.36 Kasus uji melihat detail kegiatan panen

Nama Kasus Uji	Kasus Uji Melihat Detail Kegiatan Panen
Kode Kebutuhan	SIOBTIK_F_19
Prosedur	1. Aktor menekan tombol detail pada daftar kolam yang ada pada menu panen.

**Tabel 6.36 kasus uji melihat detail kegiatan panen (lanjutan)**

Hasil yang diharapkan	Sistem menampilkan daftar kegiatan panen.
Hasil	Sistem menampilkan daftar kegiatan panen.
Status	Valid

Tabel 6.37 Kasus uji menghapus detail panen

Nama Kasus Uji	Kasus Uji Menghapus Detail Panen
Kode Kebutuhan	SIOBTIK_F_20
Prosedur	1. Aktor menekan tombol hapus yang ada pada setiap daftar panen yang ada. 2. Aktor memilih tombol "Ya".
Hasil yang diharapkan	Sistem menampilkan daftar kegiatan panen.
Hasil	Sistem menampilkan daftar kegiatan panen.
Status	Valid

Tabel 6.38 Kasus uji menambah biaya pengeluaran

Nama Kasus Uji	Kasus Uji Menambah Biaya Pengeluaran
Kode Kebutuhan	SIOBTIK_F_21
Prosedur	1. Aktor memilih menu tambah pengeluaran. 2. Aktor memilih nama kolam, memasukkan jenis pengeluarannya, memasukkan biaya. 3. Aktor menekan tombol simpan.
Hasil yang diharapkan	Sistem menambahkan biaya pengeluaran dan menampilkan pesan "berhasil ditambah".
Hasil	Sistem menambahkan biaya pengeluaran dan menampilkan pesan "berhasil ditambah".
Status	Valid

Tabel 6.39 Kasus uji menambah biaya pengeluaran Alternatif 1: bila data tidak berhasil ditambah

Nama Kasus Uji	Kasus Uji Menambah Biaya Pengeluaran
Kode Kebutuhan	SIOBTIK_F_21
Prosedur	1. Aktor memilih menu tambah pengeluaran. 2. Aktor memilih nama kolam, memasukkan jenis pengeluarannya, memasukkan biaya. 3. Aktor menekan tombol simpan.
Hasil yang diharapkan	Sistem tidak berhasil menambahkan biaya pengeluaran dan menampilkan pesan "tidak berhasil ditambah".



Tabel 6.39 kasus uji menambah biaya pengeluaran (lanjutan)

Hasil	Sistem tidak berhasil menambahkan biaya pengeluaran dan menampilkan pesan “tidak berhasil ditambah”.
Status	Valid

Tabel 6.40 Kasus uji menambah biaya pengeluaran
Alternatif 2: bila salah satu data kosong

Nama Kasus Uji	Kasus Uji Menambah biaya pengeluaran
Kode Kebutuhan	SIOBTIK_F_21
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah pengeluaran. 2. Aktor memilih nama kolam, memasukkan jenis pengeluaram, memasukkan biaya. 3. Aktor menekan tombol simpan.
Hasil yang diharapkan	Sistem akan menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Hasil	Sistem menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Status	Valid

Tabel 6.41 Kasus uji menambah pemasukan uang

Nama Kasus Uji	Kasus Uji Menambah Pemasukan Uang
Kode Kebutuhan	SIOBTIK_F_22
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah pemasukan. 2. Aktor mengisi <i>form</i> banyak ekor, hasil ekor, banyak kiloan dan hasil kiloan. 3. Aktor memilih tombol simpan.
Hasil yang diharapkan	Sistem menambahkan banyak kiloan dan banyak ekor ke dalam <i>database</i> .
Hasil	Sistem menambahkan banyak kiloan dan banyak ekor ke dalam <i>database</i> .
Status	Valid

Tabel 6.42 Kasus uji menambah pemasukan uang
Alternatif 1: bila aktor hanya mengisi data kiloan

Nama Kasus Uji	Kasus Uji Menambah Pemasukan Uang
Kode Kebutuhan	SIOBTIK_F_22
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah pemasukan. 2. Aktor mengisi <i>form</i> banyak kiloan dan hasil kiloan 3. Aktor memilih tombol simpan.



Hasil yang diharapkan	Hanya menambahkan data banyak kiloan ke dalam <i>database</i> .
Hasil	Hanya menambahkan data banyak kiloan ke dalam <i>database</i> .
Status	Valid

**Tabel 6.43 Kasus uji menambah pemasukan uang
Alternatif 2: bila aktor hanya mengisi data ekoran**

Nama Kasus Uji	Kasus Uji Menambah Pemasukan Uang
Kode Kebutuhan	SIOBTIK_F_22
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah pemasukan. 2. Aktor mengisi <i>form</i> banyak ekoran dan hasil ekoran. 3. Aktor memilih tombol simpan.
Hasil yang diharapkan	Hanya menambahkan data banyak ekoran ke dalam <i>database</i> .
Hasil	Hanya menambahkan data banyak ekoran ke dalam <i>database</i> .
Status	Valid

**Tabel 6.44 Kasus uji menambah pemasukan uang
Alternatif 3: bila salah satu data kosong**

Nama Kasus Uji	Kasus Uji Menambah Pemasukan Uang
Kode Kebutuhan	SIOBTIK_F_22
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah pemasukan. 2. Aktor mengisi <i>form</i> hasil ekoran, banyak kiloan dan hasil kiloan. 3. Aktor memilih tombol simpan.
Hasil yang diharapkan	Sistem akan menampilkan pesan pada <i>textbox</i> yang kosong berupa "data masih kosong".
Hasil	Sistem menampilkan pesan pada <i>textbox</i> yang kosong berupa "data masih kosong".
Status	Valid

Tabel 6.45 Kasus uji melihat daftar keuangan

Nama Kasus Uji	Kasus Uji Melihat Daftar keuangan
Kode Kebutuhan	SIOBTIK_F_23
Prosedur	1. Aktor memilih menu keuangan.
Hasil yang diharapkan	Sistem menampilkan daftar keuangan.
Hasil	Sistem menampilkan daftar keuangan.
Status	Valid

**Tabel 6.46 Kasus uji melihat detail keuangan**

Nama Kasus Uji	Kasus Uji Melihat Detail keuangan
Kode Kebutuhan	SIOBTIK_F_24
Prosedur	1. Aktor memilih nama kolam di daftar keuangan yang diinginkan
Hasil yang diharapkan	Sistem menampilkan menu detail keuangan berupa daftar pengeluaran, daftar pemasukan, jumlah selisih keuntungan dan status "untung".
Hasil	Sistem menampilkan menu detail keuangan berupa daftar pengeluaran, daftar pemasukan, jumlah selisih keuntungan dan status "untung".
Status	Valid

Tabel 6.47 Kasus uji melihat detail keuangan Alternatif 1: bila data pengeluaran kosong

Nama Kasus Uji	Kasus Uji Melihat Detail keuangan
Kode Kebutuhan	SIOBTIK_F_24
Prosedur	1. Aktor memilih nama kolam di daftar keuangan yang diinginkan
Hasil yang diharapkan	Sistem menampilkan menu detail keuangan berupa daftar pemasukan, jumlah selisih dan status "Untung".
Hasil	Sistem menampilkan menu detail keuangan berupa daftar pemasukan, jumlah selisih dan status "Untung".
Status	Valid

Tabel 6.48 Kasus uji melihat detail keuangan Alternatif 2: bila data pemasukan kosong

Nama Kasus Uji	Kasus Uji Melihat Detail keuangan
Kode Kebutuhan	SIOBTIK_F_24
Prosedur	1. Aktor memilih nama kolam di daftar keuangan yang diinginkan
Hasil yang diharapkan	Sistem menampilkan menu detail keuangan berupa daftar pengeluaran, jumlah selisih dan status "Rugi".
Hasil	Sistem menampilkan menu detail keuangan berupa daftar pengeluaran, jumlah selisih dan status "Rugi".
Status	Valid



**Tabel 6.49 Kasus uji melihat detail keuangan
Alternatif 3: bila selisih sama**

Nama Kasus Uji	Kasus Uji Melihat Detail keuangan
Kode Kebutuhan	SIOBTIK_F_24
Prosedur	1. Aktor memilih nama kolam di daftar keuangan yang diinginkan.
Hasil yang diharapkan	Sistem menampilkan menu detail keuangan berupa daftar pemasukan, daftar pengeluaran, jumlah selisih dan status "Netral".
Hasil	Sistem menampilkan menu detail keuangan berupa daftar pemasukan, daftar pengeluaran, jumlah selisih dan status "Netral".
Status	Valid

**Tabel 6.50 Kasus uji melihat detail keuangan
Alternatif 4: bila selisih pengeluaran lebih besar dari pemasukan.**

Nama Kasus Uji	Kasus Uji Melihat Detail keuangan
Kode Kebutuhan	SIOBTIK_F_24
Prosedur	1. Aktor memilih nama kolam di daftar keuangan yang diinginkan.
Hasil yang diharapkan	Sistem menampilkan menu detail keuangan berupa daftar pemasukan, daftar pengeluaran, jumlah selisih dan status " Rugi".
Hasil	Sistem menampilkan menu detail keuangan berupa daftar pemasukan, daftar pengeluaran, jumlah selisih dan status " Rugi".
Status	Valid

Tabel 6.51 Kasus uji menghapus keuangan

Nama Kasus Uji	Kasus Uji Menghapus Keuangan
Kode Kebutuhan	SIOBTIK_F_25
Prosedur	1. Aktor menekan tombol hapus yang ada pada setiap daftar keuangan yang ada. 2. Aktor memilih tombol "Yes".
Hasil yang diharapkan	Sistem menghapus detail hasil keuangan yang dipilih oleh aktor.
Hasil	Sistem menghapus detail hasil keuangan yang dipilih oleh aktor.
Status	Valid



**Tabel 6.52 Kasus uji menghapus keuangan
Alternatif 1: bila aktor memilih "no"**

Nama Kasus Uji	Kasus Uji Menghapus Keuangan
Kode Kebutuhan	SIOBTIK_F_25
Prosedur	1. Aktor menekan tombol hapus yang ada pada setiap daftar keuangan yang ada. 2. Aktor memilih tombol "No".
Hasil yang diharapkan	Proses menghapus keuangan akan dibatalkan.
Hasil	Proses menghapus keuangan akan dibatalkan.
Status	Valid

Tabel 6.53 Kasus uji pemberitahuan kegiatan

Nama Kasus Uji	Kasus Uji Pemberitahuan kegiatan
Kode Kebutuhan	SIOBTIK_F_26
Prosedur	1. Terdapat kegiatan pada kalender kegiatan.
Hasil yang diharapkan	Sistem memberikan pemberitahuan ikan siap dipanen.
Hasil	Sistem memberikan pemberitahuan ikan siap dipanen.
Status	Valid

Pada pengujian validasi iterasi pertama dengan menggunakan metode pengujian *Black box*, dari semua kasus uji hasilnya adalah valid. Sehingga dapat disimpulkan bahwa sistem operasional budidaya tambak ikan kerapu (SIOBTIK) dapat berjalan dengan benar sesuai dengan kebutuhan yang telah dirancang sebelumnya.

6.3 Pengujian Validasi Iterasi Kedua

Pengujian validasi iterasi kedua dilakukan pada semua kebutuhan yang ditambahkan setelah dilakukan penambahan pada sistem dengan menggunakan metode pengujian *black box*. Pengujian ini akan dilakukan dengan cara melihat apakah sistem sudah berjalan dengan benar dan tidak terdapat *error* pada sistem. Pengujian validasi dilakukan untuk mengetahui sistem sudah berjalan sesuai dengan kebutuhan fungsional yang telah dibuat.

Tabel 6.54 Kasus uji menambah jadwal persiapan budidaya

Nama Kasus Uji	Kasus Uji Menambah Jadwal Persiapan Budidaya
Kode Kebutuhan	SIOBTIK_F_27



Tabel 6.54 Kasus uji menambah jadwal persiapan budidaya (lanjutan)

Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah jadwal persiapan budidaya. 2. Aktor memilih kolam, memilih kegiatan, memilih tanggal dan mengisi keterangan kegiatan persiapan budidaya. 3. Aktor menekan tombol “simpan”
Hasil yang diharapkan	Sistem menyimpan data ke dalam database dan menampilkan pesan “berhasil ditambah”.
Hasil	Sistem menyimpan data ke dalam database dan menampilkan pesan “berhasil ditambah”.
Status	Valid

**Tabel 6.55 Kasus uji menambah jadwal kegiatan persiapan budidaya
Alternatif 1: Bila sistem gagal menyimpan**

Nama Kasus Uji	Kasus Uji Menambah Jadwal Persiapan Budidaya
Kode Kebutuhan	SIOBTIK_F_27
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah jadwal persiapan budidaya. 2. Aktor memilih kolam, memilih kegiatan, memilih tanggal dan mengisi keterangan kegiatan persiapan budidaya. 3. Aktor menekan tombol simpan
Hasil yang diharapkan	Sistem tidak berhasil menyimpan data ke dalam database dan menampilkan pesan “tidak berhasil ditambah”.
Hasil	Sistem tidak berhasil menyimpan data ke dalam database dan menampilkan pesan “tidak berhasil ditambah”.
Status	Valid

**Tabel 6.56 Kasus uji menambah jadwal kegiatan persiapan budidaya
Alternatif 2: Salah satu data kosong**

Nama Kasus Uji	Kasus Uji Menambah Jadwal Persiapan Budidaya
Kode Kebutuhan	SIOBTIK_F_27
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah jadwal persiapan budidaya. 2. Aktor memilih kolam, memilih kegiatan, memilih tanggal. 3. Aktor menekan tombol simpan
Hasil yang diharapkan	Sistem akan menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.



Tabel 6.56 Kasus uji menambah jadwal kegiatan persiapan budidaya (lanjutan)

Hasil	Sistem menampilkan pesan pada <i>textbox</i> yang kosong berupa “data masih kosong”.
Status	Valid

Tabel 6.57 Kasus uji melihat jadwal persiapan budidaya

Nama Kasus Uji	Kasus Uji Melihat Jadwal Persiapan Budidaya
Kode Kebutuhan	SIOBTIK_F_28
Prosedur	1. Aktor memilih jadwal persiapan budidaya
Hasil yang diharapkan	Sistem menampilkan daftar jadwal persiapan budidaya.
Hasil	Sistem menampilkan daftar jadwal persiapan budidaya.
Status	Valid

Tabel 6.58 Kasus uji menghapus jadwal persiapan budidaya

Nama Kasus Uji	Kasus Uji Menghapus Jadwal Persiapan Budidaya
Kode Kebutuhan	SIOBTIK_F_30
Prosedur	1. Aktor menambahkan data pembibitan.
Hasil yang diharapkan	Sistem menghapus data persiapan budidaya pada kolom yang sama.
Hasil	Sistem menghapus data persiapan budidaya pada kolom yang sama.
Status	Valid

Pada pengujian validasi iterasi kedua dengan menggunakan metode pengujian *Black box*, dari semua kasus uji yang ditambahkan hasilnya adalah valid. Sehingga dapat disimpulkan bahwa sistem operasional budidaya tambak ikan kerapu (SIOBTIK) dapat berjalan dengan benar sesuai dengan kebutuhan yang telah dirancang sebelumnya.

6.4 Pengujian Non-Fungsional

Pada pengujian non-fungsional penulis menggunakan pengujian usability, pengujian ini digunakan untuk mengetahui tingkat kepuasan pengguna sistem dalam menggunakan sistem yang dibuat. Pengujian *usability* yang diterapkan pada sistem ini yaitu dengan menyebarkan kuesioner. Kuesioner ini berupa pertanyaan tentang bagaimana sistem dapat digunakan oleh pengguna. Metode pembuatan kuesioner menggunakan metode *USE questionnaire*, metode ini dapat mengukur tingkat kepuasan pengguna dalam menggunakan sistem. Kuesioner ini akan diajukan kepada lima petani tambak ikan kerapu yang ada di kelompok tani Bhakti Usaha II, para responden akan memberikan penilaian terhadap *usability* sistem dengan memilih salah satu dari lima nilai pada masing-masing pertanyaan.



Skala penilaian terdiri dari: Sangat Tidak Setuju (STS), Tidak Setuju (ST), Kurang Setuju (KS), Setuju (S), Sangat Setuju (SS).

6.4.1 Hasil Pengujian *Usability* Iterasi Pertama

Dari hasil penyebaran kuesioner kepada lima petani yang ada di kelompok tani Bhakti Usaha II, pada perhitungan setiap sekala memiliki nilai sebagai berikut: STS = 1, TS = 2, KS = 3, S = 4, SS = 5. Nilai tersebut akan dikalikan dengan jumlah pilihan nilai dari responden. maka hasilnya dapat dilihat pada Tabel 6.59.

Tabel 6.59 Hasil penyebaran kuesioner iterasi pertama

No	Pernyataan	STS	TS	KS	S	SS	Total
<i>Usefulness</i>							
1	Aplikasi ini membantu saya menjadi lebih efektif dalam melakukan budidaya tambak ikan kerapu.				2	3	23
2	Aplikasi ini sangat berguna bagi petani tambak ikan kerapu.			1	2	2	21
3	Aplikasi ini membuat saya lebih mudah dalam melakukan operasional budidaya tambak ikan kerapu.			1	3	1	20
4	Aplikasi ini sesuai dengan yang saya harapkan dalam mengontrol operasional budidaya tambak ikan kerapu.			2	2	1	19
<i>Ease of use</i>							
5	Aplikasi ini mudah digunakan.		1	2	1	1	19
6	Aplikasi ini dapat digunakan dengan sederhana.			2	3		18
7	Aplikasi ini dapat digunakan oleh semua petani tambak ikan kerapu.				3	2	22
8	Aplikasi ini memerlukan langkah-langkah sederhana untuk dapat mencapai apa yang saya inginkan.			1	2	2	21
9	Saya dapat menggunakan aplikasi ini tanpa instruksi tertulis.	1	1	2	1		13
10	Saya tidak menemukan inkonsistensi selama menggunakan aplikasi.		1	1	2	1	16
11	Saya dapat memperbaiki kesalahan dalam penggunaan aplikasi dengan cepat dan mudah			1	3	1	20
<i>Ease of learning</i>							
12	Saya dapat belajar menggunakan aplikasi ini dengan cepat.			1	1	3	22
13	Saya mudah mengingat bagaimana menggunakan aplikasi.			2	3		18



Tabel 6.59 Hasil penyebaran kuesioner iterasi pertama (lanjutan)

No	Pernyataan	STS	TS	KS	S	SS	Total
14	Aplikasi ini dapat saya gunakan dengan mudah.			1	3	1	20
15	Saya dengan cepat menguasai penggunaan aplikasi ini.			2	3		18
<i>Satisfaction</i>							
16	Saya puas dengan aplikasi ini.		1		2	2	20
17	Saya akan merekomendasikan aplikasi ini kepada petani ikan kerapu yang lain.		1	1	2	1	18
18	Aplikasi ini menyenangkan untuk digunakan.			2	1	2	20
19	Aplikasi ini bekerja seperti yang saya inginkan.			1	1	3	22
20	Aplikasi ini sangat bagus untuk digunakan dalam operasional budidaya tambak ikan kerapu.			1	2	2	21
21	Saya ingin memiliki aplikasi ini.				3	2	22
22	Aplikasi ini nyaman digunakan			2	1	2	20

6.4.2 Analisis Hasil Pengujian *Usability* Iterasi Pertama

Dari hasil pengujian *usability* akan didapati analisis rata-rata dan persentase dari setiap aspek kuesioner. Pada proses analisis pengujian *usability* akan dilakukan interpretasi skornya. Tabel 6.60 merupakan tabel merupakan acuan tingkatan interpretasi skor dari hasil pengujian.

Tabel 6.60 Acuan tingkatan interpretasi skor

No.	Interpretasi Skor Dengan Interval = 20	Status
1	0% - 19,99%	Sangat Tidak Memuaskan
2	20% - 39,99%	Tidak Memuaskan
3	40% - 59,99%	Biasa
4	60% - 79,99%	Memuaskan
5	80% - 100%	Sangat Memuaskan

Tabel 6.61 Status hasil pengujian *usability* iterasi pertama

Aspek Penilaian	Rata-Rata Skor	Presentase (%)	Status
<i>Usefulness</i>	20,8	83	Sangat Memuaskan
<i>Ease of use</i>	18,4	73,7	Memuaskan
<i>Ease of learning</i>	19,5	78	Memuaskan
<i>Satisfaction</i>	20,4	81,7	Sangat Memuaskan
Nilai <i>Usability</i>	19,7	79,1	Memuaskan

Dari tabel 6.61 dapat di analisis bahwa pada aspek *ease of use* (kemudahan dalam penggunaan) dan *ease of learning* (kemudahan dalam belajar) memiliki rata-rata skor dibawah 20 dengan persentase 73,7% dan 78% dengan status memuaskan, sedangkan pada aspek lain memiliki rata-rata skor diatas 20 dan persentase diatas 80% dengan status sangat memuaskan. Sehingga didapati nilai *usability* sebanyak 79,1% dengan status memuaskan. Hal ini dapat disimpulkan bahwa pada iterasi pertama sistem operasional budidaya tambak ikan kerapu (SIOBTIK) memiliki kekurangan dalam kemudahan penggunaan dan kemudahan pembelajaran, sedangkan dari aspek kegunaan dan kepuasan didapatkan hasil yang sangat memuaskan.

6.4.3 Hasil Pengujian *Usability* Iterasi Kedua

Dari hasil penyebaran kuesioner pada iterasi kedua kepada lima petani yang ada di kelompok tani Bhakti Usaha II, pada perhitungan setiap sekala memiliki nilai sebagai berikut: STS = 1, TS = 2, KS = 3, S = 4, SS = 5. Nilai tersebut akan dikalikan dengan jumlah pilihan nilai dari responden. maka hasilnya dapat dilihat pada Tabel 6.62.

Tabel 6.62 Hasil penyebaran kuesioner iterasi kedua

No	pernyataan	STS	TS	KS	S	SS	Total
<i>Usefulness</i>							
1	Aplikasi ini membantu saya menjadi lebih efektif dalam melakukan budidaya tambak ikan kerapu.					5	25
2	Aplikasi ini sangat berguna bagi petani tambak ikan kerapu.				1	4	24
3	Aplikasi ini membuat saya lebih mudah dalam melakukan operasional budidaya tambak ikan kerapu.					5	25
4	Aplikasi ini sesuai dengan yang saya harapkan dalam mengontrol operasional budidaya tambak ikan kerapu.				4	1	21
<i>Ease of use</i>							

Tabel 6.62 Hasil penyebaran kuesioner iterasi kedua (lanjutan)

No	pernyataan	STS	TS	KS	S	SS	Total
5	Aplikasi ini mudah digunakan.				4	1	21
6	Aplikasi ini dapat digunakan dengan sederhana.				4	1	21
7	Aplikasi ini dapat digunakan oleh semua petani tambak ikan kerapu.				2	3	23
8	Aplikasi ini memerlukan langkah-langkah sederhana untuk dapat mencapai apa yang saya inginkan.				2	3	23
9	Saya dapat menggunakan aplikasi ini tanpa instruksi tertulis.	1		3	1		14
10	Saya tidak menemukan inkonsistensi selama menggunakan aplikasi.	1		1	2	1	17
11	Saya dapat memperbaiki kesalahan dalam penggunaan aplikasi dengan cepat dan mudah			1	3	1	20
<i>Ease of learning</i>							
12	Saya dapat belajar menggunakan aplikasi ini dengan cepat.			1	1	3	22
13	Saya mudah mengingat bagaimana menggunakan aplikasi.			1	3	1	20
14	Aplikasi ini dapat saya gunakan dengan mudah.				4	1	21
15	Saya dengan cepat menguasai penggunaan aplikasi ini.			1	3	1	20
<i>Satisfaction</i>							
16	Saya puas dengan aplikasi ini.				3	2	22
17	Saya akan merekomendasikan aplikasi ini kepada petani ikan kerapu yang lain.				2	3	23
18	Aplikasi ini menyenangkan untuk digunakan.			1	1	3	22
19	Aplikasi ini bekerja seperti yang saya inginkan.				1	4	24
20	Aplikasi ini sangat bagus untuk digunakan dalam operasional budidaya tambak ikan kerapu.				2	3	23
21	Saya ingin memiliki aplikasi ini.				1	4	24
22	Aplikasi ini nyaman digunakan				1	4	24

6.4.4 Analisis Hasil Pengujian Usability

Dari hasil pengujian *usability* iterasi kedua akan didapat analisis rata-rata dan persentase dari setiap aspek kuesioner. Pada proses analisis pengujian



usability akan dilakukan interpretasi skornya. Tabel 6.63 merupakan hasil pengujian *usability*.

Tabel 6.63 Status hasil pengujian *usability*

Aspek Penilaian	Rata-Rata Skor	Presentase (%)	Status
<i>Usefulness</i>	23,8	95	Sangat Memuaskan
<i>Ease of use</i>	19,9	79,4	Memuaskan
<i>Ease of learning</i>	20,8	83	Sangat Memuaskan
<i>Satisfaction</i>	23,1	92,6	Sangat Memuaskan
Nilai <i>Usability</i>	21,9	87,5	Sangat Memuaskan

Dari tabel 6.63 dapat di analisis bahwa pada aspek *ease of use* (kemudahan dalam penggunaan) memiliki rata-rata skor terendah yaitu 19,9 dengan persentase 79,4% dengan status memuaskan, sedangkan pada aspek lain memiliki rata-rata skor diatas 20 dan persentase diatas 80% dengan status sangat memuaskan. Sehingga didapati nilai *usability* sebanyak 87,5% dengan status sangat memuaskan. Hal ini dapat disimpulkan bahwa sistem operasional budidaya tambak ikan kerapu (SIOBTIK) memiliki sedikit kekurangan dalam kemudahan penggunaan, tetapi dari keseluruhan sistem ini dapat dinilai sangat memuaskan bagi pengguna sistem.



BAB 7 PENUTUP

7.1 Kesimpulan

Dari hasil penelitian rancang bangun sistem operasional budidaya tambak ikan kerapu berbasis android (studi kasus: kelompok tani Bhakti Usaha 2), dapat diperoleh kesimpulan sebagai berikut:

1. Berdasarkan hasil analisis kebutuhan, maka didapatkan kebutuhan fungsional sebanyak 26 kebutuhan pada iterasi pertama dan 3 tambahan kebutuhan pada iterasi kedua. Kebutuhan non fungsional yang didapat adalah 1 kebutuhan. Pada analisis kebutuhan sistem dimodelkan menggunakan *use case* diagram, pemodelan *use case* menghasilkan 26 pada iterasi pertama dan 29 *use case* pada iterasi kedua yang masing-masing dari *use case* dijelaskan secara lebih detail dengan menggunakan *use case* skenario.
2. Berdasarkan dari hasil perancangan, didapatkan perancangan *sequence* diagram yang dapat menjelaskan urutan proses serta interaksi antara objek pada ketiga sampel kegiatan fungsionalitas sistem. Perancangan basis data yang dirancangan menggunakan *Physical Data Model* (PDM), perancangan basis data ini menghasilkan enam tabel yaitu tabel kolam, kegiatan, perancangan, pengeluaran, pemasukan dan panen. Perancangan *class* diagram menghasilkan perancangan kelas yang nantinya akan diterapkan dalam pembuatan sistem. Perancangan algoritme didapatkan *pseudocode* dari tiga sampel yang digunakan pada perancangan *sequence* diagram, yang nantinya akan digunakan sebagai acuan dalam proses implementasi kode program. Sedangkan pada perancangan antarmuka menghasilkan 16 *wireframe* yang dapat menggambarkan keseluruhan antarmuka setiap halaman pada sistem yang dibuat setelah dilakukannya iterasi kedua.
3. Berdasarkan dari hasil implementasi, didapatkan spesifikasi perangkat keras dan perangkat lunak yang digunakan oleh penulis dalam melakukan implementasi sistem. dalam proses implementasi juga didapatkan implementasi basis data dari perancangan basis data, selain itu juga dihasilkan implementasi kode program yang didapatkan dari perancangan algoritma, pada implementasi antarmuka didapat juga melalui perancangan antarmuka.
4. Berdadarkan dari hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa pada pengujian unit sistem, dengan menggunakan metode *white box testing* yang dilakukan pada tiga sampel algoritme, menghasilkan nilai valid pada semua kasus uji. Pada pengujian validasi dengan menggunakan metode *black box testing* didapati hasil valid pada setiap fungsionalitas pada iterasi pertama maupun iterasi kedua. Sedangkan pada pengujian *usability* didapati bahwa sistem kurang sempurna dalam hal kemudahan penggunaan, tetapi sistem masih mendapatkan hasil sangat memuaskan dari pengguna sistem.



7.2 Saran

Saran yang dapat penulis berikan untuk pengembangan sistem operasional budidaya tambak ikan kerapu selanjutnya adalah:

1. Sistem dapat dikembangkan lagi untuk perhitungan kondisi *Power of Hydrogen* (PH) kolam untuk akurasi kegiatan persiapan budidaya.
2. Sistem dapat dikembangkan lagi dalam proses budidaya sehingga sistem dapat memberikan jadwal kegiatan yang lebih detail.
3. Sistem dapat dikembangkan dengan fitur pengecekan kondisi ikan dan pemberian obat saat ikan dalam kondisi yang tidak baik.



DAFTAR PUSTAKA

- Badan Pusat Statistik, 2016. *Statistik Perusahaan Perikanan*. Jakarta, Badan Pusat Statistik Republik Indonesia.
- Dumas, J. S. & Radish, J. C., 1999. *A Practical Guide to Usability Testing*. Revised penyunt. Oregon: Intellect.
- Kementerian Pertanian, 2013. *Peraturan Menteri Pertanian Nomor 83*. [Online] Available at: <http://perundangan.pertanian.go.id/> [Diakses 2018].
- Kementerian Kelautan dan Perikanan, 2016. *Analisis Data Pokok Kelautan dan Perikanan 2016*. Jakarta, Kementerian Kelautan dan Perikanan Republik Indonesia.
- Kementerian Kelautan dan Perikanan, 2018. *Produktivitas Perikanan Indonesia*. Jakarta, Kementerian Kelautan dan Perikanan Republik Indonesia.
- Lund, A., 2001. *Measuring Usability with the USE Questionnaire*. 8 penyunt. s.l.:Usability and User Experience Newsletter of the STC Usability SIG.
- Paruntu, C. P., 2015. Budidaya Ikan Kerapu (*Epinephelus tauvina* Forsskal, 1775) dan Ikan Beronang (*Siganus canaliculatus* Park, 1797) dalam Karamba Jaring Apung dengan Sistem Polikultur. *Jurnal Budidaya Perairan Januari 2015*, 3(Fakultas Perikanan dan Ilmu Kelautan Universitas Sam Ratulangi), pp. 1-2.
- Pressman, R., 2001. *Software Engineering*. 5 penyunt. s.l.:McGraw Hill.
- Pressman, R. S., 1997. *Rekayasa Perangkat Lunak*. 1 penyunt. Yogyakarta: Andi.
- Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach*. 7 penyunt. New York: McGraw-Hill.
- Rahman, F., Husni, M. & Hidayat, W., 2016. *Aplikasi Peternakan Ayam Broiler Berbasis Android*, s.l.: Universitas Telkom.
- Rizky, S., 2011. *Konsep Dasar Rekayasa Perangkat Lunak*. Jakarta: Prestasi Pustaka.
- Safaat, N. H., 2012. *Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Shalahuddin, M. & Rosa, A. S., 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Sidik, F., 2018. *Bisnis.com*. [Online] Available at: <http://industri.bisnis.com/read/20180201/101/733037/pengguna-perangkat-mobile-di-indonesia-semakin-tinggi-ini-datanya> [Diakses 11 Oktober 2018].
- Society, I. C., 1990. *Standart Glossary of Software Engineering Terminology*. New York, IEEE Press.



Sommerville, I., 2011. *Software Engineering*. 9 penyunt. s.l.:Pearson Education, Inc.

Usaha, B., 2014. *Bhakti Usaha II kelompok tani tambak labuhan*. [Online] Available at: <https://bhaktiusaha2.wordpress.com/2014/06/20/sejarah-dan-perkembangan-usaha/> [Diakses 29 Agustus 2018].

Utama, A. S. C., 2015. *Rancang Bangun Aplikasi Mobile Panduan Budidaya Tanaman Cabai Organik Secara Mandiri*, Malang: Filkom Universitas Brawijaya.

Whitten, J. L., Bentley, L. D. & Dittman, K. C., 2004. *Systems Analysis and Design Methods*. 6 penyunt. s.l.:McGraw-Hill.