

**PENGEMBANGAN APLIKASI *MONITORING* KANDANG BURUNG
WALET BERBASIS *PROGRESSIVE WEB APPS* TERINTEGRASI
INTERNET OF THINGS MENGGUNAKAN ICONIX PROCESS
(Studi Kasus di Startup Swiftlead)**



SKRIPSI

**Disusun Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Komputer
pada Program Studi Informatika**

Disusun oleh:

Kristian David Adi Prasetya

24060121130049

**DEPARTEMEN INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO**

2025

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : Kristian David Adi Prasetya

NIM : 24060121130049

Judul : Pengembangan Aplikasi *Monitoring* Berbasis *Progressive Web Apps* Terintegrasi *Internet of Things* Menggunakan Metode Iconix Process (Studi Kasus di Startup Swiftlead)

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan di dalam daftar pustaka.

Semarang, 31 Januari 2025

Materai
Rp. 10.000,-

Kristian David Adi Prasetya

24060121130049

HALAMAN PENGESAHAN

Judul : Pengembangan Aplikasi Monitoring Berbasis Progressive Web Apps Terintegrasi
Internet of Things Menggunakan Metode Iconix Process (Studi Kasus di Startup
Swiftlead)

Nama : Kristian David Adi Prasetya

NIM : 24060121130049

Telah diujikan pada sidang tugas akhir dan dinyatakan lulus pada tanggal

Semarang, 31 Januari 2024

Pembimbing 1,

Pembimbing 2,

Nurdin Bahtiar, S.Si., M.T.

NIP. 197907202003121002

Rismaniyati, B.Eng, M.Cs

NIP. 198511252018032001

Mengetahui,

Ketua

Departemen Informatika

Ketua

Panitia Penguji Tugas Akhir

Dr. Aris Sugiharto, S.Si., M.Kom.

NIP. 197108111997021004

Nurdin Bahtiar, S.Si., M.T.

NIP. 197907202003121002

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas berkat-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “*Pengembangan Aplikasi Monitoring Berbasis Progressive Web Apps Terintegrasi Internet of Things Menggunakan Metode Iconix Process*”. Penelitian ini berfokus pada pengembangan perangkat lunak berbasis *progressive web apps* untuk melakukan pengawasan di kandang sarang burung walet dengan integrasi *Internet of Things* sebagai sensornya. Perangkat lunak ini dikembangkan dengan metode ICONIX Process, mulai dari analisis kebutuhan hingga implementasi sistem.

Ucapan terima kasih penulis ucapkan kepada orang tua tercinta Alm. Sarimun dan Kuat Andayani atas doa yang selalu menyertai penulis. Tidak lupa terima kasih juga penulis ucapkan kepada Bapak Nurdin Bahtiar, S.Si., M.T dan Ibu Rismiyati, B.Eng, M.Cs., selaku dosen pembimbing, Bapak Prof. Dr. Kusworo Adi, S.Si., M.T., selaku Dekan Fakultas Sains dan Matematika, Bapak Dr. Aris Sugiharto, S.Si., M.Kom., selaku Ketua Program Studi Informatika, juga teman-teman seperjuangan yang tiada lelah memberikan semangat dan bantuan.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, oleh karena itu kritik dan saran yang membangun sangat diharapkan untuk menjadikan skripsi ini lebih baik. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan sistem informasi di masa depan dan menjadi kontribusi positif dalam bidang teknologi informasi.

Semarang, 20 Januari 2025

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Universitas Diponegoro, saya yang bertanda tangan di bawah ini:

Nama : Kristian David Adi Prasetya
NIM : 24060121130049
Program Studi : Informatika
Departemen : Informatika
Fakultas : Sains dan Matematika
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** kepada Universitas Diponegoro atas karya ilmiah saya yang berjudul:

Pengembangan Aplikasi Monitoring Berbasis Progressive Web Apps Terintegrasi Internet of Things Menggunakan Metode ICONIX Process (Studi Kasus di Startup Swiftlead)

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-eksklusif ini Universitas Diponegoro berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Semarang, 31 Januari 2025

Materai
Rp. 10.000,-

Kristian David Adi Prasetya
24060121130049

ABSTRAK

Bagian ini adalah abstrak yang digunakan sebagai sarana untuk menyampaikan benang merah dari isi naskah tugas akhir. Diharapkan dengan membaca abstrak ini, pembaca dapat mengerti secara umum makna apa yang terkandung dalam naskah tugas akhir yang disampaikan oleh penulis. Abstrak bukan hanya menyajikan simpulan, melainkan suatu bentuk ringkas lain dari naskah tugas akhir yang disajikan dalam sebuah paragraf. Ditulis satu spasi, tanpa adanya indentasi di baris pertama, maksimal sebanyak 300 kata. Pada umumnya, bagian abstrak ini setidaknya memiliki 4 (empat) kandungan, yaitu WHY (menjelaskan tentang urgensi atau latar belakang penulisan), WHAT (menjelaskan hal apa yang dihasilkan), HOW (menjelaskan tentang bagaimana proses implementasinya), dan CONCLUSION (menjelaskan kesimpulan yang didapat).

Kata kunci : Data Mining, Naïve Bayes, loyalitas, entropi

ABSTRACT

Bagian ini adalah abstrak yang diterjemahkan ke dalam bahasa Inggris. Bagian ini tidak ditulis miring meskipun menggunakan bahasa Inggris. Kaidah penulisannya sama persis seperti abstrak dalam bahasa Indonesia. Gunakan tata bahasa Inggris yang mudah dimengerti. Jangan mengartikan kalimat bahasa Indonesia ke dalam bahasa Inggris berdasarkan kata per kata (yang mungkin terjadi saat dilakukan transliterasi menggunakan mesin penterjemah), karena Anda akan mengalami kesulitan dalam mencari padanan kata yang cocok. Terjemahkanlah abstrak ke dalam bahasa Inggris secara maknawi dengan menggunakan kosa kata yang mudah. Hal ini dapat memudahkan Anda dalam membuat kalimat dalam bahasa Inggris yang dapat dimengerti pembaca. Kendatipun demikian, Anda tidak dilarang menggunakan alat bantu seperti mesin penterjemah, kamus, dan sebagainya.

Keywords : Data Mining, Naïve Bayes, loyalty, entropy

DAFTAR ISI

HALAMAN PERNYATAAN KEASLIAN SKRIPSI.....	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
BAB I PENDAHULUAN	1
1.1. Sub Bab 1.1	3
1.1.1. Sub Sub Bab 1.1.1. Ini contoh judul yang melewati satu baris sehingga membuat nomor halaman pada daftar isi harus turun ke bawah	5
1.1.2. Sub Sub Bab 1.1.2	7
1.2. Sub Bab 1.2	9
1.3. Sub Bab 1.3	11
1.3.1. Sub Sub Bab 1.3.1	13
1.3.1.1. Sub Sub Sub Bab 1.3.1.1	15
1.3.1.2. Sub Sub Sub Bab 1.3.1.2.	17
1.3.2. Sub Sub Bab 1.3.2	19
BAB II TINJAUAN PUSTAKA	23
BAB III METODOLOGI PENELITIAN.....	30
BAB IV HASIL DAN PEMBAHASAN.....	50
BAB V KESIMPULAN DAN SARAN.....	55
DAFTAR PUSTAKA.....	56
LAMPIRAN 1. Surat Ijin Pengambilan Data.....	57
LAMPIRAN 2. Tabel Pengujian Aplikasi.....	59

Petunjuk DAFTAR ISI

- ❖ Halaman ini harus dibuat secara otomatis menggunakan fasilitas *References* (atau sejenisnya) dalam aplikasi pengolah kata, melalui proses memasukkan **Table of Contents** (atau sejenisnya).
- ❖ Diiijinkan editing secara manual agar sesuai dengan format di atas.

Silahkan hapus tabel merah ini

Contents

KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Ruang Lingkup	3
1.6 Sistematika Penulisan	4
BAB II	5
TINJAUAN PUSTAKA	5
2.1 Progressive Web Apps	5
2.2 Internet of Things	6
2.3 Unified Modeling Language (UML)	7
2.3.1 Use Case Diagram	7
2.3.2 Robustness Diagram	8
2.3.3 Sequence Diagram	10
2.3.4 Class Diagram	11
2.4 Model View Controller (MVC)	6
2.5 ICONIX Process	12
2.5.1 Requirements	13
2.5.2 Analysis/Preliminary Design	14
2.5.3 Detailed Design	15
2.5.4 Implementation	16
2.6 Pengujian Perangkat Lunak	16
BAB III	18
METODOLOGI PENELITIAN	18

DAFTAR TABEL

Tabel 1.1. Perbandingan Nilai Koefisien x dan y pada Grafik Pengamatan	2
Tabel 1.2. Daftar Harga Komponen Komputer di Bhineka.com.....	17
Tabel 2.1. Nama-nama Petugas yang Bekerja di PT. Pos Indonesia.....	20
Tabel 4.1. Daftar Perangkat Keras yang Dibutuhkan.....	30

Petunjuk DAFTAR TABEL / DAFTAR GAMBAR

- ❖ Halaman ini harus dibuat secara otomatis menggunakan fasilitas *References* (atau sejenisnya) dalam aplikasi pengolah kata, melalui proses **Insert Table of Figures** (atau sejenisnya).
- ❖ Diijinkan editing secara manual agar sesuai dengan format di atas.

Silahkan hapus tabel merah ini

DAFTAR GAMBAR

Gambar 1.1. Tampilan Splash Screen Aplikasi Sistem Informasi Kesehatan Sapi Wilayah Propinsi Jawa Tengah	1
Gambar 2.1. Tampilan Menu Utama Perangkat Lunak SIPENDA	11
Gambar 2.2. Tampilan Menu Bantuan Perangkat Lunak SIPENDA	21
Gambar 3.1. Grafik Pengamatan Penggunaan Internet	27

BAB I

PENDAHULUAN

Bab pendahuluan ini membahas mengenai latar belakang masalah, rumusan masalah, tujuan dan manfaat, ruang lingkup, serta sistematika penulisan yang akan digunakan dalam dokumen skripsi ini.

1.1 Latar Belakang

Sarang burung walet merupakan suatu hasil ternak komoditas peternakan burung walet yang sering dijuluki sebagai emas putih Indonesia, terkenal karena memiliki khasiat yang tinggi, sarang burung walet sering digunakan sebagai bahan dalam produksi obat-obatan Tiongkok tradisional, suplemen kesehatan, kosmetik, dan produk makanan. Karena khasiat dan kelangkaan dari komoditas ini, harganya terkenal memiliki perbandingan yang signifikan dari hasil ternak komoditas peternakan lainnya, Indonesia merupakan negara penghasil utama sarang burung walet di dunia dengan persentase sebesar 78% secara global, dengan kontribusi terhadap nilai PDB (Produk Domestik Bruto) Indonesia sejumlah 590,57 juta dollar AS atau sebesar 9,7 Triliun Rupiah (BPS, 2024). Komoditas ini merupakan kontributor terbesar dalam sub-sektor ekspor hasil ternak Indonesia (Ditjen Peternakan dan Kesehatan Hewan).

Berdasarkan data yang diperoleh, peternak walet memiliki berbagai permasalahan dari sisi budidaya. Permasalahan tersebut didasari oleh faktor internal dan eksternal dari pelaku peternak walet sendiri, yaitu faktor internal mengenai manajemen yang bersifat konvensional dengan mengandalkan proses manual untuk memantau dan mengelola kandang burung, pengumpulan sarang, dan kondisi lingkungan.

Pendekatan manual memakan waktu, tidak efisien, dan rentan terhadap kesalahan, akibat dari permasalahan tersebut, tidak sedikit peternak walet yang mengalami kerugian secara finansial hingga mengalami kekosongan peternakan walet sehingga tidak mampu beroperasi kembali. Idealnya kondisi rumah burung walet yang baik mempertimbangkan variabel seperti suhu (27-29 derajat celcius); kelembaban (80-90%); tingkat kegelapan rumah yang meniru kondisi goa (80-90% atau kurang dari 10 lux; dan pemutaran suara

burung walet melalui tweeter, variabel tersebut apabila diperhatikan akan berdampak signifikan terhadap produktivitas sebuah peternak walet (Sampurna, 2017). Selain itu, faktor eksternal seperti meningkatnya permintaan pasar global namun terbatas oleh akses pemasaran atau rantai nilai yang belum komprehensif sangat berdampak bagi para peternak walet.

Dari permasalahan yang dihadapi, maka dibutuhkan sebuah sistem yang dapat membantu meningkatkan produktivitas burung walet dengan memberikan perhatian pada variabel-variabel yang berpengaruh pada tingkah laku burung walet. Dengan begitu, diusulkan sebuah aplikasi monitoring yang terintegrasi dengan sensor suhu dan kelembaban untuk meningkatkan tingkat pemantauan peternak walet terhadap kandang burung walet untuk mengoptimalkan hasil panen sarang burung walet.

Pendekatan yang dilakukan dalam pengembangan aplikasi ini yaitu dengan menggunakan metode pengembangan ICONIX Process. Metode ini menggunakan pemodelan berbasis objek dan menggunakan Unified Modelling Language sebagai alat bantu mendeskripsikan sistem perangkat lunak (Rosenberg, 2007). Metode ini dipilih karena mampu memberikan metode yang singkat sekaligus lengkap yang terstruktur dan terdokumentasi dengan baik menggunakan diagram UML yang ada. Dengan menggunakan metode ini, sistem dapat dibangun secara efisien dan meminimalisir adanya kesalahan karena memiliki langkah-langkah yang lebih terstruktur dan terdokumentasi dengan baik.

Dengan adanya aplikasi ini, diharapkan proses bisnis yang dilakukan oleh Swiftlead menjadi lebih efisien dan membantu para peternak walet dalam mengoptimalkan hasil panen sarang burung waletnya melalui pemantauan kandang burung walet yang lebih baik. Aplikasi juga diharapkan dapat membantu masyarakat secara umum dalam mengenali komoditas sarang burung walet serta dalam jangka panjang menumbuhkan ekosistem yang positif di kalangan peternak sarang burung walet.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan suatu masalah yaitu bagaimana mengembangkan sebuah aplikasi *monitoring* berbasis *progressive web apps* terintegrasi *Internet of Things* menggunakan metode ICONIX Process di Startup Swiftlead untuk membantu meningkatkan produktivitas hasil panen sarang burung walet.

1.3 Tujuan

Tujuan yang ingin dicapai adalah menghasilkan sebuah platform berbasis progressive web apps untuk membantu monitoring kandang burung walet yang terintegrasi dengan sensor suhu dan kelembaban. Platform ini dirancang untuk membantu peternak walet dalam monitoring kandang burung walet, pendataan hasil panen, artikel terkini, penjualan hasil panen, serta akses ke konten-konten membership. Pendekatan yang digunakan untuk pengembangan aplikasi ini adalah metode ICONIX Process, merupakan salah satu metode agile dan berorientasi objek yang digunakan untuk memastikan pengembangan sistem dari perancangan hingga pengujian berjalan secara terstruktur.

1.4 Manfaat

Manfaat yang didapatkan dari pengembangan aplikasi monitoring ini di antara lain sebagai berikut:

1. Bagi Perusahaan: Digitalisasi proses bisnis perusahaan, menyediakan platform untuk peternak walet, serta meningkatkan profit sesuai dengan visi perusahaan.
2. Bagi peternak walet: Meningkatkan efisiensi dan tingkat pengawasan terhadap kandang burung walet serta memudahkan penjualan sarang burung walet yang sesuai dengan harga pasar.
3. Bagi Masyarakat umum: Menyediakan platform literasi digital melalui artikel edukasi seputar burung walet.

1.5 Ruang Lingkup

Ruang lingkup bertujuan memberikan batasan agar tetap sesuai dengan tujuan penelitian dan kebutuhan pengguna. Ruang lingkup dari penelitian ini adalah sebagai berikut.

1. Sistem autentikasi untuk admin dan peternak walet.
2. Fitur monitoring mencakup fitur pengajuan instalasi, pengajuan maintenance, pengajuan uninstallasi, monitoring suhu dan kelembaban secara real-time, manajemen monitoring kandang, dan tampilan historikal suhu dan kelembaban.
3. Fitur pemanenan yang mencakup pre-harvest, post-harvest, pengajuan penjualan hasil panen, dan tampilan hasil panen secara historikal.
4. Fitur membership untuk pengguna yang terdaftar sebagai member agar dapat mengakses konten-konten premium yang disediakan.

5. Artikel edukasi seputar burung walet yang dapat diakses oleh masyarakat umum.
6. Panel admin untuk mengelola pengguna aktif, pengajuan-pengajuan, artikel, dan konten membership.
7. Fitur-fitur terkait perangkat keras IoT tidak termasuk ke dalam ruang lingkup tugas akhir ini.

1.6 Sistematika Penulisan

Sistematika penulisan sebagai acuan dalam penulisan penelitian dan mempermudah pembaca dalam memahami pembahasan yang terdapat di dalam penelitian. Sistematika penulisan dokumen ini yaitu.

BAB 1 PENDAHULUAN

Bab ini membahas latar belakang, rumusan masalah, tujuan, manfaat, ruang lingkup, serta sistematika penulisan laporan tugas akhir mengenai pengembangan aplikasi monitoring kandang burung walet.

BAB II LANDASAN TEORI

Bab ini menyajikan landasan teori yang menjadi dasar dalam penyusunan laporan tugas akhir ini, seperti bahasa pemrograman, framework yang digunakan, metode pengembangan, dan lain-lain.

BAB III METODOLOGI PENELITIAN

Bab ini berisi metode dan langkah-langkah yang dilakukan untuk menyelesaikan permasalahan pada penelitian ini yaitu metode pengembangan perangkat lunak, yang terdiri dari tahap requirement hingga tahap testing.

BAB IV HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil pembahasan yang dilakukan oleh penulis, yaitu proses pengembangan aplikasi monitoring kandang burung walet dari proses requirement hingga testing.

BAB V KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan dari bab-bab yang telah dibahas sebelumnya serta memberikan saran dan masukan untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

Bab ini menyajikan landasan teori yang menjadi dasar dalam penyusunan laporan tugas akhir ini, seperti bahasa pemrograman, framework yang digunakan, metode pengembangan, dan lain-lain.

2.1 Progressive Web Apps

Menurut Aripin (2021) Progressive Web Apps atau PWA adalah konsep penerapan pembuatan website yang menerapkan service worker, web manifest, dan cache API. Dengan membangun aplikasi menggunakan konsep PWA akan membuat aplikasi web berjalan di berbagai platform seperti website, desktop, dan platform mobile atau android. Beberapa keunggulan dari PWA sendiri memiliki berbagai keunggulan seperti:

1. PWA memiliki sifat aplikasi native layaknya android, tapi tidak memerlukan spesifikasi hardware yang tinggi dalam membuatnya.
2. PWA dapat mempercepat proses pembuatan website dan mengurangi beban server.
3. PWA dapat berfungsi secara offline, tidak seperti website yang tidak dapat diakses jika tidak ada koneksi internet.
4. PWA sudah banyak digunakan di perusahaan besar seperti X, Facebook, Tokopedia, Instagram, dan lain sebagainya.

Proses instalasi PWA pun terbilang cukup mudah, hanya dengan mengakses website PWA yang diinginkan kemudian memilih opsi install, maka aplikasi PWA pun sudah dapat digunakan dan diakses menggunakan perangkat apapun. Aplikasi PWA dibuat seringkali mungkin agar dapat menghasilkan pengalaman pengguna yang baik, pemrosesan yang cepat, dan serta menghabiskan penyimpanan yang relatif kecil. Penggunaan PWA juga disesuaikan dengan target pengguna aplikasi ini yaitu peternak walet yang sudah berumur.

Penggunaan PWA dalam penelitian ini didasari atas alasan kemudahan penggunaan dan efisiensi pengembangan. PWA dinilai dapat digunakan oleh peternak walet dimana pun karena sifatnya yang kompatibel dengan perangkat apapun, termasuk aplikasi mobile. Selain itu, pengembangan PWA juga lebih mudah dibanding dengan pengembangan aplikasi mobile itu sendiri, sehingga pengembangan PWA dinilai lebih cocok untuk startup tahap awal seperti Swiftlead.

2.2 Internet of Things

Internet of Things (IoT) merupakan sebuah konsep yang memperluas manfaat dari konektivitas internet melalui sensor cerdas yang saling berkomunikasi melalui jaringan yang tersambung. IoT dapat digunakan dalam berbagai bidang seperti monitoring suhu dan kelembaban, sensor deteksi gerakan, perekaman intensitas cahaya, dan lain sebagainya (Hidayat, 2021).

Seiring berkembangnya mikrokontroler, IoT semakin berkembang hingga modul berbasis ethernet maupun wifi pun juga semakin bervariasi. Salah satu modul berbasis wifi yang digunakan dalam penelitian ini adalah ESP32. Mikrokontroler ESP32 dipilih karena harganya yang terjangkau, fitur yang cukup lengkap, dan penggunaannya yang mudah dan sesuai kebutuhan. Selain itu, ESP32 juga dapat mengirimkan data secara realtime selama sudah tersambung ke listrik dan jaringan internet.

Sensor yang digunakan dalam penelitian ini merupakan sensor suhu dan kelembaban yang terhubung langsung ke mikrokontroler ESP32. Sensor yang digunakan mencakup DHT22, SHT40, dan HTU21D. Sensor-sensor ini menangkap suhu dan kelembaban lingkungan sekitarnya untuk dikirimkan ke dalam basis data realtime. Kemudian aplikasi akan mengambil data sensor tersebut untuk ditampilkan dalam dashboard pemantauan.

2.3 Model View Controller (MVC)

Model View Controller atau yang lebih sering disebut MVC merupakan sebuah arsitektur untuk merancang sebuah program. Tujuan utama dari arsitektur ini adalah membagi program ke dalam tiga bagian besar. Masing-masing dari bagian ini memiliki fungsi, fokus, tanggung jawab, dan logika yang berbeda sesuai dengan pembagiannya masing-masing (Kelen, 2018).

Berikut adalah detail fungsi dari masing-masing komponen pada arsitektur *Model View Controller*.

1. *Model*

Model adalah bagian dari aplikasi yang mengimplementasikan logika untuk domain data aplikasi. Umumnya, objek model digunakan untuk mengambil data dari basis data atau menyimpan data ke basis data.

2. *View*

Komponen *view* adalah komponen yang menampilkan antarmuka (*user interface*) untuk pengguna aplikasi. Antarmuka ini dibuat berdasarkan data dari *model* dan berdasarkan kenyamanan dari pengguna.

3. *Controller*

Controller adalah komponen digunakan untuk menangani interaksi dengan pengguna, bekerja dengan *model*, dan memilih *view* yang digunakan untuk menampilkan data.

Dengan memahami mekanisme MVC secara menyeluruh, pengembang perangkat lunak dapat membuat sebuah sistem dapat dengan mudah diskalakan dan adaptif terhadap perubahan dan pemeliharaan. Hal ini membuat arsitektur MVC seringkali digunakan dalam paradigma rekayasa perangkat lunak modern.

2.4 Unified Modeling Language (UML)

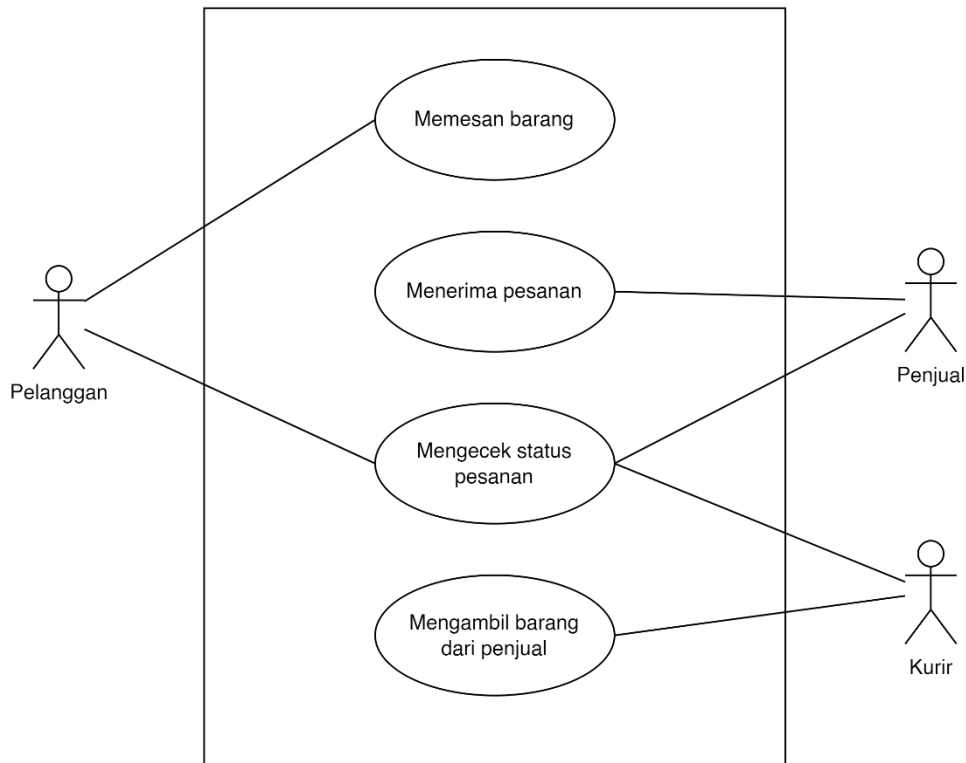
Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan dalam pengembangan sistem perangkat lunak. UML memiliki sintaks dan semantik yang harus diikuti saat membuat model serta mengatur bagaimana elemen-elemen dalam model berhubungan sesuai dengan standar yang telah ditetapkan. UML tidak hanya berfokus pada pembuatan diagram, tapi juga mampu menceritakan konteks dalam pengembangan sistem. Dengan menggunakan UML, pertanyaan mengenai penanganan kesalahan, keamanan sistem, dan aspek lainnya dapat dijawab. UML dapat memberikan pemahaman yang komprehensif dalam pengembangan perangkat lunak (Muslihudin, 2016).

2.4.1 Use Case Diagram

Use Case diagram adalah teknik yang digunakan untuk menyusun daftar functional requirements dari suatu sistem. Diagram ini berfokus pada penggambaran fungsionalitas dari suatu sistem, berfokus pada “apa” yang dilakukan oleh sebuah sistem daripada “bagaimana”. Setiap use case dalam diagram merepresentasikan hubungan antara aktor dengan sebuah sistem berupa perilaku atau tugas tertentu seperti autentikasi sebagai contohnya. Diagram ini membantu memvisualisasikan requirements suatu sistem, memfasilitasi komunikasi dengan stakeholder, serta merancang test case yang menyeluruh untuk semua fitur dalam sistem.

Dalam use case diagram, setiap simbol memiliki peran khusus dan mewakili arti tertentu. Aktor digambarkan dengan simbol manusia, merepresentasikan entitas eksternal

yang berinteraksi dengan sistem. Simbol oval melambangkan requirement atau fungsi tertentu dari sistem. Aktor dan oval dihubungkan dengan garis yang menunjukkan interaksi aktor terhadap skenario tertentu. Notasi garis ini menggambarkan hubungan antara aktor dan fungsionalitas tertentu yang relevan dengan salah satu skenario.



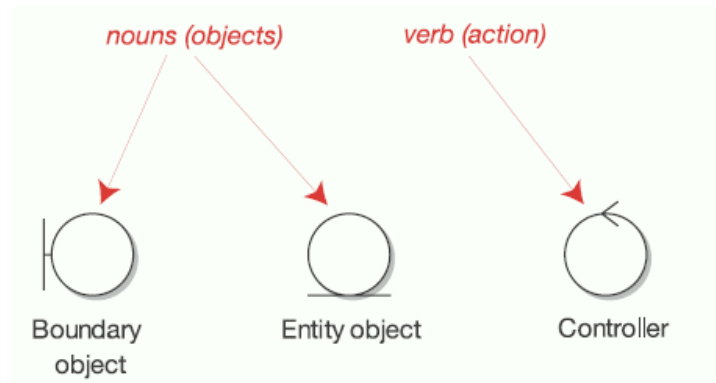
Gambar 2.1 Contoh use case diagram

Use case diagram memiliki kondisi lain selain penggunaan garis asosiasi, yaitu garis includee dan extend. Masing-masing garis ini mempunyai fungsi yang spesifik. Garis include dapat diartikan bahwa suatu use case membutuhkan fungsionalitas dari use case lain untuk melengkapi eksekusinya. Di sisi lain, extend menunjukkan suatu use case fungsionalitasnya dapat diperluas dengan use case lain atau dengan kata lain fitur opsional dari suatu use case. Notasi garis include dan extend dapat memperinci suatu use case agar pengembang dapat memahami fungsionalitas secara mendetail.

2.4.2 Robustness Diagram

Robustness diagram merupakan representasi dari perilaku yang dijelaskan oleh kasus penggunaan, menunjukkan kelas yang berpartisipasi dan perilaku perangkat lunak. Secara penggunaan, robustness diagram merupakan gabungan dari activity diagram dan class diagram. Menunjukkan perilaku suatu objek dengan kemampuan berkomunikasi dengan

objek lainnya. Garis yang menghubungkan antar objek menggambarkan interaksi atau komunikasi yang terjadi antar objek dalam robustness diagram.



Gambar 2.2 Simbol dalam robustness diagram

Pada robustness diagram, objek-objek dikelompokkan menjadi tiga jenis sesuai dengan fungsinya masing-masing.

1. Boundary Objects

Boundary objects merupakan objek yang mewakili tampilan antarmuka pengguna. Mencakup tampilan seperti formulir, dashboard, grafik, dan lain sebagainya. Objek ini berfokus pada tampilan dan disebut boundary atau batas karena merupakan batasan langsung antara sistem dengan pengguna.

2. Entity Objects

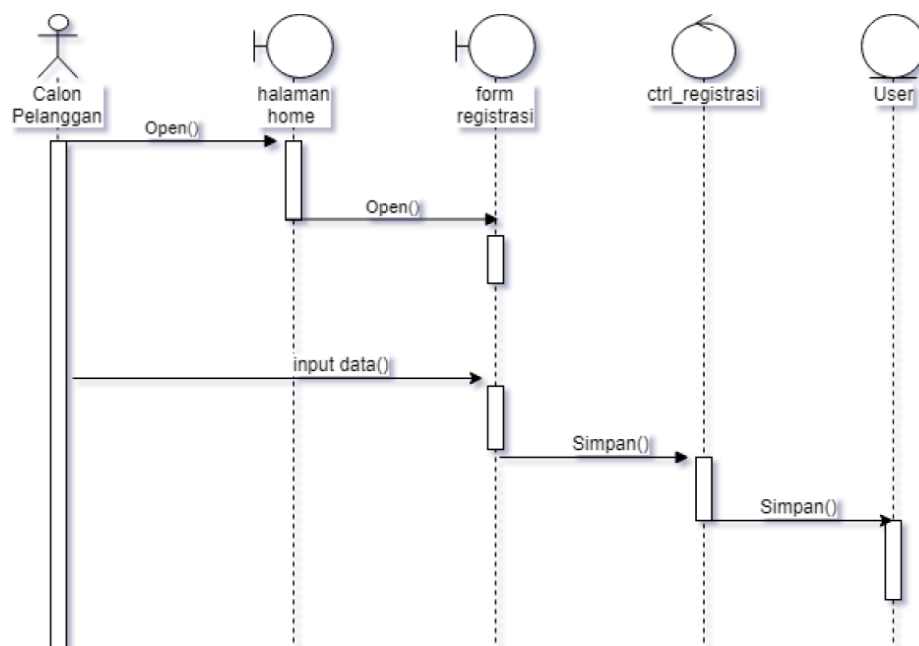
Entity Objek mewakili objek nyata dalam domain permasalahan dalam sistem yang sedang dibangun. Objek ini umumnya berkaitan satu sama lain dan membangun sebuah kesatuan sistem. Melalui objek-objek disimpan dalam bentuk entitas-entitas tertentu yang terlibat dalam sistem.

3. Controllers

Elemen objek controllers bertanggung jawab dalam mengatur aliran informasi antara boundary objects dan endtitiy objects. Controllers merupakan kumpulan fungsi-fungsi yang mengatur jalannya logika suatu sistem dalam mencapai tujuan tertentu. Controllers menginterpretasikan input dari boundary objects, memanipulasi object sesuai dengan instruksi dan logika yang diberikan, lalu mengirimkan hasil kembali kepada boundary objects.

2.4.3 Sequence Diagram

Sequence diagram merupakan salah satu diagram penting yang memvisualisasikan alur informasi mengalir melalui pertukaran antara aktor atau objek-objek dalam sistem. Jenis diagram ini memberikan gambaran yang komprehensif mengenai bagaimana pesan dikirim dan diterima dalam suatu sistem, sehingga berfokus pada aliran informasi dan interaksi antar objek dalam sistem. Keunggulan utama dari sequence diagram adalah kemampuannya untuk menunjukkan interaksi antara objek dalam urutan dan rentang waktu tertentu.

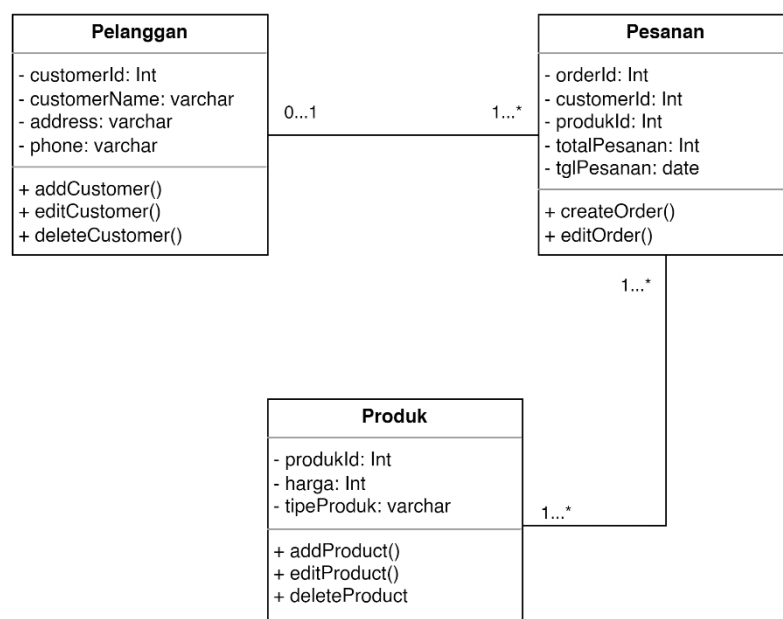


Gambar 2.3 Contoh sequence diagram

Dalam Sequence diagram, masing-masing simbol mewakili maksud tertentu. Aktor biasanya digambarkan dalam simbol orang, mewakili pengguna yang berinteraksi dengan sistem. Kemudian konsep yang sama dengan robustness, yaitu boundary objects, entity objects, dan controllers kembali dipakai untuk mewakili objek-objek yang berada dalam sistem. Selanjutnya, garis hidup berupa garis vertikal yang menandakan rentang waktu hidup suatu objek ketika berada dalam suatu interaksi. Panah horizontal antar objek menunjukkan pesan dan interaksi yang terjadi dalam rentang waktu tertentu. Pesan umumnya berisi label yang menunjukkan detail operasi atau metode yang dilakukan. Rangkaian proses ini membuat sequence diagram memfasilitasi pemahaman yang lebih baik mengenai alur kerja sebuah sistem.

2.4.4 Class Diagram

Class diagram merupakan salah satu diagram paling sering digunakan dalam pemodelan untuk pemrograman berbasis objek. Diagram ini menggambarkan suatu sistem dengan membagi-baginya menjadi objek tertentu. Class diagram memvisualisasikan konsep objek melalui kelas dengan struktur atribut, metode yang disertai dengan hubungan antar kelas. Dengan menggunakan simbol-simbol yang mencakup kotak dan garis, class diagram memberikan gambaran jelas mengenai hubungan antar kelas dalam suatu perangkat lunak tertentu.



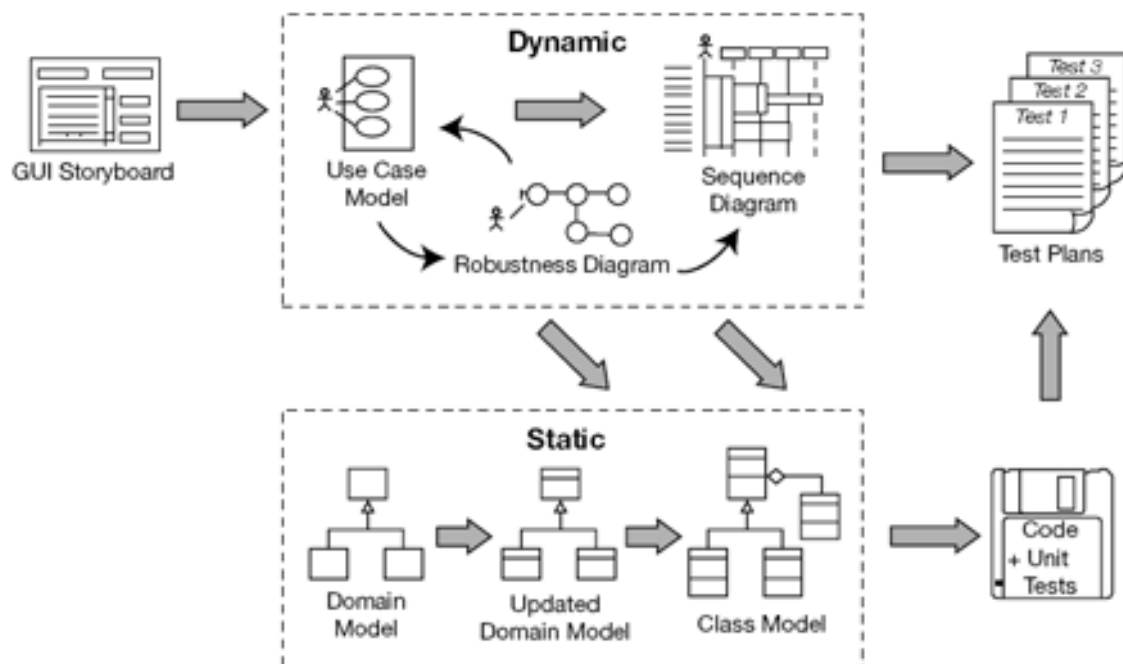
Gambar 2.4 Contoh class diagram

Class diagram umumnya digambarkan dengan kotak yang terbagi menjadi tiga bagian. Bagian atas berisi nama kelas, kotak tengah berisikan atribut yang dimiliki oleh kelas, serta baris terakhir berisi metode-metode yang dimiliki oleh kelas tersebut. Selain itu, pada class diagram juga dikenal sebuah istilah modifier yang menggambarkan kebebasan akses dari suatu atribut atau metode. Public atau simbol (+) menyimbolkan bahwa atribut atau metode bisa diakses dari mana saja. Protected disimbolkan dengan (#), artinya metode dapat diakses di dalam kelas dan oleh kelas yang berasal dari kelas tersebut. Terakhir, modifier private (-), berarti properti atau metode hanya dapat diakses di dalam kelas.

2.5 ICONIX Process

ICONIX Proses adalah salah satu metodologi perangkat lunak yang berorientasi objek dan berfokus pada penggunaan UML pada pengembangannya. ICONIX membawa prinsip analisis dan desain yang terstruktur, praktis, dan adaptif. ICONIX Process termasuk ke dalam salah satu kerangka kerja pengembangan perangkat lunak yang bersifat agile, yaitu proses pengembangan berdasarkan iterasi yang saling mendukung. Lebih lanjut, ICONIX Process dapat dilakukan seperti halnya agile dengan iterasi yang singkat dan cepat ataupun seperti waterfall yang membutuhkan kejelasan kebutuhan pengembangan perangkat lunak sebelum berpindah ke fase desain dan penulisan kode program.

Rosenberg & Stephens (2007) menerangkan bahwa fokus tersebut adalah menghilangkan kesenjangan antara konsep pemahaman awal use case dan implementasi kode program dengan fokus utamanya terletak pada tahapan analisis dan desain. ICONIX Process diharapkan dapat membantu tim pengembang lebih memahami use case dan langkah pengembang perangkat lunak sebelum memulai penulisan kode, sehingga dapat menghindari masalah dan kecacatan selama tahapan implementasi.



Gambar 2.5 ICONIX Process (Rosenberg, D., & Stephens, M., 2007)

Kelebihan dari ICONIX Process terletak pada keseimbangan antara fleksibilitas perubahan dan dokumentasi yang terstruktur. Metode ini menekankan penggunaan diagram UML untuk memodelkan berbagai komponen sistem, membuat komunikasi antar tim dan

pemangku kepentingan lebih mudah. Namun, proses ini bisa jadi terlalu berat bagi proyek kecil karena dokumentasi dan pemeliharaan diagram UML yang cukup banyak dan menyita sumber daya.

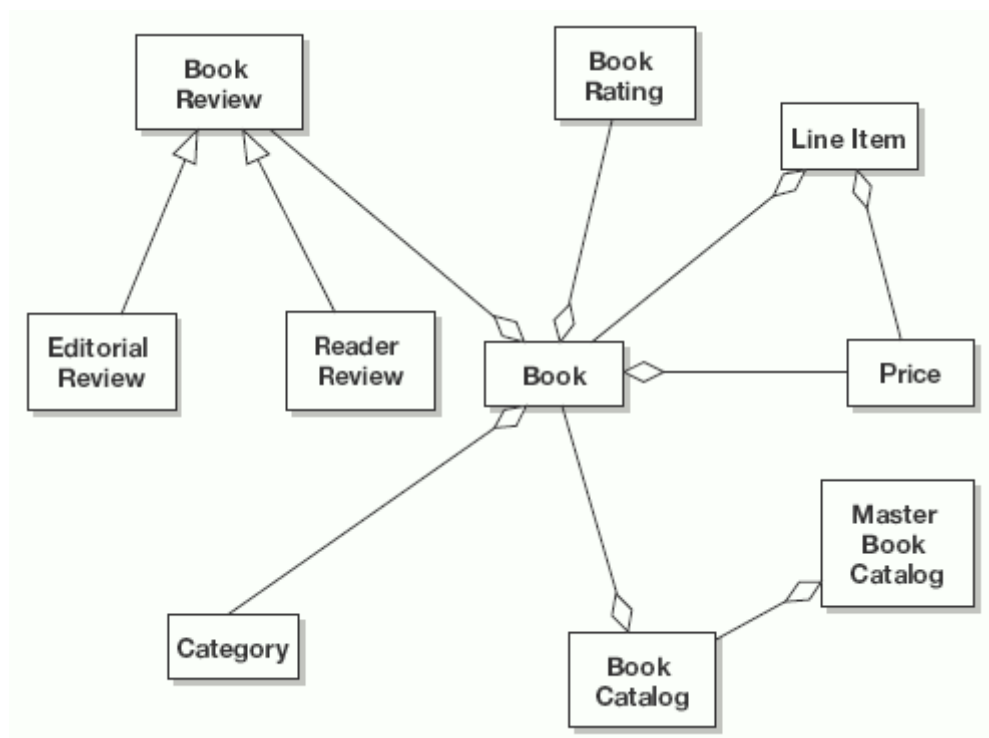
2.5.1 Requirements

Tahap pertama dalam ICONIX Process adalah pengumpulan kebutuhan pengguna. Tahap ini terbagi menjadi tiga bagian yaitu.

1. Functional Requirements

Bagian ini mendefinisikan hal-hal yang bisa dilakukan oleh sistem sesuai dengan kebutuhan oleh pengguna atau stakeholder. Kebutuhan fungsional ini berfokus pada hal-hal yang bisa dilakukan oleh sistem mencakup layanan, fitur, dan interaksi antara sistem dan pengguna. Tahap ini menjadi bagian paling penting dalam pengembangan sistem karena semua proses pengembangan sistem berawal dari tahap ini dan menjadikan tahap ini sebagai panduan. Functional requirements dirancang agar dapat diukur dan diuji selama proses pengembangan maupun pengujian akhir, serta memastikan bahwa sistem telah memenuhi kebutuhan pengguna yang telah ditetapkan. Proses ini meminimalisir adanya ketidaksesuaian kebutuhan pengguna dan hasil akhir sistem.

2. Domain Modeling



Gambar 2.6 Domain model diagram (Rosenberg, D., & Stephens, M., 2007)

Domain model merupakan tahap penentuan cakupan atau scope dari sebuah proyek guna membentuk dasar untuk membangun implementasi. Secara sederhana, domain modeling membantu pengembang dalam menghubungkan satu domain ke domain lain dalam sebuah sistem melalui representasi sebuah objek. Setiap objek dapat memiliki hubungan interaksi atau asosiasi dengan objek lain dalam sebuah sistem. Interaksi ini dapat berupa komunikasi data maupun fungsi.

3. Behavioral Requirements

Behavioral requirements mendefinisikan interaksi antara pengguna dan sistem dalam sebuah skenario tertentu. Dalam ICONIX Process, behavioral requirements berfokus pada bagaimana sistem merespon suatu perilaku pengguna untuk mencapai tujuan tertentu. Dalam mendokumentasikan behavioral requirements, use case diagram seringkali digunakan sebagai diagram yang menggambarkan antara perilaku pengguna dengan sistem.

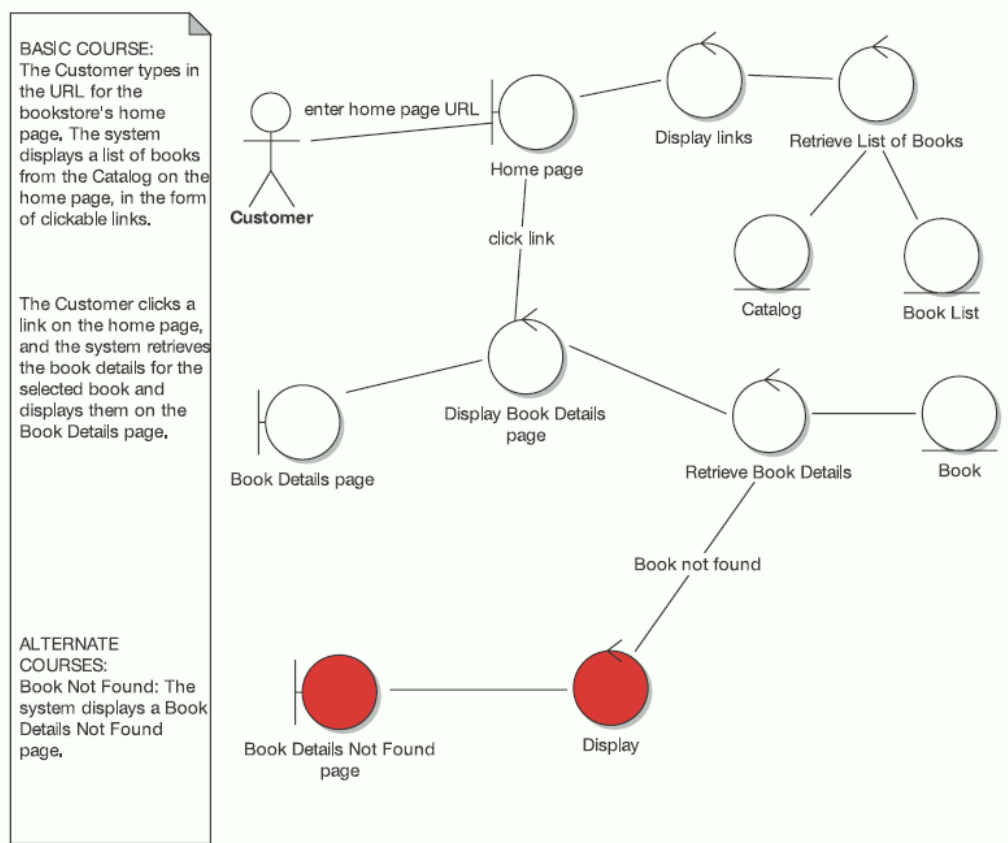
Setelah semua kebutuhan telah diidentifikasi, diperlukan requirements review yang meninjau semua use case telah memenuhi harapan stakeholder. Hal ini mencegah adanya ketidaksesuaian program akhir dengan ekspektasi stakeholder.

2.5.2 Analysis/Preliminary Design

Tahap analysis merupakan tahap yang berfokus pada desain eksploratif yang diperlukan untuk memahami kebutuhan, menyempurnakan dan menghilangkan ambiguitas dari kebutuhan tersebut. Tahap ini menjembatani antara proses analisis dengan desain agar sistem yang dibangun lebih tepat.

1. Robustness Analysis

Robustness analysis merupakan proses menjembatani kebutuhan fungsional melalui robustness diagram. Robustness diagram mengidentifikasi aktor, sistem, serta objek-objek dalam sistem kemudian memetakannya dalam sebuah interaksi antar objek. Dari robustness diagram ini sistem terbagi menjadi tiga elemen utama, yaitu entity objects, boundary objects, dan control objects. Masing-masing elemen ini memiliki fungsi yang berbeda, seperti entity objects merupakan domain atau abstraksi dari sebuah objek, boundary objects menggambarkan halaman antarmuka pengguna, dan control objects berfungsi sebagai logika, alur kerja, dan mengatur jalannya komunikasi antar objek dalam sistem.



Gambar 2.6 Robustness diagram (Rosenberg, D., & Stephens, M., 2007)

2. Updated Domain Model

Selanjutnya, tahap analysis berfungsi dalam memperbarui domain model untuk memastikan model tersebut tetap relevan dengan hasil analisis sebelumnya. Domain model diperbarui dengan menambahkan atribut-atribut yang berada dalam objek tertentu serta menjelaskan sifat dari objek itu sendiri. Hasil ini didapatkan dari analisis yang telah dilakukan sebelumnya.

2.5.3 Detailed Design

Tahap detailed design berisi pemodelan lebih lanjut menggunakan sequence diagram untuk menunjukkan bagaimana sistem bekerja serta mendefinisikan objek-objek secara lengkap melalui class diagram.

1. Sequence diagram

Sequence diagram menggambarkan sebuah use case yang dijelaskan secara mendetail melalui alokasi perilaku (behavior allocation). Dari sini akan tergambar interaksi antar objek dengan detail dan terperinci. Diagram ini akan menjelaskan setiap alur yang terjadi dalam sebuah use case diagram, sehingga perilaku pengguna dapat diketahui oleh

pengembang secara mendetail. Dalam sequence diagram alur dijelaskan melalui interaksi antar objek dan aktor melalui perpesanan dan lifelines.

2. Class Diagram

Class diagram adalah versi akhir dari pengembangan domain model dalam setiap tahap. Class diagram menggambarkan kelas, atribut, metode, dan hubungan antar kelas secara lengkap dan mendetail. Dalam class diagram, atribut merepresentasikan sifat dari sebuah kelas sedangkan metode menggambarkan fungsi logika yang dapat dilakukan oleh kelas tersebut. Hubungan antar kelas digambarkan melalui asosiasi, generalisasi, agregasi, atau komposisi. Class diagram menjadi bagian yang sangat krusial karena menjadi panduan pengembang dalam membuat basis data.

2.5.4 Implementation

Tahap implementasi dalam ICONIX Process meliputi tahapan teknis yang mengubah desain menjadi kode program yang dapat dijalankan. Tahapan ini mengimplementasikan apa yang telah dirancang pada tahap analisis dan desain sebelumnya dengan menerjemahkan use case diagram, sequence diagram, dan class diagram yang telah dibuat sebelumnya. Proses dibangun dengan arsitektur Model View Controller (MVC) dengan memperhatikan rancangan-rancangan yang telah dibuat sebelumnya. Melalui arsitektur MVC, program diharapkan menerapkan pemrograman berbasis objek yang baik seperti modularitas, enkapsulasi, serta pemisahan fungsi yang baik. Hal ini bertujuan agar sistem yang dikembangkan dapat dipelihara dan memiliki skalabilitas yang baik.

Pada tahap implementation ini, pengembang juga menerapkan logika yang efisien, memperhatikan alur kerja sistem, serta memastikan setiap unit dari sistem bekerja dengan baik. Untuk memastikan setiap unitnya berjalan dengan baik, pengembang dapat menuliskan unit testing untuk masing-masing unit yang tersedia. Proses implementasi juga mengintegrasikan semua komponen sistem baik itu antarmuka pengguna, basis data, maupun penggunaan layanan pihak ketiga seperti kerangka kerja, library, atau service yang tersedia. Pengembang harus memastikan semuanya yang terlibat dalam sistem telah berjalan dengan baik dan memenuhi requirements yang telah disusun sebelumnya.

2.6 Pengujian Perangkat Lunak

Metode pengujian perangkat lunak adalah cara atau teknik untuk menguji perangkat lunak. Metode pengujian berhubungan dengan perancang data uji yang akan dieksekusi pada perangkat lunak yang dikembangkan. Metode pengujian diharapkan mempunyai

mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan (Wibisono, 2002).

Menurut Pressman (2010), Perangkat lunak dapat diuji dengan dua cara yaitu black-box testing dan white-box testing.

1. Black-box Testing

Pengujian dilakukan dengan mengeksekusi data uji dan mengecek fungsionalitas dari perangkat lunak. Data uji diambil dari spesifikasi perangkat lunak yang menjelaskan fungsi dari perangkat lunak tanpa memperhatikan struktur kode dari sebuah sistem. Dengan pendekatan ini, pengembang dapat membuat sejumlah kondisi masuk yang secara menyeluruh mencakup semua kebutuhan fungsional program, menguji apakah perangkat lunak dapat memenuhi spesifikasi dan berfungsi dengan baik.

2. White-box Testing

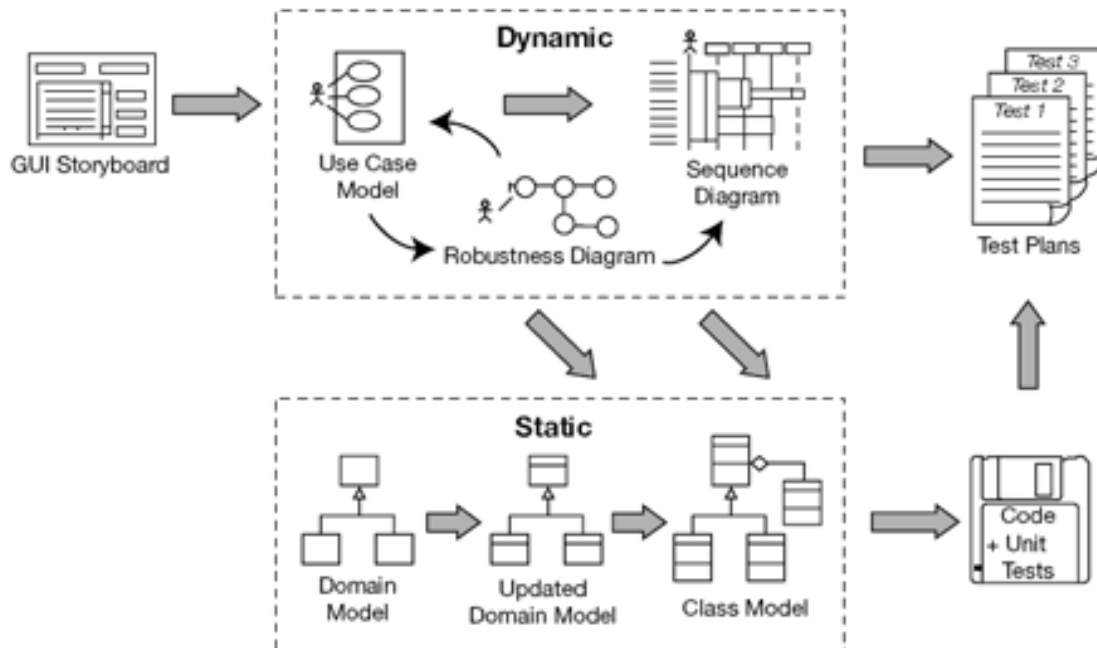
Pengujian dilakukan dengan mengenakan data uji untuk menguji semua elemen program perangkat lunak, baik itu data internal, logika keputusan, perulangan, dan lain sebagainya. Data uji dilakukan diambil dari struktur internal sebuah perangkat lunak. Dalam White-box testing, pengujian dilakukan dengan memahami struktur internal kode untuk menghasilkan test case yang melibatkan semua kemungkinan jalur eksekusinya.

BAB III

METODOLOGI PENELITIAN

Bab ini berisi metode dan langkah-langkah yang dilakukan untuk menyelesaikan permasalahan pada penelitian ini yaitu metode pengembangan perangkat lunak, yang terdiri dari tahap requirement hingga tahap testing.

Pengembangan Aplikasi *Monitoring Berbasis Progressive Web Apps* dikembangkan dengan metode ICONIX Process. Metode ini melibatkan empat tahap utama yaitu requirements, analysis/preliminary design, detailed design, dan implementation serta testing. Gambaran tahapan-tahapan dari ICONIX Process ditampilkan pada gambar 3.1.



Gambar 3.1 Tahapan pengembangan aplikasi ICONIX Process

3.1 Tahap Requirements

Perumusan kebutuhan pengguna ini didasarkan pada kebutuhan aktual dari peternak walet dan stakeholder Swiftlead. Perumusan kebutuhan pengguna ini didapatkan melalui observasi di lapangan maupun berdasarkan penelitian-penelitian sebelumnya. Proses ini menghasilkan keluaran berupa alur kerja, preferensi, permasalahan yang dihadapi calon pengguna, serta preferensi dari masing-masing stakeholder.

Tahapan ini menghasilkan sebuah landasan untuk mengembangkan sebuah sistem melalui tahap behavioral requirement dan domain modeling. Dokumen yang dihasilkan dari proses ini adalah kebutuhan fungsional dan non-fungsional, use case diagram, dan domain model. Setelah terdokumentasi dengan baik, verifikasi dilakukan terhadap stakeholder terkait untuk memastikan sistem telah memenuhi kebutuhan pengguna.

3.2 Tahap Analysis/Preliminary Design

Tahap analysis bertujuan untuk mendapatkan pemahaman yang lebih mendetail mengenai use case yang telah dihasilkan pada tahap sebelumnya. Untuk mengurangi ambiguitas dalam skenario use case, robustness analysis memungkinkan untuk meninjau kembali tahapan yang telah dilakukan oleh pengguna ketika menjalankan skenario tertentu. Selanjutnya, dilakukan pembaruan terhadap domain model yang lebih akurat dengan menambahkan atribut pada domain model. Di akhir proses, robustness diagram dibuat untuk mewakili masing-masing use case yang telah disusun. Hasil akhir dari tahapan ini ialah domain model yang telah diperbarui dan robustness diagram. Selanjutnya, peninjauan kembali dilakukan untuk memastikan tiap keluaran dari tahap ini sudah memenuhi kebutuhan pengguna yang telah disusun sebelumnya.

3.3 Tahap Detailed Design

Tujuan dari tahap Detailed Design adalah untuk menghasilkan gambaran mendetail dari sebuah sistem untuk masing-masing use case di dalamnya. Dari diagram-diagram yang telah dirancang sebelumnya, dikembangkan sebuah sequence diagram untuk menggambarkan alur informasi dari sistem yang dikembangkan secara mendetail. Kemudian, pembaruan kembali dilakukan terhadap domain model yang dihasilkan pada tahap sebelumnya dengan menambahkan metode di dalamnya. Pada akhirnya, keluaran utama dari tahapan ini adalah sequence diagram untuk masing-masing use case beserta class diagram secara lengkap.

3.4 Tahap Implementation dan Testing

Pada tahap ini sistem mulai diimplementasikan dan diterjemahkan ke dalam kode program. Semua diagram yang telah disusun sebelumnya, diterjemahkan ke dalam sebuah alur sistem yang utuh dengan menerapkan logika sesuai kebutuhan. Berikut adalah teknologi yang digunakan pada pengembangan aplikasi monitoring kandang burung walet.

1. Aplikasi Front-end

Aplikasi front-end berfokus pada penanganan masalah yang berkaitan dengan tampilan dan interaksi pengguna. Pengembangan aplikasi front-end melibatkan framework Next.js yang berjalan menggunakan bahasa Javascript. Framework ini memberikan kemudahan pengembangan melalui dukungan dynamic routing, server-side rendering, dan lain sebagainya. Selanjutnya, untuk meningkatkan estetika dari website, framework Tailwind CSS digunakan dengan bantuan library Shadcn, sebuah library berbasis komponen yang membantu pengembangan tampilan aplikasi menjadi lebih efisien.

2. Aplikasi Back-end

Aplikasi back-end menangani segala hal yang berkaitan dengan logika bisnis, pengolahan data, dan interaksi dengan basis data. Dalam proses pengembangan aplikasi monitoring kandang burung walet, framework Express.js digunakan untuk mempermudah tugas-tugas yang telah disebutkan. Express.js mendukung banyak fitur pengembangan seperti dynamic routing, middleware, dan lain sebagainya. Express.js dikembangkan dengan bahasa Javascript dan berjalan di atas lingkungan pengembangan Node.js, salah satu platform yang menyediakan fitur pengembangan dari sisi server.

3. Basis Data

Pengembangan aplikasi monitoring sarang burung walet menggunakan dua tipe Database Management System, yaitu Relational Database Management System (RDBMS) dan NoSQL. Seperti namanya, RDBMS terikat dengan tabel dan relasi antar tabel di dalamnya. Tiap tabel berisi record dalam kolom-kolom tertentu, pada penelitian ini RDBMS yang digunakan adalah MySQL. Selanjutnya, DBMS NoSQL digunakan pada penelitian ini untuk menangani data secara real-time. NoSQL bersifat tidak terikat dengan tabel dan hubungan antar tabel. DBMS ini cenderung lebih fleksibel dan beberapa di antaranya dapat menangani real-time data dengan sangat baik. Pada pengembangan aplikasi ini, DBMS NoSQL yang digunakan ialah Firebase, salah satu produk dari Google.

Selanjutnya, setelah tahap pengembangan telah selesai, maka dilanjutkan dengan tahap pengujian dengan menggunakan black-box testing. Pengujian dilakukan untuk mengecek apakah sistem telah memenuhi kebutuhan pengguna yang telah disusun sebelumnya. Fokusnya adalah memastikan bahwa semua fitur yang telah direncanakan berfungsi dengan benar, kinerja sistem stabil, dan pengalaman pengguna sesuai dengan ekspektasi.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil pembahasan yang dilakukan oleh penulis, yaitu proses pengembangan aplikasi monitoring kandang burung walet dari proses requirement hingga testing menggunakan ICONIX Process.

4.1 Tahap Requirement

Tahap requirement bertujuan untuk mengidentifikasi dan mendefinisikan kebutuhan pengguna dengan cakupan tertentu. Tahapan ini dimulai dengan melakukan observasi terhadap stakeholder terkait seperti perusahaan dan calon pengguna. Dari tahap ini dapat dirumuskan functional dan non-functional requirement. Selanjutnya, dari functional requirement disusun sebuah domain model dari sebuah sistem. Kemudian, dari domain model yang ada dapat dirumuskan menjadi behavioral requirements dan GUI storyboard.

4.1.1 Observasi

Aplikasi monitoring kandang burung walet berbasis Progressive Web Apps terintegrasi Internet of Things dirancang untuk membantu peternak walet dalam memantau keadaan aktual kandang burung walet dari segi suhu dan kelembaban. Kedua faktor ini sangat berpengaruh terhadap produktivitas sarang burung walet dan berpengaruh langsung terhadap pendapatan peternak burung walet. Dengan mengetahui tingkat suhu dan kelembaban kandang burung walet, peternak walet dapat mengambil tindakan jika mendapati suhu dan kelembaban sedang di luar kadar normal. Maka dari itu, aplikasi ini dibuat dengan mempertimbangkan aspek-aspek tersebut untuk meningkatkan produktivitas sarang burung walet.

Aplikasi monitoring ini melibatkan dua tipe pengguna atau aktor yang terlibat di dalamnya, yaitu admin dan peternak walet. Peternak walet dapat mengakses berbagai fitur dalam aplikasi seperti fitur monitoring, asisten panen, artikel, membership dan lain sebagainya. Sedangkan admin bertugas sebagai pengelola semua fitur yang terdapat dalam aplikasi mulai dari mengelola perangkat, mengelola pengguna, mengelola transaksi, serta mengelola konten. Sistem dirancang untuk memenuhi kebutuhan untuk masing-masing aktor sesuai dengan cakupan yang telah ditentukan.

4.1.2 Functional Requirements

Functional requirements berfungsi menjadi panduan utama dalam menjaga pengembangan aplikasi tetap berada pada scope yang tepat. Pada penyusunannya memperhatikan semua aspek yang dibutuhkan oleh calon pengguna berdasarkan aktor yang ada. Selain itu, Functional requirements juga menjadi acuan oleh pengembang dalam memenuhi kebutuhan pengguna tanpa ada kekurangan maupun berlebihan.

Tabel 4.1 Functional Requirements

No	Spesifikasi Sistem	Deskripsi
1	Autentikasi	Sistem mendukung fungsi autentikasi untuk pengguna.
2	Monitoring suhu dan kelembaban	Peternak dapat melakukan monitoring terhadap kandang berdasarkan sensor yang terpasang.
3	Manajemen kandang	Peternak dapat mengelola kandang, lantai, dan sensor. Peternak dapat mengajukan instalasi, pemeliharaan, dan uninstalasi IoT.
4	Asisten panen	Peternak dapat menggunakan asisten panen yaitu pra-panen, pasca-panen, pengajuan penjualan hasil panen.
5	Mengajukan Membership	Peternak dapat mengajukan akses membership dengan pembayaran melalui payment gateway. Peternak dapat mengakses konten membership.
6	Mengakses artikel	Peternak dapat mengakses artikel edukasi seputar burung walet.
7	Mengelola pengguna	Admin dapat mengelola pengguna aktif dan membership aktif.
8	Mengelola perangkat	Admin dapat mengelola perangkat IoT aktif, pengajuan instalasi, pemeliharaan, dan uninstalasi IoT.
9	Mengelola transaksi	Admin dapat memperbarui harga sarang burung walet mingguan, mengelola pengajuan penjualan hasil panen, dan transaksi membership.
10	Mengelola konten	Admin dapat mengelola konten artikel dan konten membership.

4.1.3 Non-functional Requirements

Non-functional requirements merupakan fungsi-fungsi yang tidak dilakukan oleh sistem, tapi tetap berpengaruh terhadap keberjalanan sistem. Kebutuhan ini melingkupi aspek-aspek yang berhubungan dengan pengalaman pengguna ketika menggunakan sistem, baik itu pengalaman langsung maupun tidak langsung.

Tabel 4.2 Non-functional Requirements

No	Non-functional Requirements	Deskripsi
1	Responsivity	Sistem dapat menyajikan tampilan yang responsif sesuai dengan ukuran layar perangkat dengan memastikan semua fitur berjalan dengan optimal di semua perangkat.
2	Scalability	Sistem bersifat adaptif terhadap perubahan yang akan datang untuk pengembangan lebih lanjut.
3	Usability	Sistem dapat digunakan secara nyaman sesuai dengan target pengguna dari aplikasi yaitu peternak walet.
4	Security	Sistem memastikan keamanan data melalui enkripsi data sensitif serta pemantauan keamanan secara berkala.

4.1.4 Domain Model

4.1.5 Behavioral Requirements

4.1.6 GUI Storyboard

4.1.7 Requirements Review

1.1.1 Pengertian Sistem ← (Minimum 2 kata: Diterangkan - Menerangkan)

Teks yang sedang Anda baca ini merupakan contoh isi paragraph dalam Sub Sub Bab 1.1.1. Perhatikan bahwa format pada paragraf ini sama dengan paragraf Sub Bab 1.1 di atas.

1.1.1.1 Contoh Sub Sub Sub Bab

Teks yang sedang Anda baca ini merupakan contoh isi paragraph dalam Sub Sub Sub Bab 1.1.1.1. Perhatikan bahwa format pada paragraf ini sama dengan paragraf Sub Bab 1.1 atau Sub Sub Bab 1.1.1 di atas. Sub Sub Sub Bab ini merupakan sub bab terdalam yang diperkenalkan dalam penulisan skripsi ini (tidak diperbolehkan ada Sub Sub Sub Sub Bab 1.1.1.1.1). Jika Anda membutuhkannya, gunakanlah pemerian (perincian).

1.1.1.2 Paragraf Pemerian (Perincian)

Macam-macam binatang ternak yang biasa dipelihara para petani di Indonesia sebagai berikut: (Pemerian jenis 1, poin singkat)

1. Sapi
2. Kambing

Macam-macam binatang ternak yang biasa dipelihara para petani di Indonesia sebagai berikut: (Pemerian jenis 2, poin panjang)

1. Sapi, yang merupakan binatang ternak berkaki empat yang biasa dimanfaatkan susu, daging, maupun kulitnya.
2. Kambing, yang merupakan binatang ternak berkaki empat yang biasa dimanfaatkan daging dan kulitnya, tetapi ada pula yang memanfaatkan susunya.

Macam-macam binatang ternak yang biasa dipelihara para petani di Indonesia sebagai berikut: (Pemerian jenis 3, poin memiliki paragraf di bawahnya)

1. Sapi

Sapi merupakan binatang ternak berkaki empat yang biasa dimanfaatkan susu, daging, maupun kulitnya. Dilihat dari asal negaranya, sapi memiliki jenis:

- a. Sapi India
- b. Sapi Australia

2. Kambing

Kambing merupakan binatang ternak berkaki empat yang biasa dimanfaatkan daging dan kulitnya, tetapi ada pula yang memanfaatkan susunya.

Petunjuk PARAGRAF PEMERIAN (PERINCIAN)

- ❖ Tidak diperbolehkan menggunakan simbol (bullets) pada pemerian, wajib menggunakan nomor (1, 2, 3, dst) atau huruf (a, b, c, dst).
- ❖ Pemerian (perincian) hanya memiliki tiga jenis pemformatan seperti di atas. Jika Anda membutuhkan penjelasan panjang (butuh lebih dari satu paragraf tiap poinnya), gunakanlah Sub Bab.
- ❖ Kepala pemerian (kalimat awal pemerian) wajib ada dan ditulis sebagai awal paragraf.
- ❖ Pastikan teks kepala pemerian ini tidak terpisah dengan isi pemerian.

Silahkan hapus tabel merah ini

1.1.1.3 Penampilan Tabel, Gambar, Flowchart, Bagan, atau Komponen Lain

Setiap tabel, gambar, flowchart, bagan, atau komponen lain harus disitasi oleh paragraf (terdapat penjelasan yang menyebut keberadaannya). Misal, macam-macam binatang ternak ditunjukkan pada Tabel 5.1. Contoh binatang ternak sapi ditunjukkan pada Gambar 5.10.

Tabel 5.1 Macam-macam Binatang Ternak

No	Binatang Ternak	Penjelasan
1	Sapi	Sapi merupakan binatang ternak berkaki empat yang biasa dimanfaatkan susu, daging, maupun kulitnya.
2	Kambing	Kambing merupakan binatang ternak berkaki empat yang biasa dimanfaatkan daging dan kulitnya. Tetapi ada pula yang memanfaatkan susunya.
3	Ayam	Ayam merupakan binatang ternak berkaki dua yang biasa dimanfaatkan daging dan bulunya.

Petunjuk TABEL

- ❖ Isi tabel satu spasi, lebar kolom dan *alignment*-nya dapat disesuaikan dengan isinya agar rapih dan estetik.
- ❖ Untuk tabel yang memiliki teks header 1 baris, tinggi header 1,0 cm dan warna shading abu-abu 25%. Jika teks header lebih dari 1 baris, tinggi header dapat disesuaikan.
- ❖ Jika tabel bersambung ke halaman setelahnya, header tabel harus ditampilkan ulang (*Repeat Header Rows*).
- ❖ Jika tabel memenuhi hingga lebih dari 3 halaman, sebaiknya diletakkan dalam lampiran.
- ❖ Disarankan setiap tabel memiliki kolom **No** pada kolom paling kiri, untuk membantu menunjukkan jumlah baris yang ada pada tabel tersebut.
- ❖ Garis tabel terluar tidak diperkenankan melewati margin. Jika perlu, gunakan format landscape hanya untuk halaman yang memuat tabel tersebut.
- ❖ Diperbolehkan mengurangi ukuran fontasi teks hingga 10 pt dalam tabel asalkan masih jelas terbaca.
- ❖ Nomor urut pada tabel diurutkan mulai dari no. 1 sejak muncul pertama kali dalam bab tersebut. Hal ini juga berlaku untuk Gambar, Persamaan, Source Code, dan sejenisnya.

Silahkan hapus tabel merah ini

Gambar hewan ternak sapi yang memiliki lonceng di lehernya dapat ditunjukkan pada Gambar 5.10:



Gambar 5.10 Contoh Hewan Ternak Sapi

Petunjuk GAMBAR, atau LAINNYA

- ❖ Gambar tidak diberi garis tepi.
- ❖ Tidak boleh melewati batas margin dan memiliki ukuran yang terlihat baik secara normal .
- ❖ Jika di dalam gambar terdapat teks, ukuran teks tersebut harus setara 12 pt.
- ❖ Penggunaan flowchart harus menggunakan kaidah simbol flowchart secara umum (rectangle, diamond, arrow, etc).

Silahkan hapus tabel merah ini

1.1.1.4 Penampilan Persamaan atau Rumus

Perhitungan mendapatkan akar persamaan kuadrat dapat ditunjukkan pada Persamaan 3.1:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \dots\dots\dots(3.1)$$

Keterangan:

- $x_{1,2}$: Akar persamaan yang dicari
 a : Koefesien polinomial derajat 2
 b : Koefesien polinomial derajat 1
 c : Koefesien polinomial derajat 0

Petunjuk PERSAMAAN atau RUMUS

- ❖ Persamaan ditulis rata kiri (bukan center).
- ❖ Semua simbol yang ada dalam persamaan harus dijelaskan pada bagian keterangan.

Silahkan hapus tabel merah ini

1.1.1.5 Penampilan *Source Code*

Kode sumber (*source code*) ditampilkan dalam bentuk tabel 2 baris 1 kolom dengan garis yang tidak melewati batas margin. Fontasi yang digunakan adalah Courier New 10 pt. Tidak perlu diberi Caption dan Daftar Source Code di bagian depan.

Misalnya, implementasi kode untuk menyelesaikan sistem persamaan non linier menggunakan metode Iterasi Titik - Tetap ditunjukkan pada Source Code 4.1:

Source Code 4.1: Penyelesaian sistem persamaan non linear menggunakan metode Iterasi Titik - Tetap
<pre>err = 100 iter = 0 eps = 0.00000001 while (err > eps) : x.append(f1(x[iter],y[iter])) y.append(f2(x[iter+1],y[iter])) print("%3d %12.8f %12.8f %12.8f" % (iter, x[iter], y[iter], err)) err = abs(x[iter+1]-x[iter]) iter += 1 print("Pendekatan terbaik pada x = %12.8f dan y = %12.8f" % (x[iter], y[iter]))</pre>

DAFTAR PUSTAKA

- Aripin Samsul, & Somantri. (2021). Implementasi Progressive Web Apps (PWA) pada Repository E-Portofolio Mahasiswa. Sukabumi: Jurnal Eksplora Informatika.
- Badan Pusat Statistik Indonesia. (2024). Ekspor Sarang Burung menurut Negara Tujuan Utama, 2012-2023 [Online]. Tersedia di: <https://www.bps.go.id/id/statistics-table/1/MjAyMiMx/ekspor-sarang-burung-menurut-negara-tujuan-utama--2012-2023.html> (Diakses pada 21 Januari 2025).
- Hidayat, D., & Sari, I. (2021). Monitoring suhu dan kelembaban berbasis Internet of Things (IoT). *Jurnal Penelitian Teknik Informatika Universitas Prima Indonesia*, 4(1), 525–530. <https://myusro.id/wp-content/uploads/2021/05/SENSOR13-SUHU2.pdf>.
- Kelen, Laberto. (2018). Implementasi Model-view-controller (MVC) Pada Ujian Online Melalui Penerapan Framework Codeigniter. *Jurnal Pendidikan Teknologi Informasi*, vol. 1, no. 1, 2018, pp. 10-16.
- Muslihudin, M, & Otafianto. (2016). Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML. Yogyakarta:CV. Andi Offset.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. New York: McGrawHill.
- Rosenberg, D., & Stephens, M. (2007). *Iconix Process: A Practical Approach to Developing Web Applications*. Prentice Hall.
- Sampurna, Jaya & Dedeng Hirawan. (2017). *Pembangunan Sistem Pemantauan Rumah Walet Berbasis IOT*. Bandung: Digital library - Perpustakaan Pusat Unikom.
- Wibisono, Waskitho, and Fajar Baskoro. 2002. Pengujian Perangkat Lunak Dengan Menggunakan Model Behaviour Uml. *Jurnal Ilmiah Teknologi Informasi*, vol. 1, no. 1, 1 Jul. 2002, pp. 43-50, doi:10.12962/j24068535.v1i1.a95.

- ❖ Halaman ini harus dibuat secara otomatis menggunakan fasilitas *References* (atau sejenisnya) dalam aplikasi pengolah kata, melalui sub menu **Citations & Bibliography** (atau sejenisnya).
- ❖ Style referensi yang digunakan: **APA** Seventh Edition (atau yang setara).
- ❖ Jika diperlukan, dipersilahkan edit secara manual agar tampilannya sesuai format di atas.

Silahkan hapus tabel merah ini

LAMPIRAN 1. Surat Ijin Pengambilan Data

DEPARTEMEN PENDIDIKAN NASIONAL
FAKULTAS EKONOMIKA DAN BISNIS
UNIVERSITAS DIPONEGORO
Jl. Prof Soedarto, SH Tembalang Semarang

No :
Lamp :
Hal :

Dengan hormat,

dan seterusnya...

LAMPIRAN 2. Tabel Pengujian Aplikasi