

## BAB 3 PELAKSANAAN KERJA MAGANG

### 3.1 Kedudukan dalam PT Sumber Inovasi Informatika

Program kerja magang pada PT Sumber Inovasi Informatika dilakukan dalam tim pengembang aplikasi ERP ayam layer sebagai *backend developer intern*. Tanggung jawab utama seorang *backend developer intern* pada tim tersebut yaitu melakukan implementasi rencana pengembangan aplikasi ERP ayam layer dan memastikan aplikasi ERP ayam layer berjalan dengan lancar di situs web *production*. Para *backend developer intern* juga diwajibkan untuk melakukan kegiatan *sprint planning* setiap dua minggu sekali berdasarkan *framework Agile Methodology* untuk memastikan alur pengembangan aplikasi sesuai kebutuhan pengguna [4].

Selama program kerja magang berlangsung, mahasiswa dibimbing dan dikoordinasikan oleh COO PT Sumber Inovasi Informatika dan product owner proyek peternakan ayam layer.

### 3.2 Tugas yang Dilakukan

Pada PT Sumber Inovasi Informatika, mahasiswa memiliki beberapa tugas yang harus diselesaikan pada program kerja magang yang dapat dilihat pada Tabel 3.1.

Tabel 3.1. Jadwal Tugas

<b>Tugas</b>	<b>Minggu</b>
Pembelajaran teknologi dan Agile Requirements	1
Pengumpulan Requirements	2-4
Identifikasi Isu	5-7
Pengembangan Doctype Daily Growth Recording	8-10
Pengembangan Doctype Daily Production Recording	11-13
Pengembangan Doctype House Monthly Closing	14-17
Fase <i>Testing</i>	18
<i>Release &amp; Deployment</i>	19

Berdasarkan tabel 3.1, pengembangan terbagi menjadi tiga bagian utama sebagai berikut:

- **Pengembangan Doctype Daily Production Recording**

Doctype Daily Production Recording merupakan sub-modul dari modul Poultry Farm yang digunakan oleh pengguna untuk melakukan pencatatan data pada peternakan ayam layer pada fase DOC dan Growth.

- **Pengembangan Doctype Daily Growth Recording**

Doctype Daily Growth Recording merupakan sub-modul dari modul Poultry Farm yang digunakan oleh pengguna untuk melakukan pencatatan data pada peternakan ayam, saat ayam berada pada fase Layer.

- **Pengembangan Doctype House Monthly Closing**

Doctype House Monthly Closing merupakan sub-modul dari modul Poultry Farm yang digunakan oleh pengguna untuk menutup perhitungan akuntansi setiap bulan, agar hasil perhitungan tetap akurat.

### 3.3 Uraian Pelaksanaan Magang

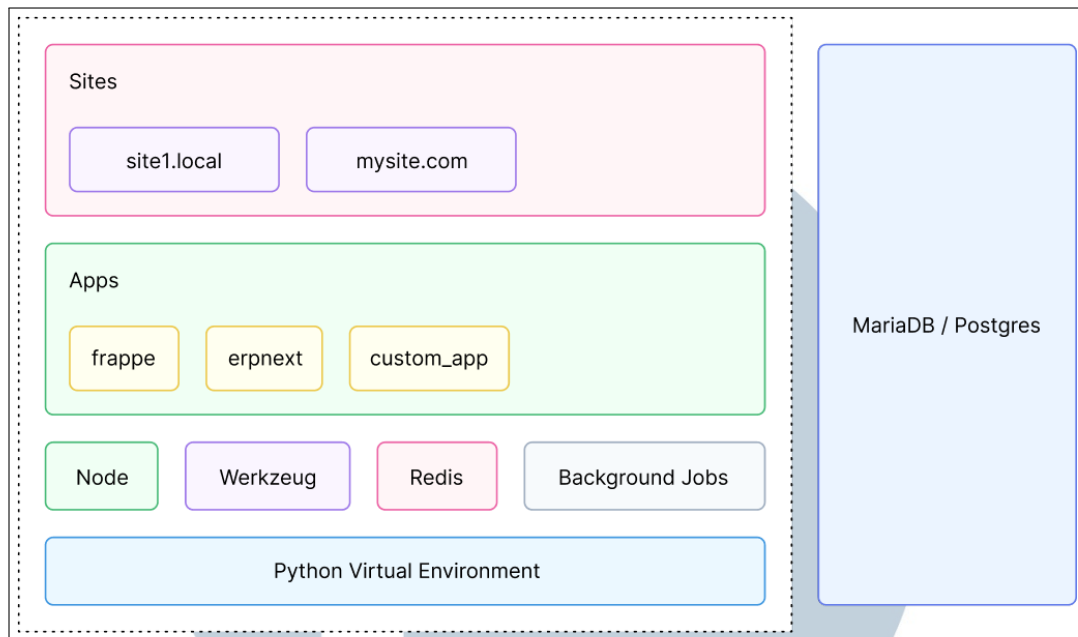
#### 3.3.1 Teknologi yang digunakan pada Pengembangan aplikasi ERP ayam layer

Dalam melakukan pengembangan aplikasi ERP ayam layer terdapat beberapa teknologi yang digunakan untuk mempercepat proses pengembangan aplikasi. Berikut merupakan beberapa teknologi dan *tools* yang digunakan pada saat pelaksanaan kegiatan magang pada PT Sumber Inovasi Informatika.

##### A. Frappe

Frappe adalah *framework* web *open-source* yang digunakan untuk membangun aplikasi web dengan Python sebagai bahasa utamanya. *framework* ini menyediakan banyak fitur bawaan, seperti manajemen pengguna, otentikasi, dan otorisasi, yang dapat menghemat waktu pengembangan aplikasi [2].

Frappe menggunakan beberapa *service* untuk mendukung fitur-fiturnya seperti pada Gambar 3.1.



Gambar 3.1. Diagram Arsitektur Frappe

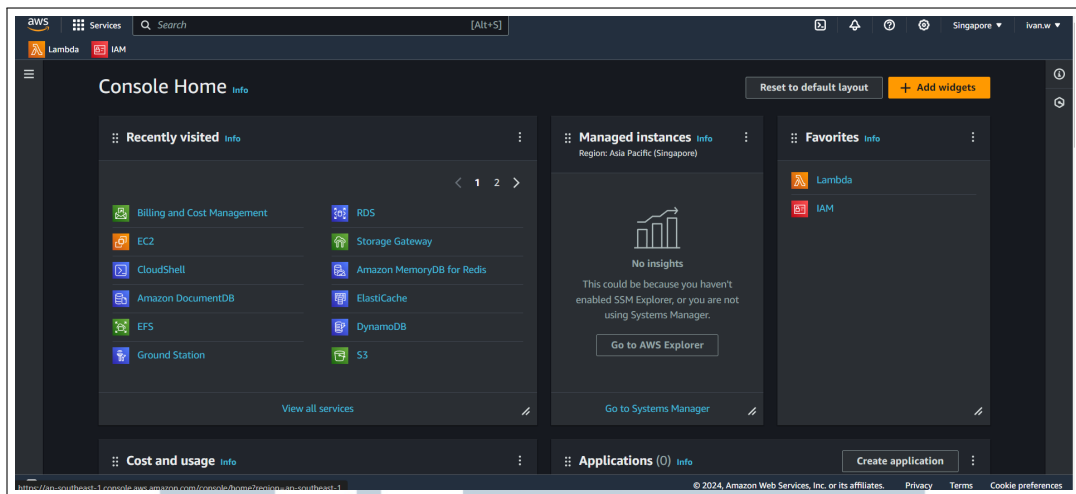
## B. ERPNext

ERPNext adalah *software* Enterprise Resource Planning (ERP) *open-source* yang dikembangkan oleh Frappé Technologies Pvt. Ltd. ERPNext merupakan salah satu contoh dari Frappe App [5]. ERPNext memiliki beberapa modul yang dapat digunakan seperti:

- Sales.
- Manufacturing.
- Asset Management.
- Accounting System.

## C. Amazon Web Service (AWS)

Amazon Web Service (AWS) merupakan layanan *cloud computing* asal Amerika. AWS memiliki banyak layanan yang dapat pengguna akses lewat *dashboardnya* seperti pada Gambar 3.2.



Gambar 3.2. Tampilan AWS Console

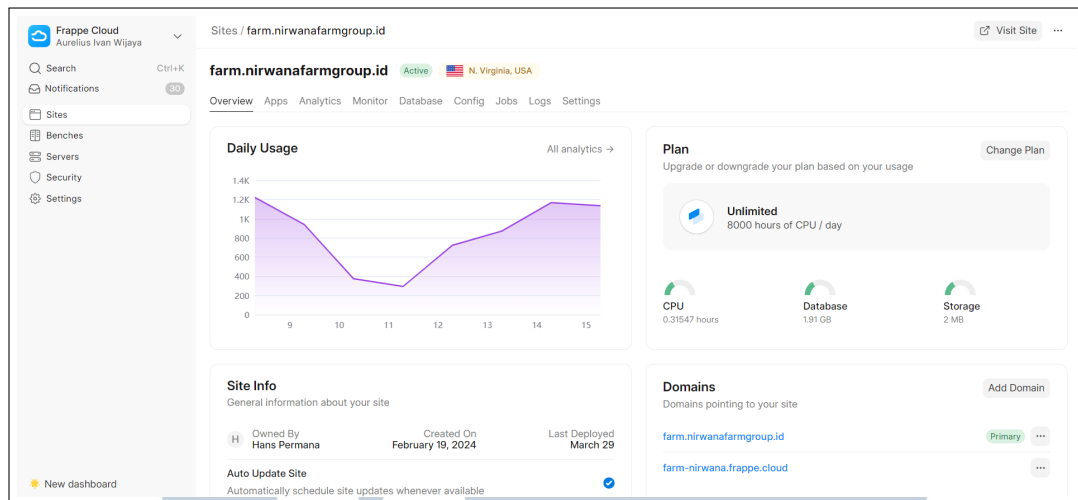
Beberapa layanan AWS yang digunakan oleh PT Sumber Inovasi Informatika:

- Amazon Elastic Cloud Compute (EC2).
- Amazon Relational Database Service (RDS).
- Amazon Simple Storage Service (S3).
- AWS Lambda.

#### D. Frappe Cloud

Frappe Cloud merupakan sebuah layanan *cloud computing* berbasis *platform as a service (PaaS)* yang mempermudah *deployment* aplikasi Frappe [6] seperti yang ditampilkan pada Gambar 3.3.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.3. Tampilan dashboard Fape Cloud

## E. Python

Python adalah bahasa pemrograman *high-level*, serbaguna, dan berorientasi objek yang dikenal akan kesederhanaan dan keterbacaannya [7].

## F. Git

Git adalah sistem kontrol versi (VCS - Version Control System). VCS adalah alat yang melacak perubahan pada kumpulan file selama jangka waktu tertentu [8].

## G. Gitlab

GitLab adalah platform yang menyediakan layanan *remote access* terhadap *repository* git. Selain itu, Gitlab juga menyediakan berbagai fitur penyokong proses pengembangan aplikasi seperti *continuous integration and continuous delivery* (ci/cd) [9].

## H. Leantime

Leantime adalah software manajemen proyek *open-source* yang dirancang khusus untuk *startup* dan tim kecil. Leantime menggabungkan konsep dari berbagai metodologi seperti Design Thinking, metode Lean, dan praktik *Agile Methodology* dalam satu *platform* terintegrasi [10].

### 3.3.2 Uraian Proses Pengembangan aplikasi ERP ayam layer

#### A. Pengenalan Peternakan Ayam Petelur (Layer)

Peternakan ayam layer merupakan salah satu pengguna dari PT Sumber Inovasi Informatika. Ayam layer memiliki beberapa daur hidup yang setiap fasenya akan selalu di catat pada aplikasi ERP ayam layer. Berikut merupakan daur hidup ayam layer:

- **Fase Day old Chick (DOC)**

Fase ini berlangsung hari pertama sampai dengan minggu ke-3. Pada fase ini ayam akan dimasukan kedalam kandang ayam grower (*growing house*). Pada fase ini pertumbuhan dan perkembangan ayam setiap harinya akan dicatat kedalam Doctype Daily Growth Recording pada aplikasi ERP ayam layer. Doctype Daily Growth Recording merupakan fitur dari aplikasi ERP ayam layer berbentuk *form* untuk melakukan pencatatan *recording* dari ayam DOC dan grower setiap harinya. Tampilan ayam pada fase ini dapat dilihat pada Gambar 3.4.



Gambar 3.4. Ayam Fase DOC

- **Fase Grower**

Fase ini berlangsung selama minggu ke-3 sampai dengan minggu ke-16 seperti pada Gambar 3.5.





Gambar 3.5. Ayam Fase Grower

- **Fase Layer**

Fase ini berlangsung selama minggu ke-16 sampai dengan minggu ke-100 seperti pada Gambar 3.6. Pada awal fase ini, ayam akan dipindahkan kedalam *laying house*. Tampilan ayam pada fase ini dapat dilihat pada Gambar 3.6.



Gambar 3.6. Ayam Fase Layer

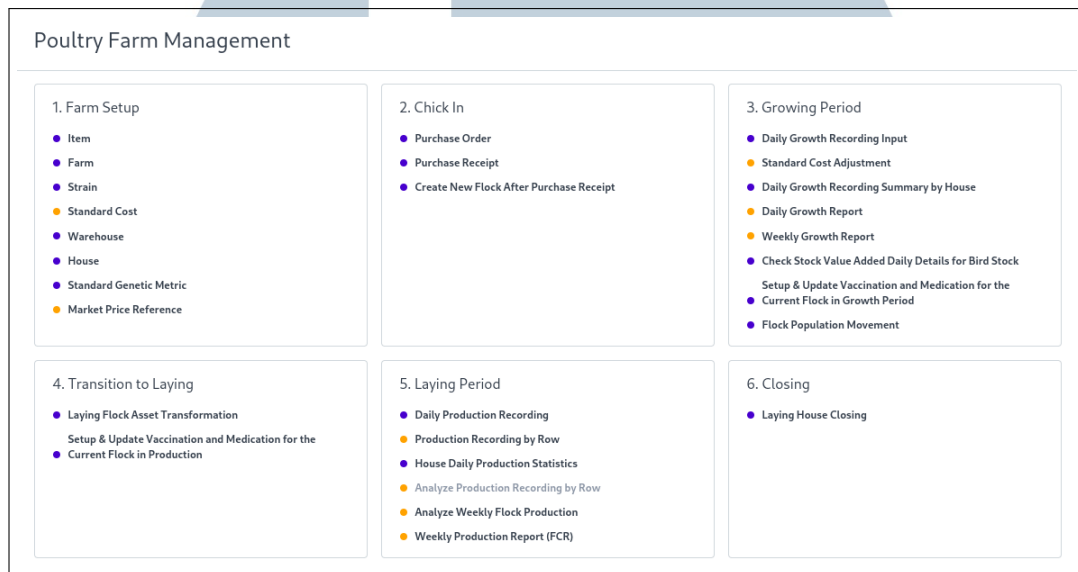
Pada fase ini juga, ayam akan mulai memproduksi telur dan setiap harinya petugas akan mulai melakukan pencatatan pada Doctype Daily Production Recording, Doctype Daily Production Recording merupakan fitur dari aplikasi ERP ayam layer berbentuk *form* untuk melakukan pencatatan *recording* dari ayam layer setiap harinya.

- **Fase Afkir**

Pada fase ini ayam dinyatakan sudah tidak produktif dan siap untuk dipotong (*culling*) sementara *house* yang telah kosong, nantinya akan digunakan untuk menampung *flock* ayam layer yang baru.

## B. Rencana pengembangan aplikasi ERP ayam layer

Aplikasi ERP ayam layer adalah alat yang penting dalam mengelola operasi peternakan ayam peternakan ayam layer secara efisien dan efektif. Aplikasi ERP ayam layer memiliki berbagai modul yang dapat membantu pengguna dalam mengelola peternakan sesuai dengan contoh pada Gambar 3.7.



Gambar 3.7. Tampilan modul-modul pada aplikasi ERP ayam layer

Berdasarkan Gambar 3.7 ikut ditampilkan daftar Doctype pada setiap modul. Doctype merupakan sistem formulir dari framework Frappe, yang memungkinkan pengguna membuat dan mengkustomisasi formulir tersebut sesuai dengan kebutuhan pengguna. Berikut merupakan contoh Doctype pada aplikasi ERP ayam layer seperti:

- Doctype Daily Production Recording
- Doctype Daily Growth Recording
- Doctype House Monthly Closing

Namun, aplikasi tentunya tidak luput dari masalah teknis yang dapat mempengaruhi kinerjanya. Untuk memastikan bahwa aplikasi berfungsi dengan optimal dan memenuhi kebutuhan pengguna, perlu dilakukan evaluasi dan perbaikan berkelanjutan. Salah satu langkah penting dalam proses ini adalah mengidentifikasi dan menyelesaikan isu-isu yang dihadapi oleh pengguna. Dalam



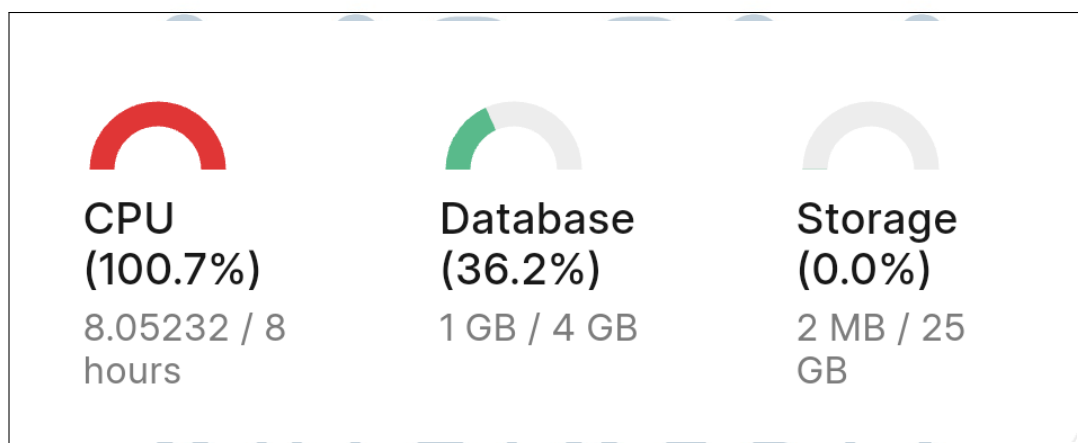
konteks ini, tim pengembang aplikasi ERP ayam layer telah melakukan serangkaian investigasi mendalam untuk memahami masalah yang dilaporkan oleh pengguna, terutama terkait dengan penggunaan CPU yang tinggi setiap kali pengguna melakukan *submit* dan *issue* jurnal depresiasi yang belum terbentuk. Jurnal depresiasi merupakan mekanisme untuk menunjukkan berapa banyak nilai aset yang telah habis digunakan dalam periode waktu tertentu, dalam hal ini ayam akan dianggap sebagai aset yang nilainya akan berkurang setiap periode tertentu.

Berikut ini uraian proses pengembangan dari aplikasi ERP ayam layer yang telah dilakukan bersama tim pengembang aplikasi ERP ayam layer.

### B.1 Investigasi *Issue* dan Solusi yang Mungkin

Pada tahap pertama, tim pengembang aplikasi ERP ayam layer melakukan investigasi dan reproduksi *issue* yang telah dilaporkan oleh pengguna untuk memberikan pemahaman yang lebih baik terkait *issue* yang dihadapi. Berdasarkan informasi dari pengguna dan hasil investigasi, didapati beberapa *issue* terkait penggunaan aplikasi ERP ayam layer.

- Penggunaan CPU yang tinggi setiap melakukan *submit* Doctype Daily Growth Recording dan Doctype Daily Production Recording seperti yang dapat dilihat pada Gambar 3.8.



Gambar 3.8. Penggunaan CPU pada panel Frappe Cloud

Untuk mengetahui performa aplikasi, dilakukan *benchmarking*. *Benchmarking* dilakukan dengan menginstall aplikasi ERP ayam layer pada *local server* dengan data situs web *production* yang telah di *backup*. Dalam melakukan *benchmark*, tim developer menggunakan *decorator function* untuk mempermudah proses *benchmark* seperti pada Kode 3.1.

```

1
2 def benchmark(func: callable):
3     """
4     Decorator to benchmark a function
5     Will print the time taken by the function
6     :param func:
7     :return:
8     """
9
10    def wrapper(*args, **kwargs):
11        if not conf.get("developer_mode"):
12            return func(*args, **kwargs)
13
14        import time
15        start = time.time()
16        result = func(*args, **kwargs)
17        end = time.time()
18        msgprint(
19            title="Benchmark",
20            msg=f"{func.__name__} took {end - start} seconds"
21        )
22        return result
23
24    return wrapper

```

Kode 3.1: Contoh Kode Unit Testing

Melalui hasil *benchmark* didapati bahwa pembentukan Doctype Daily Production Recording dan Doctype Daily Growth Recording tergolong cukup lama (lebih dari 5 menit), pada saat melakukan *submit*, sehingga menyebabkan *http request timeout*. *Issue* tersebut tidak dapat diselesaikan karena tergolong *core* aplikasi Frappe dan ERPNext itu sendiri, untuk itu tim developer menyarankan untuk melakukan percobaan upgrade versi Frappe dan ERPNext ke versi 14. Selain itu tim developer menyarankan untuk dilakukan perubahan alur pembuatan Doctype Asset Daily Depreciation dari harian menjadi bulanan agar membuat proses *submit* Doctype Daily Production Recording menjadi lebih cepat.

- Doctype Asset Daily Depreciation yang tidak terbentuk ketika pengguna membuat Doctype Daily Production Recording. Pada saat melakukan *submit* Doctype Daily Production Recording, aplikasi seharusnya secara otomatis

membuat Doctype Asset Daily Depreciation. Setelah dilakukan investigasi terhadap kode program, ditemukan bahwa ada *bug* yang membuat fitur tersebut belum berjalan pada versi aplikasi ERP ayam layer terkini. Tim developer menyarankan untuk membuat tiket perbaikan untuk Doctype Asset Daily Depreciation.

## C. Pengembangan Modul Poultry Management aplikasi ERP ayam layer

### C.1 Pengembangan sub-modul Doctype Daily Production Recording

#### 1. Pengumpulan *requirements* dan diskusi bersama stake holder

Pada tahap ini tim pengembang aplikasi ERP ayam layer melakukan pengumpulan *requirements* atau kebutuhan yang perlu dilakukan pada proyek rencana pengembangan aplikasi ERP ayam layer bersama *stake holder* terkait. Berikut merupakan daftar *requirements* yang dibutuhkan dalam sub-modul Doctype Daily Production Recording.

- Pengguna dapat melakukan *submit* Doctype Daily Production Recording tanpa mengalami *timeout*.
- Doctype Daily Production Recording dapat secara otomatis membuat jurnal perhitungan transaksi dengan benar.

#### 2. Pengembangan sub-modul Doctype Daily Production Recording

Setelah dilakukan pengumpulan *requirements*, selanjutnya dilakukan implementasi dan pengembangan sesuai dengan rancangan *flowchart* yang sudah ditentukan. tim pengembang aplikasi ERP ayam layer juga melakukan *refactor* dan *debugging* terhadap kode agar dapat menemukan akar masalah *timeout* yang dialami oleh fitur Doctype Daily Production Recording.

#### 3. Fase *testing* sub-modul Doctype Daily Production Recording

Pada fase ini dilakukan *testing* oleh tim developer dan product owner untuk memastikan semua fitur berjalan dengan baik. Terdapat sub-fase pada fase *testing* ini yaitu:

##### (a) *Local Testing*

Pada sub-fase ini fitur akan diuji secara lokal oleh tim pengembang aplikasi ERP ayam layer menggunakan *unit testing* yang telah dibuat seperti contoh pada kode 3.2 [11]. Pengetesan dilakukan dengan menggunakan fitur bawaan *framework* Frappe yaitu "bench run-tests"

yang akan otomatis menjalankan semua *file* berawalan `test_` di dalam sub-modul `Doctype Daily Production Recording`.

```
1 import frappe
2 import unittest
3
4 def create_events():
5     if frappe.flags.test_events_created:
6         return
7
8     frappe.set_user("Administrator")
9     doc = frappe.get_doc({
10         "doctype": "Event",
11         "subject": "_Test Event 1",
12         "starts_on": "2014-01-01",
13         "event_type": "Public"
14     }).insert()
15
16     doc = frappe.get_doc({
17         "doctype": "Event",
18         "subject": "_Test Event 3",
19         "starts_on": "2014-01-01",
20         "event_type": "Public"
21         "event_individuals": [{
22             "person": "test1@example.com"
23         }]
24     }).insert()
25
26     frappe.flags.test_events_created = True
27
28
29 class TestEvent(unittest.TestCase):
30     def setUp(self):
31         create_events()
32
33     def tearDown(self):
34         frappe.set_user("Administrator")
35
36     def test_allowed_public(self):
37         frappe.set_user("test1@example.com")
38         doc = frappe.get_doc("Event", frappe.db.get_value
39 ("Event",
40     {"subject": "_Test Event 1"}))
41         self.assertTrue(frappe.has_permission("Event",
```

```

doc=doc))
41
42     def test_not_allowed_private(self):
43         frappe.set_user("test1@example.com")
44         doc = frappe.get_doc("Event", frappe.db.get_value
("Event",
45             {"subject": "_Test Event 2"}))
46         self.assertFalse(frappe.has_permission("Event",
doc=doc))
47

```

Kode 3.2: Contoh Kode Unit Testing

Dari hasil pengujian akhir didapati bahwa, sub-modul Doctype Daily Production Recording dapat berjalan dengan baik tanpa mengalami *timeout* dan dapat melakukan proses *submit* dengan lebih cepat. mahasiswa juga melakukan proses *testing* secara manual untuk memastikan kecepatan dari proses *submit*.

(b) ***Staging Testing***

Pada sub-fase ini sub-modul Doctype Daily Production Recording akan diuji oleh product owner. Hasil dari pengujian didapati bahwa sub-modul Doctype Daily Production Recording dapat berjalan dengan baik dan dapat melakukan *generate* jurnal perhitungan dengan benar.

(c) ***User Acceptance Testing***

Pada sub-fase ini, sub-modul Doctype Daily Production Recording akan diuji oleh pengguna terkait. Lewat fase ini, dihasilkan kesimpulan bahwa aplikasi dapat berjalan dengan baik sehingga proyek dapat dilanjutkan ke proses *release* ke situs web *production*.

## C.2 Pengembangan sub-modul Doctype Daily Growth Recording

### 1. Pengumpulan *requirements* dan diskusi bersama stake holder

Pada tahap ini tim pengembang aplikasi ERP ayam layer melakukan pengumpulan *requirements* atau kebutuhan yang perlu dilakukan pada proyek rencana pengembangan aplikasi ERP ayam layer bersama *stake holder* terkait.

- Pengguna dapat melakukan *submit* Doctype Daily Production Recording tanpa mengalami *timeout*.

- Doctype Daily Growth Recording dapat secara otomatis membuat jurnal perhitungan transaksi dengan benar.

## 2. Pengembangan sub-modul Doctype Daily Growth Recording

Setelah dilakukan pengumpulan *requirements*, selanjutnya dilakukan *refactor* dan *debugging* terhadap kode agar dapat menemukan akar masalah *timeout* yang dialami oleh fitur Doctype Daily Production Recording. tim pengembang aplikasi ERP ayam layer juga melakukan simulasi perhitungan jurnal bersama product owner untuk memastikan fitur perhitungan jurnal berjalan dengan lancar.

## 3. Fase *testing* sub-modul Doctype Daily Growth Recording

Pada fase ini dilakukan *testing* oleh tim developer dan product owner untuk memastikan semua fitur berjalan dengan baik. Terdapat sub-fase pada fase *testing* ini yaitu:

### (a) *Local Testing*

Pada sub-fase ini fitur akan diuji secara lokal oleh tim pengembang aplikasi ERP ayam layer menggunakan *unit testing* yang telah dibuat [11]. Pengetesan dilakukan dengan menggunakan fitur bawaan *framework* Frappe yaitu "bench run-tests" yang akan otomatis menjalankan semua *file* berawalan *test\_* di dalam kode aplikasi ERP ayam layer.

Dari hasil pengujian akhir didapati bahwa, sub-modul Doctype Daily Production Recording dapat berjalan dengan baik tanpa mengalami *timeout* dan dapat melakukan proses *submit* dengan lebih cepat. tim pengembang aplikasi ERP ayam layer juga melakukan proses *testing* secara manual untuk memastikan kecepatan dari proses *submit*.

### (b) *Staging Testing*

Pada sub-fase ini sub-modul Doctype Daily Production Recording akan diuji oleh product owner. Hasil dari pengujian didapati bahwa sub-modul Doctype Daily Production Recording dapat berjalan dengan baik dan dapat melakukan *generate* jurnal perhitungan dengan benar.

### (c) *User Acceptance Testing*

Pada sub-fase ini, sub-modul Doctype Daily Production Recording akan diuji oleh pengguna terkait. Lewat fase ini, dihasilkan kesimpulan



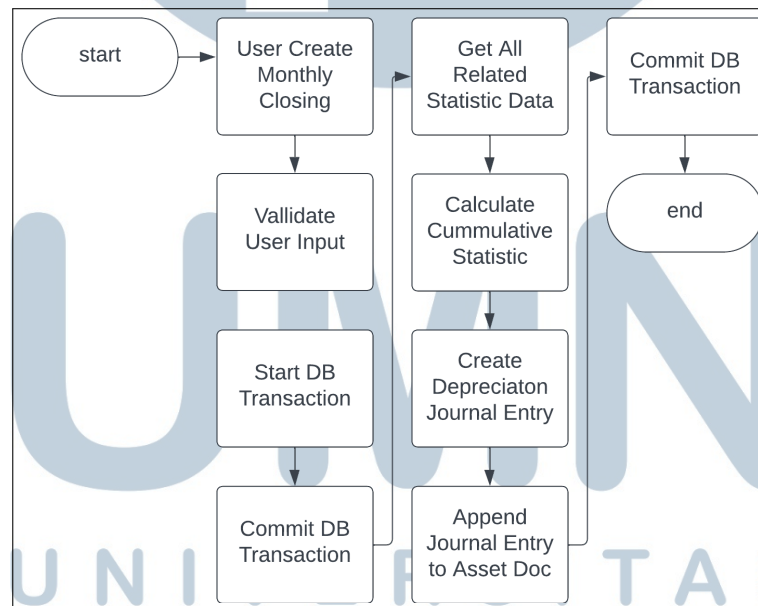
bahwa aplikasi dapat berjalan dengan baik sehingga proyek dapat dilanjutkan ke proses *release* ke situs web *production*.

### C.3 Pengembangan sub-modul Doctype House Monthly Closing

#### 1. Pengumpulan *requirements* dan diskusi bersama stake holder

Pada tahap ini tim pengembang aplikasi ERP ayam layer melakukan pengumpulan *requirements* atau kebutuhan yang perlu dilakukan pada proyek rencana pengembangan aplikasi ERP ayam layer bersama *stake holder* terkait.

- Pengguna dapat melakukan *submit* Doctype House Monthly Closing tanpa mengalami *timeout* atau *error*.
- Pengguna dapat melihat Doctype Asset Daily Depreciation yang dihasilkan dari proses *submit* Doctype House Monthly Closing setiap bulannya sesuai dengan contoh pada Gambar *flowchart* 3.9.

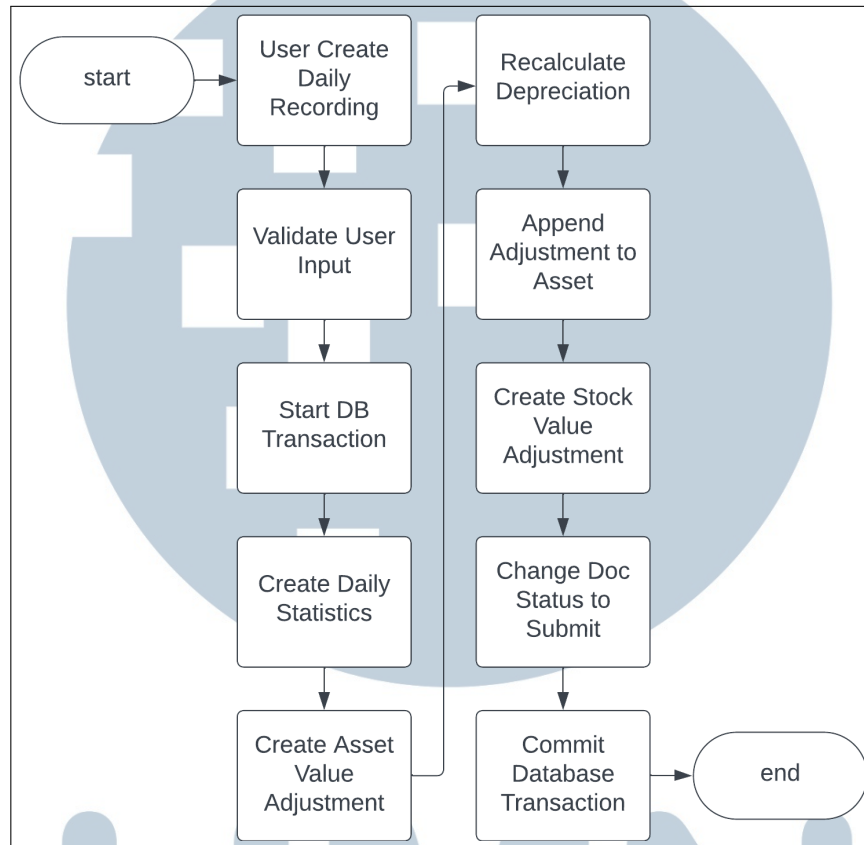


Gambar 3.9. *Flowchart* Doctype House Monthly Closing

#### 2. Pengembangan sub-modul Doctype Daily Production Recording

Setelah dilakukan pengumpulan *requirements*, selanjutnya dilakukan Pengembangan implementasi dan pengembangan sesuai dengan rancangan flowchart yang sudah ditentukan.

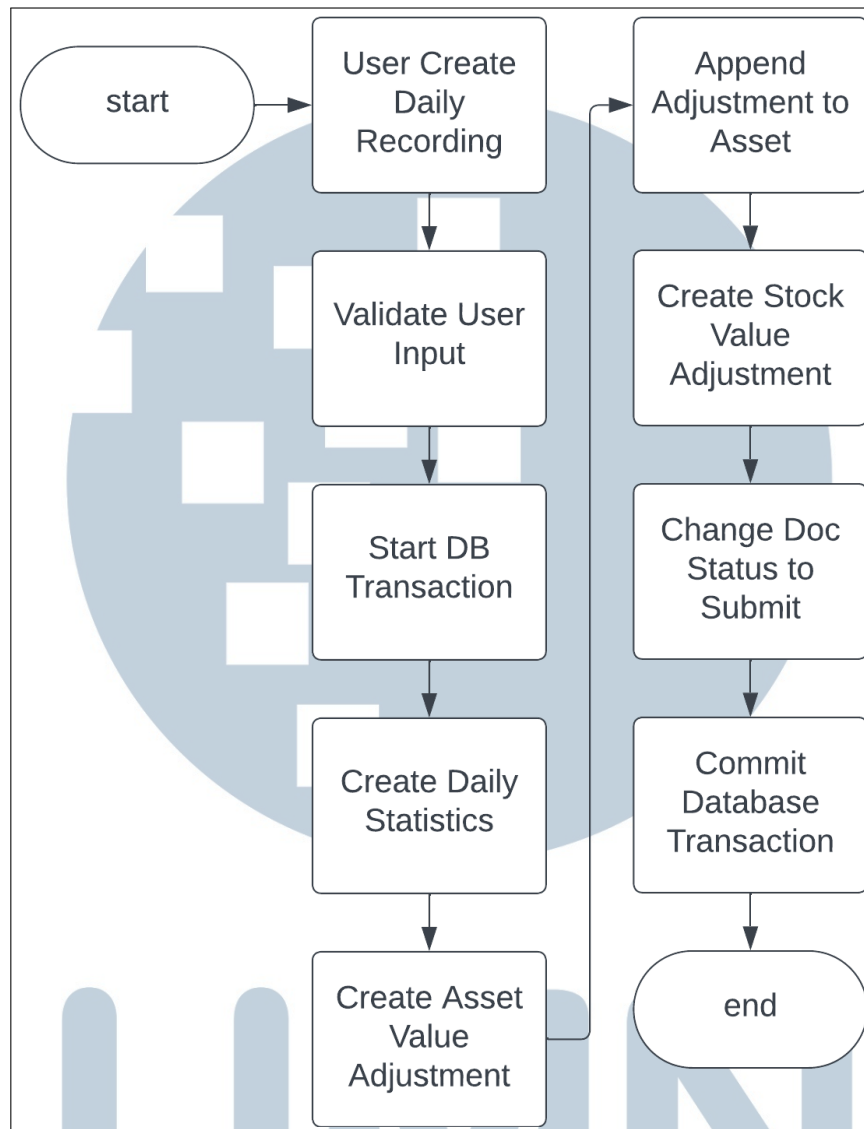
Pengembangan perubahan alur proses pada pencatatan dokumen *recording* pada Doctype Daily Production Recording. Sebelumnya ketika rekalkulasi depresiasi, Doctype Daily Production Recording tidak terbentuk.



Gambar 3.10. *Flowchart* Sistem Doctype Daily Production Recording Lama

Setelah melakukan diskusi dan simulasi jurnal, maka didapati metode pencatatan baru sesuai dengan Gambar 3.11, dengan menghilangkan bagian *recalculate* pada proses pembentukan Doctype Daily Production Recording.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.11. *Flowchart* Sistem Doctype Daily Production Recording Baru

Pada alur proses sistem baru, proses pembuatan Doctype Asset Daily Depreciation diubah dari metode harian menjadi bulanan. Perubahan ini memungkinkan pembuatan Doctype Asset Daily Depreciation yang lebih cepat dan minim *error* karena tidak perlu direkalkulasi setiap hari.

Pengembangan metode perhitungan depresiasi versi terbaru menggunakan Doctype House Monthly Closing dengan tampilan seperti pada *flowchart* 3.9. Doctype House Monthly Closing merupakan Doctype yang akan pengguna buat setiap melakukan *closing* bulanan agar jurnal depresiasi dapat terbentuk.

Doctype House Monthly Closing didesain agar pengguna dapat dengan cepat

melakukan *closing* terhadap *house*. Pengguna nantinya hanya perlu mengisi *field posting date, company, farm dan house*, sementara nantinya untuk *field* lainnya akan digenerate otomatis oleh sistem seperti pada Gambar 3.12.

Assigned To	Posting Date *
+	29-04-2024
Attachments	Company *
Attach File +	Sample Company
Shared With	Farm *
+	Sample Farm
Tags	House *
Add a tag ...	Sample House
0 · 0 FOLLOW	From Date *
You edited this 3 weeks ago	01-02-2024
You created this 3 weeks ago	To Date *
	29-02-2024
	Total Day *
	29
	<b>Statistic</b>
	Culling
	0
	Mortality
	38

Support status: Active - Complimentary | Expiry Date: null | Engineered by Agile Technica, Powered by ERPNext

Gambar 3.12. Tampilan Doctype House Monthly Closing

### 3.3.3 Release & Deployment Aplikasi ERP ayam layer Versi Terbaru

Setelah melakukan *testing* dan telah dipastikan performa yang didapati oleh aplikasi ERP ayam layer lebih baik dari versi pendahulunya serta segala fitur berjalan dengan normal, maka dilakukan *release* dan *deployment* ke situs web *production*. Langkah-Langkah yang dilakukan oleh tim pengembang aplikasi ERP ayam layer adalah sebagai berikut:

1. Informasikan kepada pengguna dan pihak *management* bahwa tim pengembang aplikasi ERP ayam layer akan melakukan *release* dan *deployment*. Informasikan juga untuk tidak menggunakan situs web *production* selama proses *release* dan *deployment* dilakukan. *Release* akan eksekusi setelah jam *office-hour* (setelah jam 18:00 WIB) untuk mencegah pengguna mengakses situs web *production* saat situs web *production* sedang *dideploy* ulang yang bisa menyebabkan *data lost*.
2. Lakukan *backup database* situs web *production*. Hal dilakukan untuk meminimalisir resiko jika terjadi *issue* setelah *upgrade* dilakukan, dengan

begitu tim pengembang aplikasi ERP ayam layer dapat dengan mudah melakukan *restore* data yang sebelumnya sudah di *backup*.

3. *Restore* data dari situs web *production* ke UAT *site* dengan versi Frappe 12.
4. Dikarenakan ada data yang *incompatible* pada versi 14, lakukan penghapusan data tersebut sesuai yang telah dilakukan investigasi sebelumnya.
5. Lakukan *backup database* yang sudah *fixed* dari *site*.
6. Lakukan *upgrade* versi Frappe dan ERPNext pada situs web *production* lewat panel Frappe Cloud.
7. *Restore* data yang telah di-*backup* dari *site* UAT ke situs web *production*.
8. Monitor jika ada *issue* yang muncul situs web *production*.

### **3.4 Kendala dan Solusi**

#### **3.4.1 Kendala**

Selama proses kerja magang di PT Sumber Inovasi Informatika terdapat kendala berupa proses *deployment* yang masih dilakukan secara manual oleh tim pengembang aplikasi ERP ayam layer, sehingga membutuhkan developer untuk selalu *stand by* pada saat *deployment*.

#### **3.4.2 Solusi**

Solusi atas kendala yang ditemukan adalah dengan mengimplementasikan *ci/cd pipeline* pada proses *deployment* aplikasi ERP ayam layer agar proses *deployment* lebih cepat dan lebih sedikit membutuhkan developer *resources*.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A