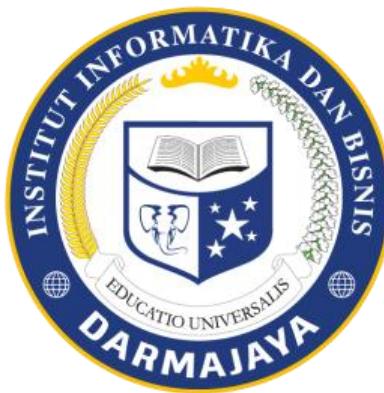


**RANCANG BANGUN APLIKASI MONITORING HASIL PRODUKSI DAN  
INFORMASI PRESERVASI (PERAWATAN) TANAMAN KARET MENGGUNAKAN  
METODE K-NEAREST NEIGHBOR**

**SKRIPSI**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Menyelesaikan Pendidikan Program Sarjana Strata Satu (S1)  
Pada Program Studi Teknik Informatika



Disusun Oleh:

**Feven Indriyani**

**NPM.1411010079**

**FAKULTAS ILMU KOMPUTER**  
**JURUSAN TEKNIK INFORMATIKA**  
**INSTITUTE INFORMATIKA DAN BISNIS DARMAJAYA**  
**BANDAR LAMPUNG**

**2018**



### **PERNYATAAN**

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi yang saya buat ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggung jawaban sepenuhnya berada di pundak saya.

Bandar Lampung, 7 September 2018



## HALAMAN PERSETUJUAN

Judul Laporan

: RANCANG BANGUN APLIKASI MONITORING  
HASIL PRODUKSI DAN INFORMASI PRESERVASI  
(PERAWATAN) TANAMAN KARET  
MENGGUNAKAN METODE K-NEAREST  
NEIGHBOR

Nama Mahasiswa

: FEVEN INDRIYANI

No. Pokok Mahasiswa : 1411010079

Jurusan

: Teknik Informatika



Dosen Pembimbing

Hariyanto Wibowo,S.kom., M.T.I  
NIK. 00051297

Ketua Jurusan  
Teknik Informatika

Yuni Arkhiansyah,S.Kom., M.Kom  
NIK. 00480802

## HALAMAN PENGESAHAN

Telah Diuji dan Dipertahankan Didepan Tim Penguji Skripsi  
Jurusan Teknik Informatika Institut Informatika dan Bisnis Darmajaya  
Bandar Lampung dan Dinyatakan Diterima untuk  
Memenuhi Syarat Guna Memperoleh  
Gelar Sarjana Komputer

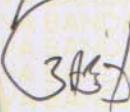
### Mengesahkan

#### 1. Tim Penguji

Ketua : Fitria, ST., MKom

Anggota : Apri Triyansyah, S.Kom., M.T.I

#### Tanda Tangan

  
.....  
  
.....

#### 2. Dekan Fakultas Ilmu Komputer

  
Srivanto, S.Kom., M.M

NIK. 00210800

## **RIWAYAT HIDUP**

1. Identitas
  - a. Nama : Feven indriyani
  - b. NPM : 1411010079
  - c. Tempat / Tanggal Lahir : Pagar Dewa, 21 Juni 1996
  - d. Agama : Islam
  - e. Alamat : Jl. Raya Prabumulih-Baturaja Kec. Lubai Ulu Kab. Muara Enim Sumatera Selatan
  - f. Suku : Palembang
  - g. Kewarganegaraan : Indonesia
  - h. E-Mail : Feven.hm97@gmail.com
  - i. HP : 082177557145
2. Riwayat pendidikan yang pernah ditempuh oleh penulis :
  - a) SDN 1 Pagar Dewa Kec. Lubai Ulu Kab. Muara Enim Sumatera Selatan, lulus tahun 2008.
  - b) SMPN 2 Kec. Lubai Ulu Kab. Muara Enim Sumatera Selatan, lulus tahun 2011.
  - c) SMA Yadika Baturaja Ogan Komering Ulu Sumatera Selatan, lulus tahun 2014.
  - d) Pada tahun 2014 penulis diterima di IIB Darmajaya Program Studi S-1 Teknik Informatika.

Bandar Lampung, 7 September 2018

Feven Indriyani  
NPM. 1411010079

## PERSEMPAHAN

*Alhamdulillahirabbil'alamin* , segala puji kepada Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Pada lembar persembahan ini penulis mengucapkan terimakasih yang sebesar-besarnya kepada :

- ✓ Allah SWT yang maha memberi petunjuk. Alhamdulillah Allah telah memberikan saya keteguhan hati dan kesabaran.
- ✓ Teruntuk kedua orang tuaku, abah dan ummi. Terimaksih untuk dua pasang mata saksi atas tumbuhnya aku hingga dewasa, terimaksih untuk dua pasang lengan penyanga saat aku kehilangan arah, dan terimaksih untuk dua kepala teladan saat aku tak tahu harus berbuat apa. Untuk setiap bulir peluh yang jatuh, untuk setiap air mata berderai di kedua pelupuk matamu, untuk setiap beban berat yang kau tanggung di balik punggungmu, untuk setiap nyala semangat yang kau percik di dalam dadaku, untuk setiap keteguhan yang kau ajarkan perlakan, untuk setiap ketegasan yang terukir dari urat menyembul di lenganmu, untuk setiap kesabaran yang kau tunjukkan, untuk setiap gelak gemetar bibirmu saat berkata, "*Maaf nak, sabarlah dahulu. Uang saku akan segera kami kirimkan. Kami sedang berusaha dirumah.*" Untuk setiap hal yang kau lakukan untuk memperjuangkan aku. Maka ku persembahkan untukmu sebuah hasil kerja keras dari jerih dan air mata, bukan untuk apa-apa selain ingin membuat mu bahagia saat kau berteriak bangga kepada siapas aja“ Perkenalkan, ini anak saya.”
- ✓ Untuk ketiga adik kandung saya; Fenni Andinie, Fania Sinta Bella, Faeza Ali Zacky, semua pencapaian ini adalah langkah awal saya untuk bisa membuka jalan baru yang saya namai kesuksesan, dan kalian adalah alasan kedua kenapa saya harus sukses, karena kalian harus terus sekolah bahkan harus melebihi saya.

- ✓ Untuk Adelia Anggraini Chaniago, Terimakasih telah menjadi saudara yang selalu mensuport, yang selalu menasehati diri yang keras kepala ini, terimakasih atas semua solusi yang selalu diberikan selama ini dan terimakasih karena selalu ada dalam setiap proses jatuh bangun saya selama menempuh pendidikan di IIB Darmajaya.
- ✓ Dosen pembimbing Bapak Hariyanto Wibowo, S.kom., M.T.I yang selalu sabar dalam meberikan arahan.
- ✓ Bapak Ketua Jurusan Teknik Informatika Institut Informatika dan Bisnis Darmajaya, Bapak Yuni Arkhiansyah, S.kom., M.kom
- ✓ Pihak Dosen pengajar yang telah memberikan ilmunya beserta Staff TU Jurusan Teknik Informatika Institut Informatika dan Bisnis Darmajaya.
- ✓ Untuk teman-teman seperjuangan dalam menyelesaikan skripsi ini (Hafsa Mukaroma, Dewi Anggraini, Mega Lestari, Ajeng, Selvi, Muji). Semoga apa yang telah kita selesaikan hari ini bisa menjadi modal awal untuk memulai target selanjutnya.
- ✓ Untuk teman- teman Teknik Informatika angkatan 2014 yang belum menyelesaikan skripsi, semoga segera terselesaikan dan dimudahkan.
- ✓ Serta semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam pengerjaan Tugas Akhir ini.

## MOTTO

*“Akan menjadi tidak mungkin jika kamu berfikir itu sulit, Tapi akan menjadi mungkin saja jika kamu berfikir itu mudah. Karena semua hal yang terjadi, berawal dari fikiran diri sendiri. So positive Thinking okay!”*

---Feven Indriyani---

*“Apapun masalahnya, makan harus tetap banyak dan tidur harus tetap nyenyak. “*

---Feven Indriyani---

## **ABSTRAK**

**Rancang Bangun Aplikasi Monitoring Hasil Produksi dan Informasi Preservasi  
(Perawatan) Tanaman Karet Menggunakan Metode K-Nearest Neighbor.**

**Oleh :**

**Feven Indriyani**

Monitoring adalah suatu proses yang dilakukan secara terus menerus yang merupakan bagian yang bersifat integral dari manajemen yang meliputi penilaian yang bersifat sistematis terhadap kemajuan suatu pekerjaan. Monitoring adalah penilaian yang sistimatis dan terus menerus terhadap kemajuan suatu pekerjaan.

Aplikasi monitoring hasil produksi PT. Perkebunan Nusantara Vii Unit Muara Enim merupakan salah satu bentuk sarana aplikasi yang bertujuan mempermudah mandor dalam penginputan hasil produksi dan mempermudah asisten kepala wilayah dalam memonitoring hasil produksi berdasarkan hasil produksi yang diinputkan, serta tersedia informasi tentang preservasi tanaman karet.

Algoritma *K-Nearest Neighbor* adalah algoritma yang digunakan dalam pengklasifikasian hasil produksi untuk menentukan katagori hasil produksi apakah termasuk kedalam hasil produksi yang rendah, sedang atau tinggi. Algoritma *K-Nearest Neigbor* merupakan sebuah metode untuk melakukan klasifikasi terhadap obyek baru berdasarkan tetangga terdekat.

**Kata kunci: Java, Android, Monitoring, K-nearest neighbor**

## **ABSTRACT**

### **DESIGNING THE K-NEAREST NEIGHBOR-BASED APPLICATION FOR MONITORING THE RUBBER PLANTATION PRODUCTION RESULTS AND INFORMATION MAINTENANCE**

**By**

**Feven Indriyani**

Nusantara Estate 7 Ltd. implements businesses in the plantation sector. The problem statement of this research was that the supervision and input on the production results in Nusantara Estate 7 Karang Agung was still done manually. To overcome this problem, an application for monitoring the production results and the information maintenance of the rubber plants was designed through K-Nearest Neighbor method.

The method to develop this application was the Waterfall system. This system was conducted through several phases i.e., planning, modeling, implementing, and testing. Moreover, the algorithmic method used in this system was K-Nearest Neighbor. This method was the algorithm used to classify the production so that the production results were determined into low, medium or high production.

This application was one of the facilities to facilitate the regional head supervisors and assistants in monitoring and inputting the production data without the conventional or procedural manuals so that the database storage media as production data was archived properly.

**Keywords:** Java, Android, Monitoring, K-nearest neighbor

## **PRAKATA**

Dengan Mengucap *Alhamdulillahirobbil;alamin*, segala puji syukur penulis panjatkan kehadiran Allah SWT, karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi ini dengan baik. Penulis menyadari bahwa dalam penyusunan skripsi ini tidak lepas dari bantuan semua pihak. Oleh karena itu dalam kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. H. Andi Desfiandi, S.E., M.A. Selaku Ketua Yayasan Alfian Husin.
2. Bapak Ir. Firmansyah Y. Alfian, MBA., MSc. Selaku Rektor Institut Informatika dan Bisnis Darmajaya Bandar Lampung.
3. Bapak Dr. RZ. Abdul Aziz, ST., M.T. Selaku Wakil Rektor Bidang Akademik dan Riset Institut Informatika dan Bisnis Darmajaya Bandar Lampung
4. Bapak Ronny Nazar, SE, MM. Selaku Wakil Rektor Bidang Sumber Daya, Keuangan dan Aset Logistik Institut Informatika dan Bisnis Darmajaya Bandar Lampung.
5. Bapak Muprihan Thaib, S.SOS., MM. Selaku Wakil Rektor Bidang Kemahasiswaan dan Pengembangan Bisnis Institut Informatika dan Bisnis Darmajaya Bandar Lampung.
6. Bapak Sriyanto, S.Kom., M.M Selaku Dekan Fakultas Ilmu Komputer Institut Informatika dan Bisnis Darmajaya.

7. Bapak Yuni Arkhiansyah, M.Kom. Selaku Ketua Jurusan Teknik Informatika Institut Informatika dan Bisnis Darmajaya Bandar Lampung.
8. Bapak Hariyanto Wibowo, S.kom., M.T.I. Selaku Dosen Pembimbing yang telah membantu membimbing dan mengarahkan serta memberikan petunjuk sehingga skripsi ini dapat diselesaikan dengan baik.
9. Seluruh dosen dan Staff Institut Informatika dan Bisnis Darmajaya Bandar Lampung, yang telah memberi bantuan baik langsung maupun tidak langsung selama saya menjadi mahasiswa.
10. Ayahanda Edi Hermanto dan Ibunda Nur Anita dan ketiga adik saya Fenni Andinie, Fania Sinta Bella serta Faeza Ali Zacky , yang menjadi motivasi bagi saya untuk selalu semangat menyelesaikan skripsi ini.
11. Semua Pihak yang telah memberikan bantuan dan petunjuk sehingga saya dapat lebih mudah dalam menyususn skripsi ini.
12. Almamaterku tercinta Institut Informatika dan Bisnis Darmajaya Bandar Lampung.

Akhir kata, bahwa tidak ada sesuatu yang sempurna melainkan kesempurnaan hanya milik allah SWT, termasuk dalam pembuatan skripsi ini, tidak luput dari kesalahan. Sehingga dibutuhkan saran dan kritik untuk membangun sebuah karya yang lebih baik lagi dimasa mendatang.

Bandar Lampung, 7 September 2018

**Feven Indriyani**

**NPM. 1411010079**

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	i
<b>PERNYATAAN.....</b>	ii
<b>HALAMAN PERSETUJUAN .....</b>	iii
<b>HALAMAN PENGESAHAN.....</b>	iv
<b>RIWAYAT HIDUP .....</b>	v
<b>HALAMAN PERSEMBAHAN .....</b>	vi
<b>MOTTO .....</b>	viii
<b>PRAKATA .....</b>	x
<b>DAFTAR ISI.....</b>	xi
<b>DAFTAR TABEL .....</b>	xv
<b>DAFTAR GAMBAR .....</b>	xvi
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan penelitian.....	3
1.5 Manfaat Penelitian .....	4
1.6 Sistematika Penelitian .....	4

## **BAB II LANDASAN TEORI**

2.1 Tanaman Karet .....	6
2.2 Aplikasi .....	6
2.3 Monitoring .....	7
2.4 Android .....	7
2.4.1 Versi Android.....	8
2.4.2 Android Studio.....	9
2.4.3 Elemen Android .....	10
2.5 Pemrograman Java .....	10
2.6 SQuite .....	11
2.7 Algoritma.....	11
2.8 Algoritma K-Nearest Neighbor.....	12
2.9 UML ( <i>Unified Modeling Language</i> ) .....	14
2.9.1 Bagian-bagian UML .....	14
2.9.2 Tujuan dan Keunggulan UML .....	15
2.9.3 Diagram UML .....	16
2.10 Pengujian <i>Black-Box Testing</i> .....	20
2.11 Penelitian Terkait.....	21

## **BAB III METODELOGI PENELITIAN**

3.1 Metode Pengumpulan Data.....	23
3.2 Metode Pengembangan Aplikasi.....	23
3.2.1 Analisis Kebutuhan Aplikasi.....	25
3.2.1.1 Analisis Sistem yang Berjalan.....	25
3.2.1.2 Kelemahan Sistem yang Berjalan.....	26

3.2.1.3 Analisis Sistem yang diajukan.....	26
3.2.2 Analisis Algoritma yang Digunakan.....	27
3.2.2.1 Algoritma K-Nearest Neighbor.....	27
3.2.3 Desain Aplikasi.....	28
3.2.3.1 Desain UML.....	28
3.3 Perancangan Antar Muka.....	33

## **BAB IV HASIL PENELITIAN DAN PEMBAHASAN**

4.1 Hasil Penelitian .....	35
4.2 Pembahasan .....	35
4.2.1 Proses Instalasi Perangkat Lunak .....	35
4.2.2 Aplikasi Selesai Diinstal .....	35
4.2.3 Halaman Pengguna Perangkat Lunak .....	36
4.2.4 Halaman Splash Screen .....	37
4.2.5 Halaman Login .....	37
4.2.6 Halaman Input Username dan Password .....	38
4.2.7 Halaman utama login sebagai mandor .....	39
4.2.8 Halaman Input Data Produksi.....	39
4.2.9 Proses Penginputan Data Produksi.....	40
4.2.10 Proses penyimpanan Data Inputan.....	41
4.2.11 Halaman Monitoring Kepala Wilayah.....	41
4.2.12 Tampilan Halaman Preservasi.....	42
4.2.13 Tampilan Halaman Klasifikasi.....	43
4.3 Pembahasan Algoritma K-Nearest Neighbor .....	44
4.4 Pengujian Perangkat Lunak .....	51
4.4.1 Pengujian Instalasi .....	52

4.4.2 Pengujian Pengunaan.....	52
4.4.3 Pengujian Algoritma K-Nearest Neighbor.....	52
4.5 Kelebihan dan Kekurangan .....	55

## **BAB V SIMPULAN DAN SARAN**

5.1 Kesimpulan .....	57
5.2 Saran .....	57

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## **DAFTAR TABEL**

	<b>Halaman</b>
Tabel 2.1 Simbol-Simbol Aktivitas .....	19
Tabel 2.2 Simbol-Simbol Aktivitas (Lanjutan) .....	19
Tabel 2.3 Penelitian Terkait .....	21
Tabel 3.1 Kebutuhan Perangkat Lunak .....	27
Tabel 3.2 Kebutuhan perangkat Keras .....	27
Tabel 4.1 Data Produksi .....	45
Tabel 4.2 Perhitungan Data Produksi.....	46
Tabel 4.3 Pengujian Instalasi .....	52
Tabel 4.4 Pengujian Input Data .....	52
Tabel 4.5 Pengujian Algoritma K-Nearest Neighbor .....	54

## DAFTAR GAMBAR

	<b>Halaman</b>
Gambar 2.1 Evolusi Android .....	8
Gambar 2.2 Logo Android Studio .....	9
Gambar 2.3 Rumus Algoritma Knn .....	13
Gambar 2.4 Ilustrasi Algoritma Knn .....	13
Gambar 2.5 <i>Use Case</i> .....	17
Gambar 2.6 Simbol Actor .....	17
Gambar 2.7 <i>Relationship</i> .....	17
Gambar 3.1 Tahapan Waterfall .....	24
Gambar 3.2 <i>Use Case</i> yang diajukan .....	28
Gambar 3.3 Activity Diagram <i>Admin</i> .....	29
Gambar 3.4 Activity Diagram user Untuk Login.....	30
Gambar 3.5 Activity Diagram Mandor Untuk penginputan Data .....	30
Gambar 3.6 Activity Diagram Kepala Wilayah untuk Monitoring.....	31
Gambar 3.7 Activity Diagram Untuk Preservasi .....	31
Gambar 3.8 Class Diagram .....	32
Gambar 3.9 Tampilan Splash Screen dan Halaman login .....	33
Gambar 3.10 Tampilan Uac Mandor .....	33
Gambar 3.11 Tampilan Uac Asisten Kepala Wilayah .....	34
Gambar 3.12 Tampilan Halaman Preservation .....	34
Gambar 4.1 Tampilan Proses Instalasi .....	35
Gambar 4.2 Tampilan Aplikasi Selesai Di Instal .....	36
Gambar 4.3 Tampilan Icon Setelah Instalasi .....	36

Gambar 4.4 Tampilan Halaman Splash Screen.....	37
Gambar 4.5 Tampilan Halaman Login User .....	38
Gambar 4.6 Tampilan Input Username dan Password.....	38
Gambar 4.7 Tampilan Awal Login Sebagai Mandor .....	39
Gambar 4.8 Tampilan Halaman Input Hasil Produksi .....	40
Gambar 4.9 Proses Penginputan Data Produksi .....	40
Gambar 4.10 Proses Penyimpanan Data Produksi .....	41
Gambar 4.11 Tampilan Pengambilan Data Dari Database .....	42
Gambar 4.12 Tampilan Data Produksi Tiap Area .....	42
Gambar 4.13 Tampilan Halaman Preservasi .....	43

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Tanaman karet adalah tanaman tahunan yang dapat tumbuh sampai umur 30 tahun. Habitus tanaman ini merupakan pohon dengan tinggi tanaman dapat mencapai 15-20 meter. Modal utama dalam pengusahaan tanaman ini adalah batang setinggi 2,5 sampai 3 meter dimana terdapat pembulu latek. Oleh karena itu fokus pengelolaan tanaman karet ini adalah bagaimana mengelola batang tanaman ini seefisien mungkin.

PTP. Nusantara VII (Persero) melakukan usaha dalam bidang perkebunan dengan beberapa komoditas andalan dan pokok yang dibudidayakan. Komoditi yang sedang dibudidayakan oleh PT Perkebunan Nusantara VII sebanyak 4 komoditas, yaitu kelapa sawit, karet, tebu, dan teh. Wilayah kerja pengelolaan tersebar di Provinsi Lampung yaitu sebanyak 10 Unit Usaha (6 Unit Usaha di Distrik Sekampung dan 4 Unit Usaha di Distrik Seputih), Sumatera Selatan sebanyak 14 Unit Usaha (7 Unit Usaha di Distrik Muara Enim dan 7 Unit Usaha di Distrik Banyuasin), dan Bengkulu sebanyak 3 Unit Usaha dibawah wilayah Distrik Bengkulu. Komoditas yang paling banyak diusahakan di PT Perkebunan Nusantara VII adalah kelapa sawit dan karet, seperti Unit Usaha di Distrik Muara Enim tepatnya di desa Karang Agung yang menjadi tempat penelitian merupakan Unit Usaha yang komoditasnya adalah karet. Pada Distrik di Desa Karang Agung ini laporan hasil produksi oleh mandor masih dilakukan dengan membuat laporan biasa dibuku mandor, pengawasan hasil produksi oleh asisten kepala wilayah juga masih dilakukan dengan cara melihat laporan dari mandor yang masih berupa catatan biasa dalam buku mandor dan saat ini masih banyak tanaman karet yang tidak berkembang dengan baik disebabkan kurangnya pengetahuan tentang cara preservasi (perawatan) tanaman karet tersebut.

Cara yang dipakai memiliki kelemahan, antara lain mandor kesulitan untuk melaporkan data hasil produksi karena jarak perkebunan dan kantor perkebunan yang jauh, selain itu asisten kepala wilayah akan sulit memonitoring hasil produksi karena

data yang diberikan masih berupa catatan dan masih sering terlambat diberikan oleh mandor. Dan kesulitan dalam mengakses informasi jika masih mencari secara manual pada sebuah website satu ke website yang lainnya.

Berdasarkan hal tersebut, maka dapat disimpulkan bahwa dibutuhkan sebuah sistem yang dapat menginputkan data hasil produksi getah karet oleh Mandor, dan asisten kepala wilayah dapat mengawasi hasil produksi tanaman karet. Dan dibutuhkan media untuk mempermudah mengakses informasi seputar tanaman karet. Untuk itu, dibangun sebuah aplikasi monitoring hasil produksi dan informasi preservasi (perawatan) tanaman karet menggunakan metode K-Nearest Neighbor yang diharapkan dapat membantu kinerja di PTP Nusantara Vii di Unit Desa Karang Agung.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, permasalahan dalam penelitian ini adalah bagaimana membangun aplikasi monitoring hasil produksi dan mengakses informasi preservasi (perawatan) tanaman karet dapat dilakukan dengan mudah. Dimana mandor bisa menginputkan laporan data hasil produksi yang bisa otomatis akan tersimpan kedalam database, sehingga asisten kepala wilayah bisa memonitoring hasil produksi berdasarkan data hasil produksi yang diinputkan oleh mandor.

### **1.3 Batasan Masalah**

Agar tujuan lebih terarah dan sesuai dengan latar belakang yang ada, maka penulis membuat batasan masalah sebagai berikut :

1. Tanaman yang digunakan pada penelitian ini adalah tanaman karet.
2. Aplikasi monitoring hasil produksi dan informasi preservasi (perawatan) tanaman karet bisa berjalan di sistem operasi android.
3. Aplikasi sebagai media untuk menginputkan hasil produksi getah karet di PTP Nusantara Vii Unit Desa Karang Agung.

4. Aplikasi dapat menampilkan data produksi setiap bulannya dan total data produksi untuk memonitoring hasil produksi getah karet di PTP Nusantara Vii Unit Usaha di Desa Karang Agung, yang dapat diakses oleh asisten kepala wilayah.
5. Aplikasi sebagai media untuk mempermudah mengakses informasi preservasi tanaman karet.
6. Metode yang diimplementasikan pada pembuatan aplikasi ini adalah metode k-nearest neighbor.

#### **1.4 Tujuan penelitian**

Adapun tujuan dari penelitian ini sesuai dengan rumusan masalah yang telah diuraikan dibagi adalah sebagai berikut:

1. Dapat mebangun aplikasi yang mempermudah Mandor dalam menginputkan laporan hasil produksi.
2. Asisten kepala wilayah Dapat memonitoring hasil produksi tanaman karet dari waktu kewaktu berdasarkan data produksi yang diinputkan oleh Mandor.
3. Tersedianya media alternatif penyedia informasi tanaman karet.

#### **1.5 Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut:

1. Mempermudah dalam penginputan data hasil produksi oleh tiap Mandor yang bertanggung jawab dalam tiap area yang ditentukan.
2. Mempermudah asisten kepala wilayah dalam memonitoring hasil produksi dan mengetahui total hasil produksi getah karet berdasarkan data yang diinputkan oleh tiap mandor dimasing-masing area.
3. Memberikan alternatif untuk mencari informasi tentang tanaman karet menjadi lebih mudah.

## **1.6 Sistematika Penulisan**

Agar memudahkan dalam memberikan gambaran secara utuh penulisan skripsi ini, maka sistematika penulisan ini dibagi menjadi 5 (lima) bab sebagai berikut :

### **BAB 1 PENDAHULUAN**

Bab ini menjelaskan latar belakang dibuatnya aplikasi monitoring hasil produksi dan preservasi (perawatan) tanaman karet dengan metode K-Nearest Neighbor, rumusan masalah yang didapat, batasan masalah yang dibuat, tujuan, manfaat dan sistematika penulisan yang diterapkan.

### **BAB II LANDASAN TEORI**

Bab ini menjelaskan mengenai teori-teori terkait aplikasi monitoring hasil produksi dan preservasi (perawatan) tanaman karet dengan metode K-Nearest Neighbor, untuk mendukung penelitian yang dilaksanakan.

### **BAB III METODOLOGI PENELITIAN**

Bab ini berisi metode yang digunakan dalam penyelesaian permasalahan yang ditanyakan dalam perumusan masalah dan analisa yang dilakukan dalam membangun aplikasi monitoring hasil produksi dan preservasi (perawatan) tanaman karet dengan metode K-Nearest Neighbor berbasis android.

### **BAB IV HASIL PENELITIAN DAN PEMBAHASAN**

Bab ini menjelaskan tentang hasil dari pengkodean yang dilakukan, sehingga yang dibahas pada bab ini adalah bagaimana tampilan sistem saat dijalankan. Selanjutnya dipaparkan tentang instalasi perangkat lunak dan bagaimana sistem ini diuji.

**BAB V SIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dan saran yang diharapkan dapat bermanfaat untuk pengembangan *Aplikasi Monitoring Hasil Produksi dan Preservasi (Perawatan) Tanaman Karet* berbasis android selanjutnya.

**DAFTAR PUSTAKA****LAMPIRAN**

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tanaman Karet**

Tanaman Karet (*Hevea Brasiliensi*) merupakan tanaman perkebunan yang penting baik dalam konteks ekonomi masyarakat maupun sumber penghasil devisa non migas bagi negara. Tanaman karet berasal dari daerah tropika lembah Amazon Brazillia dengan curah hujan 2000-3000 mm/tahun dan hari hujan antara 120-170 hari/tahun (sutardi, 1981). Pengembangan karet berkonsentrasi pada daerah 10 LU dan 10 LS (Mors, 1977). Sebagian besar areal perkebunan karet Indonesia terletak di Sumatera (70%), Kalimantan (24%) dan jawa (4%) dengan curah hujan 1500-4000 mm/tahun dengan rat-rata bulan kering 0-4 bulan pertahun dan terletak pada elevasi dibawah 500 m diatas permukaan laut. (Sabarman Damanik, 2012)

#### **2.2 Aplikasi**

Menurut Nazrudin Safaat H (2015) Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (application suite). Contohnya adalah Microsoft Office dan Open Office.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

### **2.3. Monitoring**

Monitoring adalah suatu proses yang dilakukan secara terus menerus yang merupakan bagian yang bersifat integral dari manajemen yang meliputi penilaian yang bersifat sistematis terhadap kemajuan suatu pekerjaan. Adapun beberapa tujuan dari *Monitoring* adalah mengkaji apakah kegiatan – kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan.

( Mudjahidin dan Nyoman Dita Pahang Putra, 2010)

### **2.4 Android**

Android adalah sebuah sistem operasi untuk *smartphone* dan Tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. (Yuni Puspita Sari, 2016)

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google membeli Android Inc, yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcom, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *Open source* pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler. (Nazrudin Safaat H, 2011, p.1).

### 2.4.1. Versi Android

Adapun versi-versi android yang pernah dirilis adalah sebagai berikut :

1. Android Versi 1.0 Alpha - Dirilis pada 23 September 2008.
2. Android Versi 1.1 Bender (Beta) - Dirilis pada 9 Februari 2009.
3. Android Versi 1.5 CupCake - Dirilis pada 27 April 2009.
4. Android Versi 1.6 Donut - Dirilis pada 15 September 2009.
5. Android Versi 2.0 - 2.1 Eclair - Dirilis pada 26 Oktober 2009.
6. Android Versi 2.2 Frozen Yoghurt - Dirilis pada 10 Mei 2010.
7. Android Versi 2.3 GingerBread - Dirilis pada 6 Desember 2010.
8. Android Versi 3.0-3.2 HoneyComb - Dirilis pada 22 Februari 2011.
9. Android Versi 4.0 Ice Cream Sandwich - Dirilis pada 19 Oktober 2011.
10. Android Versi 4.1 - 4.3 Jeally Bean - Dirilis pada 27 Juni 2012.
11. Android Versi 4.4 KitKat - Dirilis pada 31 Oktober 2013.
12. Android Versi 5.0 Lollipop - Dirilis pada 5 Juni 2014.
13. Android Versi 8.0 Oreo - Dirilis pada 21 Agustus 2017.



Gambar 2.1 Evolusi Android.

(sumber : <http://en.miui.com/thread-475925-1-1.html>)

#### 2.4.2. Android Studio



Gambar 2.2 Logo Android Studio

(sumber : <https://developer.android.com/studio/index.html>)

Android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools).

Android studio memiliki fitur:

- a. Projek berbasis pada Gradle Build
- b. Refactory dan pemberahan bug yang cepat.
- c. Tools baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibilitas aplikasi dengan cepat.
- d. Mendukung Proguard And App-signing untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah.
- f. Didukung oleh Google Cloud Platform untuk setiap aplikasi yang dikembangkan.

### **2.4.3. Elemen Android**

#### **1. Dalvik Virtual Machine (DVM)**

Salah satu element kunci dari Android adalah Dalvik Virtual Machine (DVM). Android berjalan di Dalvik Virtual Machine (DVM) bukan di Java Virtual Machine (JVM), sebenarnya banyak persamaannya dengan Java Virtual Machine (JVM) seperti Java ME (Java Mobile Edition), tetapi Android menggunakan virtual machine sendiri yang dirancang untuk memastikan beberapa fitur-fitur berjalan lebih efisien pada perangkat mobile.

#### **2. Android SDK (Software Development Kit)**

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi middleware dan aplikasi kunci yang di-release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan.

#### **3. ADT (Android Development Tools)**

Android development tools adalah plugin yang di desain untuk IDE Android Studio yang memberikan kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Android Studio. Dengan menggunakan ADT untuk Android Studio akan memudahkan dalam membuat aplikasi project Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya.

## **2.5 Pemrograman Java**

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystem saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi

kedalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus di desain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal juga dengan slogan “Tulis sekali, jalankan dimanapun”. Saat ini Java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi. (Moh. Rochman Wahid Maulana, 2017).

## **2.6 SQLite**

SQLite adalah suatu library yang menerapkan mesin database self-contained, serverless, zero-configuration, dan transactional. Self-contained berati SQLite membutuhkan sedikit sekali dukungan dari library eksternal atau dari sistem operasi. Serverless berati SQLite dalam mengakses database baik itu read atau write dapat secara langsung dari file database tanpa melalui proses server dan tidak mendukung pengaksesan secara remote (artinya database SQLite bisa dikendalikan dari jarak jauh dengan adanya jaringan komputer (“Computer Network”), baik melalui jaringan lokal (intranet) atau internet), dimana kebanyakan mesin SQL database diterapkan sebagai proses server yang terpisah. Zeroconfiguration menunjukkan SQLite tidak membutuhkan instalasi sebelum penggunaannya. Transactional SQLite merupakan suatu transaksional database, dimana dalam melakukan perubahan proses query menerapkan Atomic, Consistent, Isoalated, and Durable (ACID). (Angga Setiyadi & Tati Harihayati, 2013).

## **2.7 Algoritma**

Algoritma berasal dari nama seorang Ilmuan Arab yang bernama Abu Ja'far Muhammad Ibnu Muasa Al Khuwarizmi penulis buku berjudul Al Jabar Wal Muqabala (Buku Pemugaran dan Pengurangan). Kata Al Khuwarizmi dibaca orang barat menjadi Algorism yang lambat laun menjadi Algorithm diserap dalam bahasa

Indonesia menjadi Algoritma. Algoritma dapat diartikan urutan langkah-langkah (Instruksi-instruksi/aksi-aksi) terbatas untuk menyelesaikan sesuatu. (Nuraini, 2015).

Syarat-syarat Algoritma, yaitu:

1. *Finiteness* (Keterbatasan).

Algoritma harus berakhir setelah melakukan sejumlah proses.

2. *Definiteness* (Kepastian).

Setiap langkah Algoritma harus di definisikan dengan tepat dan tidak menimbulkan makna gand.

3. *Input* (Masukan).

Sebuah Algoritma memiliki nol atau lebih masukan yang diberikan kepada algoritma sebelum dijalankan.

4. *Output* (Keluaran)

Setiap Algoritma memberikan satu atau beberapa hasil keluaran

## 2.8 Algoritma Metode K-Nearest Neighbor (KNN)

KNN dilakukan dengan mencari kelompok k objek dalam data training yang paling dekat (mirip) dengan objek pada data baru atau data testing . Algoritma K-Nearest Neighbor adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Nearest Neighbor adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dan kasus lama yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada.

Klasifikasi K-NN bekerja berdasarkan pada analogi, diamana data uji dan training dikomparasikan dan ditarik kesimpulan berdasarkan kesamaan data yang dihasilkan oleh komparasi tersebut. Perhitungan dilakukan berdasarkan jarak ( kedekatan data) yang kemudian dikenal dengan nama Euclidean Distance

$$d_{ik} = \sqrt{\sum_j^m (C_{ij} - C_{kj})^2}$$

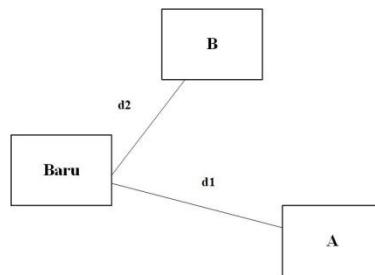
Gambar 2.3 Rumus Algoritma Knn

Dimana :

$C_{ij}$  = Sampel data

$C_{kj}$  = Data baru

K-Nearest Neighbor (KNN) merupakan sebuah metode untuk melakukan klasifikasi terhadap obyek baru berdasarkan (K) tetangga terdekatnya. KNN termasuk algoritma supervised learning, dimana hasil dari query instance yang baru, diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Kelas yang paling banyak muncul yang akan menjadi kelas hasil klasifikasi.



Gambar 2.4 Ilustrasi Algortima K-Nearest Neighbor

Neighbor adalah suatu pendekatan untuk menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. Illustrasi kedekatan kasus pada Gambar 2.3 memberikan gambaran tentang proses mencari solusi terhadap seorang pasien baru dengan mengacu pada solusi dari kasus terdahulu. Untuk mencari kasus mana yang akan digunakan, maka dihitung kedekatan antara kasus baru dengan semua kasus lama.

Kasus lama dengan kedekatan terbesar-lah yang akan diambil solusinya untuk digunakan pada kasus baru. (Toto Andri Puspito, 2017)

## 2.9 Unified Modelling Language (UML)

Menurut (Adi Nugroho, 2010) menyampaikan UML merupakan bahasa untuk membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh *Grady Booch*, *James Rumbaugh*, dan *Ivar Jacobson* di bawah bendera *Rational Software Crop*. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

### 2.9.1 Bagian-Bagian UML

UML adalah *view*, diagram, model element, dan *general mechanism*. Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain:

#### 1. Use Case Diagram

*Use case* adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe intraksi antara lain user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata user. Sedangkan *Use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta analis dan clinet.

## 2. *Class Diagram*

*Class* adalah deskripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *Class diagram* dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

## 3. *Sequence diagram*

*Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

## 4. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau intraksi.

### 2.9.2 Tujuan dan Keunggulan UML

Tujuan UML adalah sebagai berikut: Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi object, menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

Keunggulan menggunakan UML dibandingkan menggunakan metodologi terstruktur:

### 1. *Uniformity*

Pengembangan cukup menggunakan satu metodologi dari tahap analisis hingga perancangan. Memungkinkan merancang komponen antarmuka secara terintegrasi bersama perancangan perangkat lunak dan perancangan struktur data.

### 2. *Understandability*

Kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah yang sesungguhnya sehingga lebih mudah untuk dipahami.

### 3. *Stability*

Kode program yang dihasilkan relatif stabil sepanjang waktu, karena mendekati permasalahan yang sesungguhnya.

### 5. *Reusability*

Dengan metodologi berorientasi objek, dimungkinkan penggunaan ulang kode, sehingga pada akhirnya akan sangat mempercepat waktu pengembangan perangkat lunak (sistem informasi).

### **2.9.3 Diagram UML**

Model-model diagram dapat dikelompokkan berdasarkan sifatnya yaitu statis dan dinamis. Jenis diagram tersebut antara lain:

#### 1) Diagram Kelas

Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, dan relasi-relasi.

#### 2) Diagram Paket

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

#### 3) Diagram *Use case*

Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama

penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

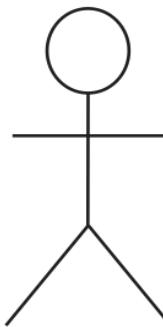
Elemen use case terdiri dari:

- a. Diagram *use case*, disertai dengan narasi dan scenario seperti pada gambar 2.2 berikut:



Gambar 2.5 *use case*

- b. Aktor, mendefinisikan entitas di luar sistem yang menggunakan system seperti gambar 2.3 sebagai berikut:



Gambar 2.6 Simbol Aktor

- c. Asosiasi, mengindikasikan aktor mana yang berinteraksi dengan *use case* dalam suatu system seperti gambar 2.4 berikut:



Gambar 2.7 *Relationship*

d. <<include>> dan <<extend>>

Merupakan indikator yang menggambarkan jenis relasi dan interaksi antar *use case*.

e. Generalisasi, menggambarkan hubungan turunan antara *use case* atau antar actor.

*Use case* mengekspresikan apa yang user harapkan terhadap sistem. Narasi *use case* menjelaskan sistem secara detail bagaimana *user* berinteraksi dengan sistem saat mengakses *use case*.

1) Diagram Interaksi atau *Sequence*

Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

2) Diagram Komunikasi

Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek – objek yang menerima serta mengirim pesan.

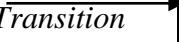
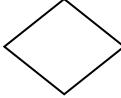
3) Diagram *Statechart*

Bersifat dinamis. Diagram ini memperlihatkan keadaan-keadaan setiap sistem, memuat status, transisi, kejadian serta aktivitas.

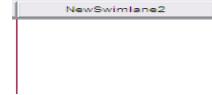
4) Diagram *Aktivitas*

Bersifat dinamis. Diagram *aktivitas* adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – sungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek. Pada tabel 2.2 berikut merupakan, symbol - simbol yang ada pada diagram aktivitas.

Tabel 2.1 Simbol –Simbol Aktivitas.

Simbol	Keterangan
<i>Start State</i> 	Menunjukkan dimulainya suatu <i>workflow</i> pada sebuah <i>activity diagram</i> .
<i>End State</i> 	Menggambarkan akhir atau terminal dari pada sebuah activity diagram.
<i>State</i> <i>Transition</i> 	Menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya.
<i>Decision</i> 	Suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.

Tabel 2.2 Simbol –Simbil Aktivitas (lanjutan).

<i>Activity</i> 	Menunjukkan aktivitas yang ada dalam system
<i>Swimlane</i> 	Membantu untuk lebih memahami siapa atau pada yang memulai aktivitas.

### 5) Diagram Komponen

Bersifat statis diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem pada komponen-komponen yang telah ada sebelumnya.

### 6) Diagram Deployment

Bersifat statis diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya.

## **2.10 Pengujian *Black-Box Testing***

Menurut Roger S. Pressman (2010) *Black Box Testing* atau Pengujian Kotak Hitam atau juga disebut *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black-Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. *Black-Box Testing* bukan merupakan alternatif dari pengujian *White Box Testing*. Sebaliknya, *Black-Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White Box Testing*.

Black Box Testing mencoba untuk menemukan kesalahan dalam kategori berikut.

1. Fungsi tidak benar atau hilang.
2. Kesalahan interface atau antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja atau perilaku.
5. Kesalahan inisialisasi dan terminasi.

## **2.11 Penelitian Terkait**

Tabel 2.3 berikut ini adalah beberapa penelitian yang berkaitan dengan Aplikasi Monitoring.

Table 2.3 Penelitian Terkait

No	Nama	Judul	Terbit	Uraian
1	Gentisyia Tri Mardiani	Sistem Monitoring Data Aset Dan Inventaris PT Telkom Cianjur Berbasis Web	Universitas Komputer Indonesia, Bandung Maret 2013	Aplikasi berbasis web ini dibangun dengan maksud dan tujuan untuk memonitoring data aset dan data distribusi perangkat seakurat mungkin sehingga dapat mengurangi masalah yang akan terjadi.

2	Mita Rohayati	Membangun Sistem Informasi Monitoring Data Inventory Di Vio Hotel Indonesia	Universitas Komputer Indonesia, Bandung Febuari 2014	Sistem monitoring data <i>Inventory</i> ini membantu memudahkan proses monitoring yang tepat dan cepat, serta manajer bisa lebih mudah dalam memantau data barang yang ada.
3	Mustakim, Giantika Oktaviani.	Algoritma K-Nearest Neighbor Classification Sebagai Sistem Prediksi Predikat Prestasi Mahasiswa	Universitas Islam Negeri Sultan Syarif Kasim Riau	pembuktian sebuah Algoritma K-NN Classification yang mampu memodelkan sebuah kasus prediksi dengan akurasi yang tinggi.

## **BAB III**

### **METODOLOGI PENELITIHAN**

#### **3.1 Metode Pengumpulan Data**

Pada bagian ini akan dijelaskan Metode yang digunakan dalam pengumpulan data penelitian yang dilaksanakan.

##### 1. Wawancara

Pada tahap ini, dilakukan wawancara terhadap narasumber yaitu Mandor PTP. Nusantara VII Unit Usaha di Kabupaten Muara Enim tepatnya di Desa Karang Agung.

##### 2. Studi Literatur

Pengumpulan data yang dilakukan dalam studi literatur ini yaitu mempelajari buku-buku serta Jurnal terkait teori mengenai android, model pengembangan perangkat lunak, dan penelitian terkait dengan judul yang diangkat.

##### 3. Pengamatan (Observasi)

Dilakukan pengamatan secara langsung dengan Mandor PT Perkebunan Nusantara Vii di Distrik Desa Karang Agung Kecamatan Muara Enim untuk mendapatkan keterangan dan melihat secara langsung perkebunan karet tersebut.

#### **3.2 Metode Pengembangan Aplikasi**

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematik dan sekuensial. Metode *Warterfall* memiliki tahapan-tahapan sebagai berikut:

##### 1) Requirements analysis and definition

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

## 2) System and software design

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

## 3) Implementation and unit testing

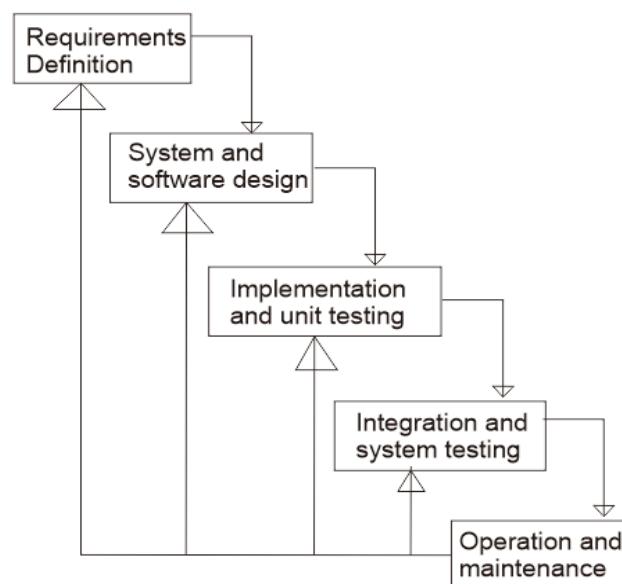
Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

## 4) Integration and system testing

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke customer

## 5) Operation and maintenance

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. Maintenance (Ginanjar Wiro Sasmito, 2017)



Gambar 3.1 Tahapan Waterfall

### **3.2.1 Analisis Kebutuhan Aplikasi (*Requirements Definition*)**

Analisa kebutuhan sistem atau aplikasi merupakan tahapan pertama yang menjadi dasar proses pembuatan sistem. Pada tahap ini dilakukan pencarian dan pengumpulan semua kebutuhan yang diperlukan untuk menunjang kelengkapan sistem atau aplikasi, kemudian mendefinisikan semua kebutuhan yang dipenuhi dalam perangkat lunak atau aplikasi yang dibuat.

#### **3.2.1.1 Analisis Sistem yang Berjalan**

PTP. Nusantara VII Unit Desa Karang Agung Kecamatan Lubai Ulu Kabupaten Muara Enim Provinsi Sumatera Selatan adalah PT. Perkebunan yang komoditas tanaman yang dihasilkan adalah getah karet. Getah karet yang dihasilkan merupakan hasil penyadapan oleh pekerja perkebunan yang dilakukan setiap hari. Dimana getah karet yang dihasilkan pekerja perkebunan itu akan dilaporankan ke kepala wilayah setiap bulannya oleh mandor yang sebelumnya menerima laporan hasil produksi dari pekerja perkebunan tersebut. Kendala saat ini yang dihadapi oleh pihak PTP. Nusantara VII yang ada di Desa Karang Agung adalah belum tersediannya sistem sebagai media yang dapat menginputkan laporan hasil produksi dan untuk memonitoring hasil produksi oleh kepala wilayah secara cepat, akurat dan tersimpan ke dalam database.

Proses sistem yang sedang berjalan saat ini di PT. Perkebunan Nusantara VII yang ada di Desa Karang Agung adalah sebagai berikut:

1. Pekerja perkebunan tiap afdiling melaporkan hasil produksi getah karet kepada Mandor yang bertanggung jawab di afdiling tersebut.
2. Mandor akan melaporkan total hasil produksi selama satu bulan sekali kepada kepala wilayah berupa buku laporan.
3. Kepala wilayah dapat memonitoring perkembangan hasil produksi dan mengetahui total hasil produksi berdasarkan data laporan dari Mandor tersebut dengan cara melihat buku laporan hasil produksi dari mandor.

### **3.2.1.2 Kelemahan Sistem Yang Berjalan**

Sistem yang berjalan saat ini masih memiliki kelemahan, antara lain, ketika melakukan laporan hasil produksi oleh Mandor kepada Asisten kepala wilayah yang masih dilakukan secara manual yaitu dengan menyerahkan catatan hasil produksi selama sebulan kepada mabes yang masih harus bertemu langsung yang terkadang memakan waktu karna asisten kepala wilayah tidak selalu ada di tempat. Ketika melakukan monitoring hasil produksi oleh asisten kepala wilayah masih dilakukan dengan cara melihat arsip data hasil produksi dari laporan mandor, hal ini tentu menyulitkan asisten kepala wilayah, karena satu asisten kepala wilayah memegang tanggung jawab empat afdiling/area.

### **3.2.1.3 Analisis Sistem Yang Diajukan**

Dari analisa sistem yang sedang berjalan maka diajukanlah sebuah Rancang Bangun Aplikasi Monitoring Hasil Produksi Dan Informasi Preservasi Tanaman Karet dengan Metode K-Nearest Neighbor. Berdasarkan hasil analisa tersebut, dibutuhkan sebuah aplikasi yang dapat mempermudah Mandor dalam melaporkan hasil produksi kepada asisten kepala wilayah secara cepat dan otomatis akan tersimpan di dalam database, selain itu aplikasi juga dapat mempermudah asisten kepala wilayah dalam memonitoring hasil produksi dari waktu kewaktu dan dapat mengetahui total hasil produksi tanpa takut data hasil produksi terhapus atau hilang karena data produksi sudah otomatis tersimpan didalam database.

Untuk membangun aplikasi tersebut, maka dibutuhkan perangkat lunak dan perangkat keras dengan spesifikasi sebagai berikut :

#### **1. Kebutuhan Perangkat Lunak**

perangkat lunak yang digunakan penulis dalam pembuatan *Aplikasi Monitoring* sebagai berikut:

Tabel 3.1 Kebutuhan Perangkat Lunak

<i>Software</i>	<i>Spesifikasi</i>
IDE	Android studio
Disain	StarUML
Web Server	Xampp
Sistem Operasi	Windows 10 x64 bit

## 2. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan penulis dalam pembuatan *Aplikasi Monitoring* sebagai berikut:

Tabel 3.2 Kebutuhan Perangkat Keras

<i>Software</i>	<i>Spesifikasi</i>
Processor	Intel Core i3
HDD	500 GB
RAM	6 GB

### 3.2.2 Analisis Algoritma yang digunakan

Pada analisis algoritma yang digunakan adalah untuk metode klasifikasi yaitu algoritma K-Nearest Neighbor. Metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya.

#### 3.2.2.1 Algoritma K-Nearest Neighbor

Nearest Neighbor adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dan kasus lama yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada.

Tahapan langkah algoritma K-Nearest Neighbor adalah sebagai berikut:

- a. Menentukan parameter k (jumlah tetangga paling dekat).
- b. Menghitung kuadrat jarak eucliden objek terhadap data training yang diberikan.
- c. Mengurutkan hasil no 2 secara berurutan.
- d. Mengumpulkan kategori Y (Klasifikasi nearest neighbor berdasarkan nilai k)
- e. Dengan menggunakan kategori nearest neighbor yang paling mayoritas maka dapat diprediksi kategori objek.

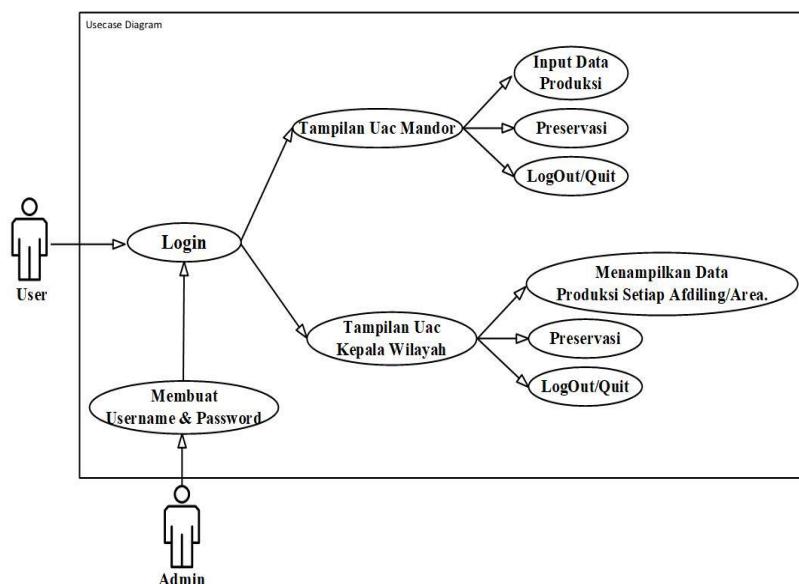
### **3.2.3 Disain Aplikasi (*Sistem And Software Design*)**

Desain aplikasi merupakan tahap perancangan sistem atau aplikasi yang meliputi penyusunan proses, data, aliran proses, dan pemenuhan kebutuhan sesuai dengan hasil analisa kebutuhan. Dokumentasi desain aplikasi yang dihasilkan dari tahapan ini adalah *use case diagram* dan *activity diagram*.

#### **3.2.3.1 Disain UML**

1. Rancangan Use Case Diagram *Aplikasi Yang Diajukan*.

Rancangan Use Case dapat dilihat pada gambar 3.2 dibawah ini.



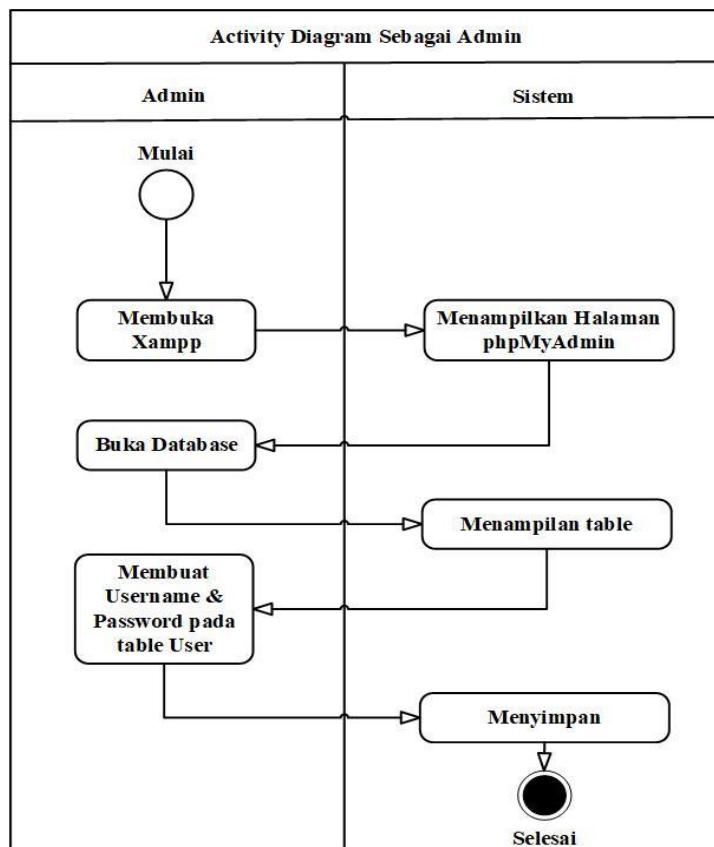
Gambar 3.2 Use Case Diagram Aplikasi Yang Diajukan

Fungsi-fungsi pada sistem berdasarkan *use case diagram* pada gambar 3.2 dapat dijelaskan secara singkat. User melakukan *Login* dengan menginputkan *username* dan *password* yang sebelumnya telah dibuat oleh admin sistem. Semua konten yang tersedia pada aplikasi hanya dapat dilakukan setelah proses *Login* berhasil. Selanjutnya barulah user dapat menggunakan aplikasi sesuai dengan jabatannya masing-masing.

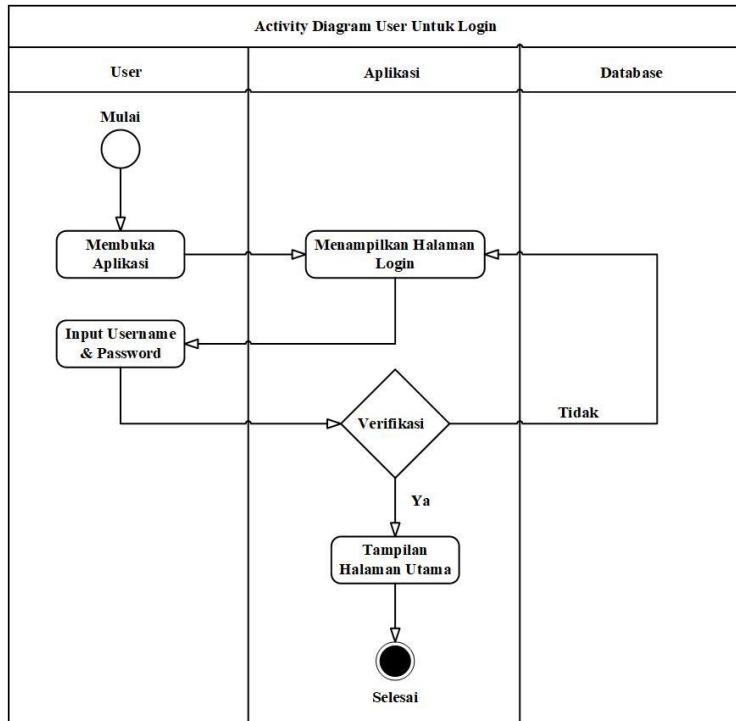
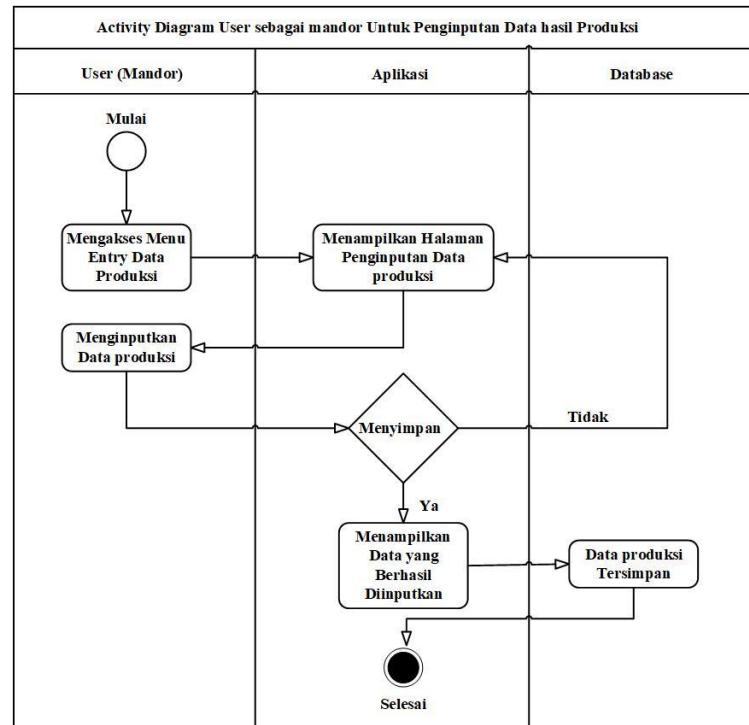
## 2. Rancangan Activity Diagram *Aplikasi*.

Activity Diagram berguna untuk memberikan visualisasi alur tindakan dalam sistem, percabangan yang mungkin terjadi, dan bagaimana alur sistem dari mulai awal mulai hingga akhir.

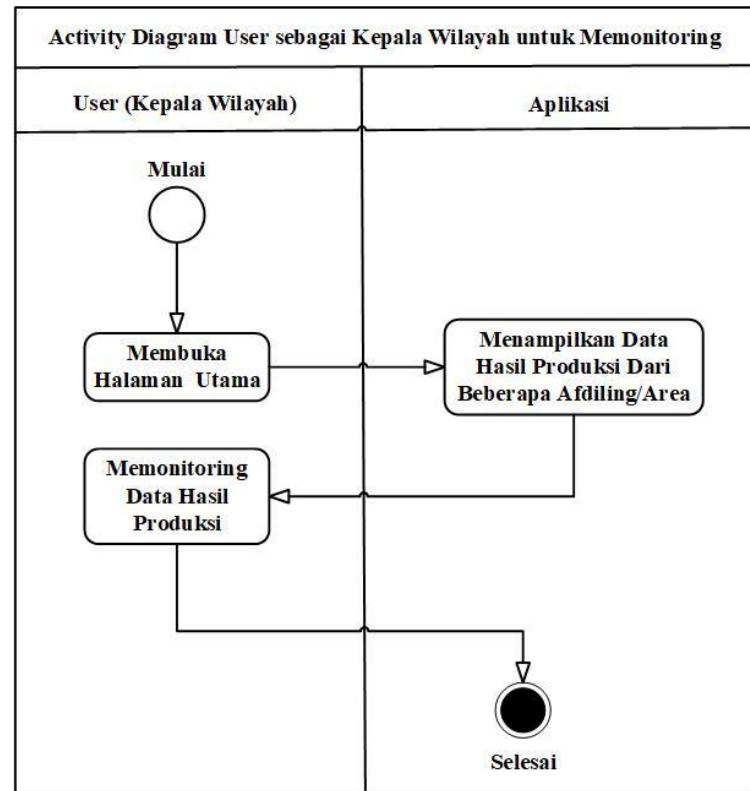
Rancangan activity diagram adalah sebagai berikut :



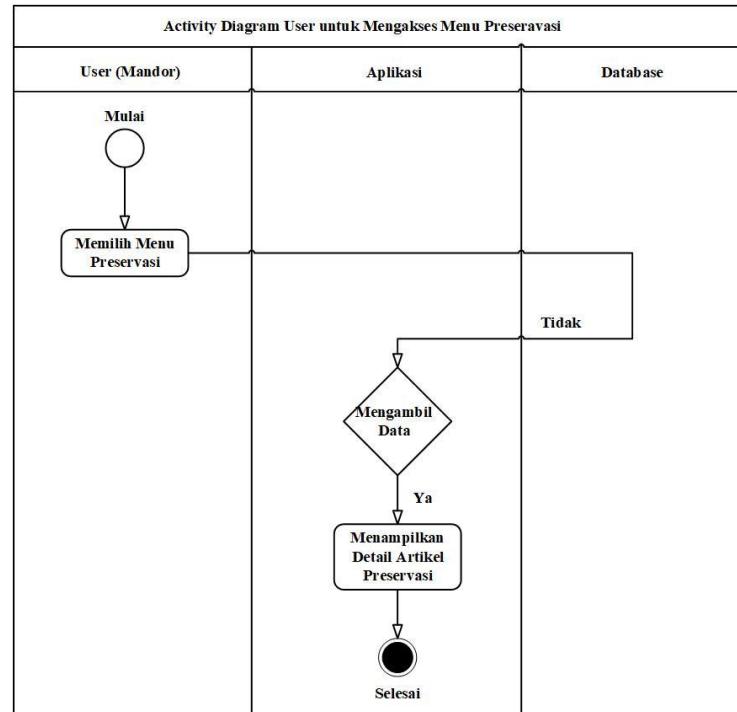
Gambar 3.3 Activity Diagram *Admin*.

Gambar 3.4 Activity Diagram *User untuk Login*

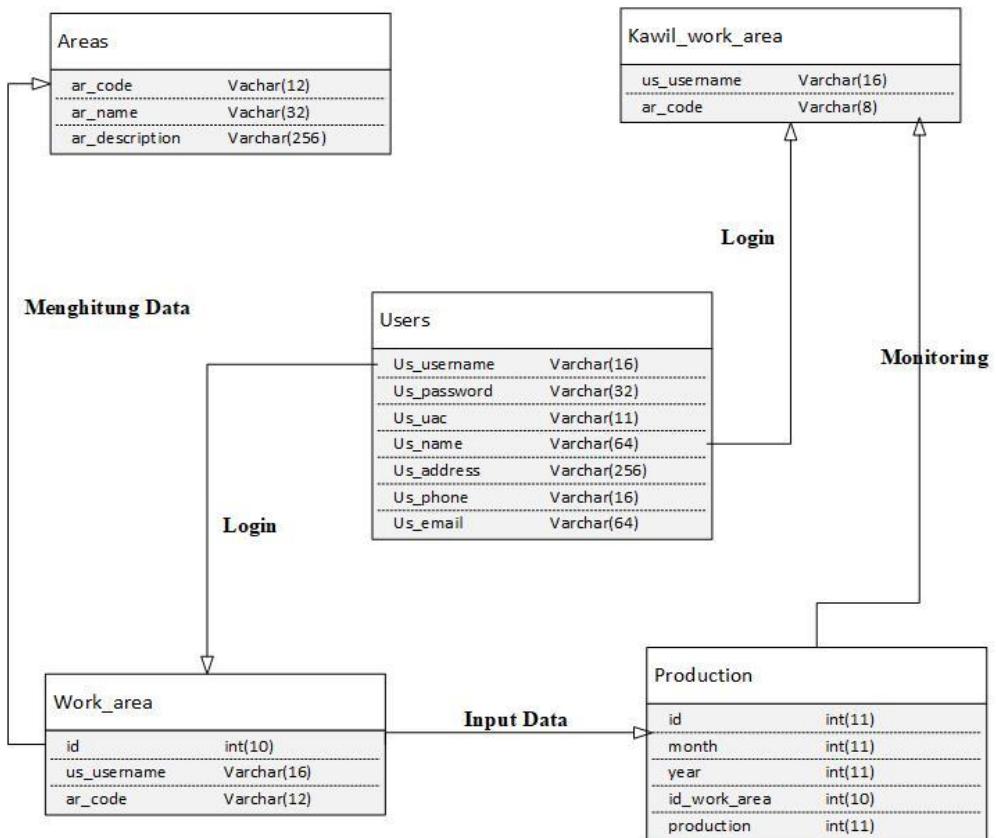
Gambar 3.5 Activity Diagram Mandor Untuk Penginputan Data.



Gambar 3.6 Activity Diagram Kepala Wilayah Untuk Monitoring



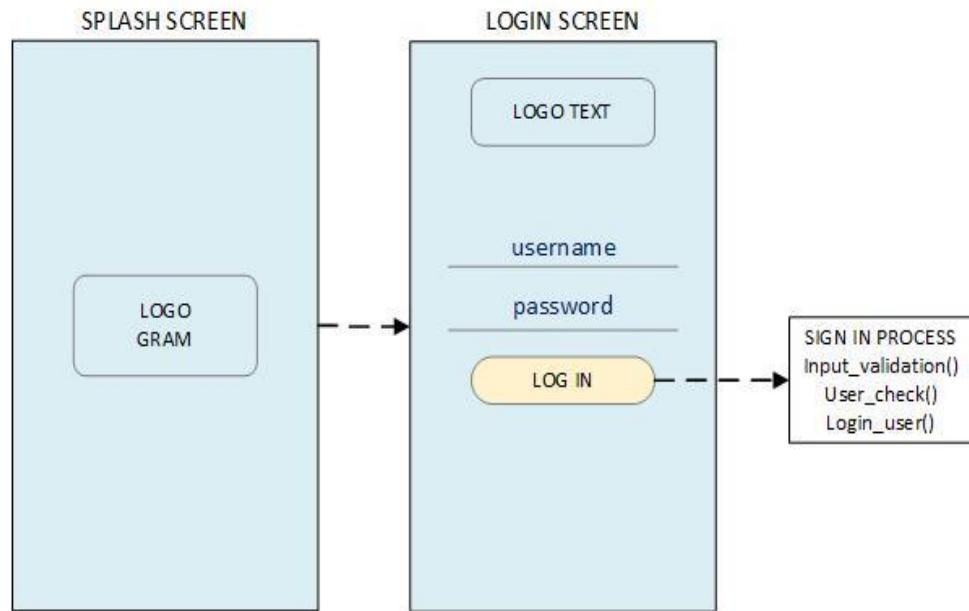
Gambar 3.7 Activity Diagram User Untuk Preservasi



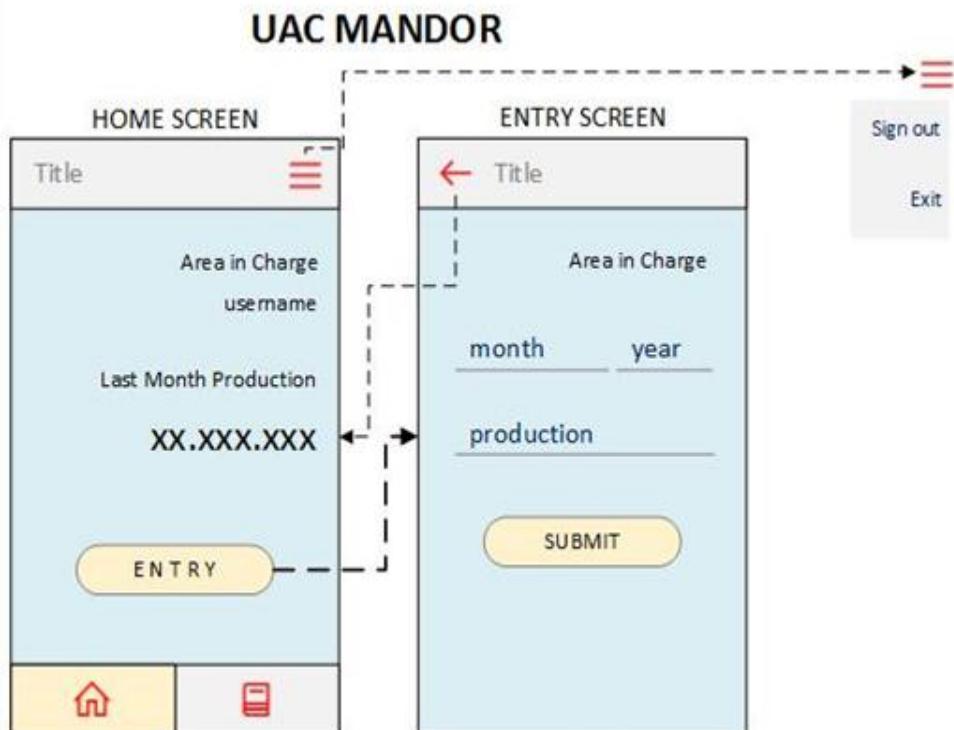
Gambar 3.8 Class Diagram

### 3.3 Perancangan Antar Muka

berikut merupakan hasil dari rancangan sistem yang dibuat :

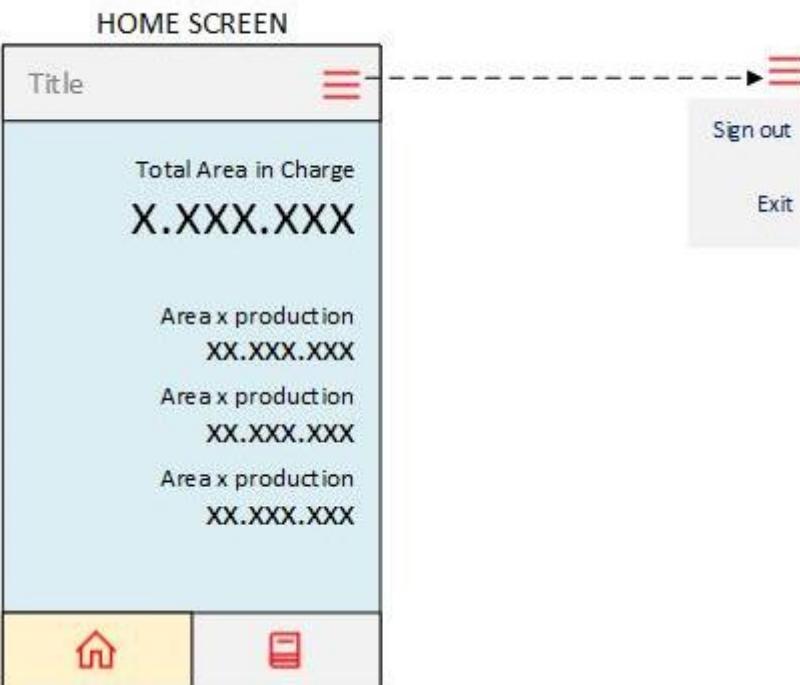


Gambar 3.9 Tampilan Splash Screen dan dan Halaman Login

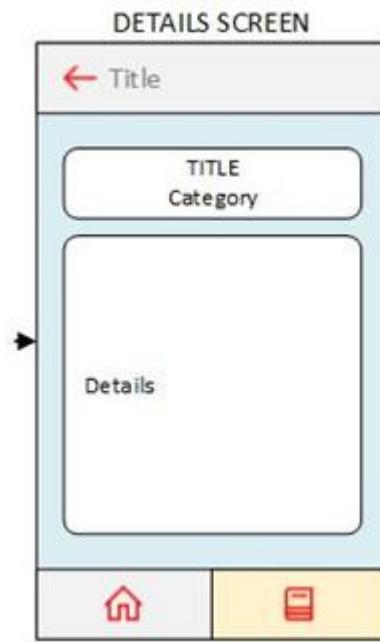


Gambar 3.10 Tampilan Uac Mandor

## UAC KEPALA WILAYAH



Gambar 3.11 Tampilan Uac Asisten Kepala Wilayah



Gambar 3.12 Tampilan Preservation

## **BAB IV**

### **HASIL PENELITIAN DAN PEMBAHASAN**

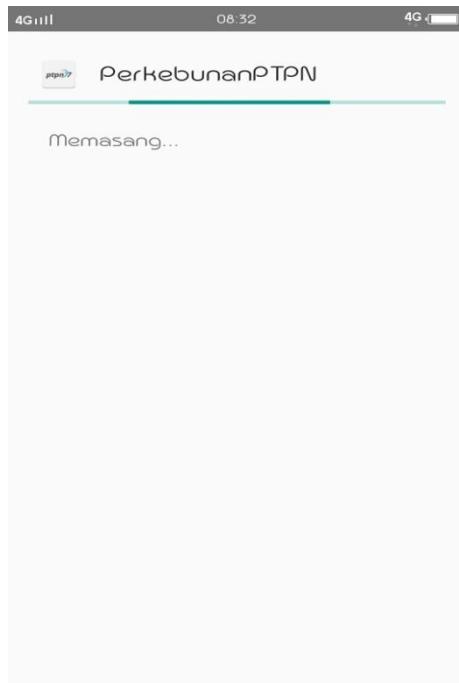
#### **4.1 Hasil Penelitian**

Setelah melalui analisa dan perancangan, maka tahapan berikutnya adalah pembuatan kode program. Tatap muka dari aplikasi ini adalah sebagai berikut.

#### **4.2 Pembahasan**

##### **4.2.1 Proses Instalasi Perangkat Lunak**

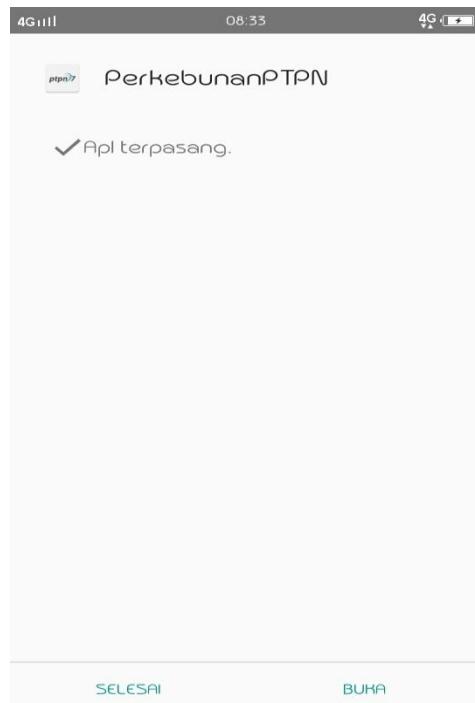
Proses Instalasi Aplikasi Monitoring Hasil Produksi dan Preservasi (perawatan) Tanaman Karet dengan metode K-Nearest Neighbor, yang dapat dilihat pada gambar 4.1



Gambar 4.1 (Tampilan Proses Instalasi)

##### **4.2.2 Aplikasi Selesai di Instal**

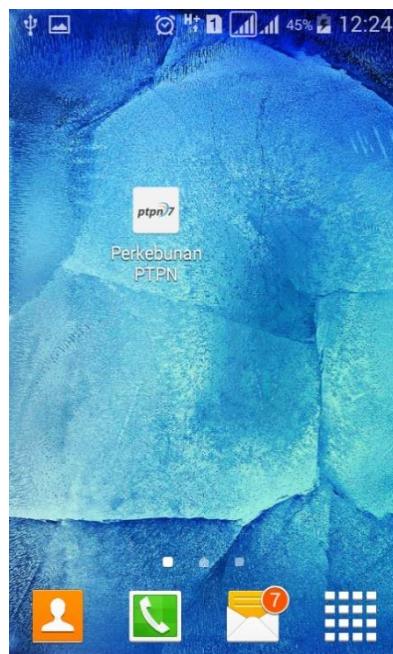
Aplikasi Monitoring Perkembangan Hasil Produksi dan Preservasi Tanaman Karet menggunakan metode algoritma K-Nearest Neighbor Berhasil di Instal Di smarphone, yang dapat dilihat pada gambar 4.2



Gambar 4.2 (Tampilan Aplikasi Selesai di Instal)

#### 4.2.3 Halaman Penggunaan Perangkat Lunak

Penggunaan adalah saat aplikasi sudah terinstal di sebuah sistem operasi Android dengan tampilan *icon* pada smartphone pengguna, yang dapat dilihat pada gambar 4.3 sebagai berikut:



Gambar 4.3 (Tampilan Icon Aplikasi Setelah Terinstal)

#### 4.2.4 Halaman Splash Screen

Pada saat aplikasi telah terinstal dan terpasang pada smartphone pengguna maka akan muncul halaman splash screen sebagai launcher tampilan pertama dari Aplikasi Monitoring Hasil Produksi dan Informasi Preservasi (Perawatan) Tanaman Karet Menggunakan Metode K-Nearest Neighbor.

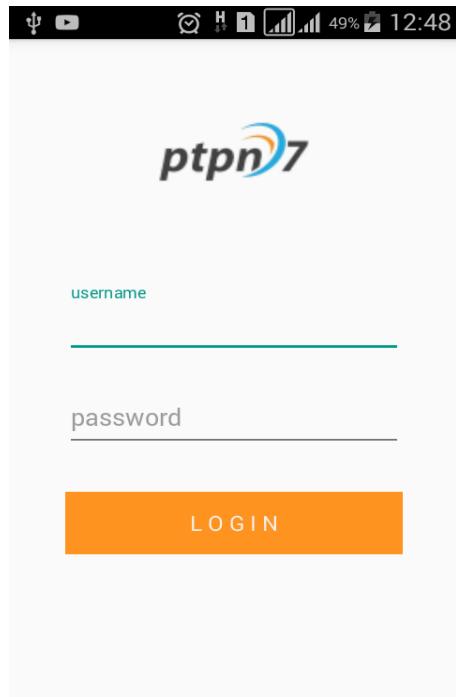
Hasil dari tampilan halaman splash screen terlihat di gambar 4.4.



Gambar 4.4 (Tampilan halaman splash screen)

#### 4.2.5 Halaman Login

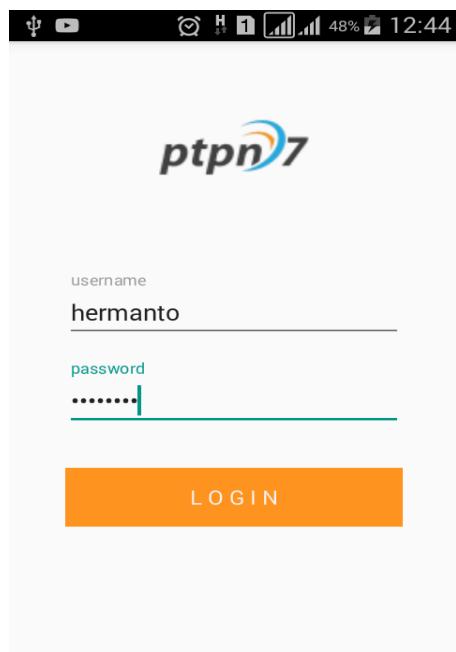
Berikut ini merupakan tampilan halaman login user untuk masuk kedalam halaman utama user. Sesuai dengan *username* dan *password* yang tentukan oleh admin. Tampilan halaman login user dapat dilihat pada gambar 4.5.



Gambar 4.5 (Tampilan halaman Login User)

#### 4.2.6 Halaman Input Username dan Password

Pengguna bisa menginputkan username dan password yang diberikan oleh admin.



Gambar 4.6 (Tampilan Input Username dan Password)

#### 4.2.7 Halaman utama login sebagai mandor

Setiap user yang login sebagai mandor akan otomatis diarahkan ke halaman utama untuk mandor, dimana pada halaman ini menampilkan data produksi bulan-bulan sebelumnya yang pernah diinputkan. Kemudian mandor bisa memilih entry data produksi untuk menginputkan data produksi baru. Halaman utama mandor dapat dilihat seperti gambar 4.7



Gambar 4.7 (Tampilan Halaman utama setelah login sebagai mandor)

#### 4.2.8 Halaman Input Data Produksi

Halaman input data produksi akan ditampilkan ketika mandor memilih entry data produksi yang tersedia pada halaman awal. Mandor dapat menginputkan data produksi getah karet. Halaman input data produksi dapat dilihat gambar 4.8



Gambar 4.8 (Tampilan Halaman Input Hasil Produksi)

#### 4.2.9 Proses Penginputan Data Produksi

Ketika sudah berada dihalaman input data produksi mandor bisa langsung menginputkan data produksi, yang dapat lihat gambar 4.9.



Gambar 4.9 (proses Penginputan data produksi)

#### 4.2.10 Proses penyimpanan Data Produksi yang diinputkan

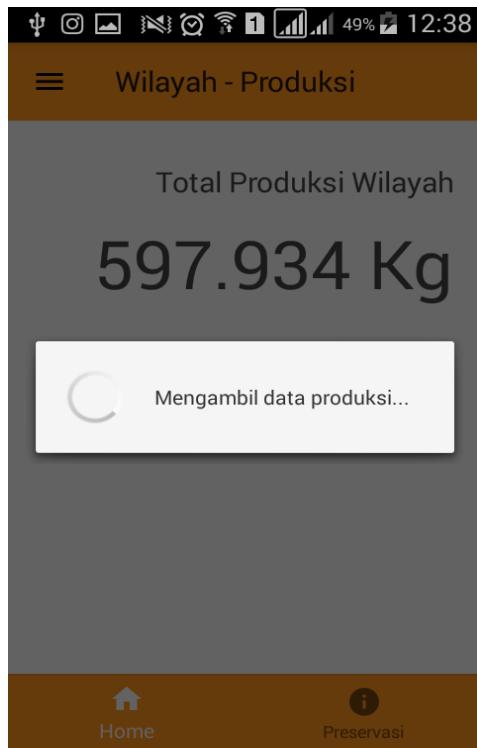
Ketika data produksi telah diinputkan, mandor bisa menyimpan data tersebut untuk kemudian dimonitoring oleh kepala wilayah.



Gambar 4.10 (proses penyimpanan data produksi)

#### 4.2.11 Halaman monitoring kepala wilayah

Halaman monitoring ini akan otomatis ditampilkan ketika user login sebagai kepala wilayah, dimana pada halaman ini menampilkan data produksi setiap area berdasarkan inputan data produksi dari masing-masing mandor yang bertanggung jawab di tiap area. Dapat dilihat pada gambar 4.11 dan gambar 4.12.



Gambar 4.11 (Tampilan ketika mengambil Data Produksi dari database)

Kemudian kepala wilayah dapat melihat hasil produksi di beberapa area, dan tampilan grafik untuk total hasil produksi setiap bulannya.



Gambar 4.12 (Tampilan data produksi setiap area)

#### 4.2.12 Tampilan Halaman Klasifikasi Hasil Produksi

Pada halaman Klasifikasi Hasil Produksi ini menampilkan hasil produksi yang telah terkласifikasi kedalam katagori rendah, sedang dan tinggi. Tampilan halaman preservasi dapat dilihat pada gambar 4.13.

The screenshot shows a mobile application interface titled "Klasifikasi". At the top, there is a header bar with the time "10:54 AM", signal strength, battery level at 94%, and a back arrow icon. Below the header, the title "Klasifikasi" is centered. The main content is a table with six rows, each representing a different month and year combination. The columns are labeled "Bulan" (Month), "Tahun" (Year), and "Kategori" (Category). The data is as follows:

Bulan	Tahun	Kategori
3	2018	<b>Tinggi</b>
2	2018	<b>Rendah</b>
2	2018	<b>Rendah</b>
2	2018	<b>Tinggi</b>
2	2018	<b>Tinggi</b>
1	2018	<b>Tinggi</b>
1	2018	<b>Rendah</b>

Gambar 4.13 (*Tampilan Halaman Klasifikasi*)

#### 4.2.13 Tampilan Halaman Preservasi

Pada halaman preservasi ini menampilkan artikel tentang preservasi tanaman karet, dan halaman preservasi ini dapat di akses asisten kepala wilayah. Tampilan halaman preservasi dapat dilihat pada gambar 4.14.



Gambar 4.13 (*Tampilan Halaman Preservasi*)

#### 4.3 Pembahasan Algoritma K-Nearest Neighbor

Algoritma K-Nearest Neighbor (K-NN) adalah sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. Termasuk dalam supervised learning, dimana hasil query instance yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam K-NN. Berikut adalah tabel hasil produksi di PTP. Nusantara VII unit Desa Karang Agung dari bulan Januari sampai Desember 2017.

Tabel 4.1 Data Produksi

Area	Hasil Produksi (Kg)											
	Januari	Februari	Maret	April	Mei	Juni	Juli	Agustus	Sept	Okt	Nov	Des
A F D I	119.912	120.216	149.291	150.221	151.967	161.755	154.211	136.132	113.194	120.125	132.321	150.112
A F D II	100.512	101.912	135.998	137.210	141.003	142.725	138.159	124.250	94.239	101.211	122.150	137.210
A F D II I	108.214	110.112	139.514	140.291	139.980	142.252	139.255	126.109	96.112	109.104	124.291	140.191
A F D IV	124.712	124.121	125.100	126.161	130.209	151.202	143.108	138.185	114.397	125.791	135.129	125.721
A F D V	114.292	113.891	172.001	175.161	175.999	172.407	163.204	133.176	101.008	115.921	130.192	172.216
A F D VI	110.912	111.132	147.412	150.126	155.461	161.222	159.301	124.341	98.143	110.213	124.912	146.421
A F D VI I	121.319	120.192	163.555	163.692	109.300	170.125	165.112	134.102	104.107	121.111	133.291	164.912
A F D VI II	100.912	101.219	132.191	135.729	136.012	135.001	130.921	114.108	86.231	100.012	115.119	132.191

Data hasil produksi dari bulan januari-desember 2017 sebanyak 96 data. Dimana pada data produksi tahun 2017 tersebut hasil produksi terendah terdapat pada area AFD II bulan september dengan hasil produksi adalah 86.231 Kg dan hasil produksi tertinggi terdapat pada area AFD V bulan mei dengan hasil produksi adalah 175.999 Kg.

Klasifikasi hasil produksi dibagi kedalam 3 katagori, yaitu :

- Katagori hasil produksi rendah : Dengan hasil produksi  $\leq 120.215$  Kg
- Katagori hasil produksi sedang : Dengan hasil produksi  $\geq 120.216$  Kg sampai dengan 140.290 Kg
- Katagori hasil produksi tinggi : Dengan hasil produksi  $\geq 140.300$  Kg

Menghitung jarak data baru dengan data training setiap Area menggunakan rumus Euclidean distance

$$d_{ik} = \sqrt{\sum_i^m (C_{ij} - C_{kj})^2}$$

Keterangan;

$d_{ik}$  = jarak

$C_{ij}$  = data training/sample

$C_{kj}$  = Data baru

Perhitungan manual menggunakan algoritma K-nearest Neighbor dengan Data baru atau data testing = 100.000 Kg dengan K=4.

Tabel 4.2 Tabel Perhitungan Data Produksi

No	Data Produksi	Hitung jarak Euclidean	Jarak Terkecil	K=4	Katagori
1	86.231	$(86.231-100.000)^2 = 13.769$	20	TIDAK	Rendah
2	94.239	$(94.239-100.000)^2 = 5.761$	11	TIDAK	Rendah
3	96.112	$(96.112-100.000)^2 = 3.888$	9	TIDAK	Rendah
4	98.143	$(98.143-100.000)^2 = 1.857$	8	TIDAK	Rendah
5	100.012	$(100.012-100.000)^2 = 12$	1	YA	Rendah
6	100.512	$(100.512-100.000)^2 = 512$	2	YA	Rendah
7	100.912	$(100.912-100.000)^2 = 912$	3	YA	Rendah
8	101.008	$(101.008-100.000)^2 = 1.008$	4	YA	Rendah
9	101.211	$(101.211-100.000)^2 = 1.211$	5	TIDAK	Rendah
10	101.219	$(101.219-100.000)^2 = 1.219$	6	TIDAK	Rendah
11	101.219	$(101.219-100.000)^2 = 1.219$	7	TIDAK	Rendah
12	104.107	$(104.107-100.000)^2 = 4.107$	10	TIDAK	Rendah
13	108.214	$(108.214-100.000)^2 = 8.214$	12	TIDAK	Rendah
14	109.104	$(109.104-100.000)^2 = 9.104$	13	TIDAK	Rendah

15	109.300	$(109.300-100.000)^2$ = 9.300	14	TIDAK	Rendah
16	110.112	$(110.112-100.000)^2$ = 10.112	15	TIDAK	Rendah
17	110.213	$(110.213-100.000)^2$ = 10.213	16	TIDAK	Rendah
18	110.912	$(110.912-100.000)^2$ = 10.912	17	TIDAK	Rendah
19	111.132	$(111.132-100.000)^2$ = 11.132	18	TIDAK	Rendah
20	113.194	$(113.194-100.000)^2$ = 13.194	19	TIDAK	Rendah
21	113.891	$(113.891-100.000)^2$ = 13.891	21	TIDAK	Rendah
22	114.108	$(114.108-100.000)^2$ = 14.108	22	TIDAK	Rendah
23	114.292	$(114.292-100.000)^2$ = 14.292	23	TIDAK	Rendah
24	114.397	$(114.297-100.000)^2$ = 14.297	24	TIDAK	Rendah
25	115.119	$(115.119-100.000)^2$ = 15.119	25	TIDAK	Rendah
26	115.921	$(115.921-100.000)^2$ = 15.921	26	TIDAK	Rendah
27	119.912	$(119.912-100.000)^2$ = 19.912	27	TIDAK	Rendah
28	120.125	$(119.912-100.000)^2$ = 19.912	28	TIDAK	Rendah
29	120.192	$(120.192-100.000)^2$ = 20.192	29	TIDAK	Rendah
30	120.216	$(120.216-100.000)^2$ = 20.216	30	TIDAK	Rendah
31	121.111	$(121.111-100.000)^2$ = 21.111	31	TIDAK	Sedang
32	121.319	$(121.319-100.000)^2$ = 21.319	32	TIDAK	Sedang
33	122.150	$(122.150-100.000)^2$	33	TIDAK	Sedang

		= 22.150			
34	124.121	$(124.121-100.000)^2$ = 24.121	34	TIDAK	Sedang
35	124.250	$(124.250-100.000)^2$ = 24.250	35	TIDAK	Sedang
36	124.291	$(124.291-100.000)^2$ = 24.291	36	TIDAK	Sedang
37	124.341	$(124.341-100.000)^2$ = 24.341	37	TIDAK	Sedang
38	124.712	$(124.712-100.000)^2$ = 24.712	38	TIDAK	Sedang
39	124.912	$(124.912-100.000)^2$ = 24.912	39	TIDAK	Sedang
41	125.100	$(125.100-100.000)^2$ = 25.100	40	TIDAK	Sedang
41	125.721	$(125.721-100.000)^2$ = 25.721	41	TIDAK	Sedang
42	125.791	$(125.791-100.000)^2$ = 25.791	42	TIDAK	Sedang
43	126.109	$(126.109-100.000)^2$ = 26.109	43	TIDAK	Sedang
44	126.161	$(126.161-100.000)^2$ = 26.161	44	TIDAK	Sedang
45	130.192	$(130.192-100.000)^2$ = 30.192	45	TIDAK	Sedang
46	130.209	$(130.209-100.000)^2$ = 30.209	46	TIDAK	Sedang
47	130.921	$(130.921-100.000)^2$ = 30.921	47	TIDAK	Sedang
48	132.191	$(132.191-100.000)^2$ = 32.191	48	TIDAK	Sedang
49	132.191	$(132.191-100.000)^2$ = 32.191	49	TIDAK	Sedang
50	132.321	$(132.321-100.000)^2$ = 32.321	50	TIDAK	Sedang
51	133.176	$(133.176-100.000)^2$ = 33.176	51	TIDAK	Sedang

<b>52</b>	133.291	$(133.291-100.000)^2$ = 33.291	52	TIDAK	Sedang
<b>53</b>	134.102	$(134.102-100.000)^2$ = 34.102	53	TIDAK	Sedang
<b>54</b>	135.001	$(135.001-100.000)^2$ = 35.001	54	TIDAK	Sedang
<b>55</b>	135.129	$(135.129-100.000)^2$ = 35.129	55	TIDAK	Sedang
<b>56</b>	135.729	$(135.729-100.000)^2$ = 35.729	56	TIDAK	Sedang
<b>57</b>	135.998	$(135.998-100.000)^2$ = 35.998	57	TIDAK	Sedang
<b>58</b>	136.012	$(136.012-100.000)^2$ = 36.012	58	TIDAK	Sedang
<b>59</b>	136.132	$(136.132-100.000)^2$ = 36.132	59	TIDAK	Sedang
<b>60</b>	137.210	$(137.210-100.000)^2$ = 37.210	60	TIDAK	Sedang
<b>61</b>	137.210	$(137.210-100.000)^2$ = 37.210	61	TIDAK	Sedang
<b>62</b>	138.159	$(138.159-100.000)^2$ = 38.159	62	TIDAK	Sedang
<b>63</b>	138.185	$(138.185-100.000)^2$ = 38.185	63	TIDAK	Sedang
<b>64</b>	139.255	$(139.255-100.000)^2$ = 39.255	64	TIDAK	Sedang
<b>65</b>	139.514	$(139.514-100.000)^2$ = 39.514	65	TIDAK	Sedang
<b>66</b>	139.980	$(139.980-100.000)^2$ = 39.980	66	TIDAK	Sedang
<b>67</b>	140.191	$(140.191-100.000)^2$ = 40.191	67	TIDAK	Tinggi
<b>68</b>	140.291	$(140.291-100.000)^2$ = 40.291	68	TIDAK	Tinggi
<b>69</b>	141.003	$(141.003-100.000)^2$ = 41.003	69	TIDAK	Tinggi
<b>70</b>	142.252	$(142.252-100.000)^2$	70	TIDAK	Tinggi

		= 42.252			
71	142.725	$(142.725-100.000)^2$ = 42.725	71	TIDAK	Tinggi
72	143.108	$(143.108-100.000)^2$ = 43.108	72	TIDAK	Tinggi
73	146.421	$(146.421-100.000)^2$ = 46.421	73	TIDAK	Tinggi
74	147.412	$(147.412-100.000)^2$ = 47.412	74	TIDAK	Tinggi
75	149.291	$(149.291-100.000)^2$ = 49.291	75	TIDAK	Tinggi
76	150.112	$(150.112-100.000)^2$ = 50.112	76	TIDAK	Tinggi
77	150.126	$(150.126-100.000)^2$ = 50.126	77	TIDAK	Tinggi
78	150.221	$(150.221-100.000)^2$ = 50.221	78	TIDAK	Tinggi
79	151.202	$(151.202-100.000)^2$ = 51.202	79	TIDAK	Tinggi
80	151.967	$(151.967-100.000)^2$ = 51.967	80	TIDAK	Tinggi
81	154.211	$(154.211-100.000)^2$ = 54.211	81	TIDAK	Tinggi
82	155.461	$(155.461-100.000)^2$ = 55.461	82	TIDAK	Tinggi
83	159.301	$(159.301-100.000)^2$ = 59.301	83	TIDAK	Tinggi
84	161.222	$(161.222-100.000)^2$ = 61.222	84	TIDAK	Tinggi
85	161.755	$(161.755-100.000)^2$ = 61.755	85	TIDAK	Tinggi
86	163.204	$(163.204-100.000)^2$ = 63.204	86	TIDAK	Tinggi
87	163.555	$(163.555-100.000)^2$ = 63.555	87	TIDAK	Tinggi
88	163.692	$(163.692-100.000)^2$ = 63.692	88	TIDAK	Tinggi

<b>89</b>	164.912	$(164.912-100.000)^2$ = 64.912	89	TIDAK	Tinggi
<b>90</b>	165.112	$(165.112-100.000)^2$ = 65.112	90	TIDAK	Tinggi
<b>91</b>	170.125	$(170.125-100.000)^2$ = 70.125	91	TIDAK	Tinggi
<b>92</b>	172.001	$(172.001-100.000)^2$ = 72.001	92	TIDAK	Tinggi
<b>93</b>	172.216	$(172.216-100.000)^2$ = 72.216	93	TIDAK	Tinggi
<b>94</b>	172.407	$(172.407-100.000)^2$ = 72.407	94	TIDAK	Tinggi
<b>95</b>	175.161	$(175.161-100.000)^2$ = 75.161	96	TIDAK	Tinggi
<b>96</b>	172.999	$(172.999-100.000)^2$ = 72.999	95	TIDAK	Tinggi

Dengan mengurutkan jarak terkecil, diambil K=4, dan ke 4 jarak paling dekat tersebut masuk dalam katagori rendah, maka dapat disimpulkan angka produksi 100.000 Kg termasuk hasil produksi yang rendah.

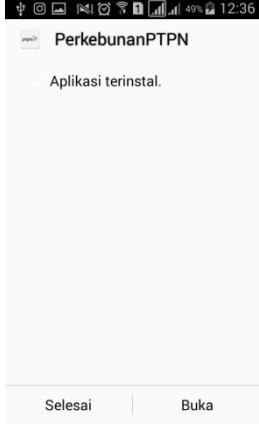
#### 4.4 Pengujian Sistem Perangkat Lunak

Proses pengujian sistem perangkat lunak dilakukan dengan menggunakan metode Metode Black Box Testing, yang merupakan salah satu cara pengujian perangkat lunak yang mengutamakan pengujian terhadap kebutuhan fungsi dari suatu program dengan menemukan kesalahan fungsi pada perangkat lunak tersebut. Dalam tahap pengujian aplikasi ini dilakukan pada berberapa perangkat mobile dengan spesifikasi yang berbeda-beda.

### 4.3.1 Pengujian Instalasi

Pengujian instalasi dilakukan apakah aplikasi yang telah dibuild dapat berjalan diatas sistem operasi Android.

Tabel 4.3 pengujian instalasi

No.	Brand	Spesifikasi	Hasil Pengujian	Keterangan
1.	Samsung J1 Ace	RAM 0.75GB Android 4.4.4 KitKat Speed HSPA 21.1/5.76 Mbps, LTE Cat4 150/50 Mbps. Ukuran Layar 4.3 inches	 PerkebunanPTPN Aplikasi terinstal.  Selesai   Buka	BERHASIL
2	Xiomi Redmi 4x		 PerkebunanPTPN Installed successfully  Details  >  Selesai   Buka	BERHASIL

### 4.3.2 Pengujian Penggunaan

Pada tahapan pengujian penggunaan, akan dilakukan pengujian dibeberapa smartphone dengan cara sebagai berikut:

Hasil dari pengujian dapat dilihat pada tabel 4.4

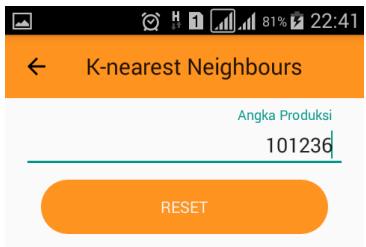
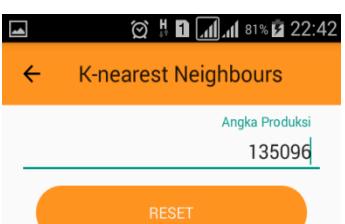
Tabel 4.4 Pengujian Pada Penginputan data produksi

No.	Brand	Spesifikasi	Hasil Pengujian	Keterangan
1.	Samsung J1 Ace	RAM 0.75GB Android 4.4.4 KitKat Speed HSPA 21.1/5.76 Mbps, LTE Ca t4 150/50 Mbps. Ukuran Layar 4.3 inches		BERHASIL
2	Xiomi Redmi 4x			BERHASIL

#### 4.3.4 Pengujian Algoritma K-Nearest Neighbor

Pengujian Algoritma K-Nearest Neighbor *perangkat lunak* dapat dilihat pada tabel 4.5

Tabel 4.5 Algoritma *perangkat lunak*

No	Brand	Spesifikasi	Hasil Pengujian	Keterangan
1	Samsung J1 Ace	RAM 0.75GB  Android 4.4.4  KitKat  Speed HSPA 21.1/5.76 Mbps, LTE Ca t4 150/50 Mbps.	 <small>Nilai k = 4 Nilai c = 101236 Masuk kedalam kategori = Rendah</small>	data baru masuk dalam katagori rendah berdasarkan perhitungan algoritma dengan mencari jarak terdekat terhadap data lama yang telah terklasifikasi kedalam katagori rendah,sedang dan tinggi.
2	Samsung J1 Ace	RAM 0.75GB	 <small>Nilai k = 4 Nilai c = 135096 Masuk kedalam kategori = Sedang</small>	data baru masuk dalam katagori sedang berdasarkan perhitungan algoritma dengan mencari jarak terdekat terhadap data lama yang telah terklasifikasi kedalam katagori

				rendah,sedang dan tinggi.
2	Samsung J1 Ace	RAM 0.75GB		data baru masuk dalam katagori Tinggi berdasarkan perhitungan algoritma dengan mencari jarak terdekat terhadap data lama yang telah terklasifikasi kedalam katagori rendah,sedang dan tinggi.

#### 4.5 Kelebihan dan Kekurangan Perangkat Lunak

Kelebihan dari Perangkat Lunak ini adalah sebagai berikut:

1. Aplikasi ini dapat mempermudah kepala wilayah dalam memonitoring hasil produksi di beberapa area perkebunan PTP Nusantara Vii.
2. Aplikasi ini mempermudah mandor dalam melaporkan hasil produksi hanya dengan menginputkan data produksi pada aplikasi.
3. Aplikasi ini dapat menjadi alternatif kepala wilayah dalam mendapatkan Informasi seputar cara preservasi tanaman karet.

Kelemahan dari Perangkat Lunak ini adalah sebagai berikut:

1. Aplikasi monitoring perkembangan hasil produksi dan cara preservasi tanaman karet ini hanya dapat dijalankan pada smartphone yang berbasis sistem operasi Android.
2. Bagian Monitoring oleh kepala wilayah belum terdapat opsi untuk memberikan masukan kepada area yang masuk kedalam katagori rendah.

## **BAB V**

### **SIMPULAN DAN SARAN**

#### **5.1 Simpulan**

Berdasarkan latar belakang serta pembahasan pada bab-bab sebelumnya maka dapat disimpulkan bahwa :

1. Sistem ini diusulkan untuk mempermudah mandor dan asisten kepala wilayah.
2. Tidak lagi dilakukan secara konvensional/manual prosedur.
3. Media penyimpanan database sebagai arsip data produksi.

#### **5.2 Saran**

Saran yang diberikan sesuai adanya penelitian yang telah di lakukan adalah :

1. Aplikasi yang telah dirancang saat ini diharapkan dapat terus dievaluasi terutama masalah monitoring.
2. Bagi peneliti selanjutnya diharapkan dapat menambahkan opsi untuk menampilkan grafik dalam memonitoring hasil produksi.
3. Desain aplikasi yang masih sederhana agar dapat terus dikembangkan.

## **DAFTAR PUSTAKA**

- Sabarman Damanik. 2012. Pengembangan Karet (*Havea Brasiliensis*) Berkelanjutan di Indonesia. Bogor : Perspektif Vol.11 No.1 Hlm 91-102 ISSN:1412-8004.
- Nazruddin, S. H. (2015). Pemrograman Aplikasi Mobile Smartphone dan Tablet Pc Berbasis Android. Bandung: Informatika
- Mudjahidin & Nyoman Dita Pahang Putra. 2010. Rancang Bangun Sistem Informasi Monitoring Perkembangan Proyek Berbasis Web Studi Kasus di Dinas Bina Marga dan Pemantusan. Surabaya : Jurnal Teknik Industri, Vol. 11, No. 1, Febuari 2010: 75-83.
- Yuni Puspita Sari. 2016. Membangun Kamus Obstetri berbasis android (studi kasus: kampus kebidanan adila bandar lampung). Bandar Lampung : Jurnal Informatika, Vol. 16, No. 1
- Moh. Rochman Wahid Maulana. 2017. Pengembangan Aplikasi Untuk Studi Bahasa Carakan Madura : PROSIDING ISSN: 2549 – 869X.
- Angga Setiyadi & Tati Hariyanti. 2016. Penerapan Sqlite Pada Aplikasi Pengaturan Waktu Ujian dan Presentasi. Bandung : Jurnal Teknik Informatika, Vol.13 No.2.
- Nuraini, R. (2015). Desain Algoritma Perkalian Matriks Menggunakan Metode Flowchart. *Jurnal Teknik Komputer Amik BSI*, 144-151.
- Toto Andri Puspito. 2017. Menentukan Pilihan Sekolah Didalam Penerimaan Peserta Didik Baru Dengan Menggunakan Metode Nave bayes dan K-Nearest Neighbor (Studi Kasus: PPDB Online Jenjang SMP Kota Metro). Bandar Lampung : Jurnal Informatika, Vol. 17, No. 1.
- Roger S.Pressman, Ph.D. (2012). Rekayasa Perangkat Lunak (pendekatan praktisi edisi 7). Yogyakarta: penerbit C.V ANDI.

Ginanjar Wiro Sasmito. 2017. Penerapan Metode *Waterfall* Pada Desain Sistem  
Informasi Geografis Industri Kabupaten Tegal. Kota Tegal : PROSIDING  
ISSN: 2477-5126 | E-ISSN: 2598-9356.



**SURAT KEPUTUSAN  
REKTOR IIB DARMAJAYA**  
**NOMOR : SK.180/DMJ/DFIK/BAAK/III-18**

Tentang  
Dosen Pembimbing Skripsi  
Program Studi S1 Teknik Informatika

REKTOR IIB DARMAJAYA

Memperhatikan : 1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam Skripsi.

Menimbang : 2. Laporan dan usulan Ketua Program Studi S1 Teknik Informatika.

Menimbang : 1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan Dosen Pembimbing Skripsi.  
2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.

Mengingat : 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.  
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi  
3. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya  
4. STATUTA IBI Darmajaya  
5. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi  
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.

**Menetapkan**

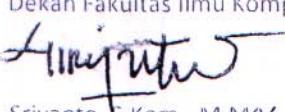
Pertama : Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Teknik Informatika.

Kedua : Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.

Ketiga : Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.

Keempat : Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung  
Pada tanggal : 12 Maret 2018  
a.n. Rektor IIB Darmajaya,  
Dekan Fakultas Ilmu Komputer

  
Sriyanto, S.Kom., M.M.  
NIK. 00210800

1. Kabiro, SDM
2. Ketua Jurusan S1 Teknik Informatika
3. Yang bersangkutan
4. Arsip



Lampiran : Surat Keputusan Rektor !IB Darmajaya  
 Nomor : SK.180/DMJ/DFIK/BAAK/II-18  
 Tanggal : 12 Maret 2018  
 Perihal : Pembimbing Penulisan Skripsi  
 Program Studi Strata Satu (S1) Teknik Informatika

#### JUDUL SKRIPSI DAN DOSEN PEMBIMBING

#### PROGRAM STUDI STRATA SATU (S1) TEKNIK INFORMATIKA

No	NAMA	NPM	JUDUL	PEMBIMBING
1	Eko Prasetyo	1411010083	Perangkat Lunak Pengukuran Audit Teknologi Informasi Pada Tingkat Kapabilitas Pengelolaan Layanan Otomasi Perusahaan Menggunakan Algoritma Knuth-Morris-Pratt pada Toko Hijab Dailyku Berbasis Android	Amnah, S.Kom, M.T.I
2	Dewi Anggraini	1411010020	Applikasi Bahasa Pemrograman Java Disertai Compiler Berbasis Android (Studi Kasus Pemrograman Menengah IIB Darmajaya)	Apri Triansah, S.Kom, M.T.I
3	Galuh Wiyenda Muhammad	1411010018	Pemanfaatan Teknologi Android Dalam Aplikasi Pembelajaran Teori Kinetik Gas Pada Fisika	
4	Nhoji Tuseno	1411010017	Implementasi Virtual Reality Sebagai Media Simulasi Tanggap Bencana Gempa Bumi Berbasis Video 360	
5	Nesa Nugraha	1411010046	Penerapan Teorema Bayes Untuk Diagnosis Pyrexia Pada Bayi Usia 1 - 6 Bulan Berbasis Android	Fitria, S.T., M.Kom
6	Febri Kusuma Putra	1311010065	Algoritma ID3 untuk Menentukan Kelayakan Pengajuan Kredit Mobil di MNC Finance Bandar Lampung	
7	Selvi Novitasari	1411010015	Klasifikasi Dokumen Penelitian dengan Menggunakan Metode K-Means Clustering (Studi Kasus: LP4M IIB Darmajaya)	
8	1411010113	Batuna Wisnu Wardana	Pembangunan Rekayasa Perangkat Lunak Untuk Mengukur Kinerja Pekerjaan PT. Bangun Cipta Karya Pamungkas (PT. BCKP) Menggunakan Metode Balance Score Card Berbasis Web	
9	1311010045	Roby Ardiansyah	Perancangan Aplikasi Penjualan Tapis Lampung Berbasis Android Menggunakan Algoritma String Matching	
10	HafsaH Mukaromah	1411010110	Rancang Bangun Aplikasi Monitoring Hasil Produksi dan Informasi Preservasi (Perawatan) Tanaman Karet Menggunakan Metode K-Nearest Neighbor	
11	Feven Indriyani	1411010079	Rancang Bangun Aplikasi Kamus Sembayan PT Kereta Api Indonesia (PERSERO) Berbasis Android	
12	Iskandar Saputro Bakri	1411010107	Media Pembelajaran Geguritan Bali Berbasis Android	
13	Komang Wijane	1411010065	Implementasi Metode Location Based Services (LBS) Pada Studi Kasus Pencarian Sanggar Bunga Di Bandar Lampung Berbasis Android	
14	M. Imroni Mustikhin	1411010042	Ketut Arraye, S.Kom, M.T.I	

**FORMULIR****BIRO ADMINISTRASI AKADEMIK KEMAHASISWAAN (BAAK)****FORM KONSULTASI/BIMBINGAN SKRIPSI/TUGAS AKHIR \*)**

NAMA : Feren Indriyani  
 TAHUN : 1911010079  
 PEMBIMBING I : Hanifanto Wibawo, S.Kom., M.TI  
 PEMBIMBING II :  
 JUMLAH LAPORAN : DANCANG BANGUN APLIKASI MONITORING HASIL  
 PRODUksi DAN INFORMASI PRESERVASI (PERAWATAN)  
 TANAMAN KAFE MENGGUNAKAN METODE FUZZIES NEIGHBOR  
 WAKTU : s.d (6+2 bulan)

NO	HARI/TANGGAL	HASIL KONSULTASI	PARAF
1	9/7/2018, Senin	+ lata beluluk, rumput laut + bahan bahan, teknik	✓
2	13/7/2018, Jumat	+ literatur, teknologi bahan + penelitian seluruh metode KNP.	✓
3	4/8/2018, Jumat	+ Metodologi penelitian + Uraian + Class diagram	✓
4	8/8/2018	+ Aplikasi <u>Metode KNP</u> .	✓
5	6/9/2018	+ Uji program	✓
6	10/8/2018	+ Sinyal dan jawaban	✓
7	14/8/2018	+ bagian berdasar	✓
8	21/8/2018	+ dugaan sidang	✓
9			
10			

\*) Coret yang tidak perlu

Bandar Lampung, .....  
Ketua Jurusan( ..... )  
NIK.

# **LAMPIRAN**

....::: Manifest.xml:::....

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.art.code.perkebunanptpn">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:name=".app.AppController"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Design.Light.NoActionBar">
        <activity
            android:name=".activity.SplashActivity"
            android:launchMode="singleTop"
            android:theme="@style/SplashTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".activity.LoginActivity" />
        <activity android:name=".activity.MandorActivity" />
        <activity android:name=".activity.KawilActivity" />
        <activity android:name=".activity.EntryDataActivity" />
        <activity android:name=".activity.KnnActivity"></activity>
    </application>

</manifest>
```

....:::EntryDataActivity.java:::....

```
package com.art.code.perkebunanptpn.activity;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.BoringLayout;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.art.code.perkebunanptpn.R;
import com.art.code.perkebunanptpn.app.AppConfig;
import com.art.code.perkebunanptpn.app.AppController;
import com.art.code.perkebunanptpn.helper.SessionManager;
import com.art.code.perkebunanptpn.model.Production;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class EntryDataActivity extends AppCompatActivity {

    private static final String TAG = LoginActivity.class.getSimpleName();
    private EditText mBulan;
    private EditText mTahun;
    private EditText mProduksi;
    private Button mBtnEntry;

    private ProgressDialog pDialog;
    private SessionManager session;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.entry_screen);

// init
session = new SessionManager(getApplicationContext());
mBulan = findViewById(R.id.bulanentry);
mTahun = findViewById(R.id.tahunentry);
mProduksi = findViewById(R.id.produksientry);
mBtnEntry = findViewById(R.id.simpanentry);
TextView mAreaInc = findViewById(R.id.area_in_charge);
mAreaInc.setText(session.getArea());

pDialog = new ProgressDialog(this);
pDialog.setCancelable(false);

mBtnEntry.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (checkData()) {
            inputData(Integer.parseInt(mBulan.getText().toString()),
            Integer.parseInt(mTahun.getText().toString()),
            Integer.parseInt(mProduksi.getText().toString()));
        } else {
            Toast.makeText(getApplicationContext(), "Pastikan Seluruh data
diisi.", Toast.LENGTH_LONG).show();
        }
    }
});

private boolean checkData() {
    return !mBulan.getText().toString().isEmpty() &&
    !mTahun.getText().toString().isEmpty() &&
    !mProduksi.getText().toString().isEmpty();
}

private void inputData(final int month, final int year, final int production) {
    String tag_string_req = "req_input";
    pDialog.setMessage("Menyimpan data...");
    showDialog();

    StringRequest strReq = new StringRequest(Request.Method.POST,
    AppConfig.URL_POST_DATA,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d(TAG, "Response: " + response.toString());
            hideDialog();
        }
    });
}

```

```

        if (response.toString().contains("true")) {
            Toast.makeText(getApplicationContext(), "Data berhasil ditambahkan.", Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(getApplicationContext(), "Data berhasil diubah.", Toast.LENGTH_LONG).show();
        }

        Intent intent = getIntent();
        setResult(Activity.RESULT_OK, intent);
        finish();

    }

    , new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e(TAG, "Login Error: " + error.getMessage());
            Toast.makeText(getApplicationContext(),
                    error.getMessage(), Toast.LENGTH_LONG).show();
            hideDialog();
        }
    });

    @Override
    protected Map<String, String> getParams() {

        Map<String, String> params = new HashMap<String, String>();
        params.put("month", String.valueOf(month));
        params.put("year", String.valueOf(year));
        params.put("production", String.valueOf(production));
        params.put("username", session.getUser().getUsername());

        return params;
    }
};

AppController.getInstance().addToRequestQueue(strReq, tag_string_req);
}

private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
}

private void hideDialog() {
    if (pDialog.isShowing())

```

```
    pDialog.dismiss();
}

}
```

### ....:::KawilActivity.java:::....

```
package com.art.code.perkebunanptpn.activity;

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Build;
import android.support.annotation.NonNull;
import android.support.design.widget.BottomNavigationView;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.content.ContextCompat;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.MenuItem;
import android.view.Window;
import android.view.WindowManager;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.art.code.perkebunanptpn.R;
import com.art.code.perkebunanptpn.adapter.AdapterProduksiMandor;
import com.art.code.perkebunanptpn.app.AppConfig;
import com.art.code.perkebunanptpn.app.AppController;
import com.art.code.perkebunanptpn.helper.SessionManager;
import com.art.code.perkebunanptpn.model.Production;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
```

```
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class KawilActivity extends AppCompatActivity {

    private static final String TAG = LoginActivity.class.getSimpleName();
    private ProgressDialog pDialog;
    private SessionManager session;
    private DrawerLayout mDrawerLayout;

    private List<Production> productionList;
    private List<Production> productionListBulanIni;
    private String year;

    private int iTotalBulanIni;
    private TextView total;
    Fragment fragmentProduksi, fragmentPreservasi;
    private ActionBar actionBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.kawil_home_screen);

        // init
        session = new SessionManager(getApplicationContext());
        pDialog = new ProgressDialog(this);
        pDialog.setCancelable(false);

        productionList = new ArrayList<>();
        productionListBulanIni = new ArrayList<>();
        iTotalBulanIni = 0;

        year = getYear();

        BottomNavigationView navigation = findViewById(R.id.btm_nav_kawil);
        navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelected
            Listener);

        setToolBar();
        setStatusBarColor();

        loadData(session.getUser().getUsername());
    }
}
```

```

    }

private BottomNavigationView.OnNavigationItemSelectedListener
mOnNavigationItemSelectedListener
= new BottomNavigationView.OnNavigationItemSelectedListener() {

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.home:
                actionBar.setTitle("Wilayah - Produksi");
                fragmentProduksi = ProFrag.newInstance(productionList,
session.getUser().getUac(), setTotal());
                loadFragment(fragmentProduksi);
                return true;
            case R.id.preserve:
                actionBar.setTitle("Preservasi");
                fragmentPreservasi = PreFrag.newInstance();
                loadFragment(fragmentPreservasi);
                return true;
        }
        return false;
    }
};

private void loadFragment(Fragment fragment) {
    // load fragment
    FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.frag_container, fragment);
    transaction.addToBackStack(null);
    transaction.commit();
}

private String getYear() {
    Calendar c = Calendar.getInstance();
    int year = c.get(Calendar.YEAR);
    return String.valueOf(year);
}

private int getMo() {
    Calendar c = Calendar.getInstance();
    int mo = c.get(Calendar.MONTH);
    return (mo + 1);
}

private void setToolBar() {
    Toolbar mToolbar = findViewById(R.id.main_toolbar);
}

```

```

setSupportActionBar(mToolbar);
ActionBar = getSupportActionBar();
ActionBar.setDisplayHomeAsUpEnabled(true);
ActionBar.setHomeAsUpIndicator(R.drawable.ic_menu_black_24dp);
ActionBar.setTitle("Wilayah - Produksi");

mDrawerLayout = findViewById(R.id.main_drawer_layout);

    NavigationView mNavigationView =
findViewById(R.id.main_drawer_nav);
    mNavigationView.inflateMenu(R.menu.drawer_kawil);

    mNavigationView.setNavigationItemSelectedListener(
        new NavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem
item) {
                switch (item.getItemId()) {
                    case R.id.logout:
                        session.setLogin(false);
                        Intent intentLogout = new Intent(KawilActivity.this,
                            LoginActivity.class);
                        startActivity(intentLogout);
                        finish();
                        break;
                    case R.id.knn:
                        Intent knnAct = new Intent(KawilActivity.this,
                            KnnActivity.class);
                        startActivity(knnAct);
                        break;
                    case R.id.exit:
                        System.exit(0);
                        break;
                }
                item.setChecked(false);
                mDrawerLayout.closeDrawers();
                return true;
            }
        );
    }

private void setStatusBarColor() {
    Window window = this.getWindow();

    window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
}

```

```

window.addFlags(WindowManager.LayoutParams.FLAG_DRAW_S_SYSTEM_B
AR_BACKGROUNDS);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        window.setStatusBarColor(ContextCompat.getColor(this,
R.color.colorSecondary));
    }
}

private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
}

private void hideDialog() {
    if (pDialog.isShowing())
        pDialog.dismiss();
}

private void loadData(final String username) {

    String tag_string_req = "req_perkebunan";

    pDialog.setMessage("Mengambil data produksi...");
    showDialog();

    StringRequest strReq = new StringRequest(Request.Method.POST,
AppConfig.URL_TARIK_DATA_PRODUKSI,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d(TAG, "Response: " + response.toString());
            hideDialog();

            try {
                JSONObject jObj = new JSONObject(response);
                boolean error = jObj.getBoolean("error");
                JSONArray objArr = jObj.getJSONArray("perkebunan");

                if (!error) {
                    for (int y = 0; y < objArr.length(); y++) {
                        JSONObject perkebunanObj = null;
                        perkebunanObj = objArr.getJSONObject(y);

                        String area = perkebunanObj.getString("ar");
                        String mandor = perkebunanObj.getString("mandor");
                        int mon = perkebunanObj.getInt("month");
                        int pro = perkebunanObj.getInt("pro");
                
```

```

Production tmp = new Production();
tmp.setProduction(pro);
tmp.setMonth(mon);
tmp.setYear(Integer.parseInt(year));
tmp.setAreaname(area);
tmp.setUsername(mandor);

productionList.add(tmp);

productionListBulanIni.add(tmp);
// }
}

String ttl = setTotal();
fragmentProduksi = ProFrag.newInstance(productionList,
session.getUser().getUac(), ttl);
loadFragment(fragmentProduksi);

} else {
String errorMsg = jObj.getString("error_msg");
Toast.makeText(getApplicationContext(),
errorMsg, Toast.LENGTH_LONG).show();
}

} catch (JSONException e) {
e.printStackTrace();
Toast.makeText(getApplicationContext(), "Json error: " +
e.getMessage(), Toast.LENGTH_LONG).show();
}
}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
Log.e(TAG, "Login Error: " + error.getMessage());
Toast.makeText(getApplicationContext(),
error.getMessage(), Toast.LENGTH_LONG).show();
hideDialog();
}
}) {

@Override
protected Map<String, String> getParams() {
//post param
Map<String, String> params = new HashMap<String, String>();
params.put("username", username);
params.put("year", year);

return params;
}
}

```

```

        }
    };

    AppController.getInstance().addToRequestQueue(strReq,
tag_string_req);
}

private String setTotal() {
    iTotalBulanIni = 0;
    for (int i = 0; i < productionListBulanIni.size(); i++) {
        iTotalBulanIni += productionListBulanIni.get(i).getProduction();
    }
    String ttl = iTotalBulanIni + " kg";
//    total.setText(ttl);
    return ttl;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            mDrawerLayout.openDrawer(GravityCompat.START);
            return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

....:::KnnActivity.java:::....

```

package com.art.code.perkebunanptpn.activity;

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

```

```
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.art.code.perkebunanptpn.R;
import com.art.code.perkebunanptpn.app.AppConfig;
import com.art.code.perkebunanptpn.app.AppController;
import com.art.code.perkebunanptpn.model.Production;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class KnnActivity extends AppCompatActivity {

    private EditText knnNumber;
    private TextView knnDetail;
    private String iYear;
    private ProgressDialog pDialog;
    private List<Production> productionList;
    private int k = 4;
    private String category;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.knn_test);

        pDialog = new ProgressDialog(this);
        pDialog.setCancelable(false);
        setToolBar();
        setStatusBarColor();

        knnNumber = findViewById(R.id.knn_number);
        final Button knnButton = findViewById(R.id.knn_button);
        knnDetail = findViewById(R.id.knn_detail);

        productionList = new ArrayList<>();
```

```

knnButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (knnButton.getText().toString().toLowerCase().equals("hitung
kategori")) {
            if (knnNumber.getText().toString().isEmpty()) {
                Toast.makeText(KnnActivity.this, "Ketikan angka produksi
yang akan dihitung", Toast.LENGTH_LONG).show();
            } else {
                calculateKnn();
            }
            knnButton.setText("Reset");
            knnNumber.setText("");
        } else {
            knnButton.setText("Hitung Kategori");
            knnDetail.setText(R.string.knn_detail);
        }
    }
});

iYear = getYear();
loadProData();
}

private String getYear() {
    Calendar c = Calendar.getInstance();
    int year = c.get(Calendar.YEAR);
    return String.valueOf(year - 1);
}

private void loadProData() {

String tag_string_req = "req_perkebunan";

pDialog.setMessage("Mengambil data produksi...");
showDialog();

StringRequest strReq = new StringRequest(Request.Method.POST,
AppConfig.URL_TARIK_DATA_PRODUKSI_ALL,
new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {

hideDialog();

try {
    JSONObject jObj = new JSONObject(response);
}

```

```

boolean error = jObj.getBoolean("error");

if (!error) {
    JSONArray objArr = jObj.getJSONArray("perkebunan");
    for (int y = 0; y < objArr.length(); y++) {
        JSONObject perkebunanObj = null;
        perkebunanObj = objArr.getJSONObject(y);

        String area = perkebunanObj.getString("ar");
        String mandor = perkebunanObj.getString("mandor");
        int mon = perkebunanObj.getInt("month");
        int pro = perkebunanObj.getInt("pro");

        Production tmp = new Production();
        tmp.setProduction(pro);
        tmp.setMonth(mon);
        tmp.setYear(Integer.parseInt(iYear));
        tmp.setAreaname(area);
        tmp.setUsername(mandor);

        productionList.add(tmp);
    }

} else {
    String errorMsg = jObj.getString("error_msg");
    Toast.makeText(getApplicationContext(),
        errorMsg, Toast.LENGTH_LONG).show();
}

} catch (JSONException e) {
    e.printStackTrace();
    Toast.makeText(getApplicationContext(), "Json error: " +
e.getMessage(), Toast.LENGTH_LONG).show();
}
}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {

    Toast.makeText(getApplicationContext(),
        error.getMessage(), Toast.LENGTH_LONG).show();
    hideDialog();
}
}) {

@Override
protected Map<String, String> getParams() {
    //post param
}
}

```

```

        Map<String, String> params = new HashMap<String, String>();
        params.put("year", String.valueOf(iYear));

        return params;
    }
};

AppController.getInstance().addToRequestQueue(strReq, tag_string_req);
}

private void calculateKnn() {
    Collections.sort(productionList);
    /* Menghitung jarak c ke setiap titik */
    int iC = Integer.parseInt(knnNumber.getText().toString());
    List<Integer> dist = new ArrayList<>();
    for (int i = 0; i < productionList.size(); i++) {
        int temp = productionList.get(i).getProduction();
        int tempDist = Math.abs(iC - temp);
        dist.add(tempDist);
    }
    /* Mengurutkan dari yang jarak paling kecil */
    List<Production> rank = new ArrayList<>();
    for (int i = 0; i < dist.size(); i++) {
        int y = i;
        for (int j = i + 1; j < dist.size(); j++) {
            if (dist.get(y) > dist.get(j)) {
                y = j;
            }
        }
        rank.add(productionList.get(y));
        int tempDistance = dist.get(y);
        if (y != i) {
            dist.remove(y);
            dist.add(0, tempDistance);
        }
    }

    /* Cek KNN dengan k = 4 */
    int category1 = 0;
    int category2 = 0;
    int category3 = 0;
    for (int y = 0; y < k; y++) {
        int pro = rank.get(y).getProduction();
        int comp = productionList.get(productionList.size()/3).getProduction();
        int comp2 =
productionList.get(2*productionList.size()/3).getProduction();
        Log.d("TAG", "comp1: "+comp+"; comp2: "+comp2+"; pro: "+pro);
        if (iC < comp) {

```

```

        category1 += 1;
    } else if (iC > comp && iC < comp2) {
        category2 += 1;
    } else if (iC > comp2){
        category3 += 1;
    }
}
// Log.d("TAG", "calculateKnn: "+ category1 + " cat2: " + category2 + "
cat3: "+ category3);
if (category1 > category2 && category1 > category3) {
    category = "Rendah";
} else if (category2 > category1 && category2 > category3) {
    category = "Sedang";
} else {
    category = "Tinggi";
}
String result = "Nilai k = "+ k + "\nNilai c = " + iC + "\nMasuk kedalam
kategori = " + category;
knnDetail.setText(result);
}

private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
}

private void hideDialog() {
    if (pDialog.isShowing())
        pDialog.dismiss();
}

private void setToolBar() {
    Toolbar mToolbar = findViewById(R.id.main_toolbar);
    setSupportActionBar(mToolbar);
    ActionBar actionBar = getSupportActionBar();
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setHomeAsUpIndicator(R.drawable.ic_arrow_back_black_24dp);
    actionBar.setTitle("K-nearest Neighbor");
}

private void setStatusBarColor() {
    Window window = this.getWindow();

    window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);

    window.addFlags(WindowManager.LayoutParams.FLAG_DRAW_SYSTEM_BAR_BACKGROUNDS);
}

```

```

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    window.setStatusBarColor(ContextCompat.getColor(this,
        R.color.colorSecondary));
}
}

@Override
public boolean onSupportNavigateUp() {
    onBackPressed();
    return true;
}
}

```

....:::LoginActivity.java:::....

```

package com.art.code.perkebunanptpn.activity;

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Build;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;

import com.art.code.perkebunanptpn.R;
import com.art.code.perkebunanptpn.app.AppConfig;
import com.art.code.perkebunanptpn.app.AppController;
import com.art.code.perkebunanptpn.helper.SQLiteHandler;
import com.art.code.perkebunanptpn.helper.SessionManager;
import com.art.code.perkebunanptpn.model.Users;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;

```

```
import java.util.Map;

public class LoginActivity extends AppCompatActivity {

    private static final String TAG = LoginActivity.class.getSimpleName();
    private Button btnLogin;
    private EditText textUsName;
    private EditText textPasswd;
    private ProgressDialog pDialog;
    // private SQLiteHandler db;
    private SessionManager session;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.login_screen);

        pDialog = new ProgressDialog(this);
        pDialog.setCancelable(false);

        session = new SessionManager(getApplicationContext());

        if (session.isLoggedIn()) {

            if (session.getUser().getUac() == 1) {
                Intent mandorHome = new Intent(LoginActivity.this,
                        MandorActivity.class);
                startActivity(mandorHome);
                finish();
            } else {
                Intent mabesHome = new Intent(LoginActivity.this,
                        KawilActivity.class);
                startActivity(mabesHome);
                finish();
            }
        }

        Window window = this.getWindow();

        window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);

        window.addFlags(WindowManager.LayoutParams.FLAG_DRAW_SYSTEM_BAR_BACKGROUNDS);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
```

```
        window.setStatusBarColor(ContextCompat.getColor(this,
                android.R.color.black));
    }

btnLogin = findViewById(R.id.btnLoginScr);
textUsName = findViewById(R.id.usernameLoginScr);
textPasswd = findViewById(R.id.passwordLoginScr);

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        String usTmp = textUsName.getText().toString().trim();
        String pwTmp = textPasswd.getText().toString().trim();

        if ( !usTmp.isEmpty() && !pwTmp.isEmpty() ) {

            checkLogin(usTmp, pwTmp);
        } else {
            Toast.makeText(LoginActivity.this,
                    "Silahkan masukkan username dan password!!!",
                    Toast.LENGTH_LONG).show();
        }
    }
});

private void checkLogin(final String username, final String password) {

    String tag_string_req = "req_login";

    pDialog.setMessage("Mencoba masuk ...");
    showDialog();

    StringRequest strReq = new StringRequest(Request.Method.POST,
            AppConfig.URL_LOGIN, new Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
            Log.d(TAG, "Login Response: " + response.toString());
            hideDialog();

            try {
                JSONObject jObj = new JSONObject(response);
                boolean error = jObj.getBoolean("error");
            }
        }
    });
}
```

```

if (!error) {

    JSONObject user = jObj.getJSONObject("user");
    String username = user.getString("username");
    int level = Integer.parseInt( user.getString("uac") );
    String name = user.getString("name");
    String address = user.getString("address");
    String phone = user.getString("phone");
    String email = user.getString("email");

    session.setLogin(true)
    session.setUserCredentials(new Users(username, level, name,
address, phone, email));
    if (level == 1) {
        String afd = user.getString("area");
        session.setArea(afd);
        Intent madorHome = new Intent(LoginActivity.this,
MandorActivity.class);
        startActivity(madorHome);
        finish();
    } else {
        Intent mabesHome = new Intent(LoginActivity.this,
KawilActivity.class);
        startActivity(mabesHome);
        finish();
    }
} else {

    String errorMsg = jObj.getString("error_msg");
    Toast.makeText(getApplicationContext(),
        errorMsg, Toast.LENGTH_LONG).show();
}

catch (JSONException e) {

    e.printStackTrace();
    Toast.makeText(getApplicationContext(), "Json error: " +
e.getMessage(), Toast.LENGTH_LONG).show();
}

}, new Response.ErrorListener() {

    @Override
public void onErrorResponse(VolleyError error) {
        Log.e(TAG, "Login Error: " + error.getMessage());
        Toast.makeText(getApplicationContext(),
            error.getMessage(), Toast.LENGTH_LONG).show();
}
}

```

```

        hideDialog();
    }
}) {

    @Override
    protected Map<String, String> getParams() {

        Map<String, String> params = new HashMap<String, String>();
        params.put("username", username);
        params.put("password", password);

        return params;
    }

};

AppController.getInstance().addToRequestQueue(strReq, tag_string_req);

}

private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
}

private void hideDialog() {
    if (pDialog.isShowing())
        pDialog.dismiss();
}

}

```

...:::MandorActivity.java:::...

```

package com.art.code.perkebunanptpn.activity;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Build;
import android.support.annotation.NonNull;
import android.support.design.widget.BottomNavigationView;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.content.ContextCompat;
import android.support.v4.view.GravityCompat;

```

```
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.art.code.perkebunanptpn.R;
import com.art.code.perkebunanptpn.adapter.AdapterProduksiMandor;
import com.art.code.perkebunanptpn.app.AppConfig;
import com.art.code.perkebunanptpn.app.AppController;
import com.art.code.perkebunanptpn.helper.SessionManager;
import com.art.code.perkebunanptpn.model.Production;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class MandorActivity extends AppCompatActivity {

    private static final String TAG = LoginActivity.class.getSimpleName();
    private ProgressDialog pDialog;
    private SessionManager session;
    private DrawerLayout mDrawerLayout;
    private AdapterProduksiMandor apm;

    private List<Production> productionList;
    private String year;
```

```
private static final int REQUEST_INPUT = 555;
private Button mBtnEntryData;
private TextView mNamaArea;
private TextView mNamaUser;
Fragment fragmentProduksi, fragmentPreservasi;
private ActionBar actionBar;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mandor_home_screen);

    // init
    session = new SessionManager(getApplicationContext());
    pDialog = new ProgressDialog(this);
    pDialog.setCancelable(false);
    productionList = new ArrayList<>();

    mBtnEntryData = findViewById(R.id.tombol_entry_data);
    mNamaArea = findViewById(R.id.nama_area);
    mNamaUser = findViewById(R.id.nama_user);
    Recycler View rvProduksi = findViewById(R.id.produksi_RV);
    LinearLayoutManager llm = new LinearLayoutManager(this);
    rvProduksi.setLayoutManager(llm);
    apm = new AdapterProduksiMandor(productionList,
    session.getUser().getUac());
    rvProduksi.setAdapter(apm);
    BottomNavigationView navigation = findViewById(R.id.btm_nav);

    navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelected
    Listener);
    navigation.setVisibility(View.GONE);

    mNamaUser.setText(session.getUser().getName());
    mNamaArea.setText(session.getArea());

    mBtnEntryData.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent entryIntent = new Intent(MandorActivity.this,
            EntryDataActivity.class);
            startActivityForResult(entryIntent, REQUEST_INPUT);
        }
    });

    setToolBar();
    setStatusBarColor();
```

```
year = getYear();

        loadData(session.getUser().getUsername());
    }

private BottomNavigationView.OnNavigationItemSelectedListener
mOnNavigationItemSelectedListener
    = new BottomNavigationView.OnNavigationItemSelectedListener() {

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.home:
                actionBar.setTitle("Mandor - Produksi");

                return true;
            case R.id.preserve:
                actionBar.setTitle("Preservasi");

                return true;
        }
        return false;
    }
};

private void loadFragment(Fragment fragment) {
    //load fragment
    FragmentTransaction transaction =
    getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.frag_container, fragment);
    transaction.addToBackStack(null);
    transaction.commit();
}

private String getYear() {
    Calendar c = Calendar.getInstance();
    int year = c.get(Calendar.YEAR);
    return String.valueOf(year);
}

private void setToolBar() {
    Toolbar mToolbar = findViewById(R.id.main_toolbar);
    setSupportActionBar(mToolbar);
    actionBar = getSupportActionBar();
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setHomeAsUpIndicator(R.drawable.ic_menu_black_24dp);
```

```
mDrawerLayout = findViewById(R.id.main_drawer_layout);

    NavigationView mNavigationView =
findViewById(R.id.main_drawer_nav);
    mNavigationView.inflateMenu(R.menu.drawer_mandor);

    mNavigationView.setNavigationItemSelectedListener(
        new NavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem
item) {
                switch (item.getItemId()) {
                    case R.id.logout:
                        session.setLogin(false);
                        Intent intentLogout = new Intent(MandorActivity.this,
                            LoginActivity.class);
                        startActivity(intentLogout);
                        finish();
                        break;
                    case R.id.exit:
                        System.exit(0);
                        break;
                }
                item.setChecked(false);
                mDrawerLayout.closeDrawers();
                return true;
            }
        );
    );
}

private void setStatusBarColor() {
    Window window = this.getWindow();

    window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_S
TATUS);

    window.addFlags(WindowManager.LayoutParams.FLAG_DRAW_S
YSTEM_B
AR_BACKGROUNDS);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        window.setStatusBarColor(ContextCompat.getColor(this,
R.color.colorSecondary));
    }
}

private void showDialog() {
    if (!pDialog.isShowing())
        pDialog.show();
```

```

}

private void hideDialog() {
    if (pDialog.isShowing())
        pDialog.dismiss();
}

private void loadData(final String username) {
    String tag_string_req = "req_perkebunan";

    pDialog.setMessage("Mengambil data produksi...");
    showDialog();

    StringRequest strReq = new StringRequest(Request.Method.POST,
AppConfig.URL_TARIK_DATA_PERKEBUNAN,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d(TAG, "Response: " + response.toString());
            hideDialog();

            try {
                JSONObject jObj = new JSONObject(response);
                boolean error = jObj.getBoolean("error");
                JSONArray objArr = jObj.getJSONArray("perkebunan");

                if (!error) {
                    for (int y = 0; y < objArr.length(); y++) {
                        JSONObject perkebunanObj = null;
                        perkebunanObj = objArr.getJSONObject(y);

                        int id = perkebunanObj.getInt("pro_id");
                        int mon = perkebunanObj.getInt("month");
                        int yr = perkebunanObj.getInt("year");
                        int wa = perkebunanObj.getInt("wa_id");
                        int pro = perkebunanObj.getInt("pro");
                        String us = perkebunanObj.getString("user");
                        String ar = perkebunanObj.getString("area");

                        Production tmp = new Production(id, mon, yr, wa, pro, us,
ar);
                        mNamaArea.setText(ar);
                        productionList.add(tmp);
                }
            }

            apm.notifyDataSetChanged();
        }
    }
}

```

```

    } else {
        String errorMsg = jObj.getString("error_msg");
        Toast.makeText(getApplicationContext(),
                errorMsg, Toast.LENGTH_LONG).show();
    }

} catch (JSONException e) {
    e.printStackTrace();
    Toast.makeText(getApplicationContext(), "Json error: " +
e.getMessage(), Toast.LENGTH_LONG).show();
}

}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    Log.e(TAG, "Login Error: " + error.getMessage());
    Toast.makeText(getApplicationContext(),
            error.getMessage(), Toast.LENGTH_LONG).show();
    hideDialog();
}
}) {

@Override
protected Map<String, String> getParams() {
    Map<String, String> params = new HashMap<String, String>();
    params.put("username", username);
    params.put("year", year);

    return params;
}
};

AppController.getInstance().addToRequestQueue(strReq, tag_string_req);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            mDrawerLayout.openDrawer(GravityCompat.START);
            return true;
    }
    return super.onOptionsItemSelected(item);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_INPUT && resultCode ==

```

```
Activity.RESULT_OK) {  
    productionList.clear();  
    loadData(session.getUser().getUsername());  
}  
}  
}  
}
```

....::PreFragActivity.java:::....

```
package com.art.code.perkebunanptpn.activity;  
  
import android.content.Context;  
import android.net.Uri;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import com.art.code.perkebunanptpn.R;  
import com.art.code.perkebunanptpn.adapter.AdapterProduksiMandor;  
import com.github.barteksc.pdfviewer.PDFView;  
  
public class PreFrag extends Fragment {  
  
    public PreFrag() {  
        // Required empty public constructor  
    }  
  
    public static PreFrag newInstance() {  
        PreFrag fragment = new PreFrag();  
        return fragment;  
    }  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        View view = inflater.inflate(R.layout.pdf_fragment, container, false);  
        PDFView pdf_PdfView = view.findViewById(R.id.pdf);  
    }  
}
```

```
        pdf_PdfView.fromAsset("pre.pdf").load();
        return view;
    }

}
```

....:::ProFragActivity.xml:::....

```
package com.art.code.perkebunanptpn.activity;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.art.code.perkebunanptpn.R;
import com.art.code.perkebunanptpn.adapter.AdapterProduksiMandor;
import com.art.code.perkebunanptpn.model.Production;

import java.io.Serializable;
import java.util.List;

public class ProFrag extends Fragment {

    private List<Production> productionList;
    private int uac;
    private String total;

    public ProFrag() {

    }

    public static ProFrag newInstance(List<Production> productionList, int uac,
String total) {
        ProFrag fragment = new ProFrag();
        Bundle args = new Bundle();
        args.putSerializable("proList", (Serializable) productionList);
        args.putInt("uac", uac);
        args.putString("total", total);
    }
}
```

```

        fragment.setArguments(args);
        return fragment;
    }

    @Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        productionList = (List<Production>)
getArguments().getSerializable("proList");
        uac = getArguments().getInt("uac");
        total = getArguments().getString("total");
    }
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.kawil_fragment, container, false);

    AdapterProduksiMandor apm = new
AdapterProduksiMandor(productionList, uac);
    TextView total = view.findViewById(R.id.askep_total);
    RecyclerView askepSum = view.findViewById(R.id.askep_summary);
    LinearLayoutManager llm = new LinearLayoutManager(getActivity());
    askepSum.setLayoutManager(llm);
    askepSum.setAdapter(apm);
    total.setText(this.total);

    return view;
}
}

```

### ....:::SplashActivity.Java:::....

```

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class SplashActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

Intent intent = new Intent(SplashActivity.this, LoginActivity.class);
startActivity(intent);
finish();
}
}

```

....:::AdepterPreservasi.Java:::....

```

package com.art.code.perkebunanptpn.adapter;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.art.code.perkebunanptpn.R;
import com.art.code.perkebunanptpn.model.Preservation;
import com.art.code.perkebunanptpn.model.Production;

import java.util.List;

public class AdapterPreservasi extends
RecyclerView.Adapter<AdapterPreservasi.PreservasiViewHolder> {

    private List<Preservation> preservationList;

    public AdapterPreservasi(List<Preservation> preservationList) {
        this.preservationList = preservationList;
    }

    public class PreservasiViewHolder extends RecyclerView.ViewHolder {
        TextView judul;
        public PreservasiViewHolder(View itemView) {
            super(itemView);
            judul = itemView.findViewById(R.id.judul);
        }
    }

    @Override
    public PreservasiViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View v =
LayoutInflator.from(parent.getContext()).inflate(R.layout.item_preservasi, parent,
false);
        PreservasiViewHolder pvh = new PreservasiViewHolder(v);
        return pvh;
    }
}

```

```

@Override
public void onBindViewHolder(PreservasiViewHolder holder, int position) {
    holder.judul.setText(preservationList.get(position).getJudul());
}

@Override
public int getItemCount() {
    return preservationList.size();
}

}

```

....:::AdapterProduksiMandor.Java:::....

```

package com.art.code.perkebunanptpn.adapter;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.art.code.perkebunanptpn.R;
import com.art.code.perkebunanptpn.model.Production;

import java.util.List;

public class AdapterProduksiMandor extends
RecyclerView.Adapter<AdapterProduksiMandor.ProduksiViewHolder> {

    private List<Production> daftarProduksi;
    private int uac;

    public AdapterProduksiMandor(List<Production> daftarProduksi, int uac) {
        this.daftarProduksi = daftarProduksi;
        this.uac = uac;
    }

    public class ProduksiViewHolder extends RecyclerView.ViewHolder {
        TextView proBulan, proValue, proArea;
        public ProduksiViewHolder(View itemView) {
            super(itemView);
            proBulan = itemView.findViewById(R.id.produksi_bulan);
            proValue = itemView.findViewById(R.id.produksi_value);
            proArea = itemView.findViewById(R.id.produksi_area);
        }
    }
}

```

```

@Override
public ProduksiViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
    View v =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_produksi_mandor,
parent, false);
    ProduksiViewHolder pvh = new ProduksiViewHolder(v);
    return pvh;
}

@Override
public void onBindViewHolder(ProduksiViewHolder holder, int position) {
    String pro = daftarProduksi.get(position).getProduction() + " kg";
    holder.proValue.setText(pro);
    String sBulan = "Produksi Bulan " +
daftarProduksi.get(position).getMonth() + " Tahun " +
daftarProduksi.get(position).getYear();
    holder.proBulan.setText(sBulan);
    if (uac == 1) {
        holder.proArea.setVisibility(View.GONE);
    } else {
        holder.proArea.setText(daftarProduksi.get(position).getAreaname());
    }
}

@Override
public int getItemCount() {
    return daftarProduksi.size();
}
}

```

....:::AppConfig.Java:::....

```

package com.art.code.perkebunanptpn.app;

public class AppConfig {

    private static final String server =
"http://perkebunanptpn.000webhostapp.com/perkebunan/";

    public static final String URL_TARIK_DATA_PRODUKSI_ALL = server +
"tarikDataProduksi.php";

    public static String URL_LOGIN = server+"login.php";
}

```

```

public static String URL_TARIK_DATA_PERKEBUNAN =
server+"tarikDataPerkebunan.php";

public static String URL_TARIK_DATA_PRODUKSI =
server+"tarikDataProduksiKawil.php";

public static String URL_POST_DATA = server+"input.php";
}

```

....:::AppController.Java:::....

```

package com.art.code.perkebunanptpn.app;

import android.app.Application;
import android.text.TextUtils;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;

public class AppController extends Application {

    public static final String TAG = AppController.class.getSimpleName();

    private RequestQueue mRequestQueue;

    private static AppController mInstance;

    @Override
    public void onCreate() {
        super.onCreate();
        mInstance = this;
    }

    public static synchronized AppController getInstance() {
        return mInstance;
    }

    public RequestQueue getRequestQueue() {
        if (mRequestQueue == null) {
            mRequestQueue = Volley.newRequestQueue(getApplicationContext());
        }

        return mRequestQueue;
    }

    public <T> void addToRequestQueue(Request<T> req, String tag) {

```

```

        req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
        getRequestQueue().add(req);
    }

public <T> void addToRequestQueue(Request<T> req) {
    req.setTag(TAG);
    getRequestQueue().add(req);
}

public void cancelPendingRequests(Object tag) {
    if (mRequestQueue != null) {
        mRequestQueue.cancelAll(tag);
    }
}

}

```

....:::Preservasion.Java:::....

```

package com.art.code.perkebunanptpn.model;

import java.io.Serializable;

public class Preservation implements Serializable {
    private int id;
    private String judul;
    private String isi;

    public Preservation() {
    }

    public Preservation(int id, String judul, String isi) {
        this.id = id;
        this.judul = judul;
        this.isi = isi;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getJudul() {
        return judul;
    }
}

```

```
public void setJudul(String judul) {
    this.judul = judul;
}

public String getIsi() {
    return isi;
}

public void setIsi(String isi) {
    this.isi = isi;
}
```

....:::Production.Java:::....

```
package com.art.code.perkebunanptpn.model;

import android.support.annotation.NonNull;

import java.io.Serializable;

public class Production implements Serializable, Comparable {
    private int id;
    private int month;
    private int year;
    private int id_workarea;
    private int production;
    private String username;
    private String areaname;

    public Production() {
    }

    public Production(int id, int month, int year, int id_workarea, int production,
String username, String areaname) {
        this.id = id;
        this.month = month;
        this.year = year;
        this.id_workarea = id_workarea;
        this.production = production;
        this.username = username;
        this.areaname = areaname;
    }

    public int getId() {
        return id;
    }
```

```
public void setId(int id) {
    this.id = id;
}

public int getMonth() {
    return month;
}

public void setMonth(int month) {
    this.month = month;
}

public int getId_workarea() {
    return id_workarea;
}

public void setId_workarea(int id_workarea) {
    this.id_workarea = id_workarea;
}

public int getProduction() {
    return production;
}

public void setProduction(int production) {
    this.production = production;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getAreaname() {
    return areaname;
}

public void setAreaname(String areaname) {
    this.areaname = areaname;
}

public int getYear() {
    return year;
}
```

```

public void setYear(int year) {
    this.year = year;
}

@Override
public int compareTo(@NonNull Object o) {
    int comp = ((Production)o).getProduction();
    return this.getProduction()-comp;
}
}

```

....:::Users.Java:::....

```

package com.art.code.perkebunanptpn.model;

import java.io.Serializable;

public class Users implements Serializable{
    private String username;
    private int uac;
    private String name;
    private String address;
    private String phone;
    private String email;

    public Users(String username, int uac, String name, String address, String
    phone, String email) {
        this.username = username;
        this.uac = uac;
        this.name = name;
        this.address = address;
        this.phone = phone;
        this.email = email;
    }

    public Users() {

    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }
}

```

```

}

public int getUac() {
    return uac;
}

public void setUac(int uac) {
    this.uac = uac;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getPhone() {
    return phone;
}

public void setPhone(String phone) {
    this.phone = phone;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}
}

....::: Detail_Screen.xml:::...

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

`xmlns:android="http://schemas.android.com/apk/res/android"`

```
xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">>

    <android.support.v7.widget.Toolbar
        android:id="@+id/preservasi_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:layout_alignParentTop="true"
        android:background="@color/colorSecondary"
        android:theme="@style/ThemeOverlay.AppCompat.ActionBar" />

    <TextView
        android:id="@+id/judul_preservasi"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:textSize="32sp"
        android:gravity="center"
        android:text="Title"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_marginTop="20dp"
        android:layout_height="350dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:padding="8dp"
        app:cardCornerRadius="8dp"
        android:backgroundTint="@color/colorLight">

        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/detail_preservasi"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_margin="8dp"
                android:padding="8dp"
                android:textSize="18sp"
                android:text="detail"/>

```

```
</ScrollView>

</android.support.v7.widget.CardView>

</LinearLayout>
....::: Entry_Screen.xml:::...

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/area_in_charge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="24dp"
        android:layout_marginTop="32dp"
        android:text="AFD-1"
        android:textAlignment="viewEnd"
        android:textSize="24sp"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <LinearLayout
        app:layout_constraintTop_toBottomOf="@+id/area_in_charge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="45dp"
        android:padding="16dp"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <android.support.design.widget.TextInputLayout
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight=".6"
                android:layout_marginStart="16dp"
                android:layout_marginEnd="4dp"
                android:layout_marginBottom="8dp"
                android:layout_marginTop="8dp">
```

```
<EditText
    android:id="@+id/bulanentry"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:hint="bulan"
    android:inputType="number" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight=".4"
    android:layout_marginStart="4dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp">

<EditText
    android:id="@+id/tahunentry"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:hint="tahun"
    android:inputType="number" />

</android.support.design.widget.TextInputLayout>

</LinearLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp">

<EditText
    android:id="@+id/produksientry"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:hint="produksi"
    android:inputType="number" />

</android.support.design.widget.TextInputLayout>
```

```
<Button
    android:id="@+id/simpanentry"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="45dp"
    android:layout_marginStart="24dp"
    android:layout_marginEnd="24dp"
    android:background="@color/colorSecondary"
    android:padding="8dp"
    android:text="S I M P A N"
    android:textSize="24sp"
    android:textColor="@color/colorLight"/>

</LinearLayout>
</android.support.constraint.ConstraintLayout>
```

....::: Item\_Preservation.xml:::....

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/produksi_cv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="3dp"
    android:backgroundTint="@color/colorThird"
    android:padding="8dp"
    app:cardCornerRadius="6dp"
    app:cardElevation="2dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="12dp">

        <TextView
            android:id="@+id/judul"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_centerVertical="true"
            android:layout_centerHorizontal="true"
            android:gravity="center"
            android:textColor="@color/colorLight"
            android:text="Judul"
            android:textSize="16sp"
```

```
        android:textStyle="bold" />  
  
</RelativeLayout>  
</android.support.v7.widget.CardView>
```

....::: Item\_Produksi\_Mandor.xml:::....

```
<?xml version="1.0" encoding="utf-8"?>  
<android.support.v7.widget.CardView  
    xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:app="http://schemas.android.com/apk/res-auto"  
            android:id="@+id/produksi_cv"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:layout_margin="3dp"  
            android:backgroundTint="@color/colorThird"  
            android:padding="8dp"  
            app:cardCornerRadius="6dp"  
            app:cardElevation="2dp">  
  
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:padding="12dp">  
  
<TextView  
    android:id="@+id/produksi_area"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentStart="true"  
    android:layout_centerVertical="true"  
    android:gravity="start"  
    android:textColor="@color/colorLight"  
    android:text="Area"  
    android:textSize="16sp"  
    android:textStyle="bold" />  
  
<TextView  
    android:id="@+id/produksi_bulan"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentTop="true"  
    android:gravity="end"  
    android:text="Bulan ke-#"  
    android:textColor="@color/colorDark"  
    android:textSize="12sp"  
    android:textStyle="italic" />
```

```
<TextView
    android:id="@+id/produksi_value"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/produksi_bulan"
    android:gravity="end"
    android:text="90.000.000"
    android:textColor="@color/colorLight"
    android:textSize="24sp"
    android:textStyle="bold" />
</RelativeLayout>
</android.support.v7.widget.CardView>
```

....::: Kawil\_Pragment.xml:::....

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorLight">

    <TextView
        android:id="@+id/askep"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/main_toolbar"
        android:layout_marginEnd="20dp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:gravity="end"
        android:text="Total Produksi Wilayah"
        android:textColor="@color/colorDark"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/askep_total"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/askep"
        android:layout_marginEnd="20dp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="12dp"
```

```

        android:gravity="end"
        android:text="597.934 Kg"
        android:textColor="@color/colorDark"
        android:textSize="48sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/askep" />

<android.support.v7.widget.RecyclerView
    android:id="@+id/askep_summary"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/btm_nav_kawil"
    android:layout_below="@+id/askep_total"
    android:layout_marginEnd="20dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="16dp" />

</RelativeLayout>

```

....::: Kawil\_HomeScreen.xml:::....

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main_drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/colorLight">

        <android.support.v7.widget.Toolbar
            android:id="@+id/main_toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:layout_alignParentTop="true"
            android:background="@color/colorSecondary"
            android:theme="@style/ThemeOverlay.AppCompat.ActionBar" />

        <FrameLayout
            android:id="@+id/frag_container"
            android:layout_below="@+id/main_toolbar"
            android:layout_above="@+id/btm_nav_kawil"

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

<android.support.design.widget.BottomNavigationView
    android:id="@+id/btm_nav_kawil"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:background="@color/colorSecondary"
    android:foreground="?attr/selectableItemBackground"
    app:itemBackground="@color/colorSecondary"
    app:menu="@menu/navigation" />

</RelativeLayout>

<android.support.design.widget.NavigationView
    android:id="@+id/main_drawer_nav"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true">

</android.support.design.widget.NavigationView>

</android.support.v4.widget.DrawerLayout>

```

....::: Knn\_Test.xml:::....

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.Toolbar
        android:id="@+id/main_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/colorSecondary"
        android:theme="@style/ThemeOverlay.AppCompat.ActionBar"/>

    <android.support.design.widget.TextInputLayout
        android:id="@+id/input_container"
        android:layout_below="@+id/main_toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"

```

```

        android:layout_marginEnd="16dp"
        android:layout_marginBottom="8dp"
        android:layout_marginTop="8dp">

    <EditText
        android:id="@+id/knn_number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingStart="12dp"
        android:paddingEnd="12dp"
        android:gravity="end"
        android:hint="Angka Produksi"
        android:inputType="number" />

</android.support.design.widget.TextInputLayout>

<Button
    android:id="@+id/knn_button"
    android:layout_below="@id/input_container"
    android:background="@drawable/rounded_button"
    android:layout_marginStart="32dp"
    android:layout_marginEnd="32dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/colorLight"
    android:text="Hitung Kategori"/>

<TextView
    android:id="@+id/knn_detail"
    android:layout_margin="32dp"
    android:layout_below="@id/knn_button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:text="@string/knn_detail"
    android:textSize="12sp"
    android:textStyle="italic"
    android:fontFamily="monospace"/>

</RelativeLayout>

```

..... Login\_Screen.xml:....

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteY="25dp">>

<ImageView
    android:contentDescription="logo_login_screen"
    android:id="@+id/logoLoginScr"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:minWidth="200dp"
    android:scaleType="centerCrop"
    android:src="@mipmap/ic_launcher_foreground"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginBottom="24dp"
    android:layout_marginTop="32dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:orientation="vertical"
    android:padding="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/logoLoginScr">
```

```
<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp">
```

```
<EditText
    android:id="@+id/usernameLoginScr"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="start"
    android:hint="username"
    android:inputType="textAutoComplete" />
```

```

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp">

    <EditText
        android:id="@+id/passwordLoginScr"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="start"
        android:hint="password"
        android:inputType="textPassword" />

</android.support.design.widget.TextInputLayout>

<Button
    android:id="@+id/btnLoginScr"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="24dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:padding="8dp"
    android:background="@color/colorSecondary"
    android:textColor="@color/colorLight"
    android:textSize="16sp"
    android:text="L O G I N" />

</LinearLayout>

</android.support.constraint.ConstraintLayout>

```

....::: Mandor\_Home\_Screen.xml:::....

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main_drawer_layout"
    android:layout_width="match_parent"

```

```
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <android.support.v7.widget.Toolbar
                android:id="@+id/main_toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:background="@color/colorSecondary"
                android:theme="@style/ThemeOverlay.AppCompat.ActionBar"/>

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="fill_parent"
                android:paddingTop="24dp"
                android:orientation="vertical" >

                <TextView
                    android:layout_alignParentTop="true"
                    android:id="@+id/nama_area"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginStart="32dp"
                    android:layout_marginEnd="32dp"
                    android:layout_marginTop="30dp"
                    android:gravity="end"
                    android:textSize="20sp"
                    android:textColor="@color/colorDark"
                    android:text="AFD-1"/>

                <TextView
                    android:id="@+id/nama_user"
                    android:layout_below="@+id/nama_area"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="6dp"
                    android:layout_marginStart="32dp"
                    android:layout_marginEnd="32dp"
                    android:gravity="end"
                    android:textSize="20sp"
```

```
        android:textColor="@color/colorDark"
        android:text="username" />

    <LinearLayout
        android:id="@+id/rec_cont"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/tombol_entry_data"
        android:layout_below="@+id/nama_user"
        android:layout_gravity="end"
        android:layout_marginHorizontal="32dp"
        android:layout_marginTop="16dp"
        android:orientation="vertical">

        <android.support.v7.widget.RecyclerView
            android:id="@+id/produksi_RV"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

    </LinearLayout>

    <Button
        android:id="@+id/tombol_entry_data"
        android:layout_alignParentBottom="true"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="12dp"
        android:layout_gravity="center"
        android:padding="16dp"
        android:background="@color/colorSecondary"
        android:textColor="@color/colorLight"
        android:textSize="24sp"
        android:text="ENTRY DATA PRODUKSI"/>

    <android.support.design.widget.BottomNavigationView
        android:id="@+id/btm_nav"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:background="@color/colorSecondary"
        android:foreground="?attr/selectableItemBackground"
        app:itemBackground="@color/colorSecondary"
        app:menu="@menu/navigation" />

</RelativeLayout>

</LinearLayout>
```

```
</RelativeLayout>

<android.support.design.widget.NavigationView
    android:id="@+id/main_drawer_nav"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true">

</android.support.design.widget.NavigationView>

</android.support.v4.widget.DrawerLayout>
```

..... Pdf\_Fragment.xml:::....

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.github.barteksc.pdfviewer.PDFView
        android:id="@+id/pdf"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>
```

..... Pdf\_Fragment.xml:::....

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="24dp"
```

```
        android:layout_marginEnd="24dp"
        android:layout_marginTop="16dp"
        android:orientation="vertical">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_marginTop="20dp"
        android:layout_height="75dp"
        android:padding="8dp"
        app:cardCornerRadius="8dp"
        android:backgroundTint="@color/colorThird">

        <android.support.constraint.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:padding="8dp">

            <TextView
                android:id="@+id/preservation_title"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="8dp"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintLeft_toLeftOf="parent"
                android:textSize="24sp"
                android:textColor="@color/colorLight"
                android:text="Title"/>

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="8dp"

                app:layout_constraintTop_toBottomOf="@+id/preservation_title"
                app:layout_constraintLeft_toLeftOf="parent"
                android:textSize="16sp"
                android:textColor="@color/colorLight"
                android:text="Kategori"/>

            <ImageButton
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                app:layout_constraintRight_toRightOf="parent"
                android:background="@color/colorThird"
                android:tint="@color/colorLight"
                android:layout_marginEnd="8dp"
                android:src="@drawable/ic_info_outline_black_24dp"/>

```

```
</android.support.constraint.ConstraintLayout>

</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:layout_width="match_parent"
    android:layout_marginTop="20dp"
    android:layout_height="75dp"
    android:padding="8dp"
    app:cardCornerRadius="8dp"
    android:backgroundTint="@color/colorThird">

    <android.support.constraint.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="8dp">

        <TextView
            android:id="@+id/preservation_title"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            android:textSize="24sp"
            android:textColor="@color/colorLight"
            android:text="Title"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"

            app:layout_constraintTop_toBottomOf="@+id/preservation_title"
            app:layout_constraintLeft_toLeftOf="parent"
            android:textSize="16sp"
            android:textColor="@color/colorLight"
            android:text="Kategori"/>

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:layout_constraintRight_toRightOf="parent"
            android:background="@color/colorThird"
            android:tint="@color/colorLight"
            android:layout_marginEnd="8dp"
            android:src="@drawable/ic_info_outline_black_24dp"/>
    
```

```
</android.support.constraint.ConstraintLayout>

</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:layout_width="match_parent"
    android:layout_marginTop="20dp"
    android:layout_height="75dp"
    android:padding="8dp"
    app:cardCornerRadius="8dp"
    android:backgroundTint="@color/colorThird">

    <android.support.constraint.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="8dp">

        <TextView
            android:id="@+id/preservation_title"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            android:textSize="24sp"
            android:textColor="@color/colorLight"
            android:text="Title"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"

            app:layout_constraintTop_toBottomOf="@+id/preservation_title"
            app:layout_constraintLeft_toLeftOf="parent"
            android:textSize="16sp"
            android:textColor="@color/colorLight"
            android:text="Kategori"/>

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:layout_constraintRight_toRightOf="parent"
            android:background="@color/colorThird"
            android:tint="@color/colorLight"
            android:layout_marginEnd="8dp"
            android:src="@drawable/ic_info_outline_black_24dp"/>
    
```

```
</android.support.constraint.ConstraintLayout>

</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:layout_width="match_parent"
    android:layout_marginTop="20dp"
    android:layout_height="75dp"
    android:padding="8dp"
    app:cardCornerRadius="8dp"
    android:backgroundTint="@color/colorThird">

    <android.support.constraint.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="8dp">

        <TextView
            android:id="@+id/preservation_title"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            android:textSize="24sp"
            android:textColor="@color/colorLight"
            android:text="Title"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"

            app:layout_constraintTop_toBottomOf="@+id/preservation_title"
            app:layout_constraintLeft_toLeftOf="parent"
            android:textSize="16sp"
            android:textColor="@color/colorLight"
            android:text="Kategori"/>

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            app:layout_constraintRight_toRightOf="parent"
            android:background="@color/colorThird"
            android:tint="@color/colorLight"
            android:layout_marginEnd="8dp"
            android:src="@drawable/ic_info_outline_black_24dp"/>
    
```

```
</android.support.constraint.ConstraintLayout>

</android.support.v7.widget.CardView>

</LinearLayout>

</ScrollView>

</android.support.constraint.ConstraintLayout>
```