



**IMPLEMENTASI METODE NAÏVE BAYES CLASSIFIER DAN
ECONOMYC ORDER QUANTITY (EOQ) PADA FITUR PENENTUAN
KUALITAS BIJI JAGUNG DALAM SISTEM INFORMASI MANAJEMEN
PERSEDIAAN BIJI JAGUNG DI UD. SUNAN**

SKRIPSI

Oleh

REGA CHANDIKA PRATAMA

NIM 132410101032

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2020



**IMPLEMENTASI METODE *NAÏVE BAYES CLASSIFIER* DAN
ECONOMYC ORDER QUANTITY (EOQ) PADA FITUR PENENTUAN
KUALITAS BIJI JAGUNG DALAM SISTEM INFORMASI MANAJEMEN
PERSEDIAAN BIJI JAGUNG DI UD. SUNAN JEMBER**

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan pendidikan Sarjana (S1) Program Studi Sistem Informasi
Fakultas Ilmu Komputer Universitas Jember dan mendapat gelar Sarjana
Komputer

Oleh

REGA CHANDIKA PRATAMA

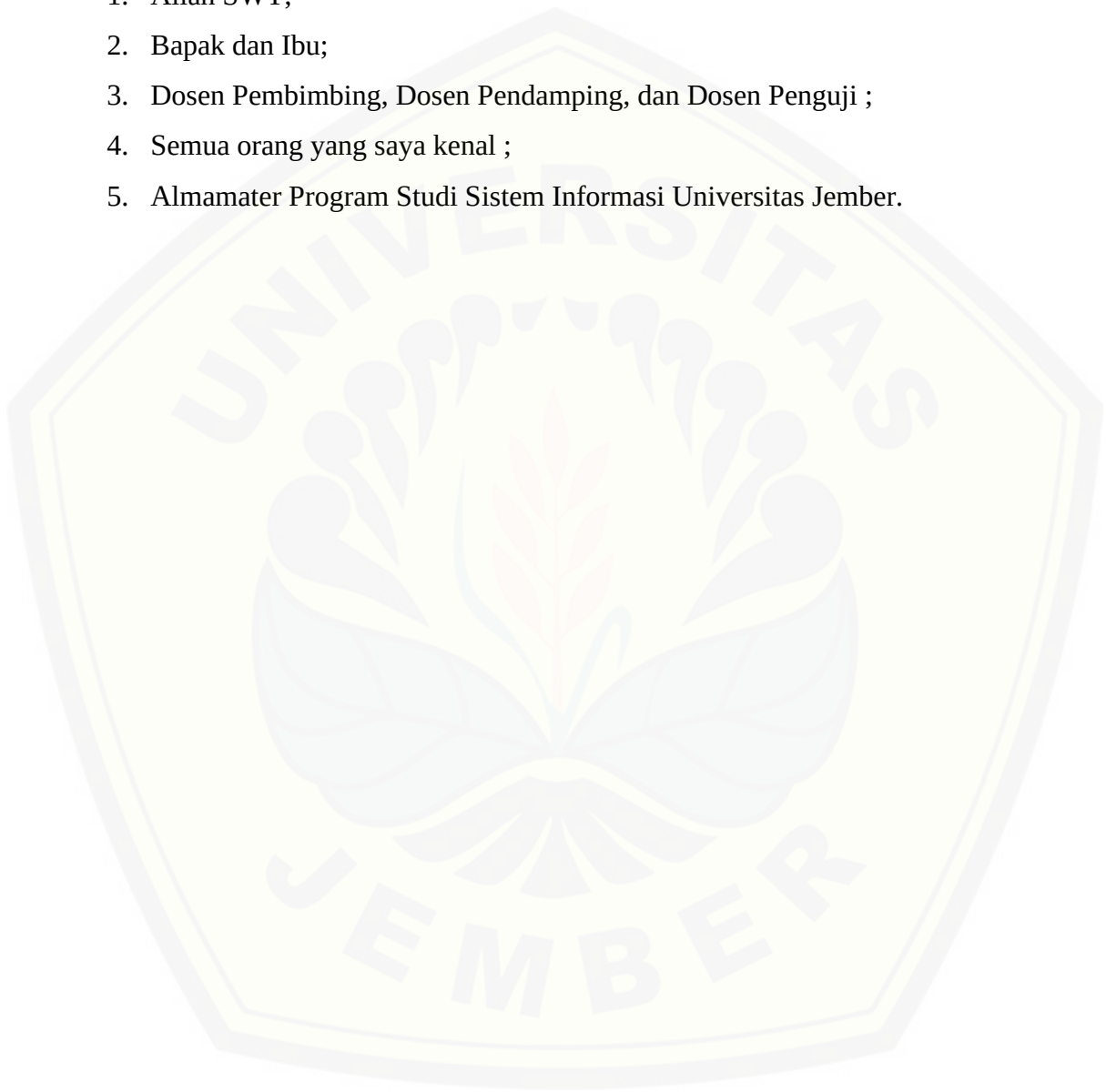
NIM 132410101032

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2020**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT;
2. Bapak dan Ibu;
3. Dosen Pembimbing, Dosen Pendamping, dan Dosen Penguji ;
4. Semua orang yang saya kenal ;
5. Almamater Program Studi Sistem Informasi Universitas Jember.



MOTO

“Waktu terus berjalan tanpa memberi tahu, termasuk waktu antara fajar dan senja. Dia tidak akan pernah bisa kembali lagi. Oleh karena itu, pergunakanlah dia dengan bijaksana”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Rega Chandika Pratama

NIM : 132410101032

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi Metode *Naïve Bayes Classifier* dan *Economyc Order Quantity* (EOQ) Pada Fitur Penentuan Kualitas Biji Jagung Dalam Sistem Informasi Manajemen Persediaan Biji Jagung di UD.Sunan”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 8 Januari 2020

Yang menyatakan,

Rega Chandika Pratama

NIM 132410101032

SKRIPSI

**Implementasi Metode *Naïve Bayes Classifier* dan *Economyc Order Quantity*
(EOQ) Pada Fitur Penentuan Kualitas Biji Jagung Dalam Sistem Informasi
Manajemen Persediaan Biji Jagung di UD.SUNAN**

Oleh :

Rega Chandika Pratama

NIM 132410101032

Pembimbing

Dosen Pembimbing Utama : Prof. Dr. Saiful Bukhori, ST., M.Kom

Dosen Pembimbing Pendamping : Fajrin Nurman Arifin, ST., M,Eng

PENGESAHAN PEMBIMBING

Skripsi berjudul "*Implementasi Metode Naive Bayes Classifier dan Economyc Order Quantity (EOQ) Pada Fitur Penentuan Kualitas Biji Jagung Dalam Sistem Informasi Manajemen Persediaan Biji Jagung di UD.Sunan*", telah diuji dan disahkan pada:

hari, tanggal : Jum'at, 26 Juni 2020

tempat : Fakultas Ilmu Komputer, Universitas Jember

Disetujui oleh:

Pembimbing I,



Prof. Dr. Saiful Bukhori, ST., M.Kom
NIP. 196811131994121001

Pembimbing II,



Fajrin Nurman Arifin, ST., M.Eng
NIP. 198511282015041002

PENGESAHAN PENGUJI

Skripsi berjudul “Implementasi *Naïve Bayes Classifier* dan *Economye Order Quantity (EOQ)* Pada Fitur Penentuan Kualitas Biji Jagung Dalam Sistem Informasi Manajemen Persediaan Biji Jagung di UD.Sunan”, telah diuji dan disahkan pada:

hari, tanggal : Jum'at, 26 Juni 2020

tempat : Fakultas Ilmu Komputer, Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,



Prof. Drs. Slamin, M.Comp.Sc., Ph.D

NIP. 196704201992011001



Diksy Media Firmansyah, S.Kom., M.Kom

NIP. 760016853

Mengesahkan

Penjabat Dekan



Prof. Dr. Saiful Bukhori, S.T., M.Kom

NIP. 196811131994121001

RINGKASAN

Implementasi Metode *Naïve Bayes Classifier* dan *Economic Order Quantity* (EOQ) Pada Fitur Penentuan Kualitas Biji Jagung Dalam Sistem Informasi Manajemen Persediaan Biji Jagung di UD. Sunan Jember, Rega Chandika Pratama, 132410101032; 2020, 121 halaman; Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

Jagung (*Zea mays*) merupakan tanaman penting di Indonesia, selain sebagai tanaman bahan pangan pokok pengganti beras dalam upaya diversifikasi pangan, jagung juga merupakan pakan ternak. Jenis-jenis jagung yang dikembangkan di Indonesia yaitu jagung hibrida, jagung komposit dan Jagung transgenik. Dari beberapa jenis jagung memiliki pasar yang berbeda diantaranya sebagai bahan konsumtif. Jagung merupakan salah satu tanaman pangan yang memiliki nilai ekonomis serta mempunyai peluang untuk dikembangkan. Permintaan jagung semakin meningkat sejalan dengan beragamnya kegunaan jagung. Salah satu wilayah penghasil jagung di Jawa Timur adalah Kabupaten Jember. Tingginya permintaan jagung harus sejalan dengan pengadaan kualitas biji yang baik seperti yang dilakukan di UD.Sunan.

Permintaan yang besar membuat perusahaan sebisa mungkin terus memberikan produk kualitas terbaik yang didapatkan dari petani. Pengujian harus dilakukan untuk menentukan klasifikasi biji jagung yang masuk. Kendala yang sering terjadi adalah lamanya pengujian membuat stok biji jagung menumpuk dan sulit untuk dianalisa secara cepat dan tepat. Solusi dari kendala diatas adalah dengan membuat sistem penunjang pengambilan keputusan yang menerapkan algoritma dari metode yang telah terbukti keakuratannya.

Penelitian ini mengimplementasikan metode algoritma *Naive Bayes Classifier* dalam fitur klasifikasi kualitas biji jagung dan metode *Economic Order Quantity* dalam pengembangan sistem informasi manajemen stok biji jagung di UD.Sunan.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi Metode *Naïve Bayes Classifier* dan *Economy Order Quantity (EOQ)* Pada Fitur Penentuan Kualitas Biji Jagung Dalam Sistem Informasi Manajemen Persediaan Biji Jagung di UD.Sunan”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Prof. Dr. Saiful Bukhori, ST., M.Kom selaku Dosen Pembimbing Utama yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
2. Fajrin Nurman Arifin, ST., M,Eng selaku Dosen Pembimbing II yang telah memberikan saran dan arahan dalam penulisan skripsi ini;
3. Kepada kedua orang tua dan seluruh keluarga;
4. Semua pihak yang tidak dapat disebutkan satu persatu.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 8 Januari 2020

Penulis

DAFTAR ISI

SKRIPSI	i
PERSEMBAHAN	ii
MOTO	iii
PERNYATAAN	iv
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI	vii
RINGKASAN	viii
PRAKATA	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	4
1.5 Sitematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1 Persyaratan Kualitas Biji Jagung	6
2.2 <i>Naive Bayes Classifier</i>	7
2.3 <i>Economyc Order Quanttity (EOQ)</i>	9
2.4 Re – Order Point	11
2.5 Hubungan EOQ dengan Re – Order Point	12

2.6	UD.Sunan Jember	13
2.7	<i>Extreme Programming (XP)</i>	13
BAB III	METODOLOGI PENELITIAN	18
3.1	Jenis Penelitian	18
3.2	Tempat dan Waktu Penelitian	18
3.3	Tahapan Penelitian	18
BAB IV	PENGEMBANGAN SISTEM	22
4.1.	Planning	22
4.2.	Design	25
4.3	<i>Implementation</i>	31
4.4	<i>Testing System</i>	38
BAB V	HASIL DAN PEMBAHASAN	40
5.1	Hasil Implementasi Kode Program Pada Fitur Penentuan Kualitas Biji Jagung Dalam Sistem Informasi Manajemen Persediaan Biji Jagung	40
5.2	Implementasi Penghitungan manual algoritma <i>Naive Bayes</i>	47
5.3	Penghitungan <i>error</i>	52
5.4	Penerapan ROP dalam sistem manajemen biji jagung	54
BAB VI	PENUTUP	56
6.1	Kesimpulan	56
6.2	Saran	56
DAFTAR PUSTAKA	58
LAMPIRAN	59

DAFTAR TABEL

BAB II

Tabel 2.1 Persyaratan mutu jagung berdasarkan SNI 3920:2013 6

BAB IV

Tabel 4. 1 Data Kriteria 23

Tabel 4. 2 Data Jagung 23

Tabel 4. 3 Fitur pada sistem 24

Tabel 4. 4 Definisi aktor sistem 26

Tabel 4. 5 Definisi usecase..... 26

Tabel 4. 6 Definisi CRC Card 28

Tabel 4. 7 Testing design 30

Tabel 4. 8 Tabel *Blackbox Tesring* 38

BAB V

Tabel 5.1 Data Training 47

Tabel 5.2 Probabilitas Kriteria Kadar Air 50

Tabel 5.3 Probabilitas Kriteria Butir Rusak 50

Tabel 5.4 Probabilitas Kriteria Warna Lain 50

Tabel 5.5 Probabilitas Kriteria Butir Pecah 51

Tabel 5.6 Probabilitas Kriteria Kadar Kotoran 51

Tabel 5.7 Probabilitas Kriteria Alfatoxin 51

Tabel 5.8 Probabilitas Grade 49

Tabel 5.9 Uji Kasus 51

Tabel 5.10 Hasil Penghitungan 52

Tabel 5.11 Hasil Uji Data Training 52

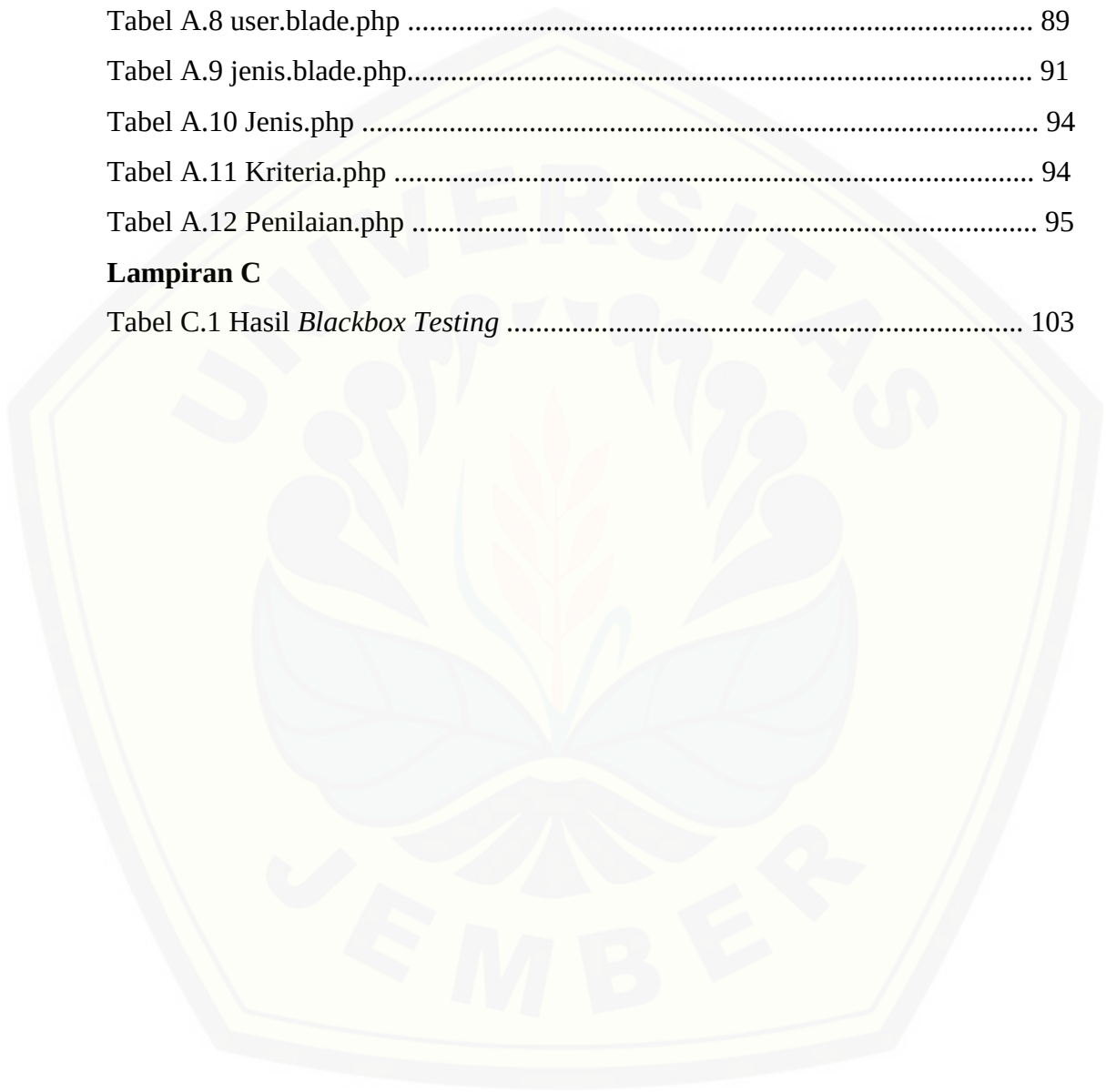
Lampiran A

Tabel A.1 hasilKriteria.php 59

Tabel A.2 kriteria.blade.php..... 60

Tabel A.3 penjualan.blade.php..... 65

Tabel A.4 persediaan.blade.php	71
Tabel A.5 range.blade.php	78
Tabel A.6 rop.blade.php	79
Tabel A.7 subkriteria.blade.php	85
Tabel A.8 user.blade.php	89
Tabel A.9 jenis.blade.php.....	91
Tabel A.10 Jenis.php	94
Tabel A.11 Kriteria.php	94
Tabel A.12 Penilaian.php	95
Lampiran C	
Tabel C.1 Hasil <i>Blackbox Testing</i>	103



DAFTAR GAMBAR

BAB II

Gambar 2.1 Life Cycle XP (Aji, 2015) 14

Gambar 2.2 Grafik EOQ 9

BAB IV

Gambar 4.1 *Entity Relationship Diagram*.....

Gambar 4. 1 Requirment Sistem 22

Gambar 4. 2 Usecase Sistem 27

Gambar 4.3 CRC card sistem 29

Gambar 4.4 Class Diagram Sistem 29

Gambar 4. 5 Entity Relationship Diagram 31

Gambar 4. 6 Kode program Hitung-grade 35

Gambar 4.7 Tampilan hitung grade 35

Gambar 4. 8 Kode pogram Data Jagung 36

Gambar 4.9 Tampilan data jagung 36

Gambar 4. 10 Kode program Hitung ROP 37

Gambar 4.11 Tampilan data ROP 38

BAB V

Gambar 5.1 Tampilan Login 40

Gambar 5.2 Tampilan Admin Tambah Stoke Grade 41

Gambar 5.3 Tampilan Admin Hitung Grade..... 41

Gambar 5.4 Tampilan Admin Hasil Hitung Grade 42

Gambar 5.5 Tampilan Admin Lihat Data Persediaan 42

Gambar 5.6 Tampilan Admin Melihat Data Stok 43

Gambar 5.7 Tampilan Admin Melihat Data ROP 43

Gambar 5.8 Tampilan Admin Melihat Data Training 44

Gambar 5.9 Tampilan Admin Melihat Data Kriteria Jagung 44

Gambar 5.10 Tampilan data penjualan 45

Gambar 5.11 Tampilan Manajer Data Persediaan 45

Gambar 5.12 Tampilan Manajer Input Data ROP 46

Gambar 5.13 Tampilan Manajer Lihat Data ROP	46
Gambar 5.14 Tampilan Pembeli Input Data Pembelian	47
Gambar 5.15 Hasil Penghitungan <i>grade</i>	52

Lampiran

Gambar B.1 Tampilan Login	96
Gambar B.2 Tampilan Admin Tambah Stok Grade.....	96
Gambar B.3 Tampilan Admin Hitung Grade	96
Gambar B.4 Tampilan Admin Lihat Penentuan Grade	97
Gambar B.5 Tampilan Admin Lihat Data Persediaan	97
Gambar B.6 Tampilan Admin Lihat Data Stok	97
Gambar B.7 Tampilan Admin Lihat Data ROP	98
Gambar B.8 Tampilan Admin Lihat Data Training	98
Gambar B.9 Tampilan Admin Lihat Data Kriteria	98
Gambar B.10 Tampilan Admin Validasi Data Penjualan	99
Gambar B.11 Tampilan Manager Lihat Data Persediaan	99
Gambar B.12 Tampilan Manager Lihat Data Jagung	99
Gambar B.13 Tampilan Manager Tambah Data Jagung	100
Gambar B.14 Tampilan Manager Lihat Data ROP	100
Gambar B.15 Tampilan Manager Lihat Data Training	100
Gambar B.16 Tampilan Manager Edit Data Kriteria	101
Gambar B.17 Tampilan Manager Tambah Data Kriteria.....	101
Gambar B.18 Tampilan Manager Lihat Data Kriteria	101
Gambar B.19 Tampilan Pembeli Tambah Pembelian	102
Gambar B.20 Tampilan Pembeli Lihat Validasi Pembelian	102
Gambar B.21 Tampilan Manager Tambah Data ROP	102
Gambar B.22 Tampilan Lihat Data Penjualan	103

DAFTAR LAMPIRAN

A. Kode Program	59
B. Tampilan Sistem 96



BAB I PENDAHULUAN

Bab ini menjelaskan hal-hal yang menjadi dasar dalam penelitian dan implementasi metode *Naïve Bayes Classifier* dan *Economic Order Quantity* (EOQ) pada fitur penentuan kualitas biji jagung dalam membangun Sistem Informasi Manajemen Persediaan Biji Jagung di UD.Sunan. Adapun yang akan dijelaskan antara lain adalah latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, serta sistematika penulisan.

1.1 Latar Belakang

Jagung (*Zea mays*) merupakan tanaman serealia penting di Indonesia, selain sebagai tanaman bahan pangan pokok pengganti beras dalam upaya diversifikasi pangan, jagung juga merupakan pakan ternak. Jenis-jenis jagung yang dikembangkan di Indonesia yaitu jagung hibrida, jagung komposit dan Jagung transgenik. Jagung hibrida merupakan keturunan pertama dari persilangan dua tetua yang memiliki karakter/sifat yang unggul. Shull (1908) merupakan orang yang pertama kali menemukan bahwa hasil persilangan sendiri tanaman jagung mengakibatkan terjadinya depresi *inbreeding*, dan persilangan dua tetua yang *homozigot* menghasilkan F1 yang sangat *vigor*. Jagung komposit atau biasanya disebut jagung lokal adalah jenis jagung yang pada jaman dulu ditanam petani setempat yang menyerbuk sendiri tanpa bantuan manusia. Jagung transgenik merupakan jenis jagung hasil dari penyisipan gen seperti gen tahan penyakit, gen tahan hama maupun gen tahan obat kimia yang berasal dari makhluk hidup atau non-makhluk hidup sehingga tanaman itu menjadi tanaman super.

Jagung merupakan salah satu tanaman pangan yang memiliki nilai ekonomis serta mempunyai peluang untuk dikembangkan. Permintaan jagung semakin meningkat sejalan dengan beragamnya kegunaan jagung. Salah satu wilayah penghasil jagung di Jawa Timur adalah Kabupaten Jember. Tingginya permintaan jagung harus sejalan dengan pengadaan kualitas biji yang baik seperti yang dilakukan di UD.Sunan.

UD.Sunan merupakan salah satu perusahaan yang bergerak dalam bidang agribisnis (Pengolahan Jagung) dan merupakan salah satu unit bisnis yang terletak

di Kabupaten Jember. Mengusung visi untuk menjadi perusahaan yang menyediakan biji jagung bahan konsumsi dengan kualitas dan keuntungan tertinggi dan misi mengembangkan biji jagung terbaik dengan kualitas unggul, meningkatkan kepuasan pelanggan dengan menyediakan produk berkualitas tinggi dan terjangkau, mengembangkan sumber daya manusia dengan meningkatkan keterampilan dan kemampuan kerja mereka dan membantu pemerintah dan lembaga swasta dalam program produksi dan pengembangan jagung konsumtif mereka. Permintaan yang besar membuat perusahaan sebisa mungkin terus memberikan produk kualitas terbaik yang didapatkan dari petani. Pengujian harus dilakukan untuk menentukan klasifikasi biji jagung yang masuk. Kendala yang sering terjadi adalah lamanya pengujian membuat stok biji jagung menumpuk dan sulit untuk dianalisa secara cepat dan tepat. Solusi dari kendala diatas adalah dengan membuat sistem penunjang pengambilan keputusan yang menerapkan algoritma dari metode yang telah terbukti keakuratannya.

Naive Bayes Classifier merupakan sebuah metode klasifikasi menggunakan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. *Naive Bayes Classifier* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dr *Naive Bayes Classifier* ini adalah asumsi yg sangat kuat (naïf) akan independensi dari masing-masing kondisi / kejadian. *Naive Bayes Classifier* bekerja sangat baik dibanding dengan model classifier lainnya karena memiliki tingkat akurasi yang tinggi. Keuntungan penggunaan adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*data training*) yang kecil untuk menentukan estimasi parameter yg diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai variabel independent, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians. Penelitian terdahulu yang dilakukan oleh Cahya dsn dengan judul “Implementasi Metode *Naive Bayes Classifier* (NBC) untuk Memprediksi Besarnya Penggunaan Listrik Rumah Tangga” Penerapan *Naive Bayes* menghasilkan prediksi dengan tingkat ke akuratan sebesar 75%.

Economic Order Quantity (EOQ) merupakan salah satu metode manajemen persediaan yang cukup banyak dikenal. Metode ini dapat digunakan baik untuk barang yang diproduksi sendiri. Model EOQ biasa digunakan untuk menentukan kuantitas pesanan persediaan yang meminimumkan biaya langsung, penyimpanan persediaan dan biaya kebalikannya pemesanan persediaan. Dalam model EOQ didapatkan nilai *Re-Order Point* untuk memajemen persediaan stok agar lebih efisien. Penelitian terdahulu yang dilakukan oleh Bagoes Setya Wardhana dengan judul “Rancang Bangun Sistem Informasi Manajemen Stok Perusahaan Susu Bestcow Menggunakan Metode *Economic Order Quantity* (EOQ)” dengan proses perhitungan yang baik dan berhasil.

Penelitian ini mengimplementasikan metode algoritma *Naive Bayes Classifier* dalam fitur klasifikasi kualitas biji jagung dan metode *Economic Order Quantity* dalam pengembangan sistem informasi manajemen stok biji jagung di UD.Sunan.

1.2 Rumusan Masalah

Berdasarkan uraian di atas, permasalahan yang muncul adalah:

- 1) Bagaimana cara membangun sistem informasi manajemen persediaan biji jagung dengan fitur penentuan kualitas biji jagung di UD.Sunan menggunakan metode *Economic Order Quantity*?
- 2) Bagaimana implementasi metode *Naive Bayes Classifier* pada fitur klasifikasi kualitas biji jagung di sistem informasi manajemen persediaan biji jagung menggunakan metode *Economic Order Quantity* di UD.Sunan?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini antara lain :

- 1) Membangun sistem informasi manajemen persediaan biji jagung dengan fitur penentuan kualitas biji jagung di UD.Sunan Indonesia menggunakan metode *Economic Order Quantity*.
- 2) Mengetahui hasil implementasi metode *Naive Bayes Classifier* pada fitur klasifikasi kualitas biji jagung di sistem informasi manajemen persediaan biji jagung di UD.Sunan.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

- 1) Sistem informasi ini hanya sebagai alat bantu bagi perusahaan dalam menentukan pembelian jagung terbaik terhadap petani berdasarkan kriteria yang ditentukan oleh perusahaan dan keputusan akhir tetap berada di tangan perusahaan serta membantu mengatur persediaan barang dan manajemen biji jagung pada perusahaan.
- 2) Penentuan hasil kualitas jagung mengacu pada SNI 3920:2013 tentang jagung.
- 3) Hasil dari sistem klasifikasi ini adalah *grade* biji jagung yang sesuai untuk mengisi persediaan oleh perusahaan mulai dari *grade* yang bagus sampai terendah.

1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

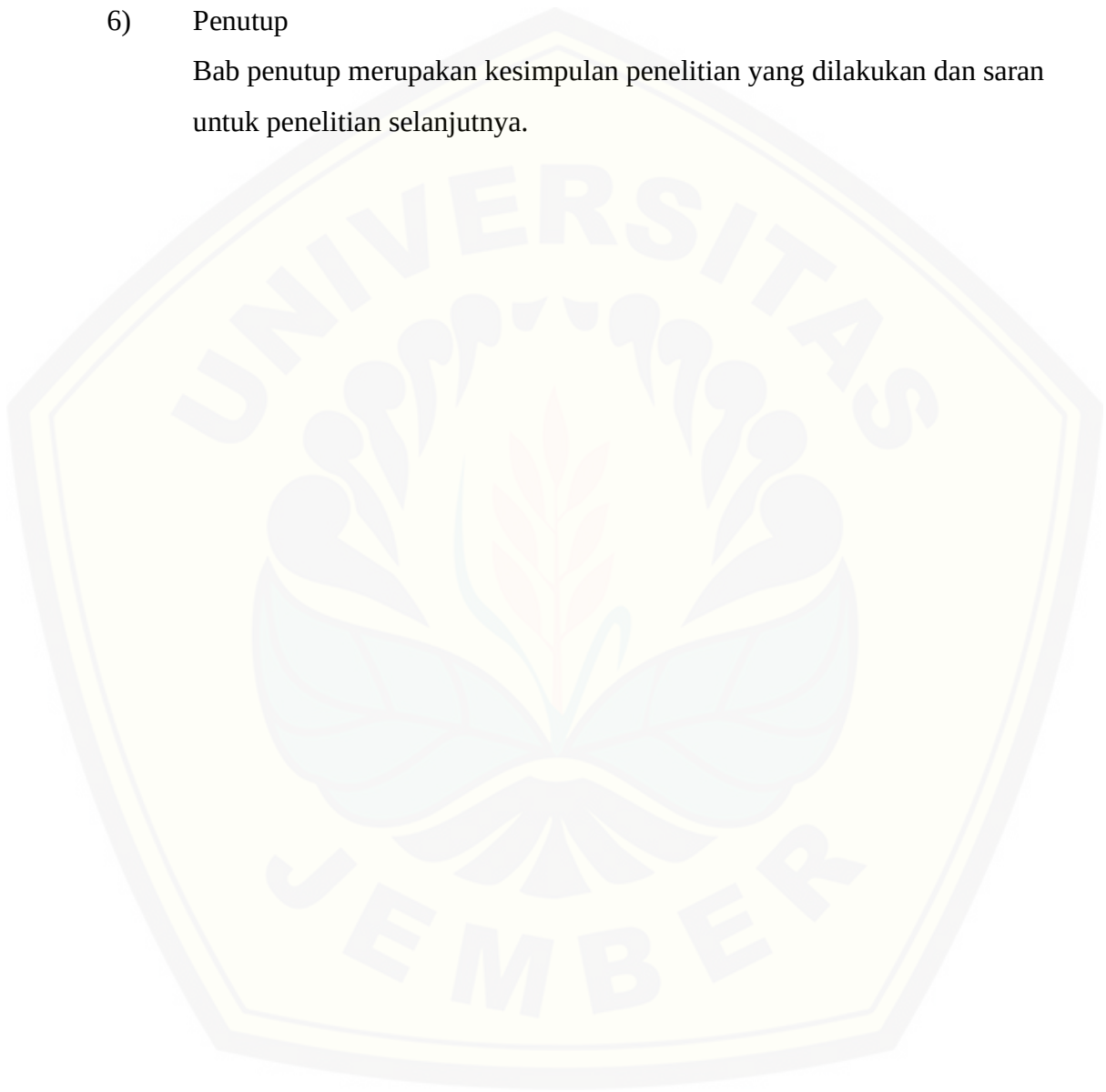
- 1) **Pendahuluan**
Bab pendahuluan merupakan langkah awal dari penyusunan tugas akhir yang membahas latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan.
- 2) **Tinjauan Pustaka**
Bab tinjauan pustaka merupakan bab yang menjelaskan teori-teori yang melandasi penelitian, tinjauan pustaka, dan studi terdahulu yang menjadi acuan dalam penelitian.
- 3) **Metodologi Penelitian**
Bab metodologi penelitian jenis penelitian dan tahapan penelitian sesuai dengan model pengembangan sistem yang dipakai.
- 4) **Pengembangan Sistem**
Bab pengembangan sistem adalah bab yang berisi proses pengembangan sistem yang dibangun dalam penelitian. Adapun tahapan-tahapan pengembangan sistem ini didasarkan pada metodologi penelitian yang telah ditulis pada bab sebelumnya.

5) Hasil dan Pembahasan

Bab hasil dan pembahasan adalah bab yang menjelaskan secara rinci pemecahan masalah yang ada pada rumusan masalah yang telah dijelaskan pada bab pendahuluan.

6) Penutup

Bab penutup merupakan kesimpulan penelitian yang dilakukan dan saran untuk penelitian selanjutnya.



BAB II TINJAUAN PUSTAKA

Bab ini menguraikan teori serta konsep yang menjadi kerangka pemikiran dalam penelitian. Teori dan konsep tersebut didapatkan dari penelitian terdahulu serta kajian pustaka terkait penelitian.

2.1 Persyaratan Kualitas Biji Jagung

Jagung (*Zea mays*) merupakan tanaman penting di Indonesia, selain sebagai tanaman bahan pangan pokok pengganti beras dalam upaya diversifikasi pangan, jagung juga merupakan pakan ternak. Jenis-jenis jagung yang dikembangkan di Indonesia yaitu jagung hibrida, jagung komposit dan Jagung transgenik. Dari beberapa jenis jagung memiliki pasar yang berbeda diantaranya sebagai bahan konsumtif.

Jagung bahan konsumtif adalah jagung pipilan hasil tanaman jagung berupa biji kering yang sudah dilepaskan dan dibersihkan dari tongkolnya. Berdasarkan Standart Nasional Indonesia (SNI 3920:2013) tentang persyaratan mutu jagung yang dipasarkan untuk konsumsi dan bahan baku industri pangan. Jagung digolongkan ke dalam 4 kelas mutu seperti pada tabel 2.1.

Tabel 2.1 Persyaratan mutu jagung berdasarkan SNI 3920:2013

No.	Komponen Mutu	Satuan	Persyaratan Mutu			
			Mutu I	Mutu II	Mutu III	Mutu IV
1	Kadar Air	%	14	14	14	15
2	Butir Rusak	%	2	4	6	8
3	Butir Warna Lain	%	1	3	7	10
4	Butir Pecah	%	1	2	3	3
5	Kadar Kotoran	%	1	1	2	2
6	Alfatoxin	%	5	5	15	20

1. Kadar Air

Jumlah kandungan air dalam jagung dinyatakan dalam kandungan presentase berat bersih. Kadar air dalam jagung merupakan faktor mutu utama, menentukan kondisi kritis dimana mikroorganisme dapat tumbuh dan merusak jagung. Dalam

SNI Jagung Kadar maksimum 14 % untuk mutu I, II, III dan mutu IV maksimal 15%.

2. Butir Rusak

Butir rusak pada jagung adalah butir jagung yang utuh maupun pecah yang mengalami kerusakan mekanis, biologis, fisik maupun enzimatis. Kerusakan mekanis maupun fisik dapat berupa pecahan $\leq 0,6$ bagian biji jagung utuh. Kerusakan biologis dapat terjadi karena mikroorganisme atau serangga.

3. Butir Warna Lain

Jagung memiliki warna lain disebabkan oleh faktor genetis atau tercampurnya varietas lain. Butir warna lain pada jagung juga dapat disebabkan oleh kurang meratanya tingkat kematangan jagung pasca panen sehingga pada proses pengeringan bisa menjadi pucat.

4. Butir Pecah

Butir pecah pada biji jagung dapat terjadi karena perlakuan selama pasca panen baik secara fisik dan mekanis, hal tersebut juga dapat dipengaruhi oleh interaksi faktor genetis biologis dan enzimatis. Jagung pecah dikategorikan cacat dikarenakan tidak dapat dikategorikan sebagai jagung utuh.

5. Kadar Kotoran

Kadar kotoran pada jagung adalah kandungan benda-benda seperti tanah, batu/kerikil, sisa batang/tongkol dan juga benda lain selain jagung. Kadar kotoran menunjukkan kebersihan fisik pasca panen yang berpengaruh pada kualitas maupun kuantitas jagung.

2.2 *Naive Bayes Classifier*

Naive Bayes Classifier merupakan sebuah metode klasifikasi menggunakan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. *Naive Bayes Classifier* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dari *Naive Bayes Classifier* ini adalah asumsi yg sangat kuat (naïf) akan independensi dari masing-masing kondisi / kejadian.

Naive Bayes Classifier bekerja sangat baik dibanding dengan model *classifier* lainnya karena memiliki tingkat akurasi yang tinggi. Keuntungan penggunaan adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*data training*) yang kecil untuk menentukan estimasi parameter yg diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai variabel independent, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

Tahapan dari proses algoritma *Naive Bayes* adalah:

1. Membaca Data Training
2. Menghitung Jumlah dan Probabilitas, namun jika data numerik maka
 - a. Menghitung nilai *mean* dan Standar Deviasi dari masing-masing parameter yang merupakan numerik. Adapun persamaan untuk mencari nilai rata-rata hitung (*mean*) adalah seperti dalam persamaan 1 berikut ini:

$$\mu = \frac{\sum_{i=1}^n X_i}{n} \dots\dots\dots(1)$$

Keterangan

- μ : nilai rata-rata hitung (mean)
- x_i : nilai x ke-i
- n : jumlah sampel

Sedangkan persamaan untuk menghitung nilai Nilai Simpangan Baku (Standar Deviasi) dirumuskan sebagai berikut :

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \mu)^2}{n-1}} \dots\dots\dots(\text{Persamaan 2})$$

Keterangan

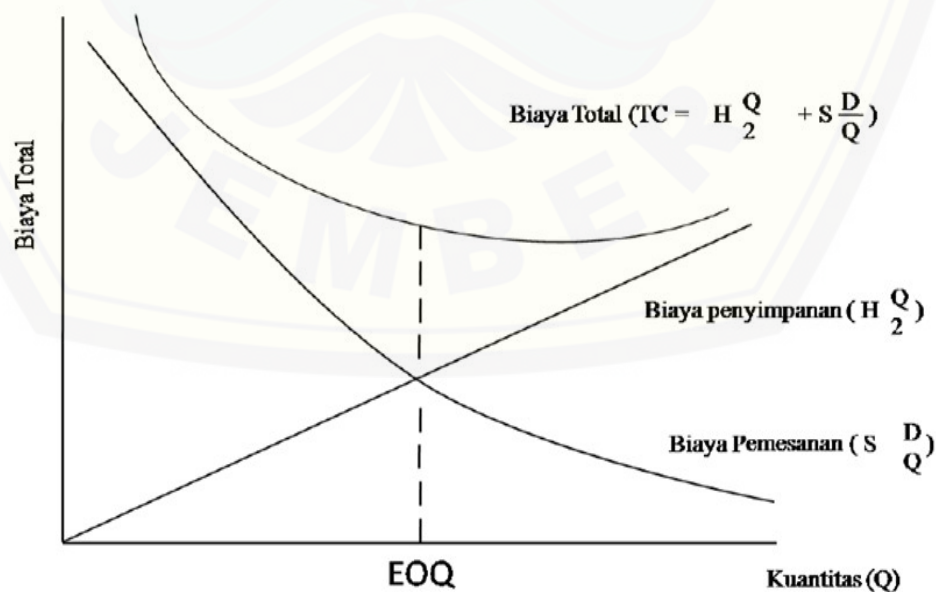
- σ :standar deviasi
- x_i : nilai x ke-i

- μ : nilai rata-rata hitung (mean)
 - n : jumlah sampel
- b. Menghitung nilai probabilitas dengan cara menghitung jumlah data yang sesuai dari kategori yang sama dibagi dengan jumlah data pada kategori tersebut.
 3. Mendapatkan nilai dalam tabel mean, Standar Deviasi dan Probabilitas
 4. Menghitung nilai akhir dengan mengalikan probabilitas kriteria dengan probabilitas grade
 5. Menghasilkan Solusi

2.3 *Economic Order Quantity (EOQ)*

EOQ merupakan salah satu metode manajemen persediaan. Metode ini dapat digunakan baik untuk barang yang diproduksi sendiri. Model EOQ biasa digunakan untuk menentukan kuantitas pesanan persediaan yang meminimumkan biaya langsung, penyimpanan persediaan dan biaya kebalikannya pemesanan persediaan.

Beberapa variabel yang terdapat pada metode EOQ yang dijelaskan oleh Jose L. Gonzales and Daniel Gonzales (2012) digambarkan dalam sebuah grafik sebagai berikut :



Gambar 2.1 Grafik EOQ

Definisi EOQ menurut Sutrisno (2001) EOQ adalah jumlah kuantitas barang yang dapat diperoleh dengan biaya minimal, atau sering dikatakan sebagai jumlah pembelian yang optimal. Ada beberapa faktor yang berpengaruh pada penentuan EOQ. Penghitungan EOQ dapat dilihat pada persamaan 5.

$$EOQ = \sqrt{\frac{2DS}{h}} \dots\dots\dots(\text{Persamaan 3})$$

Keterangan :

- EOQ = Kualitas Pembelian Optimal
- S = Biaya Pemesanan setiap kali pesan
- D = Penggunaan barang pertahun
- H = Biaya penyimpanan per unit

Demand selama periode perencanaan dapat di asumsikan linier kemudian output EOQ, yaitu persediaan (Q) atau jumlah barang yang dipesan menjadi pembagi (D) untuk menemukan berapa banyak N atau frekuensi pemesanan barang, dapat dilihat pada persamaan 4.

$$N = \frac{D}{EOQ} \dots\dots\dots(\text{Persamaan 4})$$

Kesimpulan yang dapat diambil dari definisi diatas adalah bahwa EOQ merupakan suatu metode yang digunakan untuk mengoptimalkan pembelian barang yang dapat menekan biaya-biaya persediaan sehingga efisiensi persediaan barang dalam perusahaan dapat berjalan dengan baik.

Jarak tiap kali pesan dalam *Economic Order Quantity* (EOQ) adalah menentukan berapa hari sekali perusahaan melakukan *restock* dalam frekuensi yang telah ditentukan. Dapat dilihat dalam persamaan 5.

$$\text{Jarak tiap kali pesan} = \frac{\text{Jumlah Tahun kerja}}{\text{Frekuensi}} \dots\dots\dots \text{Persamaan (5)}$$

Kebutuhan barang perhari dalam *Economic Order Quantity* (EOQ) adalah menentukan berapa banyak barang keluar dalam sehari dalam perusahaan. Dapat dilihat dalam persamaan 6.

$$\text{Kebutuhan barang perhari} = \frac{\text{Kebutuhan Bahan Baku per Periode}}{\text{Frekuensi}} \dots\dots \text{Persamaan (6)}$$

Kebutuhan bahan baku selama *Lead Time* dalam *Economic Order Quantity* (EOQ) adalah kebutuhan barang yang perlu disiapkan atau yang harus ada selama pengiriman barang yang dipesan belum datang. Dapat dilihat dalam persamaan 7.

$$\text{Penggunaan Barang selama Lead Time} = \text{Kebutuhan Barang perhari} \times \text{Lead Time} \dots\dots \text{Persamaan (7)}$$

Penelitian “Rancang Bangun Sistem Informasi Manajemen Stok Barang Berbasis Web Toko Besi Bangunan Barokah Material Menggunakan Metode EOQ” oleh Sadam Husen (2015) menunjukkan pengujian pada semen didapat hasil frekuensi pembelian yang optimal dan lebih efektif dari sebelumnya, barang yang diolah juga bersifat konstan selalu ada. Perbedaan dengan penelitian selanjutnya adalah bahan yang digunakan bersifat musiman sehingga tidak konstan dan terdapat pengujian kualitas terlebih dahulu sebelum barang dibeli.

2.4 Re – Order Point

Re – Order Point adalah titik/batas jumlah pemesanan kembali. ROP berguna untuk mengetahui kapan suatu perusahaan untuk melakukan pemesanan kembali suatu barang. Apabila jumlah persediaan yang terdapat pada stok barang berkurang terus menerus maka terjadilah ROP atau pemesanan kembali dengan jumlah yang sesuai batas minimal tingkat persediaan agar tidak terjadi kekurangan persediaan.

Re – Order Point dalam suatu perusahaan sangat penting agar stok yang tersedia tidak mengalami kekurangan selama suatu perusahaan beroperasi, karena

re – order point berarti memesan kembali. Suad Husnan, (2001) mengatakan, “*re – order point* adalah saat yang tepat dimana persediaan harus dilakukan kembali”.

Jumlah pemesanan kembali dalam *Re – Order Point* (ROP) dapat ditentukan oleh kebutuhan stok saat masa tenggang yang ditambahkan dengan stok aman (*safety stock*) dalam suatu perusahaan yang biasanya mengacu pada tingkat probabilitas atau kemungkinan terjadinya kekurangan stok selama masa tenggang (*lead time*).

Dalam manajemen persediaan agar pelayanan dapat melaksanakan siklus persediaan yang baik diperlukan pengolahan data kondisi permintaan dan stok aman yang sedang terjadi, karena semakin besar angka tingkat permintaan dan masa tenggang maka jumlah *safety stock* harus lebih banyak, agar dapat memenuhi tingkat permintaan pelanggan yang fluktuatif.

2.5 Hubungan EOQ dengan Re – Order Point

Hubungan *Economic Order Quantity* (EOQ) dengan *Re – Order Point* (ROP) dapat ditentukan menggunakan persamaan :

$$ROP = \text{Safety Stock} + \text{Penggunaan Bahan Baku Selama Lead Time} \dots\dots$$

Persamaan (8)

Dalam persamaan diatas dapat diketahui suatu perusahaan harus melakukan pemesanan kembali barang – barang persediaan pada saat jumlah persediaan dapat menjamin persediaan aman atau *safety stock* dan barang yang dibutuhkan disaat waktu tunggu pemesanan atau disebut sebagai *lead time*. Sedangkan *Economic Order Quantity* (EOQ) jumlah barang yang dipesan saat melakukan proses *re – order point* agar dapat mencapai biaya minimal / ekonomis.

Setelah proses *Re – Order Point* dilakukan maka suatu perusahaan dapat mengetahui kapan harus melakukan pemesanan kembali karena akan diketahui batas persediaan barang agar tidak terjadi keterlambatan atau kekosongan stok barang dalam suatu perusahaan maupun pabrik.

Re – Order Point dapat dilakukan atau dapat dihitung hanya untuk satu barang sedangkan apabila barang beragam dapat dihitung satu persatu. Hasil dari perhitungan *Re – Order Point* nantinya juga akan menghasilkan satu persatu nilai

dari tiap jenis barang. Hasil dari ROP yang nantinya digunakan untuk melakukan *Restock* setiap jenis barang.

2.6 UD.Sunan Jember

UD.Sunan Jember merupakan salah satu perusahaan yang bergerak dalam bidang agribisnis (Jagung dan Padi) dan merupakan salah satu unit bisnis yang terletak di pinggiran kota Jember tepatnya di Kelurahan Banjarsengon, didirikan pada tahun 2013 oleh Bapak Sunan selaku owner, yang memiliki tujuan untuk membantu masyarakat khususnya petani untuk menjaga kualitas harga agar sesuai dengan standart.

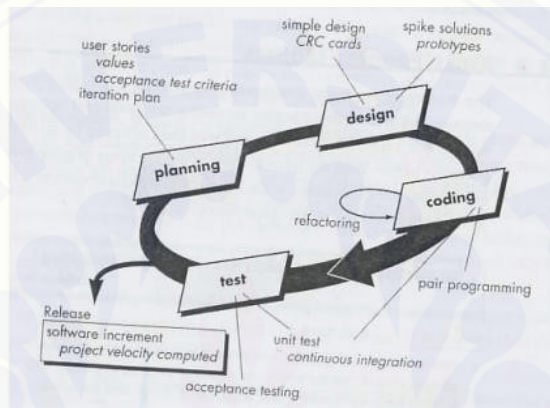
Mengusung visi untuk menjadi perusahaan yang menyediakan dan menampung biji jagung bahan pakan kualitas dan keuntungan tertinggi dan misi mengembangkan biji jagung terbaik dengan kualitas unggul, meningkatkan kepuasan pelanggan dengan menyediakan produk berkualitas tinggi dan terjangkau, mengembangkan sumber daya manusia dengan meningkatkan keterampilan dan kemampuan kerja mereka dan membantu pemerintah dan lembaga swasta dalam program produksi dan pengembangan biji jagung.

2.7 *Extreme Programming (XP)*

Extreme Programming (XP) adalah proses pengembangan perangkat lunak dirancang untuk diterapkan oleh pengembang perangkat lunak secara individu. XP menjaga prinsip dasar mengurangi jumlah dokumentasi dan upaya pemeliharaan. Proses pengembangan XP bersifat fleksibel dan responsif untuk perubahan. Metodologi ini ditujukan untuk meningkatkan kinerja *programmer* dan mempersingkat pengkodean dan waktu yang digunakan untuk mendukung sistem perangkat lunak. Metodologi XP didasarkan pada prinsip-prinsip berikut (Rizal, 2014):

- 1) XP memerlukan pendekatan disiplin, pengembang bertanggung jawab untuk mengikuti proses dan menerapkan praktik XP
- 2) Pengembang harus mengukur, melacak dan menganalisis pekerjaan sehari-hari mereka.

- 3) Pengembang harus belajar dari variasi kinerja mereka dan memperbaiki proses berdasarkan data proyek yang terkumpul
- 4) XP melibatkan pengujian terus menerus
- 5) Perbaikan cacat harus terjadi pada tahap pengembangan awal, bila biaya itu lebih rendah
- 6) Pengembang harus mencoba mengotomatisasi sebanyak mungkin hari mereka kerja



Gambar 2.2 Life Cycle XP (Aji, 2015)

Gambar 2.2 merupakan alur dari metode *Extreme Programming* dengan rincian sebagai berikut:

1) *Planning*

Pada tahap *planning* atau perencanaan, perencanaan fungsionalitas keseluruhan yang akan dikembangkan dalam sistem. Seperti rencana pengolahan data, rencana fitur, rencana alur sistem, rencana pengerjaan sistem.

2) *Design*

Tahap ini merupakan pembuatan desain dari sistem. Desain yang dipakai dalam PXP adalah *Usecase Diagram*, *CRC Card*, dan *Class Diagram*. *Usecase diagram* adalah *diagram* yang digunakan untuk menunjukkan hubungan antara fitur dan aktor yang dipakai pada sistem, sedangkan *CRC Card* adalah kartu yang menunjukkan hubungan antara *class* dengan fungsi dan tanggungjawabnya serta menunjukkan kolaborasinya dengan *class* lain, dan *Class diagram* adalah *diagram* yang menunjukkan hubungan antar *class*, dengan atribut dan *method* pada *coding*.

3) *Coding*

Pada tahapan ini desain sistem yang telah dibuat akan diimplementasikan secara langsung dalam sebuah sistem informasi berbasis *website*. Pengkodean program menggunakan bahasa pemrograman PHP dengan *tool Sublime Text* serta mengatur manajemen basis data menggunakan *tool Xampp*. Apabila kode selesai kemudian kode tersebut diuji dalam unit *testing*, apabila ada kesalahan maka dilakukan koreksi ulang atau *refactor* pada tahap dimana kesalahan tersebut bermula, apabila tidak ada kesalahan maka dilanjutkan ke unit selanjutnya.

4) *Test*

Pengujian yang dilakukan pada penelitian ini yaitu dengan menggunakan pengujian *black box*. *Black Box Testing* merupakan uji fungsionalitas apakah masih ada kekurangan atau sudah cukup sesuai. Apabila terdapat pengujian yang tidak sesuai maka akan dilakukan proses perbaikan.

Extreme programming memiliki 12 *core practices* (Ariaji et al., 2014) yaitu:

1) *Planning game*

Planning game merupakan *practice* yang digunakan untuk melakukan perencanaan dan melakukan prioritas terhadap fitur-fitur yang dituliskan pada tabel kebutuhan oleh *customer*.

2) *Small release*

Small releases adalah rilis yang dihasilkan untuk setiap iterasi sangat pendek dan dengan umpan balik terhadap perubahan dari *customer* juga sangat cepat.

3) *Metaphor*.

Metaphor adalah semacam *simple guidance* bagi proses pengembangan dari fase paling awal hingga terakhir.

4) *Simple design*.

Simple design merupakan rancangan yang sederhana untuk diproses pada setiap iterasi.

5) *Testing*.

Testing yang dilakukan adalah menguji performa sistem dan pengujian fitur.

6) *Refactoring*.

Refactoring adalah proses untuk memperbaiki *code* selain untuk menghindari berbagai redundansi yang mungkin terjadi.

7) *Pair programming*

Pair programming yaitu 2 orang *programmer* bekerja dalam 1 komputer. Model kerja ini dapat meningkatkan kualitas perangkat lunak tanpa mempengaruhi waktu pengerjaan. Dengan meningkatnya kualitas maka dapat meningkatkan penghematan proyek.

8) *Collective ownership*

Collective ownership adalah keadaan dimana semua anggota tim harus dapat menanggulangi semua hal yang berkaitan dengan proses pengembangan.

9) *Continuous integration*

Continuous integration bahwa proses pengembangan setiap hari bahkan setiap saat terdapat perubahan harus segera diintegrasikan.

10) *40-hour week*

40-hour week adalah jumlah jam kerja selama satu minggu. Hal ini sebenarnya tidak mutlak 40 jam, namun intinya adalah bahwa proses pengembangan tidak mengenal lembur, semua harus diselesaikan pada saat jam kerja. Semua masalah dan pekerjaan harus dioptimalkan pada waktu jam kerja tersebut.

11) *On-site customer*

On-site customer merupakan salah satu keunggulan bagi metodologi ini karena XP memerlukan satu orang dari pihak pemesan yang dibawa dalam proses pengembangan dari awal sampai berakhir. Kecepatan dalam mengatasi berbagai *requirements* yang mungkin berubah atau bertambah.

12) *Coding Standard*

Coding Standard adalah menstandarkan proses *coding* bagi *programmer* terutama karena hal ini berkaitan dengan *practice pair programming*

Extreme Programming memiliki nilai-nilai dalam pengembangannya, yang berguna sebagai pedomannya.

- 1) Kesederhaan, diterapkan dalam melakukan pengembangan sistem, terutama saat melakukan *coding*. Susunan coding yang sederhana dengan menggunakan method yang pendek namun efisien.
- 2) Komunikasi, XP sangat memperhatikan komunikasi antar setiap anggota tim sendiri maupun dengan pihak klien. Komunikasi yang baik antar tim akan mempermudah proses kerja proyek dimana setiap anggota tim bisa saling bantu dalam memecahkan sebuah masalah, terutama menghilangkan ego dari programmer yang cukup tinggi.
- 3) Umpan Balik, sangat diandalkan dalam proses model XP. Setiap anggota tim wajib memberikan setiap masalah kepada tim yang lainnya. Masukan, kritikan dan ide baru yang didapat bisa menjadi sebuah perbaikan maupun peningkatan dalam menyelesaikan setiap tugas.
- 4) Keberanian, sebuah proyek yang memiliki tenggat waktu akan menghadapi tekanan dari berbagai pihak, terutama klien. XP menerapkan nilai keberanian untuk tetap memiliki integritas pada setiap anggota tim. Dengan integritas tersebut anggota-anggota tim akan memiliki keyakinan dapat menyelesaikan proyek sesuai dengan yang diharapkan. Kepercayaan itulah yang dibangun sehingga semangat untuk segera dapat menyelesaikan dapat terus terbentuk.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang sekumpulan metode yang digunakan dalam penelitian pengembangan sistem. Pada bab ini juga dijelaskan mengenai langkah dan prosedur yang dilakukan dalam pengumpulan data atau informasi guna memecahkan permasalahan dalam penelitian.

3.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian kuantitatif. Jenis penelitian kuantitatif digunakan untuk mendapatkan kebutuhan sistem melalui wawancara dan studi literatur serta perhitungan manual, yang kemudian dianalisa secara kuantitatif untuk mendapatkan data biji jagung dan manajemen persediaan yang selanjutnya digunakan dalam proses pengembangan sistem.

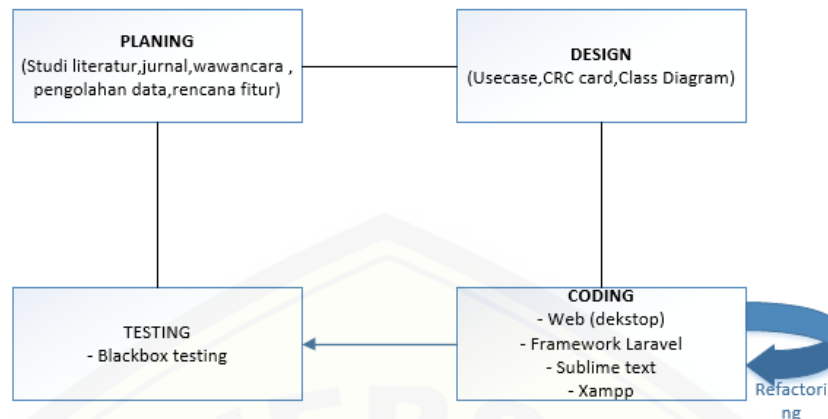
3.2 Tempat dan Waktu Penelitian

Tempat dan Waktu Penelitian Penelitian dilakukan di UD.Sunan Jember. Tepatnya berada di kelurahan Banjarsengon, Kecamatan Patrang, Kabupaten Jember. Waktu penelitian sekitar 2 bulan, dimulai bulan Desember 2019 – Februari 2020. Penelitian dilakukan secara bertahap yang melibatkan pimpinan serta surveyor kualitas biji jagung.

3.3 Tahapan Penelitian

Tahapan penelitian pengembangan sistem ini dilakukan dengan menggunakan model *Personal Extreme Programming* (PXP), model PXP dipilih karena pada penelitian ini membutuhkan waktu yang cepat dalam pengembangan untuk mengetahui kekurangan sistem agar memudahkan memperbaiki sistem ketika ada kesalahan.

Adapun model PXP memiliki tahapan-tahapan seperti yang dapat dilihat pada bab dua. Alur dari metode *Personal Extreme Programming* yang akan digunakan pada penelitian ini dengan rincian sebagai berikut:



4.3.1 Planning

Pada tahap ini bertujuan untuk memperoleh data atau informasi yang dibutuhkan dalam pengembangan sistem. Pengumpulan data (*requirement*) pada penelitian ini dilakukan melalui beberapa cara yaitu:

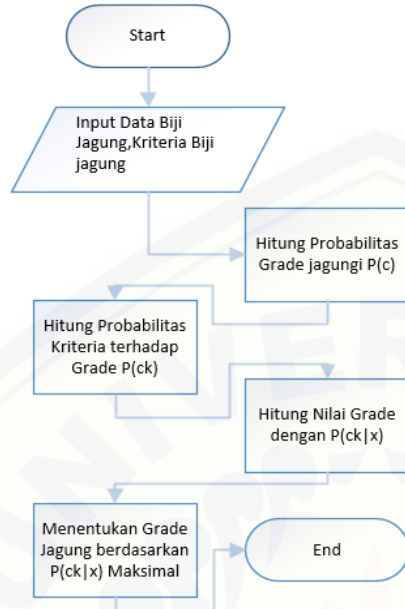
- a. Studi literatur, jurnal, serta mencari informasi menggunakan media internet mengenai biji jagung, *Naïve Bayes*, *EOQ* dan *ROP*, dan pengembangan sistem.
- b. Wawancara kepada kepala UD. Sunan Jember atau yang mewakili, yang nantinya akan diolah dalam penelitian. Data yang didapat meliputi : data jagung, data persediaan, data histori penjualan dan pembelian serta segmentasi kualitas jagung.

Adapun data dan informasi yang didapatkan dari tahap ini akan digunakan dalam penelitian ini.

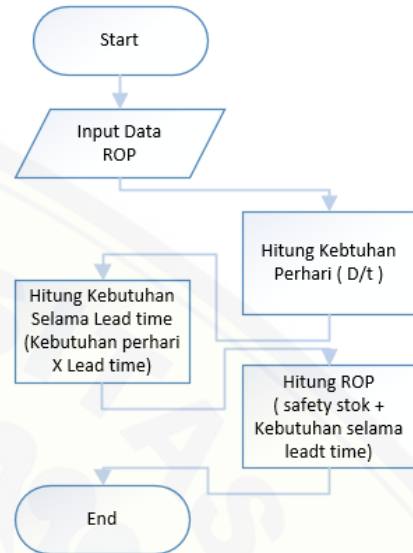
Tahap selanjutnya yaitu melakukan perencanaan fungsionalitas keseluruhan yang akan dikembangkan dalam sistem. Beberapa perencanaan yang dibuat meliputi:

- 1) Rencana pengolahan data: data yang didapat seperti data jagung dan data persediaan dikumpulkan untuk dihitung menggunakan metode *Naïve Bayes* dan *Economic Order Quantity* kemudian dianalisa sebagai bahan penyusun sistem.

Selanjutnya yaitu menganalisa menggunakan metode *Naïve Bayes* dan *Economic Order Quantity* pada gambar.



Gambar Flowchart *Naïve Bayes*



Gambar Flowchart ROP

2) Penentuan Grade Jagung

Dalam menentukan *grade* dilakukan wawancara dengan beberapa surveyor biji jagung dari beberapa pabrik. Dilakukan studi literatur sebagai faktor pendukung tentang bobot setiap kriteria yang digunakan dalam menentukan kualitas biji jagung. Bobot kriteria seperti pada tabel 3.1.

Tabel 3.1 Bobot Kriteria Penentuan kualitas Jagung

No.	Nama Kriteria	Bobot Kriteria
1	Kadar Air	20%
2	Butir Rusak	20%
3	Butir Warna Lain	10%
4	Butir Pecah	15%
5	Kadar Kotoran	15%
6	Aflatoxin	20%

3) Rencana fitur: melakukan perencanaan fitur yang diperoleh dari wawancara.

4.3.2 *Design*

Tahap ini merupakan pembuatan desain dari sistem. Desain yang dipakai dalam PXP adalah *Usecase Diagram*, *CRC Card*, dan *Class Diagram*.

4.3.3 *Implementation*

Pada tahapan ini desain sistem yang telah dibuat akan diimplementasikan secara langsung dalam sebuah sistem informasi berbasis *web (desktop)*. Pengkodean program menggunakan *framework laravel* dengan *tool Sublime Text* serta mengatur manajemen basis data menggunakan *tool Xampp*. Apabila kode selesai kemudian kode tersebut diuji dalam unit *testing*, apabila ada kesalahan maka dilakukan koreksi ulang atau *refactor* pada tahap dimana kesalahan tersebut bermula, apabila tidak ada kesalahan maka dilanjutkan ke unit selanjutnya.

4.3.4 *System Testing*

Pengujian yang dilakukan pada penelitian ini yaitu dengan menggunakan pengujian *black box*. *Black Box Testing* merupakan uji fungsionalitas apakah masih ada kekurangan atau sudah cukup sesuai. Testing juga dilakukan pada tahapan *design* dan *implementation*.

BAB IV PENGEMBANGAN SISTEM

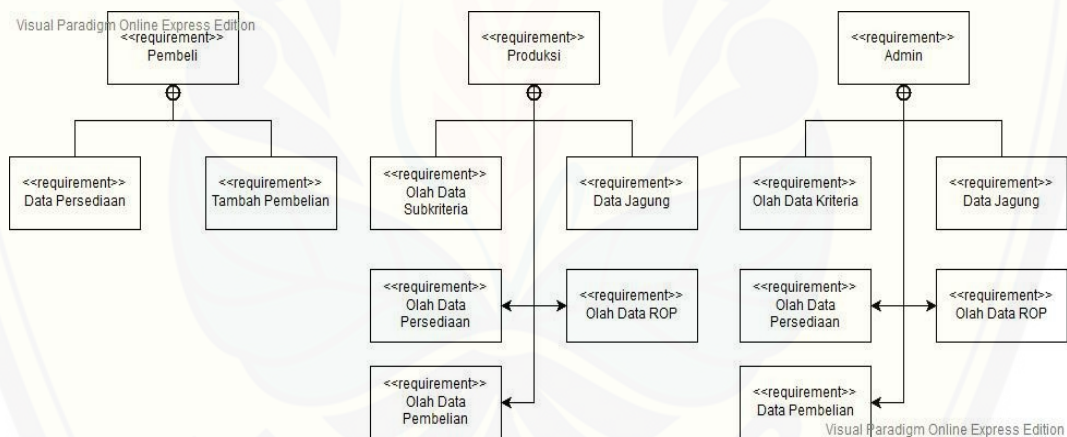
Bab ini menjelaskan tahapan pengembangan sistem. Tahapan-tahapan perancangan dilaksanakan berdasarkan model *Personal Extreme Programming* seperti yang telah dijelaskan pada bab tiga.

4.1. Planning

Tahap ini menggunakan hasil dari pengumpulan data yang telah dijabarkan pada bab sebelumnya sebagai dasar penentuan kebutuhan sistem.

4.1.1 Requirement

Kebutuhan sistem pada tahapan ini didefinisikan melalui proses pengolahan hasil wawancara dengan pihak UD. Sunan Jember yang bertujuan untuk menentukan apa saja fungsi dari setiap fitur sistem yang dikembangkan. Kebutuhan sistem dari *requirement* bisa dilihat di Gambar 4.1 :



Gambar 4. 1 Requirment Sistem

4.1.2 Perencanaan sistem

Tahapan perencanaan sistem digunakan untuk menyusun perencanaan yang dibutuhkan dalam pengembangan sistem informasi manajemen biji jagung. Setelah mendapatkan kebutuhan selanjutnya membuat beberapa perencanaan yang digunakan dalam PXP, yaitu perencanaan pengolahan data, perencanaan fitur, dan perencanaan pengerjaan sistem :

1. Perencanaan pengolahan data

Perencanaan pengolahan data dari wawancara adalah data *sampling*. Data *sampling* yang dimaksud adalah data contoh yang digunakan untuk mengembangkan sistem informasi manajemen biji jagung. Data yang diperoleh dari wawancara adalah data jagung, data kriteria, data subkriteria, dan data pembelian.

Tabel 4. 1 Data Kriteria

Kadar Air		Butir Rusak		Butir Warna Lain		Butir Pecah		Kadar Kotoran		Aflatoxin	
<1-4%	Rendah	<1-2%	Sedikit	<=1%	Sedikit	<1%	Sedikit	<1%	Rendah	<1-5%	Rendah
5-9%	Sedang	3-4%	Sedang	2-3%	Sedang	1%	Sedang	1%	Sedang	6-15%	Sedang
10-14%	Tinggi	5-7%	Banyak	4-9%	Banyak	2%	Banyak	2%	Tinggi	16-20%	Tinggi
>=15%	Sangat Tinggi	>=8%	Sangat Banyak	>=10%	Sangat Banyak	>2%	Sangat Banyak	>2%	Sangat Tinggi	>20%	Sangat Tinggi

Berdasarkan tabel 4.2 ketentuan nilai masing-masing sub kriteria sesuai dengan isian tabel berdasarkan yang telah dituliskan

Tabel 4. 2 Data Jagung

No	Kadar Air	Butir Rusak	Butir Warna Lain	Butir Pecah	Kadar Kotoran	Aflatoxin	Grade
1	rendah	Sedang	Banyak	Sedang	Rendah	Rendah	A
2	tinggi	Sedang	Sedikit	Sedang	Tinggi	Sedang	C
3	rendah	Banyak	Sedang	sangat banyak	Tinggi	Tinggi	C
4	rendah	Sedang	Sedang	Sedikit	Rendah	Sedang	B
5	rendah	Sedang	Banyak	Sedikit	Rendah	Tinggi	B
6	tinggi	Sedang	Sedikit	banyak	Rendah	Sedang	C
7	rendah	Banyak	Sedang	Sedikit	Tinggi	Sedang	D
8	sangat tinggi	Sedikit	Sedang	banyak	Rendah	Tinggi	C
9	tinggi	Sedang	Sedikit	banyak	Sedang	Rendah	C
10	sedang	Sedang	Sedang	banyak	Rendah	Rendah	B
11	sedang	Banyak	Sedang	Sedikit	Tinggi	Rendah	C
12	sedang	Rendah	Banyak	Sedang	Rendah	Tinggi	B
13	rendah	Sedang	Banyak	Sedang	Rendah	Rendah	A
14	tinggi	Sedang	Sedikit	Sedang	Sedang	Sedang	B
15	sedang	Banyak	Sedang	banyak	Rendah	Sedang	C
16	sedang	Banyak	Sedang	Sedikit	Tinggi	Rendah	C
17	sedang	Rendah	Banyak	Sedang	Rendah	Tinggi	B

No	Kadar Air	Butir Rusak	Butir Warna Lain	Butir Pecah	Kadar Kotoran	Aflatoxin	Grade
18	tinggi	Sedang	Sedikit	Sedang	Tinggi	Sedang	C
19	rendah	Sedang	Banyak	Sedang	Rendah	Sedang	B
20	sedang	Banyak	Banyak	sangat banyak	Sedang	Sedang	D
21	sedang	Rendah	Sedikit	Sedang	sangat tinggi	Sedang	C
22	tinggi	Sedang	Sedang	banyak	Rendah	Tinggi	D
23	sangat tinggi	Sedang	Sedang	Sedikit	Sedang	Tinggi	D
24	rendah	Sedang	Sedikit	Sedang	Sedang	Rendah	A
25	sedang	Banyak	sangat banyak	Sedang	Tinggi	Sedang	C
26	sedang	Sedikit	Sedang	banyak	Tinggi	Sedang	C
27	tinggi	Sedang	Sedang	banyak	Rendah	Tinggi	D
28	rendah	Sedang	Sedang	Sedikit	Sedang	Rendah	A
29	tinggi	Sedang	Banyak	Sedang	Sedang	Tinggi	D
30	sedang	Banyak	Sedikit	Sedang	Singggi	Rendah	C
31	sangat tinggi	Banyak	Banyak	Sedang	Rendah	Tinggi	D
32	rendah	Sedang	Sedang	Sedikit	Sedang	sangat tinggi	C
33	sedang	Sedang	Sedang	banyak	Sedang	Rendah	C
34	tinggi	Sedang	Banyak	banyak	Tinggi	Sedang	D
35	rendah	Banyak	Sedikit	Sedikit	Tinggi	Sedang	C
36	sedang	Banyak	Sedang	Sedang	Rendah	Rendah	B
37	sedang	Sedikit	Sedang	Sedang	Sedang	Tinggi	B
38	tinggi	Sedang	Sedang	Sedikit	Sedang	Sedang	C
39	sangat tinggi	Sedang	Banyak	sangat banyak	sangat tinggi	Sedang	D
40	rendah	Banyak	Sedikit	Sedang	Rendah	Sedang	C

2) Perencanaan Fitur

Perencanaan fitur digunakan untuk memberikan pedoman fitur yang akan dibangun di sistem berdasarkan *requirement* sistem.

Tabel 4. 3 Fitur pada sistem

No	Fitur	Aktor	Fungsi
1	Login	Manajer, Admin Pegawai, User Pembeli	Untuk bisa masuk ke sistem
2	Lihat data kriteria	Manajer, Admin Pegawai	Untuk melihat data kriteria
3	Tambah data kriteria	Manajer	Untuk menambah data kriteria

4	Edit data kriteria	Manajer	Untuk mengubah data kriteria
5	Hapus data kriteria	Manajer	Untuk menghapus data kriteria
6	Lihat Data Training	Manajer, Admin Pegawai	Untuk melihat data Data Training
7	Lihat data rop	Manajer, Admin Pegawai	Untuk melihat data rop
8	Tambah data rop	Manajer	Untuk menambah data rop
9	Hapus data rop	Manajer	Untuk menghapus data rop
10	Lihat data stok jagung	Manajer, Admin Pegawai	Untuk melihat data stok jagung
11	Lihat data persediaan	Manajer, Admin Pegawai	Untuk melihat data persediaan
12	Validasi data persediaan	Manajer	Untuk memvalidasi persediaan
13	Lihat data penjualan	Manajer, Admin Pegawai	Untuk melihat data penjualan
14	Lihat data penentuan grade	Admin Pegawai	Untuk melihat data penentuan grade
15	Hitung grade	Admin Pegawai	Untuk menghitung grade
16	Tambah stok grade	Admin Pegawai	Untuk menambah data stok grade
17	Tambah data pembelian	User Pembeli	Untuk menambah data pembelian
18	Lihat data validasi	User Pembeli	Untuk melihat data validasi

4.2. Design

Seperti yang telah dijelaskan pada bab sebelumnya, desain sistem pada penelitian ini meliputi *usecase diagram* untuk membuat fitur pada sistem, *collaboration responsibility class (CRC Card)* untuk menentukan tanggung jawab kelas yang dibuat, dan *class diagram* untuk menentukan hubungan antara kelas yang akan dibuat.

4.2.1 Usecase Diagram

Hasil dari tahapan *planning* sistem diperoleh keputusan bahwa sistem ini memiliki 3 aktor yang bisa mengakses karena merupakan sistem yang ditujukan untuk umum. Penentuan hak akses diperoleh dari proses *planning*. *Usecase diagram* digunakan untuk menggambarkan interaksi antara aktor-aktor tersebut dengan fitur sistem yang dibuat. Selain menyajikan *usecase diagram* pada Gambar 4.1, sub bab ini akan memberikan penjelasan tentang setiap aktor dan *usecase* yang ada.

1. Definisi Aktor

Berikut merupakan penjelasan aktor atau pengguna pada sistem. Terdapat dua aktor yang dijelaskan pada Tabel 4.4 berikut ini.

Tabel 4. 4 Definisi aktor sistem

No.	Aktor	Deskripsi
1.	Manajer	Aktor Manajer bisa mengakses seluruh fitur.
2.	Admin Pegawai	Aktor Admin Pegawai bisa mengakses seluruh fitur kecuali fitur validasi pesanan
3.	User Pembeli	Aktor User Pembeli bisa mengakses fitur tambah data pembelian dan lihat data validasi

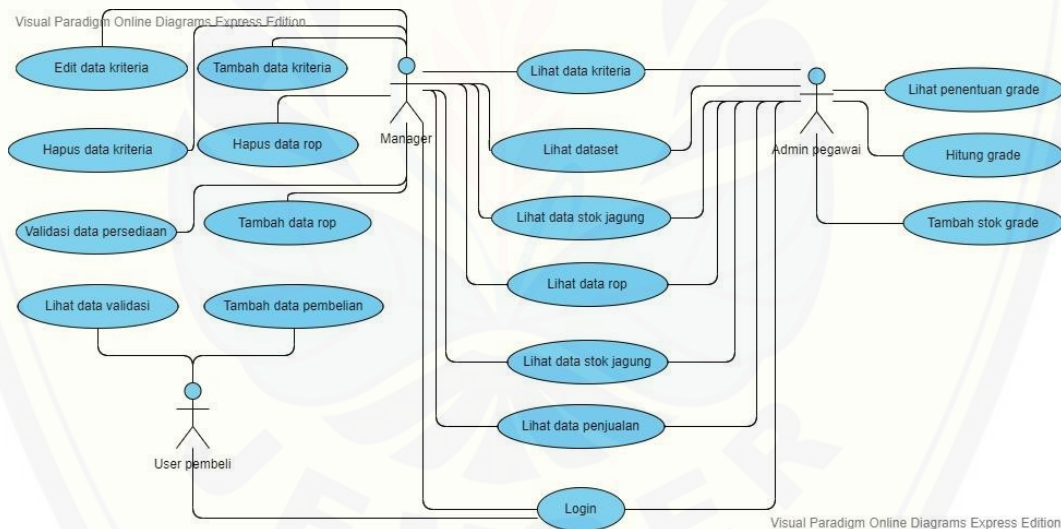
2. Definisi *Usecase*

Definisi *usecase* menjelaskan setiap *usecase* dalam sistem informasi pemetaan flora dan fauna. Definisi *usecase* ini dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Definisi usecase

No	Fitur	Definisi
1	Login	Menggambarkan fitur masuk ke sistem
2	Lihat data kriteria	Menggambarkan fitur melihat data kriteria
3	Tambah data kriteria	Menggambarkan fitur menambah data kriteria
4	Edit data kriteria	Menggambarkan fitur mengubah data kriteria
5	Hapus data kriteria	Menggambarkan fitur menghapus data kriteria
6	Lihat Data Training	Menggambarkan fitur melihat data Data Training
7	Lihat data rop	Menggambarkan fitur melihat data rop
8	Tambah data rop	Menggambarkan fitur menambah data rop
9	Hapus data rop	Menggambarkan fitur menghapus data rop
10	Lihat data stok jagung	Menggambarkan fitur melihat data stok jagung
11	Lihat data persediaan	Menggambarkan fitur melihat data persediaan

No	Fitur	Definisi
12	Validasi data persediaan	Menggambarkan fitur memvalidasi persediaan
13	Lihat data penjualan	Menggambarkan fitur melihat data penjualan
14	Lihat data penentuan grade	Menggambarkan fitur melihat data penentuan grade
15	Hitung grade	Menggambarkan fitur menghitung grade
16	Tambah stok grade	Menggambarkan fitur menambah data stok grade
17	Tambah data pembelian	Menggambarkan fitur menambah data pembelian
18	Lihat data validasi	Menggambarkan fitur melihat data validasi



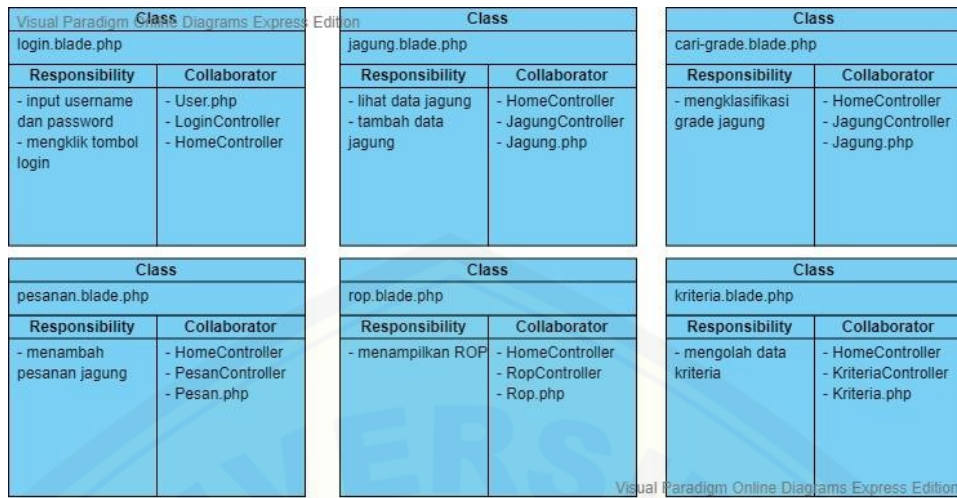
Gambar 4. 2 Usecase Sistem

4.2.2 Collaboration Responsibility Class (CRC Card)

CRC Card diperoleh dari penjabaran hubungan *responsibility* antar *class* di sistem sesuai dengan yang telah digambarkan dalam *Usecase Diagram*. *Class* pada sistem dibuat sesuai fitur dan definisi CRC Card dapat dilihat pada tabel 4.6, dengan penjelasan nama *class* dan deskripsi dari setiap kolom CRC Card.

Tabel 4. 6 Definisi CRC Card

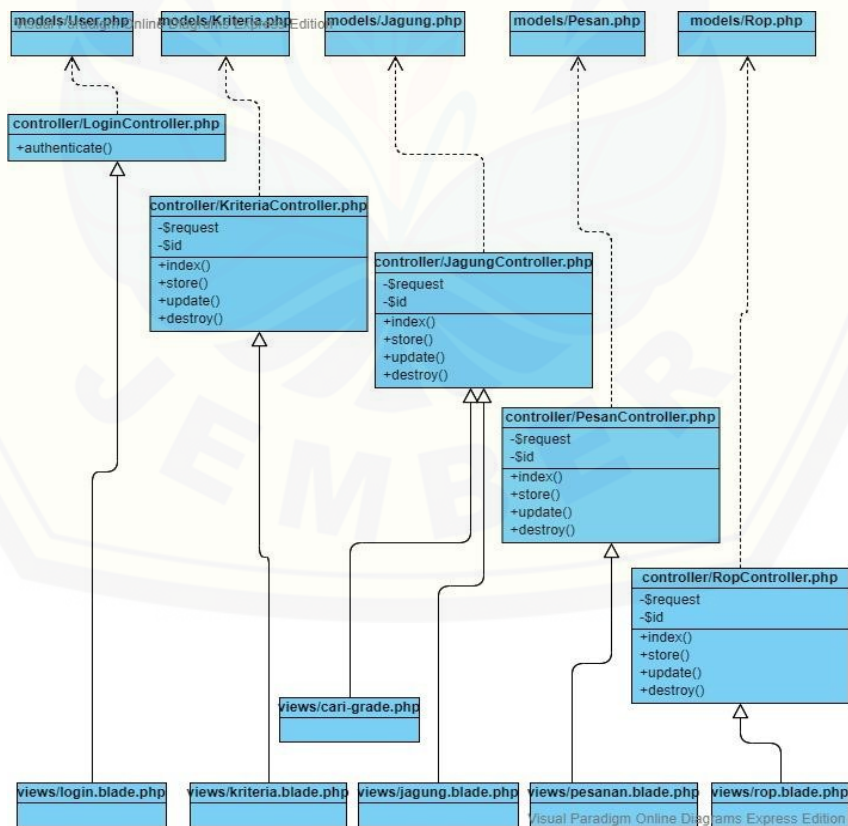
No.	Nama Class	Deskripsi
1.	login.blade.php	Class ini memiliki <i>responsibility</i> untuk input username dan password, mengklik tombol login. Class ini berkolaborasi dengan User.php, LoginController.php, dan HomeController.php
2.	jagung.blade.php	Class ini memiliki <i>responsibility</i> untuk melihat dan menambah data jagung. Class ini berkolaborasi dengan HomeController.php, JagungController.php, dan Jagung.php
3.	cari-grade.blade.php	Class ini memiliki <i>responsibility</i> untuk mengklasifikasi grade jagung. Class ini berkolaborasi dengan HomeController.php, JagungController.php, dan Jagung.php
4.	pesanan.blade.php	Class ini memiliki <i>responsibility</i> untuk menambah pesanan jagung. Class ini berkolaborasi dengan HomeController.php, PesanController.php, dan Pesan.php
5.	rop.blade.php	Class ini memiliki <i>responsibility</i> untuk melihat data ROP. Class ini berkolaborasi dengan HomeController.php, RopController.php, dan Rop.php
6.	kriteria.blade.php	Class ini memiliki <i>responsibility</i> untuk melihat data Kriteria. Class ini berkolaborasi dengan HomeController.php, KriteriaController.php, dan Kriteria.php



Gambar 4.3 CRC card sistem

4.2.3 Class Diagram

Class Diagram menggambarkan hubungan antarkelas yang ada dalam suatu sistem. Class Diagram sistem pada penelitian ini dapat dilihat pada Gambar 4.5.



Gambar 4.4 Class Diagram Sistem

4.2.4 Testing design sistem

Testing dari design yang telah dibuat sebelumnya bertujuan untuk mengetahui kesesuaian dari desain *Usecase*, *CRC card*, dan *Class Diagram* seperti yang dijabarkan pada tabel 4.8.

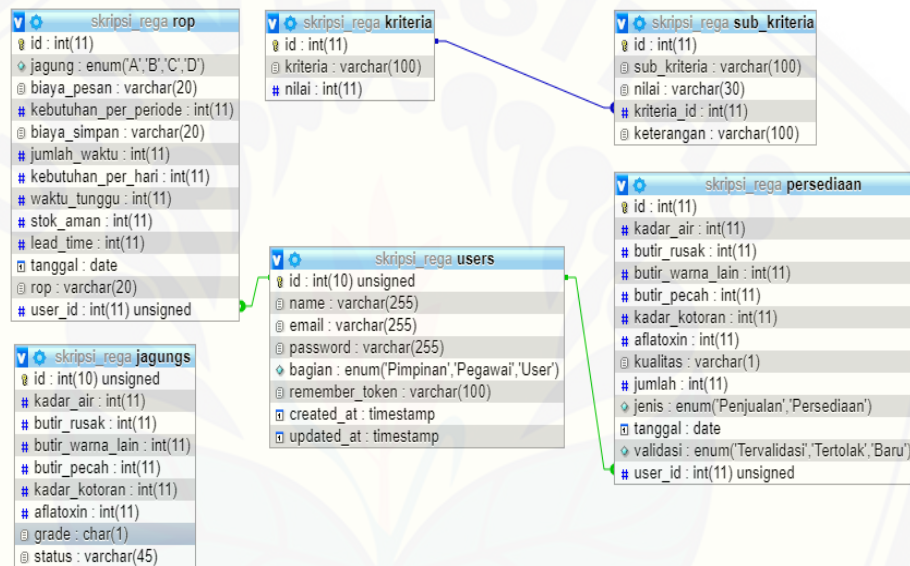
Tabel 4. 7 Testing design

Fitur	<i>Usecase</i>	<i>CRC Card</i>	<i>Class Diagram</i>	Status
Login	Login	login.blade.php	login.blade.php	Sesuai
Lihat data kriteria	Lihat data kriteria	kriteria.blade.php	kriteria.blade.php	Sesuai
Tambah data kriteria	Tambah data kriteria	kriteria.blade.php	kriteria.blade.php	Sesuai
Edit data kriteria	Edit data kriteria	kriteria.blade.php	kriteria.blade.php	Sesuai
Hapus data kriteria	Hapus data kriteria	kriteria.blade.php	kriteria.blade.php	Sesuai
Lihat Data Training	Lihat Data Training	jagung.blade.php	jagung.blade.php	Sesuai
Lihat data rop	Lihat data rop	rop.blade.php	rop.blade.php	Sesuai
Tambah data rop	Tambah data rop	rop.blade.php	rop.blade.php	Sesuai
Hapus data rop	Hapus data rop	rop.blade.php	rop.blade.php	Sesuai
Lihat data stok jagung	Lihat data stok jagung	jagung.blade.php	jagung.blade.php	Sesuai
Lihat data persediaan	Lihat data persediaan	jagung.blade.php	jagung.blade.php	Sesuai
Validasi data persediaan	Validasi data persediaan	pesanan.blade.php	pesanan.blade.php	Sesuai
Lihat data penjualan	Lihat data penjualan	pesanan.blade.php	pesanan.blade.php	Sesuai
Lihat data penentuan grade	Lihat data penentuan grade	cari-grade.blade.php	cari-grade.blade.php	Sesuai
Hitung grade	Hitung grade	cari-grade.blade.php	cari-grade.blade.php	Sesuai
Tambah stok grade	Tambah stok grade	cari-grade.blade.php	cari-grade.blade.php	Sesuai

Fitur	Usecase	CRC Card	Class Diagram	Status
Tambah data pembelian	Tambah data pembelian	pesanan.blade.php	pesanan.blade.php	Sesuai
Lihat data validasi	Lihat data validasi	pesanan.blade.php	pesanan.blade.php	Sesuai

4.2.5 Entity Relationship Diagram

ERD merupakan gambaran komponen dan struktur *database* yang digunakan dalam pembangunan sistem.



Gambar 4. 5 Entity Relationship Diagram

4.3 Implementation

Penulisan kode program merupakan tahap pengimplementasian desain yang telah dibuat sebelumnya. Setiap fitur yang dimiliki oleh sistem dituliskan dalam beberapa *class*. Tahap implementasi dan hasil tampilan lainnya bisa dilihat pada lampiran.

4.3.1 Kode program data jagung dan hitung grade

Kode program menambah data jagung terletak pada class jagung.blade.php, JagungController.php, dan Jagung.php.

Kode program Hitung-grade

```
53 Route::post('/HitungGrade','JagungController@hitungGrade')->name('HitungGrade');

49 public function hitungGrade(Request $request)
50 {
51     $ka = $request->kadar_air;
52     $br = $request->butir_rusak;
53     $bw = $request->butir_warna_lain;
54     $bp = $request->butir_pecah;
55     $kk = $request->kadar_kotoran;
56     $af = $request->aflatoxin;
57
58     $jagungs = Jagung::whereNull('status')->get();
59     $grade['A'] = 0;
60     $grade['B'] = 0;
61     $grade['C'] = 0;
62     $grade['D'] = 0;
63     $grade_count = 0;
64     foreach ($jagungs as $jagung) {
65         $grade_count++;
66         switch ($jagung->grade) {
67             case 'A':
68                 $grade['A']++;
69                 break;
70             case 'B':
71                 $grade['B']++;
72                 break;
73             case 'C':
74                 $grade['C']++;
75                 break;
76             case 'D':
77                 $grade['D']++;
78                 break;
79             default:
```

```
84 $p_grade = array();
85 $p_grade['A'] = $grade['A'] / $grade_count;
86 $p_grade['B'] = $grade['B'] / $grade_count;
87 $p_grade['C'] = $grade['C'] / $grade_count;
88 $p_grade['D'] = $grade['D'] / $grade_count;
89 $p_grade_attr = null;
90 $p_grade_attr['kadar_air']['A'][0] = 0;
91 $p_grade_attr['kadar_air']['B'][0] = 0;
92 $p_grade_attr['kadar_air']['C'][0] = 0;
93 $p_grade_attr['kadar_air']['D'][0] = 0;
94 $p_grade_attr['kadar_air']['A'][1] = 0;
95 $p_grade_attr['kadar_air']['B'][1] = 0;
96 $p_grade_attr['kadar_air']['C'][1] = 0;
97 $p_grade_attr['kadar_air']['D'][1] = 0;
98
99 $p_grade_attr['butir_rusak']['A'][0] = 0;
00 $p_grade_attr['butir_rusak']['B'][0] = 0;
01 $p_grade_attr['butir_rusak']['C'][0] = 0;
02 $p_grade_attr['butir_rusak']['D'][0] = 0;
03 $p_grade_attr['butir_rusak']['A'][1] = 0;
04 $p_grade_attr['butir_rusak']['B'][1] = 0;
05 $p_grade_attr['butir_rusak']['C'][1] = 0;
06 $p_grade_attr['butir_rusak']['D'][1] = 0;
07
08 $p_grade_attr['butir_warna_lain']['A'][0] = 0;
09 $p_grade_attr['butir_warna_lain']['B'][0] = 0;
10 $p_grade_attr['butir_warna_lain']['C'][0] = 0;
11 $p_grade_attr['butir_warna_lain']['D'][0] = 0;
12 $p_grade_attr['butir_warna_lain']['A'][1] = 0;
13 $p_grade_attr['butir_warna_lain']['B'][1] = 0;
14 $p_grade_attr['butir_warna_lain']['C'][1] = 0;
15 $p_grade_attr['butir_warna_lain']['D'][1] = 0;
```

```
117 $p_grade_attr['butir_pecah']['A'][0] = 0;
118 $p_grade_attr['butir_pecah']['B'][0] = 0;
119 $p_grade_attr['butir_pecah']['C'][0] = 0;
120 $p_grade_attr['butir_pecah']['D'][0] = 0;
121 $p_grade_attr['butir_pecah']['A'][1] = 0;
122 $p_grade_attr['butir_pecah']['B'][1] = 0;
123 $p_grade_attr['butir_pecah']['C'][1] = 0;
124 $p_grade_attr['butir_pecah']['D'][1] = 0;
125
126 $p_grade_attr['kadar_kotoran']['A'][0] = 0;
127 $p_grade_attr['kadar_kotoran']['B'][0] = 0;
128 $p_grade_attr['kadar_kotoran']['C'][0] = 0;
129 $p_grade_attr['kadar_kotoran']['D'][0] = 0;
130 $p_grade_attr['kadar_kotoran']['A'][1] = 0;
131 $p_grade_attr['kadar_kotoran']['B'][1] = 0;
132 $p_grade_attr['kadar_kotoran']['C'][1] = 0;
133 $p_grade_attr['kadar_kotoran']['D'][1] = 0;
134
135 $p_grade_attr['aflatoxin']['A'][0] = 0;
136 $p_grade_attr['aflatoxin']['B'][0] = 0;
137 $p_grade_attr['aflatoxin']['C'][0] = 0;
138 $p_grade_attr['aflatoxin']['D'][0] = 0;
139 $p_grade_attr['aflatoxin']['A'][1] = 0;
140 $p_grade_attr['aflatoxin']['B'][1] = 0;
141 $p_grade_attr['aflatoxin']['C'][1] = 0;
142 $p_grade_attr['aflatoxin']['D'][1] = 0;
143 $grades = ['A', 'B', 'C', 'D'];
```

```

144     foreach ($jagungs as $jagung) {
145         foreach ($p_grade_attr as $key => $value) {
146             foreach ($value as $k => $v) {
147                 if ($jagung->{$key} == $request->{$key} && $jagung->grade == $k) {
148                     $p_grade_attr[$key][$k][0]++;
149                     $p_grade_attr[$key][$k][1] = round($p_grade_attr[$key][$k][0] / $grade[$k], 3);
150                 }
151             }
152         }
153     }
154     $pxg = null;
155     $pxg['A'] = round($p_grade_attr['kadar_air']['A'][1] * $p_grade_attr['butir_rusak']['A'][1], 7);
156     $pxg['B'] = round($p_grade_attr['kadar_air']['B'][1] * $p_grade_attr['butir_rusak']['B'][1], 7);
157     $pxg['C'] = round($p_grade_attr['kadar_air']['C'][1] * $p_grade_attr['butir_rusak']['C'][1], 7);
158     $pxg['D'] = round($p_grade_attr['kadar_air']['D'][1] * $p_grade_attr['butir_rusak']['D'][1], 7);
159     $px = array();
160     $px['A'] = round($pxg['A'] * $p_grade['A'], 7);
161     $px['B'] = round($pxg['B'] * $p_grade['B'], 7);
162     $px['C'] = round($pxg['C'] * $p_grade['C'], 7);
163     $px['D'] = round($pxg['D'] * $p_grade['D'], 7);
164     $hasil_grade = array_keys($px, max($px));
165
166     $array = [
167         'kadar_air' => $request->kadar_air,
168         'butir_rusak' => $request->butir_rusak,
169         'butir_warna_lain' => $request->butir_warna_lain,
170         'butir_pecah' => $request->butir_pecah,
171         'kadar_kotoran' => $request->kadar_kotoran,
172         'aflatoxin' => $request->aflatoxin,
173         'grade' => $hasil_grade[0],
174         'status' => 'perhitungan',
175     ];

```

Gambar 4. 6 Kode program Hitung-grade

Program diatas menjelaskan tentang proses perhitungan *grade* menggunakan metode *naïve bayes* dengan kriteria kadar air, butir rusak, butir warna lain, kadar kotoran, dan *aflatoxin* yang menghasilkan antara *grade* A, B, C, dan D.

ID	Kadar Air	Butir Rusak	Butir Warna Lain	Butir Pecah	Kadar Kotoran	Aflatoxin	Grade	Aksi
44	Rendah	Sedikit	Sedang	Sangat Banyak	Rendah	Tinggi	C	Tambah Stok

Gambar 4.7 Tampilan hitung grade

Kode program Data Jagung

```

40 Route::get('DataJagung','PersediaanController@range')->name('DataJagung');
41
19 public function index()
20 {
21     $kriteria = Kriteria::all();
22     $data = Persediaan::where('jenis','Persediaan')->get();
23     $rg = VwRange::all();
24     // dd($rg);
25     return view('admin.persediaan',compact('data','kriteria','rg'));
26 }
    
```

Gambar 4. 8 Kode program Data Jagung

Kode program diatas menjelaskan tentang proses didaptkannya data jagung yang tersedia.

No	Kualitas	Jumlah Stok
1	Kualitas A	44
2	Kualitas B	1003
3	Kualitas C	1325
4	Kualitas D	171

Gambar 4.9 Tampilan data jagung

4.3.2 Kode program Hitung data ROP

Kode program melihat data ROP terletak pada class rop.blade.php, RopController.php, dan Rop.php.

Kode program Hitung ROP

```

43 Route::post('/Rop/Create','RopController@store')->name('createRop');
    
```

```

8 class RopController extends Controller
9 {
10     public function index()
11     {
12         $data = Rop::orderBy('id','desc')->get();
13         return view('admin.rop',compact('data'));
14     }
15
16     public function store(Request $request)
17     {
18         $request->offsetUnset('_token');
19         $data = $request->all();
20
21         $idBarang=$data['jagung'];
22         $biayapesan=$data['biaya_pesan'];
23         $biayasimpan=$data['biaya_simpan'];
24         $tahunkerja=$data['jumlah_waktu'];
25         $bbperhari=$data['kebutuhan_per_hari'];
26         $bbperperiode=$data['kebutuhan_per_periode'];
27         $tunggubarang=$data['waktu_tunggu'];
28         $stokaman=$data['stok_aman'];
29         $leadtime=$data['lead_time'];
30
31         $EOQ=(2*$biayapesan*$bbperperiode)/$biayasimpan;
32         $FP=$bbperperiode/$EOQ;
33         $bbsehari=$bbperperiode/$tahunkerja;
34         $bbselamaLeadTime=$bbperhari*$leadtime;
35         $rop=$stokaman+$bbselamaLeadTime;
36         $data['rop'] = $rop;
37
38         Rop::create($data);
39         return back();
40     }

```

Gambar 4. 10 Kode program Hitung ROP

Kode program diatas menjelaskan tentang proses perhitungan ROP dengan atribut diantaranya jenis jagung, biaya pesan, biaya simpan, periode, kebutuhan per hari, kebutuhan per periode, waktu tunggu, stok aman, dan *lead time*.

ID	Nama Barang	Biaya Pesan	Kebutuhan Bahan Baku Per Periode	Biaya Simpan	Jumlah Tahun Kerja	Kebutuhan Bahan Baku Per Hari	Waktu Tunggu Barang (hari)	Stok Aman	Lead Time	Rop
5	B	10000	5000	5000	1	100	1	500	0	500
4	A	500000	14	12000	1	1	2	5	3	8

Gambar 4.11 Tampilan data ROP

4.4 Testing System

Tahap pengujian dilakukan untuk mengevaluasi sistem informasi yang telah selesai dan telah diimplementasikan. Seperti yang telah dijelaskan pada bab sebelumnya, tahapan ini dilakukan metode pengujian *black box* yang dapat dilihat pada tabel 4.15

Tabel 4. 8 Tabel *Blackbox Tesring*

Fitur	Usecase	CRC Card	Class Diagram	Tampilan	Status
Login	Login	login.blade.php	login.blade.php	Gambar B.1	Sesuai
Lihat data kriteria	Lihat data kriteria	kriteria.blade.php	kriteria.blade.php	Gambar B.18, Gambar B.9	Sesuai
Tambah data kriteria	Tambah data kriteria	kriteria.blade.php	kriteria.blade.php	Gambar B.17	Sesuai
Edit data kriteria	Edit data kriteria	kriteria.blade.php	kriteria.blade.php	Gambar B.16	Sesuai
Hapus data kriteria	Hapus data kriteria	kriteria.blade.php	kriteria.blade.php	Gambar B.18	Sesuai
Lihat Data Training	Lihat Data Training	jagung.blade.php	jagung.blade.php	Gambar B.15, Gambar B.8	Sesuai
Lihat data rop	Lihat data rop	rop.blade.php	rop.blade.php	Gambar B.14, Gambar B.7	Sesuai

Fitur	Usecase	CRC Card	Class Diagram	Tampilan	Status
Tambah data rop	Tambah data rop	rop.blade.php	rop.blade.php	Gambar B.21	Sesuai
Hapus data rop	Hapus data rop	rop.blade.php	rop.blade.php	Gambar B.14	Sesuai
Lihat data stok jagung	Lihat data stok jagung	jagung.blade.php	jagung.blade.php	Gambar B.12, Gambar B.6	Sesuai
Lihat data persediaan	Lihat data persediaan	jagung.blade.php	jagung.blade.php	Gambar B.11, Gambar B.5	Sesuai
Validasi data persediaan	Validasi data persediaan	pesanan.blade.php	pesanan.blade.php	Gambar B.10	Sesuai
Lihat data penjualan	Lihat data penjualan	pesanan.blade.php	pesanan.blade.php	Gambar B.22	Sesuai
Lihat data penentuan grade	Lihat data penentuan grade	cari-grade.blade.php	cari-grade.blade.php	Gambar B.4	Sesuai
Hitung grade	Hitung grade	cari-grade.blade.php	cari-grade.blade.php	Gambar B.3	Sesuai
Tambah stok grade	Tambah stok grade	cari-grade.blade.php	cari-grade.blade.php	Gambar B.2	Sesuai
Tambah data pembelian	Tambah data pembelian	pesanan.blade.php	pesanan.blade.php	Gambar B.19	Sesuai
Lihat data validasi	Lihat data validasi	pesanan.blade.php	pesanan.blade.php	Gambar B.20	Sesuai

BAB VI PENUTUP

Bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan. Kesimpulan dan saran ini diharapkan mampu menjadi acuan untuk melakukan penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

- 1) Penerapan metode *Economy Order Quantity* dalam membangun sistem manajemen persediaan biji jagung menghasilkan hasil yang sama dengan perhitungan secara manual. Metode EOQ menggunakan beberapa inputan diantaranya : biaya pesan, kebutuhan barang per periode, jumlah tahun kerja, kebutuhan barang per hari, biaya simpan, waktu tunggu (*Lead Time*), dan stok aman (*Safety Stock*). Dari beberapa inputan tersebut diproses sehingga menghasilkan inputan untuk melakukan proses *Re – Order Point* (ROP) .Uji coba perhitungan dilakukan selama 2 kali dan menghasilkan hasil yang sama.
- 2) Implementasi Metode *Naïve Bayes* pada sistem informasi manajemen biji jagung memiliki tingkat akurasi 80% dalam penentuan kualitas biji jagung dengan pengklasifikasian yang melibatkan beberapa kriteria diantaranya : kadar air, butir rusak, butir warna lain, butir pecah, kadar kotoran, dan aflatoxin. Yang sesuai dengan (SNI 3920:2013).

6.2 Saran

Saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

- 1) Sistem yang dikembangkan hanya dapat memesan dengan kualitas yang telah ditentukan oleh pihak perusahaan, sehingga diharapkan pada penelitian selanjutnya pembeli dapat melihat secara detail nilai setiap kriteria pada kualitas jagung yang dipilih.
- 2) Sistem ini dikembangkan dengan metode EOQ yang menitik beratkan pada *Re – Order Point* untuk *restock* barang. Untuk pengembangan selanjutnya dapat ditambahkan fitur dimana dapat memprediksi kapan proses *restock* dapat dilakukan sebelum menentukan titik *Re – Order Point*.

- 3) Penggunaan Data Training dalam metode *Naïve Bayes* berjumlah 40 sampel yang relatif sedikit sehingga berpengaruh pada tingkat akurasi. Diharapkan untuk pengembangan penelitian selanjutnya dapat menambah jumlah sampel data training untuk meningkatkan tingkat akurasi.



DAFTAR PUSTAKA

- Wardhana, Bagoes. 2017. “Rancang Bangun Sistem Informasi Manajemen Stok Perusahaan Susu Bestcow Menggunakan Metode Economic Order Quantity (EOQ)”. Skripsi. Sistem Informasi, Universitas Jember, Jember.
- Daniela, X., Christopher, J. & Roger, G., 2009. “ Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages”. International Journal of Computer Science Issues.P.8
- Dsn, Cahya. 2018. “ Naïve Bayes Classifier (NBC)”, cahyadsn.phpindonesia.id, Diakses pada 17 Oktober 2019.
- Husen, S. 2015. “ Rancang Bangun Sistem Informasi Manajemen Stok Barang Berbasis Web Toko Besi Bangunan Barokah Material Menggunakan Metode EOQ”. Skripsi. Sistem Informasi, Universitas Jember, Jember.
- Lutfi, 2020. *Cara menentukan Kualitas Jagung* [Interview] (11 Januari 2020).
- Nugroho, Y. S., n.d. “Data Mining Menggunakan Algoritma Naive Bayes Untuk Klasifikasi Kelulusan Mahasiswa”. P. 11.
- Sunan, 2019. *Cara menentukan Kualitas Jagung dan Manajemen Persediaan* [Interview] (17 10 2019).
- Imron, 2020. *Cara menentukan grade jagung* [Interview] (13 02 2020).
- Dzhurov, Y., Krasteva, I., & Ilieva, S. (2009). Personal Extreme Programming–An Agile Process for Autonomous Developers. *International Conference on Software, Services & Semantic Technologies*, (August 2016), 252–259.
- Rizal, H. et al. (2014). *Perancangan dan Pembuatan Mobile Learning Interaktif Berbasis Android dengan Metode Personal Extreme Programming*

LAMPIRAN

A. Kode Program

Tabel A.1 hasilKriteria.php

```

@extends('layouts.app2')

@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-11">
      <div class="panel panel-default">
        <div class="panel-heading">Range Pembelian
Tanah
        </div>
        <div class="panel-body">
          <table class="table table-responsive">
            <thead>
              <tr>
                <th>Jenis Kategori</th>
                <th>Range</th>
                <th>Keterangan</th>
              </tr>
            </thead>
            <tbody>
              <?php $i = 1;$prev = ""; ?>
                @foreach($data as $row)
                  <tr>
                    <td>Range {{ $i++ }}</td>
                    <td>{{ $row->nilai }}</td>
                    <td>

```

```

@if($i==2)
    Nilai > {{$row->range}}
@else
    Nilai > {{$row->range}} & Nilai < {{$prev}}
@endif
</td>
<?php $prev = $row->range; ?>
    </tr>
@endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>

```

Tabel A.2 kriteria.blade.php

```

@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
                <div class="panel-heading">Kriteria
                    @if(Auth::user()->bagian == 'Pimpinan')

```

```

<button type="button" class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#create">Tambah Kriteria</button>

    @endif

</div>

<div class="panel-body">
    @if (session('status'))
<div class="alert alert-success">
    {{ session('status') }}
</div>
    @endif
    <table class="table table-responsive">
        <thead>
            <tr>
                <th>ID</th>
                <th>Kriteria</th>
                <th>Sub</th>
                <th>Nilai</th>
                <th>Aksi</th>
            </tr>
        </thead>
        <tbody>
            @foreach($data as
            $row)
                <tr>
                    <td>{{ $row->id }}</td>
                    <td>{{ $row->kriteria }}</td>
                    <td>
                        @foreach($row->sub as $r)
                            {{ $r->sub_kriteria }} <br>
                        @endforeach
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>

```

```

</td>
@foreach($row->sub as $r)
    {{$r->nilai}} <br>
    @endforeach
</td>
<td>
<a href="{{route('SubKriteria',['id' => $row->id])}}"><button
type="button" class="btn-sm btn-info"><i class="fa fa-list"></i></button></a>

@if(Auth::user()->bagian == 'Pimpinan')
<button type="button" class="btn-sm btn-warning" data-toggle="modal"
data-target="#update"          onClick='setData({{$row->id}},{{$row-
>kriteria}},{{$row->nilai}});'><i class="fa fa-pencil"></i></button>

<a href="{{route('deleteKriteria',['id' => $row->id])}}"><button type="button"
class="btn-sm btn-danger"><i class="fa fa-trash"></i></button></a>

@endif

</td>
</tr>

@endforeach

</tbody>
</table>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">

```

```

<button type="button" class="close" data-dismiss="modal">&times;</button>
    <h4 class="modal-title">Tambah Data</h4>
</div>
<form action="{{route('createKriteria')}}" method="post" accept-charset="utf-8">
    {{csrf_field()}}
    <div class="modal-body">
        <label>Nama Kriteria</label>
        <input type="text" name="kriteria" value="" placeholder="nama kriteria"
        class="form-control">
        <label>Bobot/Nilai</label>
        <input type="text" name="nilai" value="" placeholder="Nilai Kriteria"
        class="form-control">
    </div>
    <div class="modal-footer">
        <button class="btn-sm btn-primary" type='submit'>Simpan</button>
        <button type="button" class="btn btn-default" data-
        dismiss="modal">Close</button>
    </div>
</form>
</div>
</div>
<div id="update" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Update Data</h4>
            </div>

```



```

<form action="{{route('updateKriteria')}}" method="post" accept-charset="utf-
8">
    {{csrf_field()}}
<div class="modal-body">
<input class="hidden" name="id" type="hidden" id="id"></input>
    <label>Nama Kriteria</label>
<input type="text" id="kriteria" name="kriteria" value="" placeholder="nama
kriteria" class="form-control">
    <label>Bobot/Nilai</label>
<input type="text" name="nilai" id="nilai" value="" placeholder="Nilai
Kriteria" class="form-control">
</div>
<div class="modal-footer">
    <button class="btn-sm btn-primary" type='submit'>Simpan</button>
    <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
</div>
    </form>
</div>
</div>
</div>
<script>
function setData(id,kriteria,nilai){
    console.log(id+kriteria+nilai);
    $('#id').val(id);
    $('#kriteria').val(kriteria);
    $('#nilai').val(nilai);
}
</script>
@endsection

```

Tabel A.3 penjualan.blade.php

```

@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
                <div class="panel-heading">Data Persediaan
                    @if(Auth::user()->bagian == 'User')
                        <button type="button" class="btn-sm btn-
primary pull-right" data-toggle="modal" data-target="#create">Tambah Data
Pembelian</button>
                    @endif
                </div>

                <div class="panel-body">
                    @if (session('status'))
                        <div class="alert alert-success">
                            {{ session('status') }}
                        </div>
                    @endif

                    <table class="table table-responsive">
<thead>
<tr>
<th>ID</th>
<th>Tanggal</th>
<th>Kadar Air</th>
<th>Butir Rusak</th>
<th>Butir Warna Lain</th>
<th>BUtir Pecah</th>

```

```

        <th>Kadar Kotoran</th>
        <th>Aflatoxin</th>
        <th>Total</th>
        <th>Range</th>
        <th>Jumlah</th>
        <th>Aksi</th>
    </tr>
</thead>
<tbody>
    @foreach($data as $row)
        <tr>
            <td>{{ $row->id }}</td>
            <td>{{ $row->tanggal }}</td>
            <td>{{ $row->kadarAir->sub_kriteria }}</td>
            <td>{{ $row->butirRusak->sub_kriteria }}</td>
            <td>{{ $row->butirWarnaLain->sub_kriteria }}</td>
            <td>{{ $row->butirPecah->sub_kriteria }}</td>
            <td>{{ $row->kadarKotoran->sub_kriteria }}</td>
            <td>{{ $row->vaflatoxin->sub_kriteria }}</td>
            <td>
                <?php
                    $range = 0;
                    $tot = $row->kadarAir->hasil+$row->butirRusak->hasil+$row-
>butirWarnaLain->hasil+$row->butirPecah->hasil+$row->kadarKotoran-
>hasil+$row->vaflatoxin->hasil;
                    echo $tot;
                    if($tot <= $rg[3]->range){
                        $range = 4;
                    }else if($tot <= $rg[2]->range){
                        $range = 3;
                    }
                </php>
            </td>
        </tr>
    @endforeach
</tbody>
</table>

```

```

}else if($tot <= $rg[1]->range){
    $range = 2;
    }else{
    $range = 1;
    }
?>
</td>
<td>{{ $range }}</td>
<td>{{ abs($row->jumlah) }}</td>
<td>
@if(Auth::user()->bagian == 'User')
@if($row->validasi == 'Baru')
<button type="button" class="btn-sm btn-warning" data-
toggle="modal" data-target="#update" onClick='setData({{ $row-
>id }},'{{ $row->jumlah }})';><i class="fa fa-pencil"></i></button>
<a href="{{ route('deletePersediaan',['id' => $row->id]) }}"><button
type="button" class="btn-sm btn-danger"><i class="fa fa-
trash"></i></button></a>
@else
{{ $row->validasi }}
@endif
@endif

@if(Auth::user()->bagian == 'Pegawai')
@if($row->validasi == 'Baru')
<a href="{{ route('validasiPersediaan',['id' => $row->id,'v' => 'Tervalidasi']) }}"
class="btn-sm btn-success" ><i class="fa fa-check"></i></a>

<a href="{{ route('validasiPersediaan',['id' => $row->id,'v' => 'Tertolak']) }}"
class="btn-sm btn-danger"><i class="fa fa-close"></i></a>
@else

```

```

        {{ $row->validasi }}

    @endif
    @endif

        </td>
    </tr>
@endforeach

</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Tambah Data</h4>
            </div>
            <form action="{{ route('createPersediaan') }}"
method="post" accept-charset="utf-8">
                {{ csrf_field() }}
                <div class="modal-body">
                    <label for="">Pilihan</label>
                    <select id="pilihan" class="form-control" onchange="pilih();">
                        <option value="">Pilih Kualitas Jagung</option>
                        <option value="1">Jagung 1</option>
                        <option value="2">Jagung 2</option>
                        <option value="3">Jagung 3</option>
                        <option value="4">Jagung 4</option>
                    </select>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

        </select>
                <?php $i = 1;$j = 1; ?>
                @foreach($kriteria as $row2)
                <label for="">{{ $row2->kriteria }}</label>
<select name="{{ str_replace(' ','_',strtolower($row2->kriteria)) }}"
id="abc{{ $j }}" class="form-control" required>
<option value="">----</option>
        <?php $i = 1; ?>
        @foreach($row2->sub as $row3)
<option value="{{ $row3->id }}" id="a{{ $j }}b{{ $i++ }}">{{ $row3-
>sub_kriteria }}</option>
        @endforeach
        <?php $j++; ?>
</select>
        @endforeach
        <label for="">Jumlah</label>
        <input type="number" name="jumlah" value="" placeholder="Jumlah
Stok Input" class="form-control">
        <label>Tanggal</label>
        <input type="date" name="tanggal" value="" placeholder="Tanggal"
class="form-control">
</div>
<div class="modal-footer">
<button class="btn-sm btn-primary" type='submit'>Simpan</button>
<button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
        </div>
</form>
</div>
</div>
</div>

```

```
<script>
    function pilih(){
        var jg = $('#pilihan').val();
        var jj = <?php echo json_encode($j); ?>;
        var ji = <?php echo json_encode($i); ?>;

        for (var j = 1; j <=jj;j++) {
            for (var i = 1; i <=ji;i++) {
                a = '#a'+j+"b"+i;
                $(a).removeAttr('selected');
            }
        }
        for (var j = 1; j <=jj;j++) {
            a = '#a'+j+"b"+jg;
            $(a).attr('selected',true);
            // alert(a);
        }
        // alert(jg);
    }
</script>

<div id="update" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal">&times;</button>
                <h4 class="modal-title">Update Data</h4>
            </div>
            <form action="{{ route('updatePersediaan') }}" method="post" accept-
            charset="utf-8">
```

```

        {{csrf_field()}}
    <div class="modal-body">
    <input class="hidden" name="id" type="hidden" id="id"></input>
        <label>Jumlah</label>
    <input type="text" name="jumlah" id="jumlah" value="" placeholder="Jumlah
    Stok" class="form-control">
    </div>
        <div class="modal-footer">
    <button class="btn-sm btn-primary" type='submit'>Simpan</button>
    <button type="button" class="btn btn-default" data-
    dismiss="modal">Close</button>
        </div>
    </form>
</div>
</div>
</div>
<script>
    function setData(id,jumlah){
        // console.log(id+kriteria+nilai);
        $('#id').val(id);
        // $('#kriteria').val(kriteria);
        $('#jumlah').val(jumlah);
    }
</script>
@endsection

```

Tabel A.4 persediaan.blade.php

```

@extends('layouts.app2')

@section('content')

```



```

<div class="container">
  <div class="row">
    <div class="col-md-11">
      <div class="panel panel-default">
        <div class="panel-heading">Data Persediaan
          <button type="button" class="btn-sm btn-
primary pull-right" data-toggle="modal" data-target="#create">Tambah Data
Stok</button>
        </div>

        <div class="panel-body">
          @if (session('status'))
          <div class="alert alert-success">
            {{ session('status') }}
          </div>
          @endif

          <table class="table table-responsive">
            <thead>
              <tr>
                <th>ID</th>
                <th>Tanggal</th>
                <th>Kadar Air</th>
                <th>Butir Rusak</th>
                <th>Butir Warna Lain</th>
                <th>BUtir Pecah</th>
                <th>Kadar Kotoran</th>
                <th>Aflatoxin</th>
                <th>Total</th>
                <th>Range</th>
                <th>Stok</th>
                <th>Aksi</th>
            
```

```

        </tr>
    </thead>
<tbody>
    @foreach($data as $row)
<tr>
    <td>{{ $row->id }}</td>
    <td>{{ $row->tanggal }}</td>
    <td>{{ $row->kadarAir->sub_kriteria }}</td>
    <td>{{ $row->butirRusak->sub_kriteria }}</td>
    <td>{{ $row->butirWarnaLain->sub_kriteria }}</td>
    <td>{{ $row->butirPecah->sub_kriteria }}</td>
    <td>{{ $row->kadarKotoran->sub_kriteria }}</td>
    <td>{{ $row->vaflatoxin->sub_kriteria }}</td>
    <td>
        <?php
            $range = 0;
            $tot = $row->kadarAir->hasil+$row->butirRusak->hasil+$row->
            >butirWarnaLain->hasil+$row->butirPecah->hasil+$row->kadarKotoran-
            >hasil+$row->vaflatoxin->hasil;
            echo $tot;

            if($tot <= $rg[3]->range){
                $range = 4;
            }else if($tot <= $rg[2]->range){
                $range = 3;
            }else if($tot <= $rg[1]->range){
                $range = 2;
            }else{
                $range = 1;
            }
        </td>
    </tr>
    </tbody>
</table>

```

```

?>
    </td>

    <td>{{ $range }}</td>
    <td>{{ $row->jumlah }}</td>
    <td>
        <!-- <a href="{{ route('SubKriteria',['id' => $row->id]) }}"><button
type="button" class="btn-sm btn-info"><i class="fa fa-list"></i></button></a>
-->

        @if(Auth::user()->bagian == 'Pegawai')
            @if($row->validasi == 'Baru')

                <button type="button" class="btn-sm btn-warning" data-toggle="modal" data-
target="#update" onClick='setData({{ $row->id }},{{ $row->jumlah }})';><i
class="fa fa-pencil"></i></button>

                <a href="{{ route('deletePersediaan',['id' => $row->id]) }}"><button
type="button" class="btn-sm btn-danger"><i class="fa fa-
trash"></i></button></a>

                @else

                    {{ $row->validasi }}

                                                    @endif
                                                    @endif

        @if(Auth::user()->bagian == 'Pimpinan')
            @if($row->validasi == 'Baru')

                <a href="{{ route('validasiPersediaan',['id' => $row->id,'v' =>
'Tervalidasi']) }}" class="btn-sm btn-success" ><i class="fa fa-
check"></i></a>

                <a href="{{ route('validasiPersediaan',['id' => $row->id,'v' =>
'Tertolak']) }}" class="btn-sm btn-danger"><i class="fa fa-close"></i></a>

                @else
    
```

```

        {{ $row->validasi }}
                                                    @endif
                                                    @endif
                                                    </td>
        </tr>
    @endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-
dismiss="modal">&times;</button>
                <h4 class="modal-title">Tambah Data</h4>
            </div>
            <form action="{{ route('createPersediaan') }}"
method="post" accept-charset="utf-8">
                {{ csrf_field() }}
                <div class="modal-body">
                    @foreach($kriteria as $row2)
                        <label for="">{{ $row2-
>kriteria }}</label>
                            <select name="{{ str_replace(
','_', strtolower($row2->kriteria) }}" id="{{ str_replace(
','_', strtolower($row2-
>kriteria) }}" 1" class="form-control" required>

```

```

<option value="">----</option>
@foreach($row2->sub as $row3)
<option value="{{ $row3-
>id }}">{{ $row3->sub_kriteria }}</option>
@endforeach
</select>
@endforeach
<label for="">Jumlah</label>
<input type="number" name="jumlah"
value="" placeholder="Jumlah Stok Input" class="form-control">
<label>Tanggal</label>
<input type="date" name="tanggal"
value="" placeholder="Tanggal" class="form-control">
<input type="hidden" name="jenis"
value="Persediaan">
</div>
<div class="modal-footer">
<button class="btn-sm btn-primary"
type='submit'>Simpan</button>
<button type="button" class="btn btn-
default" data-dismiss="modal">Close</button>
</div>
</form>
</div>
</div>
</div>
<div id="update" class="modal fade" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">

```

```

        <button type="button" class="close" data-
dismiss="modal">&times;</button>
        <h4 class="modal-title">Update Data</h4>
    </div>
    <form action="{{route('updatePersediaan')}}"
method="post" accept-charset="utf-8">
        {{csrf_field()}}
        <div class="modal-body">
            <input class="hidden" name="id"
type="hidden" id="id"></input>
            <label>Jumlah</label>
            <input type="text" name="jumlah"
id="jumlah" value="" placeholder="Jumlah Stok" class="form-control">
        </div>
        <div class="modal-footer">
            <button class="btn-sm btn-primary"
type='submit'>Simpan</button>
            <button type="button" class="btn btn-
default" data-dismiss="modal">Close</button>
        </div>
    </form>
</div>
</div>
</div>
</script>
function setData(id,jumlah){
    // console.log(id+kriteria+nilai);
    $('#id').val(id);
    $('#kriteria').val(kriteria);
    $('#jumlah').val(jumlah);
}

```

```
</script>
@endsection
```

Tabel A.5 range.blade.php

```
@extends('layouts.app2')

@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-11">
      <div class="panel panel-default">
        <div class="panel-heading">DATA JAGUNG
          <!-- <button type="button" class="btn-sm
btn-primary pull-right" data-toggle="modal" data-target="#create">Tambah
Kriteria</button> -->
        </div>
        <div class="panel-body">
          @if (session('status'))
            <div class="alert alert-success">
              {{ session('status') }}
            </div>
          @endif
        <table class="table table-responsive">
          <thead>
            <tr>
              <th>No</th>
              <th>Identitas</th>
              <th>Nilai Range</th>
              <th>Jumlah Stok</th>
```

```

        </tr>
    </thead>
    <tbody>
        <?php $i = 1; ?>
        @foreach($data as $row)
    <tr>
        <td>{{ $i }}</td>
        <td>Kualitas {{ $i++ }}</td>
        <td>> {{(isset($data[$i-1])?$data[$i-1]->range:0)}} dan
<= {{ $row->range }}</td>
        <td>{{(isset($st[$i-1])?$st[( $i-1)]:0)}}</td>
    </tr>
        @endforeach
    </tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
@endsection

```

Tabel A.6 rop.blade.php

```

@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">

```



```

<div class="panel-heading">ROP
    @if(Auth::user()->bagian != 'User')
    <button type="button" class="btn-sm btn-
primary pull-right" data-toggle="modal" data-target="#create">Tambah
ROP</button>
    @endif
</div>

<div class="panel-body">
    @if (session('status'))
    <div class="alert alert-success">
        {{ session('status') }}
    </div>
    @endif
    <table class="table table-responsive">
    <thead>
    <tr>
    <th>ID</th>
    <th>Nama Barang</th>
    <th>Biaya Pesan</th>
    <th>Kebutuhan Bahan Baku Per Periode</th>
    <th>Biaya Simpan</th>
    <th>Jumlah Tahun Kerja</th>
    <th>Kebutuhan Bahan Baku Per Hari</th>
    <th>Waktu Tunggu Barang (hari)</th>
    <th>Stok Aman</th>
    <th>Lead Time</th>
    <th>Rop</th>
    <th>Aksi</th>
    </tr>
    </thead>

```

```
|  |
| --- |
| @foreach($data as $row)                 <tr>                 <td>{{ $row->id }}</td>                 <td>{{ $row->jagung }}</td>                 <td>{{ $row->biaya_pesan }}</td>                 <td>{{ $row->kebutuhan_per_periode }}</td>                 <td>{{ $row->biaya_simpan }}</td>                 <td>{{ $row->jumlah_waktu }}</td>                 <td>{{ $row->kebutuhan_per_hari }}</td>                 <td>{{ $row->waktu_tunggu }}</td>                 <td>{{ $row->stok_aman }}</td>                 <td>{{ $row->lead_time }}</td>                 <td>{{ $row->rop }}</td>                 <td>                     @if(Auth::user()->bagian != 'User')                     <!-- <button type="button" class="btn-sm btn-warning" data-                     toggle="modal" data-target="#update" onClick='setData({{ $row-                     >id }},{{ $row->kriteria }},{{ $row->nilai }});'><i class="fa fa-                     pencil"></i></button> -->                     <a href="{{ route('deleteRop',['id' => $row->id]) }}"><button                     type="button" class="btn-sm btn-danger"><i class="fa fa-                     trash"></i></button></a>                     @endif                 </td>                 </tr>                 @endforeach             </tbody>         </table>     </div> </div> |

```

```

        </div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
<button type="button" class="close" data-dismiss="modal">&times;</button>
<h4 class="modal-title">Tambah Data</h4>
            </div>
<form action="{{ route('createRop') }}" method="post" accept-charset="utf-8">
                {{ csrf_field() }}
                <div class="modal-body">
                    <label>Nama Barang</label>
                    <select name="jagung" id="" class="form-control">
                        <option value="">Pilih Kategori Jagung</option>
                        <option value="Jagung 1">Jagung 1</option>
                        <option value="Jagung 2">Jagung 2</option>
                        <option value="Jagung 3">Jagung 3</option>
                        <option value="Jagung 4">Jagung 4</option>
                    </select>
                    <label>Biaya Pesan</label>
                    <input type="number" class="form-control" name="biaya_pesan"
placeholder="Dalam Rupiah">
                    <label>Kebutuhan Bahan Baku Per Periode</label>
                    <input
                        type="number"
                        class="form-control"
name="kebutuhan_per_periode" placeholder="Dalam Kg">
                    <label>Biaya Simpan</label>
                    <input type="number" class="form-control" name="biaya_simpan"
placeholder="Dalam Rupiah">
                    <label>Jumlah Tahun Kerja</label>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        <input type="number" class="form-control" name="jumlah_waktu"
placeholder="">
        <label>Kebutuhan Bahan Baku Per Hari</label>
        <input type="number" class="form-control"
name="kebutuhan_per_hari" placeholder="Dalam Kg">
        <label>Waktu Tunggu Barang (hari)</label>
        <input type="number" class="form-control" name="waktu_tunggu"
placeholder="Dalam Hari">
        <label>Stok Aman</label>
        <input type="number" class="form-control" name="stok_aman"
placeholder="Dalam Kg">
        <label for="">Lead Time</label>
        <input type="number" class="form-control" name="lead_time"
placeholder="Dalam Hari">
        <input type="hidden" name="tanggal" value="{{ date('Y-m-d') }}">
        </div>
        <div class="modal-footer">
        <button class="btn-sm btn-primary" type='submit'>Simpan</button>
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
        </div>
        </form>
        </div>
</div>

<div id="update" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">

```

```

<button type="button" class="close" data-
dismiss="modal">&times;</button>
<h4 class="modal-title">Update Data</h4>
</div>
<form action="{{route('updateRop')}}" method="post" accept-
charset="utf-8">
    {{csrf_field()}}
    <div class="modal-body">
        <input class="hidden" name="id" type="hidden" id="id"></input>
        <label>Nama Barang</label>
        <select name="jagung" id="" class="form-control">
            <option value="">-----</option>
            <option value="Jagung 1">Jagung 1</option>
            <option value="Jagung 2">Jagung 2</option>
            <option value="Jagung 3">Jagung 3</option>
            <option value="Jagung 4">Jagung 4</option>
        </select>
        <label>Biaya Pesan</label>
        <input type="number" class="form-control" name="biaya_pesan">
        <label>Kebutuhan Bahan Baku Per Periode</label>
        <input type="number" class="form-control"
name="kebutuhan_per_periode">
        <label>Biaya Simpan</label>
        <input type="number" class="form-control" name="biaya_simpan">
        <label>Jumlah Tahun Kerja</label>
        <input type="number" class="form-control" name="jumlah_waktu">
        <label>Kebutuhan Bahan Baku Per Hari</label>
        <input type="number" class="form-control"
name="kebutuhan_per_hari">
        <label>Waktu Tunggu Barang (hari)</label>
        <input type="number" class="form-control" name="waktu_tunggu">
    
```

```

<label>Stok Aman</label>
<input type="number" class="form-control" name="stok_aman">
    </div>
<div class="modal-footer">
<button class="btn-sm btn-primary" type='submit'>Simpan</button>
<button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
</div>
    </form>
</div>
</div>
</div>
<script>
    function setData(id,kriteria,nilai){
        console.log(id+kriteria+nilai);
        $('#id').val(id);
        $('#kriteria').val(kriteria);
        $('#nilai').val(nilai);}
</script>
@endsection

```

Tabel A.7 subkriteria.blade.php

```

@extends('layouts.app2')
@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
                <div class="panel-heading">Sub Kriteria "{{ $kriteria->kriteria }}"

```

```

        <button type="button" class="btn-sm btn-primary pull-right" data-
toggle="modal" data-target="#create">Tambah Sub Kriteria</button>
        <a href="{{route('Kriteria')}}"><button type="button" class="btn-sm
btn-danger pull-right">Kembali</button></a>
    </div>

    <div class="panel-body">
        @if (session('status'))
        <div class="alert alert-success">
            {{ session('status') }}
        </div>
        @endif
        <table class="table table-responsive">
<thead>
    <tr>
        <th>ID</th>
        <th>Kriteria</th>
        <th>Bobot</th>
        <th>Keterangan</th>
        <th>Aksi</th>
    </tr>
</thead>
<tbody>
    @foreach($data as $row)
        <tr>
            <td>{{ $row->id }}</td>
            <td>{{ $row->sub_kriteria }}</td>
            <td>{{ $row->nilai }}</td>
            <td>{{ $row->keterangan }}</td>
            <td><button type="button" class="btn-sm btn-warning" data-
toggle="modal" data-target="#update" onClick='setData({{ $row-

```

```

>id}}, "{{ $row->sub_kriteria}}", "{{ $row->nilai}}", "{{ $row-
>keterangan}}");'>Update</button>
<a href="{{ route('deleteSubKriteria',['id' => $row->id]) }}"><button
type="button" class="btn-sm btn-danger">Delete</button></a>
</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<div id="create" class="modal fade" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-
dismiss="modal">&times;</button>
<h4 class="modal-title">Tambah Data</h4>
</div>
<form action="{{ route('createSubKriteria') }}" method="post" accept-
charset="utf-8">
{{ csrf_field() }}
<div class="modal-body">
<input type="hidden" name="kriteria_id" value="{{ $id }}"
placeholder="">
<label>Nama Kriteria</label>
<input type="text" name="sub_kriteria" value="" placeholder="Nama
SubKriteria" class="form-control">

```



```

        <label>Bobot/Nilai</label>
        <input type="text" name="nilai" value="" placeholder="Nilai
SubKriteria" class="form-control">
        <label>Keterangan</label>
        <textarea name="keterangan" class="form-control"></textarea>
        </div>
        <div class="modal-footer">
        <button class="btn-sm btn-primary" type='submit'>Simpan</button>
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
        </div>
        </form>
    </div>
</div>
</div>
<div id="update" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-
dismiss="modal">&times;</button>
                <h4 class="modal-title">Update Data</h4>
            </div>
            <form action="{{route('updateSubKriteria')}}" method="post" accept-
charset="utf-8">
                {{csrf_field()}}
            <div class="modal-body">
                <input class="hidden" name="id" type="hidden" id="id"></input>
                <label>Nama Kriteria</label>
    
```

```

        <input type="text" id="kriteria" name="sub_kriteria" value=""
placeholder="nama kriteria" class="form-control">
        <label>Bobot/Nilai</label>
        <input type="text" name="nilai" id="nilai" value="" placeholder="Nilai
Kriteria" class="form-control">
        <label>Keterangan</label>
        <textarea name="keterangan" id="keterangan" class="form-
control"></textarea>
    </div>
<div class="modal-footer">
    <button class="btn-sm btn-primary" type='submit'>Simpan</button>
    <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
    </div>
</form>
</div>
</div>
</div>
<script>
    function setData(id,kriteria,nilai,keterangan){
        console.log(id+kriteria+nilai);
        $('#id').val(id);
        $('#kriteria').val(kriteria);
        $('#nilai').val(nilai);
        $('#keterangan').val(keterangan);
    }
</script>
@endsection

```

Tabel A.8 user.blade.php

```

@extends('layouts.app2')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-11">
            <div class="panel panel-default">
                <div class="panel-heading">Data User
            </div>

            <div class="panel-body">
                @if (session('status'))
                <div class="alert alert-success">
                    {{ session('status') }}
                </div>
                @endif
            </div>
        </div>
    </div>
</div>

<table class="table table-responsive">

<thead>
    <tr>
        <th>ID</th>
        <th>Username</th>
        <th>Email</th>
        <th>Bagian</th>
        <th>Aksi</th>
    </tr>
</thead>
<tbody>

```

```

                @foreach($data as $row)

                <tr>

                <td>{{ $row->id }}</td>

                <td>{{ $row->name }}</td>

                <td>{{ $row->email }}</td>

                <td>{{ $row->bagian }}</td>

</td>

<a href="{{ route('deleteUser',['id' => $row->id]) }}"><button type="button"
class="btn-sm btn-danger">Delete</button></a>

</td>

</tr>

                @endforeach

                </tbody>

                </table>

                </div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

                @endsection
    
```

Tabel A.9 jenis.blade.php

```

    @extends('layouts.app2')

    @section('content')

    <div class="container">

        <div class="row">

            <div class="col-md-8 col-md-offset-2">

                <div class="panel panel-default">

                    <div class="panel-heading">Data Master Jenis
    
```

```

        <button type="button" class="btn btn-info btn-sm pull-right" data-
toggle="modal" data-target="#tambahData">Tambah Data</button>
    </div>
    <div class="panel-body">
        @if (session('status'))
        <div class="alert alert-success">
            {{ session('status') }}
        </div>
        @endif

        <table class="table table-responsive">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Jenis</th>
                </tr>
            </thead>
            <tbody>
                @foreach($data as $row)
                <tr>
                    <td>{{ $row->id }}</td>
                    <td>{{ $row->jenis }}</td>
                    <td>
                        <a href="{{ route('JenisDelete',['id' => $row->id]) }}"
title="Hapus" class="btn btn-danger btn-sm">Hapus</a>
                    </td>
                </tr>
                @endforeach
            </tbody>
        </table>
    </div>

```

```

        </div>
    </div>
</div>
</div>
<div id="tambahData" class="modal fade" role="dialog">
    <form action="{{route('JenisCreate')}}" method="post" accept-charset="utf-8">
        {{csrf_field()}}
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <button type="button" class="close" data-dismiss="modal">&times;</button>
                    <h4 class="modal-title">Tambah Data</h4>
                </div>
                <div class="modal-body">
                    <label>Jenis</label>
                    <input type="text" name="jenis" class="form-control">
                </div>
                <div class="modal-footer">
                    <button type="submit" class="btn btn-Primary">Simpan</button>
                    <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
                </div>
            </div>
        </div>
    </div>
</form>
</div>
@endsection

```

Tabel A.10 Jenis.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Jenis extends Model
{
    protected $table = 'jenis';
    protected $fillable = ['jenis'];
    public $timestamps = false;
}
```

Tabel A.11 Kriteria.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Kriteria extends Model
{
    protected $table = 'kriteria';
    protected $fillable = ['kriteria','nilai'];
    public $timestamps = false;

    public function sub(){
        return $this->hasMany('App\VwSubKriteria','kriteria_id');
    }
}
```

```
public function vwN(){
    return $this->belongsTo('App\VwNormalisasi','id','id');
}
}
```

Tabel A.12 Penilaian.php

```
<?php

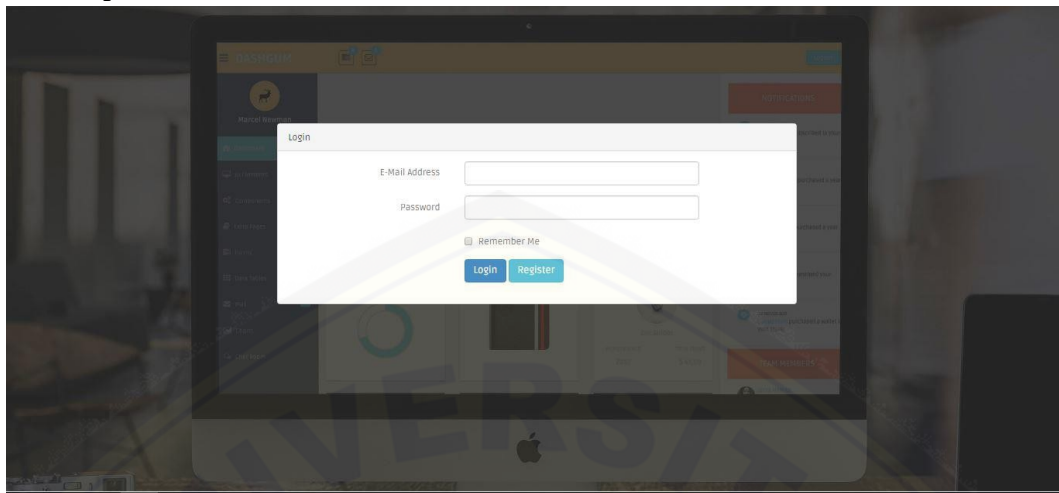
namespace App;

use Illuminate\Database\Eloquent\Model;

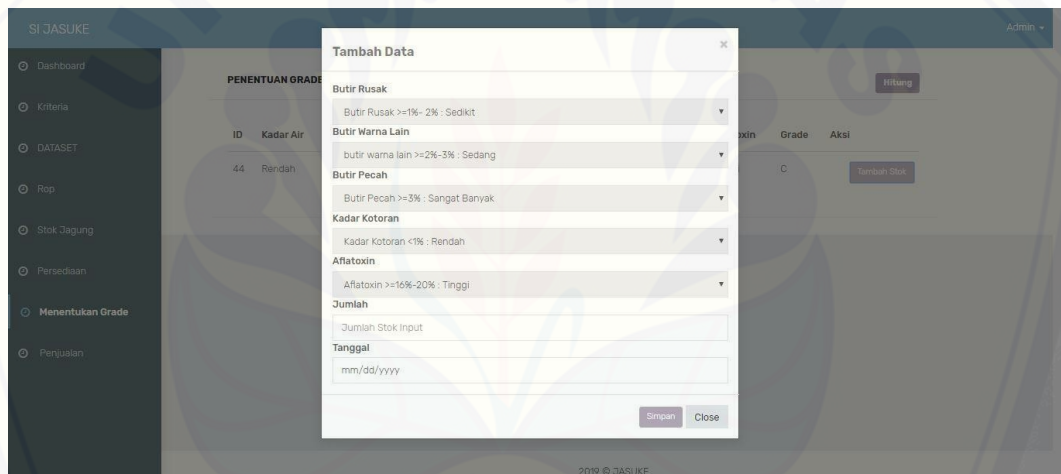
class Penilaian extends Model
{
    protected $table = 'keuangan';
    public $timestamps = false;
    protected $fillable = ['user_id','jenis','jumlah','keterangan'];

    function user(){
        return $this->belongsTo('App\User','user_id','id');
    }
}
```

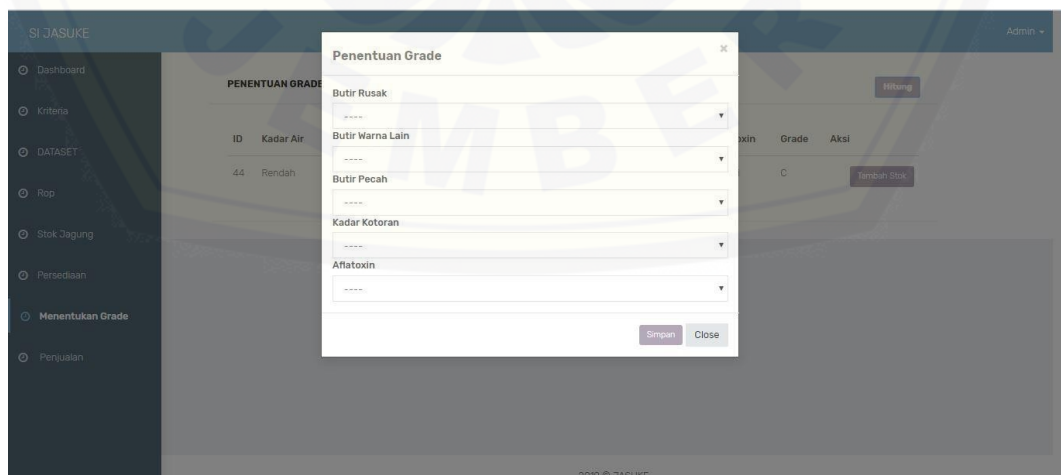

B. Tampilan Sistem



Gambar B.1 Tampilan Login



Gambar B.2 Tampilan Admin Tambah Stok Grade



Gambar B.3 Tampilan Admin Hitung Grade

PENENTUAN GRADE

ID	Kadar Air	Butir Rusak	Butir Warna Lain	Butir Pecah	Kadar Kotoran	Aflatoxin	Grade	Aksi
44	Rendah	Sedikit	Sedang	Sangat Banyak	Rendah	Tinggi	C	Tambah Stok

Gambar B.4 Tampilan Admin Lihat Penentuan Grade

DATA PERSEDIAAN

ID	Tanggal	Kadar Air	Butir Rusak	Butir Warna Lain	Butir Pecah	Kadar Kotoran	Aflatoxin	Kualitas	Stok	Aksi
18	2018-12-21	Kadar Air >=15%	Butir Rusak >=8%	butir warna lain >=10%	Butir Pecah >=3%	Kadar Kotoran >=2%	Aflatoxin > 20%	A	1	Tervalidasi
19	2018-12-12	Kadar Air >=1%-4%	Butir Rusak >=1%- 2%	butir warna lain <=1%	Butir Pecah <=1%	Kadar Kotoran <=1%	Aflatoxin <=1%- 5%	D	1	Tervalidasi
20	2018-12-22	Kadar Air >=1%-4%	Butir Rusak >=6%- 7%	butir warna lain >=2%- 3%	Butir Pecah >=1%	Kadar Kotoran >=2%	Aflatoxin >=16%- 20%	D	3	Tervalidasi
21	2018-12-14	Kadar Air >=10%-14%	Butir Rusak >=8%	butir warna lain >=4%-7%	Butir Pecah >=1%	Kadar Kotoran >=2%	Aflatoxin > 20%	B	3	Tervalidasi
24	2018-12-12	Kadar Air >=10%-14%	Butir Rusak >=6%- 7%	butir warna lain >=4%-7%	Butir Pecah >=2%	Kadar Kotoran >=2%	Aflatoxin >=6%-15%	C	100	Tervalidasi
25	2019-03-14	Kadar Air >=1%-4%	Butir Rusak >=1%- 2%	butir warna lain <=1%	Butir Pecah <=1%	Kadar Kotoran <=1%	Aflatoxin <=1%- 5%	D	165	Tervalidasi
27	2019-07-10	Kadar Air >=10%-14%	Butir Rusak >= 3%- 4%	butir warna lain <=1%	Butir Pecah >= 3%	Kadar Kotoran >=1%	Aflatoxin <=1%- 5%	C	1222	Tervalidasi

Gambar B.5 Tampilan Admin Lihat Data Persediaan

DATA JAGUNG

No	Kualitas	Jumlah Stok
1	Kualitaa A	44
2	Kualitas B	993
3	Kualitas C	1325
4	Kualitas D	171

Gambar B.6 Tampilan Admin Lihat Data Stok

SI JASUKE Admin

Dashboard

Kriteria

DATASET

ROP

Stok Jagung

Persediaan

Menentukan Grade

Penjualan

2019 © JASUKE

ID	Nama Barang	Biaya Pesan	Kebutuhan Bahan Baku Per Periode	Biaya Simpan	Jumlah Tahun Kerja	Kebutuhan Bahan Baku Per Hari	Waktu Tunggu Barang (hari)	Stok Aman	Lead Time	Rop
5	B	10000	5000	5000	1	100	1	500	0	500
4	A	500000	14	12000	1	-1	2	5	3	8

Gambar B.7 Tampilan Admin Lihat Data ROP

SI JASUKE Admin

Dashboard

Kriteria

DATASET

ROP

Stok Jagung

Persediaan

Menentukan Grade

Penjualan

ID	Kadar Air	Butir Rusak	Butir Warna Lain	Butir Pecah	Kadar Kotoran	Aflatoxin	Grade
1	Rendah	Sedang	Banyak	Sedang	Rendah	Rendah	A
2	Tinggi	Sedang	Sedikit	Sedang	Tinggi	Sedang	C
3	Rendah	Banyak	Sedang	Sangat Banyak	Tinggi	Tinggi	C
4	Rendah	Sedang	Sedang	Sedikit	Rendah	Sedang	B
5	Rendah	Sedang	Banyak	Sedikit	Rendah	Tinggi	B
6	Tinggi	Sedang	Sedikit	Banyak	Rendah	Sedang	C
7	Rendah	Banyak	Sedang	Sedikit	Tinggi	Sedang	D
8	Sangat Tinggi	Sedikit	Sedang	Banyak	Rendah	Tinggi	C
9	Tinggi	Sedang	Sedikit	Banyak	Sedang	Rendah	C
10	Sedang	Sedang	Sedang	Banyak	Rendah	Rendah	B

Gambar B.8 Tampilan Admin Lihat Data Training

SI JASUKE Admin

Dashboard

Kriteria

DATASET

ROP

Stok Jagung

Persediaan

Menentukan Grade

Penjualan

ID	Kriteria	Sub	Nilai	Aksi
2	Butir Rusak	Butir Rusak >=1%- 2% Butir Rusak >=3%- 4% Butir Rusak >=6%- 7% Butir Rusak >=8%	Sedikit Sedang Banyak Sangat Banyak	
3	Butir Warna Lain	butir warna lain <=1% butir warna lain >=2%-3% butir warna lain >=4%-7% butir warna lain >=10%	Sedikit Sedang Banyak Sangat Banyak	
4	Butir Pecah	Butir Pecah <1% Butir Pecah =1% Butir Pecah >2% Butir Pecah >=3%	Sedikit Sedang Banyak Sangat Banyak	
5	Kadar Kotoran	Kadar Kotoran <1% Kadar Kotoran =1% Kadar Kotoran >2% Kadar Kotoran >2%	Rendah Sedang Tinggi Sangat Tinggi	
6	Aflatoxin	Aflatoxin <=1%- 5% Aflatoxin >=6%- 15%	Rendah Sedang	

Gambar B.9 Tampilan Admin Lihat Data Kriteria

DATA PENJUALAN

ID	No User	Nama User	Tanggal	Kualitas	Jumlah	Aksi
22	2	Admin	2018-12-12	A	5	Tervalidasi
23	2	Admin	2018-12-27	D	5	Tervalidasi
26	2	Admin	2019-07-19	D	100	Tertolak
35	2	Admin	2019-11-21	A	56	Tervalidasi
36	2	Admin	2019-11-25	A	5	Tervalidasi
37	2	Admin	2019-11-21	A	3	Tervalidasi
38	3	Pembeli 1	2020-03-19	B	10	Tervalidasi
39	2	Admin	2020-03-19	C	1	<input type="checkbox"/> <input type="checkbox"/>

Gambar B.10 Tampilan Admin Validasi Data Penjualan

DATA PERSEDIAAN

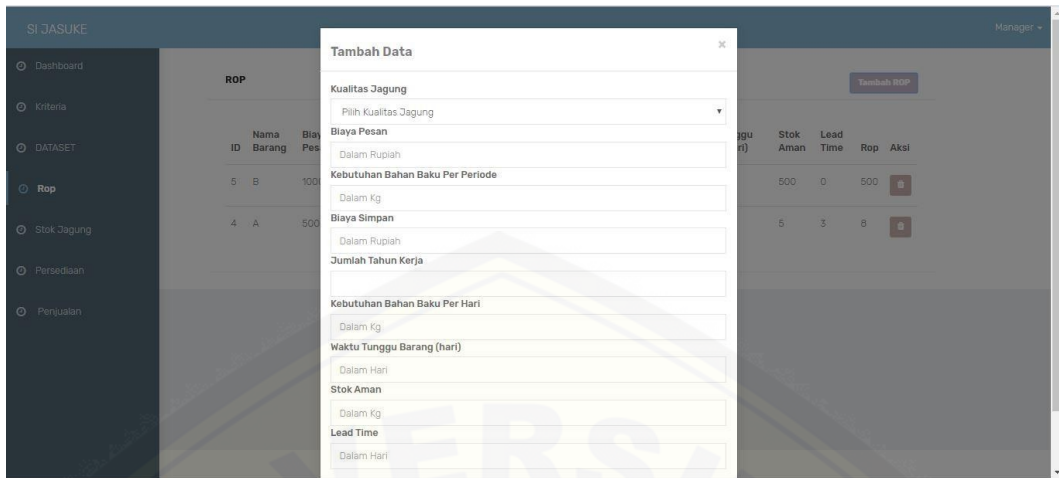
ID	Tanggal	Kadar Air	Butir Rusak	Butir Warna Lain	Butir Pecah	Kadar Kotoran	Aflatoxin	Kualitas	Stok	Aksi
18	2018-12-21	Kadar Air >=15%	Butir Rusak >=8%	butir warna lain >=10%	Butir Pecah >=3%	Kadar Kotoran >2%	Aflatoxin > 20%	A	1	Tervalidasi
19	2018-12-12	Kadar Air >=1%-4%	Butir Rusak >=1%-2%	butir warna lain <=1%	Butir Pecah <=1%	Kadar Kotoran <=1%	Aflatoxin <=1%-5%	D	1	Tervalidasi
20	2018-12-22	Kadar Air >=1%-4%	Butir Rusak >=6%-7%	butir warna lain >=2%-3%	Butir Pecah >=2%	Kadar Kotoran >2%	Aflatoxin >=16%-20%	C	3	Tervalidasi
21	2018-12-14	Kadar Air >=10%-14%	Butir Rusak >=8%	butir warna lain >=4%-7%	Butir Pecah >=1%	Kadar Kotoran >=2%	Aflatoxin > 20%	B	3	Tervalidasi
24	2018-12-12	Kadar Air >=10%-14%	Butir Rusak >=6%-7%	butir warna lain >=4%-7%	Butir Pecah >=2%	Kadar Kotoran >=2%	Aflatoxin >=6%-15%	C	100	Tervalidasi
25	2019-03-14	Kadar Air >=1%-4%	Butir Rusak >=1%-2%	butir warna lain <=1%	Butir Pecah <=1%	Kadar Kotoran <=1%	Aflatoxin <=1%-5%	D	165	Tervalidasi
27	2019-07-19	Kadar Air >=11%-14%	Butir Rusak >=5%-6%	butir warna lain <=1%	Butir Pecah >=5%	Kadar Kotoran >=1%	Aflatoxin <=1%-5%	C	1222	Tervalidasi

Gambar B.11 Tampilan Manager Lihat Data Persediaan

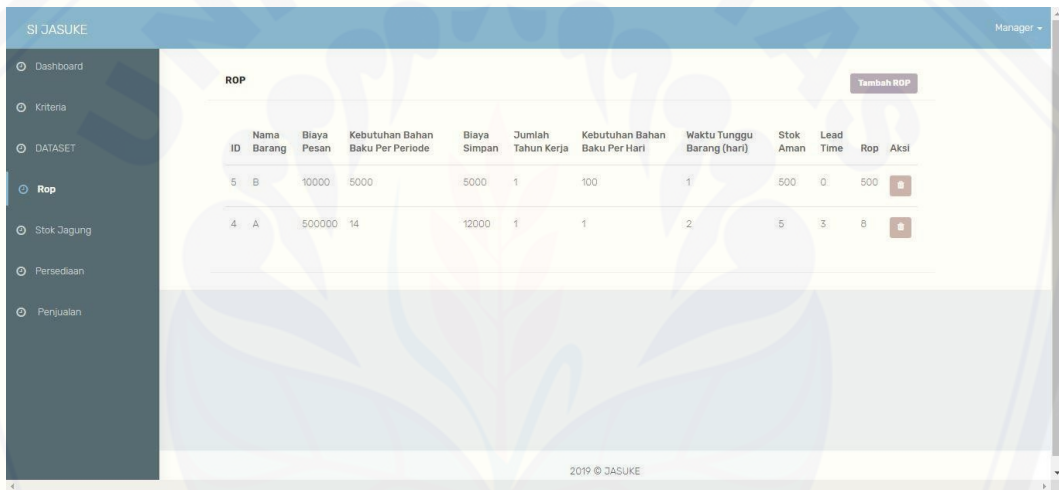
DATA JAGUNG

No	Kualitas	Jumlah Stok
1	Kualitas A	44
2	Kualitas B	993
3	Kualitas C	1325
4	Kualitas D	171

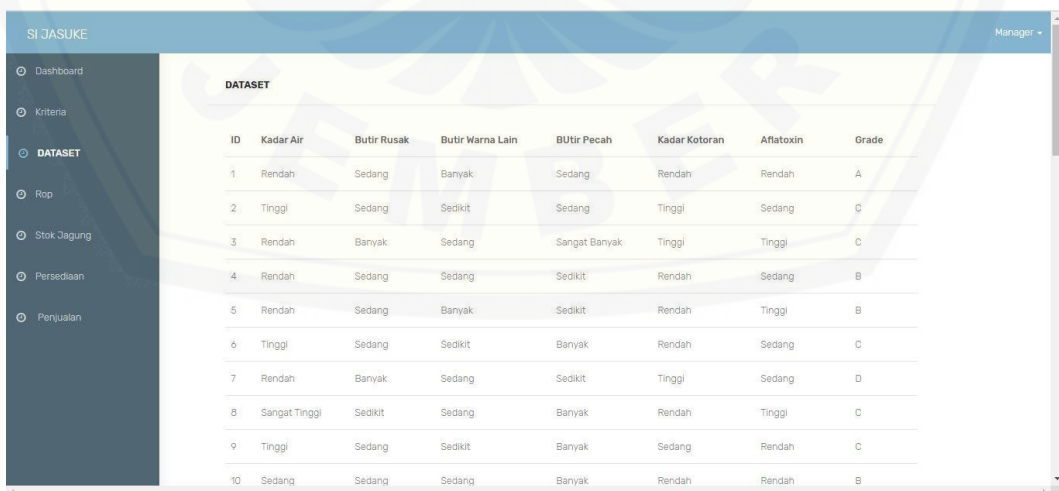
Gambar B.12 Tampilan Manager Lihat Data Jagung



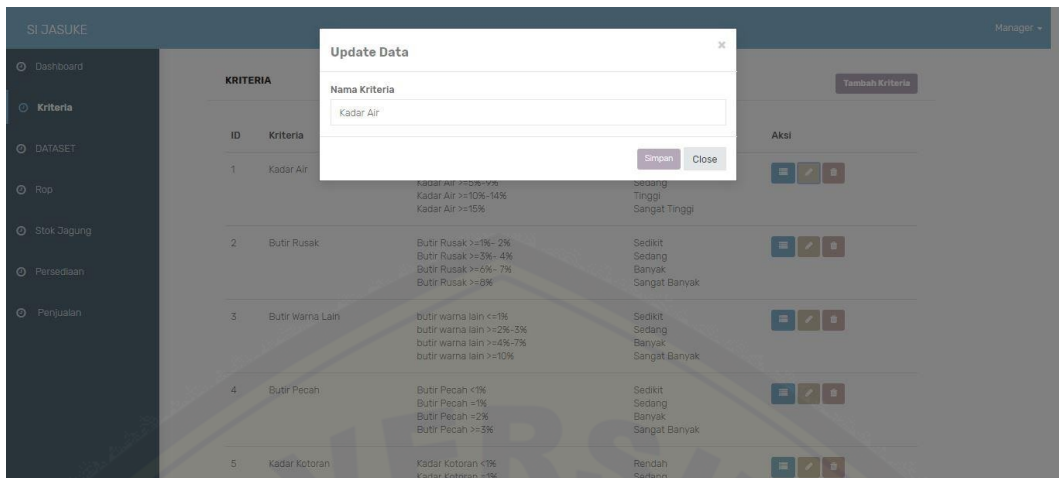
Gambar B.13 Tampilan Manager Tambah Data Jagung



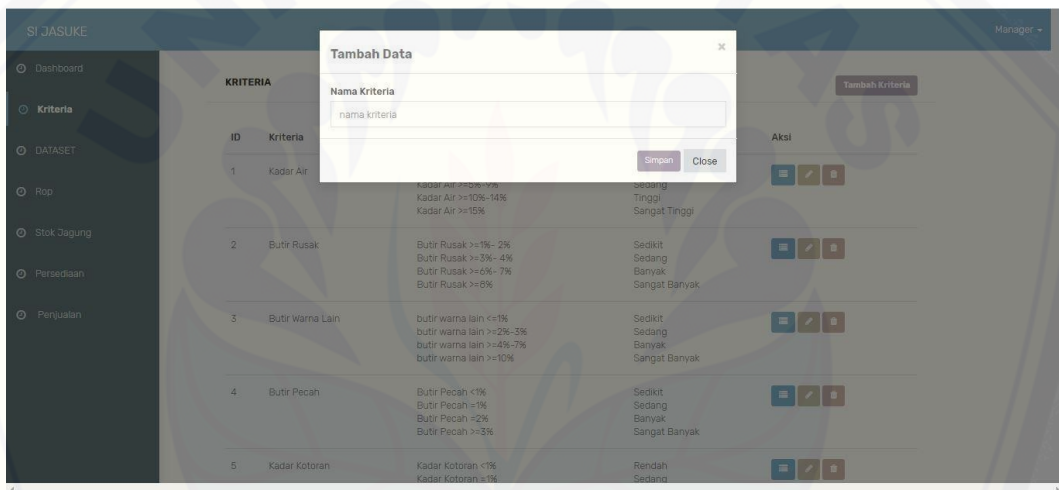
Gambar B.14 Tampilan Manager Lihat Data ROP



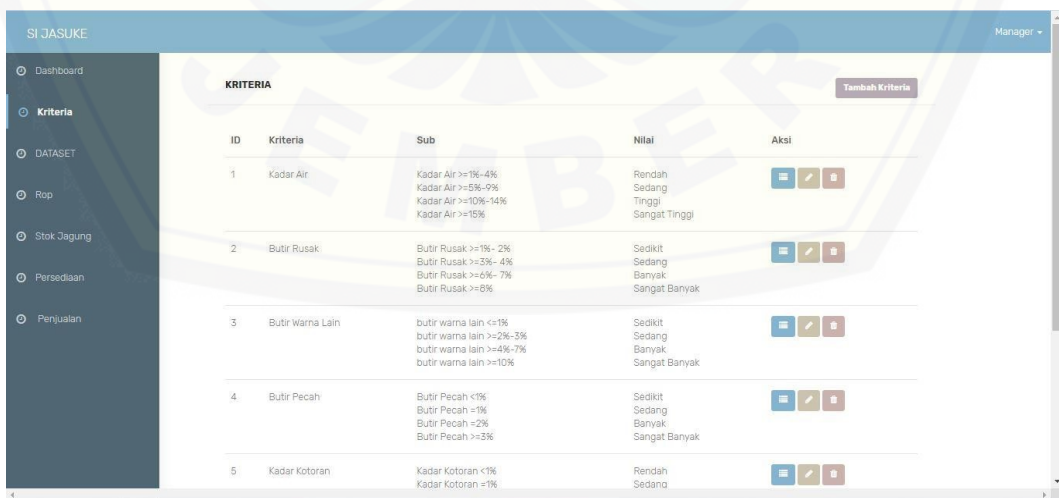
Gambar B.15 Tampilan Manager Lihat Data Training



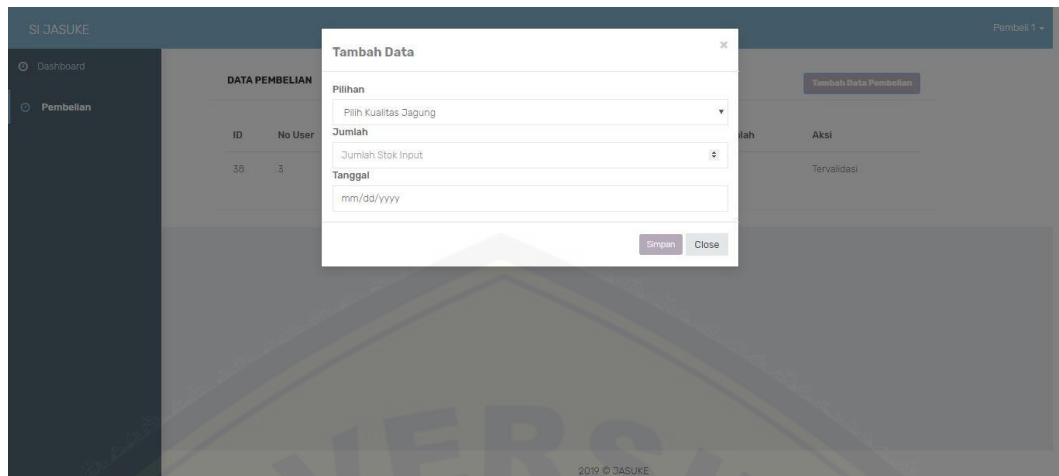
Gambar B.16 Tampilan Manager Edit Data Kriteria



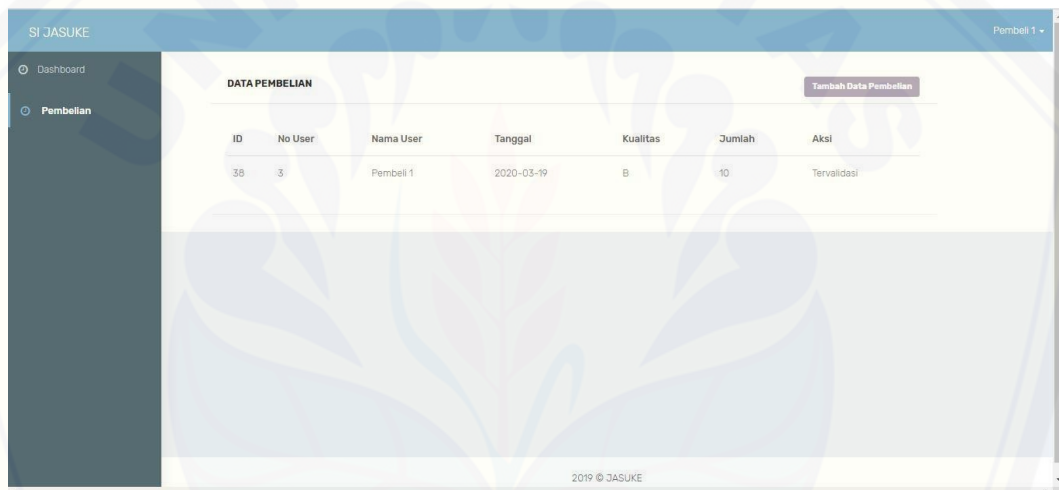
Gambar B.17 Tampilan Manager Tambah Data Kriteria



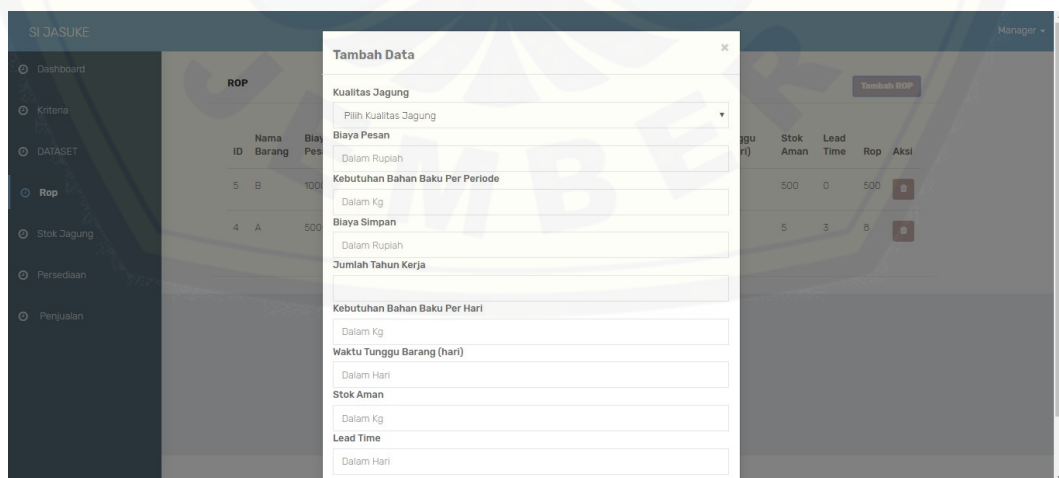
Gambar B.18 Tampilan Manager Lihat Data Kriteria



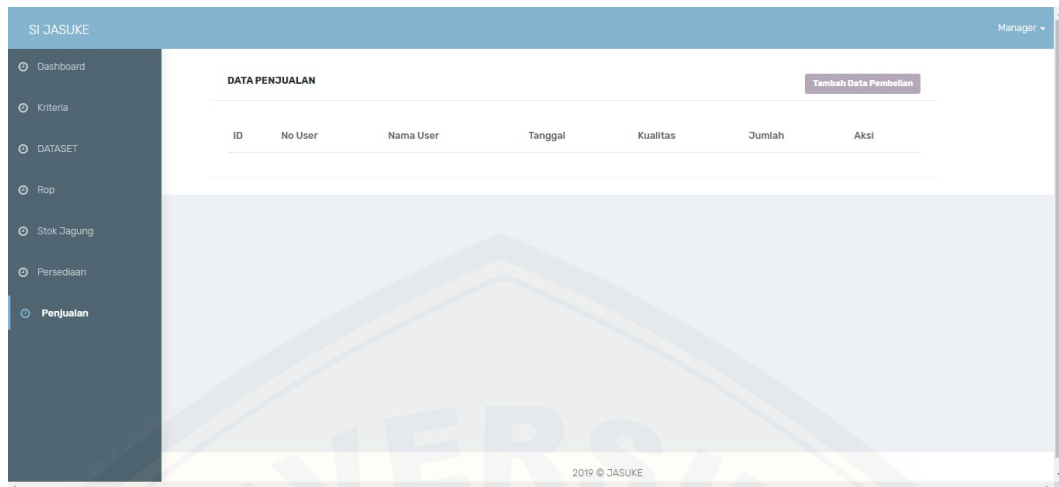
Gambar B.19 Tampilan Pembeli Tambah Pembelian



Gambar B.20 Tampilan Pembeli Lihat Validasi Pembelian



Gambar B.21 Tampilan Manager Tambah Data ROP



Gambar B.22 Tampilan Lihat Data Penjualan

C. Blackbox Testing

Tabel C.1 Hasil *Blackbox Testing*

Fitur	Usecase	CRC Card	Class Diagram	Tampilan	Status
Login	Login	login.blade.php	login.blade.php	Gambar B.1	Sesuai
Lihat data kriteria	Lihat data kriteria	kriteria.blade.php	kriteria.blade.php	Gambar B.18, Gambar B.9	Sesuai
Tambah data kriteria	Tambah data kriteria	kriteria.blade.php	kriteria.blade.php	Gambar B.17	Sesuai
Edit data kriteria	Edit data kriteria	kriteria.blade.php	kriteria.blade.php	Gambar B.16	Sesuai
Hapus data kriteria	Hapus data kriteria	kriteria.blade.php	kriteria.blade.php	Gambar B.18	Sesuai
Lihat Data Training	Lihat Data Training	jagung.blade.php	jagung.blade.php	Gambar B.15, Gambar B.8	Sesuai
Lihat data rop	Lihat data rop	rop.blade.php	rop.blade.php	Gambar B.14, Gambar B.7	Sesuai
Tambah data rop	Tambah data rop	rop.blade.php	rop.blade.php	Gambar B.21	Sesuai

Fitur	<i>Usecase</i>	<i>CRC Card</i>	<i>Class Diagram</i>	Tampilan	Status
Hapus data rop	Hapus data rop	rop.blade.php	rop.blade.php	Gambar B.14	Sesuai
Lihat data stok jagung	Lihat data stok jagung	jagung.blade.php	jagung.blade.php	Gambar B.12, Gambar B.6	Sesuai
Lihat data persediaan	Lihat data persediaan	jagung.blade.php	jagung.blade.php	Gambar B.11, Gambar B.5	Sesuai
Validasi data persediaan	Validasi data persediaan	pesanan.blade.php	pesanan.blade.php	Gambar B.10	Sesuai
Lihat data penjualan	Lihat data penjualan	pesanan.blade.php	pesanan.blade.php	Gambar B.22	Sesuai
Lihat data penentuan grade	Lihat data penentuan grade	cari-grade.blade.php	cari-grade.blade.php	Gambar B.4	Sesuai
Hitung grade	Hitung grade	cari-grade.blade.php	cari-grade.blade.php	Gambar B.3	Sesuai
Tambah stok grade	Tambah stok grade	cari-grade.blade.php	cari-grade.blade.php	Gambar B.2	Sesuai
Tambah data pembelian	Tambah data pembelian	pesanan.blade.php	pesanan.blade.php	Gambar B.19	Sesuai
Lihat data validasi	Lihat data validasi	pesanan.blade.php	pesanan.blade.php	Gambar B.20	Sesuai