**TUGAS AKHIR**



Oleh:

JEVON ANANTA GUNAWAN NRP: 160420097

**DATA SCIENCE & ARTIFICIAL INTELLIGENCE PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS**

**TEKNIK**

**UNIVERSITAS SURABAYA**

**2024**

**TUGAS AKHIR**

Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana



Oleh:

**JEVON ANANTA GUNAWAN NRP: 160420097**

**DATA SCIENCE & ARTIFICIAL INTELLIGENCE PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS**

**TEKNIK**

**UNIVERSITAS SURABAYA**

**2024**

**HALAMAN PENGESAHAN**

Skripsi ini diajukan oleh :

Nama : Jevon Ananta Gunawan

NRP : 160420097

Program Studi : Teknik Informatika

Judul Skripsi : Pembuatan Aplikasi Sistem Pengenalan Aksara Bali Menggunakan Metode *Convolutional Neural Network*

**Telah diperiksa Dosen Pembimbing dan berhasil dipertahankan dihadapan Tim Penguji untuk diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Surabaya**

Dosen Pembimbing I Dosen Pembimbing II Tanda tangan Tanda tangan

(……….Nama………) (……….Nama………)

Dosen Penguji I Dosen Penguji II Tanda tangan Tanda tangan

(……….Nama………) (……….Nama………)

Ditetapkan di : …………….. Tanggal : …………….

Mengetahui

Ketua Program Studi

Tanda tangan

(……….Nama………)

iii

**PERNYATAAN KEASLIAN**

Nama : Jevon Ananta Gunawan

NRP : 160420097

Program Studi : Teknik Informatika

Fakultas : Teknik

Judul Skripsi : Pembuatan Aplikasi Sistem Pengenalan Aksara Bali Menggunakan Metode *Convolutional Neural Network*

Skripsi/Tesis ini adalah hasil karya saya sendiri, semua sumber kutipan dan rujukan telah saya tulis dengan benar dan disebutkan dalam sumber kutipan dan daftar pustaka. Apabila dikemudian hari penulisan Skripsi/Tesis ini merupakan hasil plagiat atau jiplakan terhadap karya orang lain, maka saya bersedia bertanggung jawab atas nama diri sendiri dan menerima sanksi berdasarkan ketentuan yang berlaku di Universitas Surabaya

Tanggal : …………… Materai

Nama : ……………..

iv

JUDUL : PEMBUATAN APLIKASI SISTEM PENGENALAN AKSARA BALI MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK*

Nama : Jevon Ananta Gunawan

Jurusan/ Program Studi : Teknik Informatika

Pembimbing 1 : Mohammad Farid Naufal, S.Kom., M.Kom. Pembimbing 2 : Dr. Joko Siswantoro, S.Si., M.Si.

**ABSTRAK**

Aksara Bali adalah suatu bentuk tulisan tradisional yang berasal dari Bali dan digunakan sebagai media komunikasi tertulis. Akan tetapi, seiring perkembangan waktu, tingkat literasi masyarakat Bali dalam mengenali tulisan aksara Bali semakin rendah. Sebagian besar orang membutuhkan waktu yang sangat lama serta melalui proses yang panjang dan berulang-ulang dalam melakukan proses pengenalan dari tulisan aksara Bali ke tulisan latin. Masalah ini dapat diselesaikan menggunakan algoritma *Convolutional Neural Network* (CNN) dengan memanfaatkan berbagai macam arsitektur untuk melakukan pengenalan karakter pada tulisan aksara Bali dengan batasan uji hanya dapat mengenali aksara *wreastra*, *pengangge suara* dan *tengenan*. Sistem akan melakukan proses segmentasi untuk mendapatkan masing-masing karakter menggunakan salah satu teknik *Pre-processing* citra, yaitu *projection profile segmentation*. Tiga model klasifikasi tulisan aksara Bali yang masing-masing dibuat menggunakan arsitektur MobileNet V2, ResNet50, dan Xception memperoleh tingkat akurasi sebesar 100%. Sistem transliterasi aksara Bali yang dibuat kemudian diuji menggunakan 100 citra uji coba untuk menghitung *Levenshtein Distance* dari hasil transliterasi. Nilai *Levenshtein Distance* yang didapat adalah 84.7 untuk model MobileNet V2, 85.68 untuk model ResNet50, dan 85.01 untuk model Xception. Sistem divalidasi oleh 15 responden dan disimpulkan bahwa aplikasi dapat membantu pengguna dalam melakukan pengenalan dari tulisan aksara Bali ke tulisan latin. Hasil dari uji hipotesis t-*test* yang membandingkan skor hasil transliterasi yang dilakukan masyarakat Bali sebelum dan setelah menggunakan aplikasi transliterasi menunjukkan bahwa terdapat perbedaan berupa meningkatnya tingkat literasi masyarakat secara signifikan setelah menggunakan aplikasi sehingga dapat disimpulkan bahwa aplikasi yang dibuat dapat membantu meningkatkan literasi pengguna dalam mengenali tulisan aksara Bali.

**Kata Kunci:** **Aksara Bali, *Convolutional Neural Network*, Pengenalan, *Pre-processing* Citra**

v

TITLE: CREATING BALINESE SCRIPT RECOGNITION SYSTEM APPLICATION USING CONVOLUTIONAL NEURAL NETWORK METHOD

Name: Jevon Ananta Gunawan

Dicipline/Study Programme: Informatics Engineering Contributor 1: Mohammad Farid Naufal, S.Kom., M.Kom. Contributor 2: Dr. Joko Siswantoro, S.Si., M.Si.

**ABSTRACT**

Balinese script is a form of traditional script that originated from Bali and used as a medium of written communication. But, over time, the Balinese people’s literacy level in recognizing Balinese script is getting lower. Most people need a lot of time and go through a long and repetitive process in recognizing from Balinese script to latin script. This problem can be solved using Convolutional Neural Network algorithm by using various architecture to perform Balinese script character recognition with limitation that it’s only able to recognize wreastra script, pengangge suara and tengenan. The system will conduct segmentation process to get each character that is applied using an image pre-processing technique called projection profile segmentation. Three Balinese script classification models that are each built using MobileNet V2, ResNet50, and Xception architecture obtain 100% accuracy rate. The Balinese script transliteration system then tested using 100 testing images to calculate the Levenshtein Distance of the transliteration results. The Levenshtein Distance score obtained is 84.7 for MobileNet V2 model, 85.68 for ResNet50 model, and 85.01 for Xception model. The system was validated by 15 respondents and it was concluded that the application could help users in recognizing the Balinese script into Latin script. The result of the t-test hypothesis which compared the transliteration score of the Balinese people before and after using the transliteration application showed that there was a difference in the form of a significant increase in the community's literacy level after using the application, so it was concluded that the application can help improve user’s literacy in recognizing Balinese script.

**Keywords: Balinese script, Convolutional Neural Network, image pre-processing, recognition**

vi

**KATA PENGANTAR**

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat serta kesempatan bagi saya untuk menyelesaikan tugas akhir yang berjudul “Pembuatan Aplikasi Sistem Pengenalan Aksara Bali Menggunakan Metode *Convolutional Neural Network*”. Penulisan tugas akhir ini tidak dapat berjalan dengan baik tanpa adanya kontribusi dari berbagai pihak. Oleh karena itu, melalui kata pengantar ini, saya ingin menyampaikan rasa terima kasih yang sebesar-besarnya dan penghargaan yang mendalam kepada:

1. Bapak Dr. Joko Siswantoro, S.Si., M.Si., selaku Ketua Jurusan Teknik Informatika Universitas Surabaya, dosen wali, dan dosen pembimbing kedua tugas akhir saya yang telah memberikan bimbingan, arahan, serta masukan yang sangat berharga selama perkuliahan serta proses pembuatan dan penulisan tugas akhir ini.
2. Bapak Mohammad Farid Naufal, S.Kom., M.Kom., selaku dosen pembimbing pertama tugas akhir saya yang telah memberikan bimbingan, arahan, serta masukan yang sangat berharga dalam pembuatan serta penulisan tugas akhir ini.
3. Ibu I Gusti Ayu Made Sintia Dewi, S.Pd., selaku guru Bahasa Bali SMAK Santo Yoseph Denpasar yang telah bersedia meluangkan waktu menjadi validator dalam penelitian ini.
4. Seluruh responden yang telah berkenan meluangkan waktu serta memberikan tanggapan yang sangat berharga terhadap penelitian ini.

vii

1. Kedua orang tua saya yang telah memberikan semangat serta dukungan sehingga saya dapat menyelesaikan penelitian ini dengan baik.
2. Dan kepada semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa masih terdapat banyak kekurangan dan keterbatasan dalam penyusunan skripsi ini. Oleh karena itu, kami mengharapkan segala saran, kritik, dan masukan yang membangun untuk pengembangan di masa mendatang.

Akhir kata, penulis berharap skripsi ini dapat berkontribusi serta memberikan manfaat bagi pengembangan ilmu pengetahuan dan pembelajaran di bidang teknologi.

Surabaya, 3 Januari 2024

Penulis

viii

**DAFTAR ISI**

HALAMAN PENGESAHAN................................................................................ iii PERNYATAAN KEASLIAN................................................................................ iv ABSTRAK .............................................................................................................. v KATA PENGANTAR .......................................................................................... vii DAFTAR ISI.......................................................................................................... ix DAFTAR GAMBAR ........................................................................................... xiii DAFTAR TABEL................................................................................................ xvi DAFTAR LISTING ............................................................................................. xix BAB I PENDAHULUAN.....................................................................................I-1

1.1 LATAR BELAKANG ..........................................................................I-1

1.2 RUMUSAN MASALAH......................................................................I-4

1.3 TUJUAN ...............................................................................................I-5

1.4 MANFAAT...........................................................................................I-5

1.5 RUANG LINGKUP..............................................................................I-5

1.6 RENCANA KEGIATAN......................................................................I-6

1.7 SISTEMATIKA PENULISAN.............................................................I-7

BAB II DASAR TEORI...................................................................................... II-1

2.1 Analisis Sentimen................................................................................ II-1

2.2 *Crawling*.............................................................................................. II-2

2.3 *Preprocessing*...................................................................................... II-3

2.3.1 Case Folding.................................................................................... II-3

ix

2.3.2 Data Cleaning.................................................................................. II-4

2.3.3 Slang Words.................................................................................... II-4

2.3.4 Stopword Removal.......................................................................... II-5

2.3.5 Negation .......................................................................................... II-5

2.3.6 Stemming ........................................................................................ II-6

2.4 Pembobotan TF-IDF ........................................................................... II-6

2.5 Support Vector Machine ..................................................................... II-9

2.6 Evaluasi............................................................................................. II-14

2.7 Twitter ............................................................................................... II-17

2.8 Youtube............................................................................................. II-18

BAB III ANALISIS SISTEM ............................................................................ III-1

3.1 Analisis Kondisi Saat Ini.................................................................... III-1

3.2 Analisis Sistem Sejenis ...................................................................... III-9

3.3 Analisis Permasalahan ..................................................................... III-17

3.4 Analisis Kebutuhan Sistem .............................................................. III-17

BAB IV DESAIN SISTEM ...............................................................................IV-1

4.1 Desain Data ........................................................................................IV-1

4.1.1 Tabel Data *Users*............................................................................IV-2

4.1.2 Tabel Data Komentar .....................................................................IV-2

4.1.3 Tabel Data Komentar Sementara ...................................................IV-3

4.2 Desain Proses .....................................................................................IV-4

4.2.1 Desain Proses Umum .....................................................................IV-4

4.2.2 Desain Proses Ambil Data (Crawling)...........................................IV-5

x

4.2.3 Desain Proses Preprocessing..........................................................IV-5

4.2.4 Desain Proses Feature Extraction...................................................IV-6

4.2.5 Desain Proses Klasifikasi...............................................................IV-6

4.3 Desain Tampilan ................................................................................IV-7

4.3.1 Desain Tampilan Halaman Login ..................................................IV-8

4.3.2 Desain Tampilan Halaman Beranda...............................................IV-9

4.3.3 Desain Tampilan Halaman Ambil data ........................................IV-11

4.3.4 Desain Tampilan Halaman Komentar..........................................IV-14

4.3.5 Desain Tampilan Halaman Cek Komentar ..................................IV-15

4.3.6 Desain Tampilan Halaman Validasi.............................................IV-16

BAB V IMPLEMENTASI SISTEM................................................................... V-1

5.1 Implementasi Data .............................................................................. V-1

5.1.1 Implementasi Data Komentar ......................................................... V-1

5.1.2 Implementasi Data User.................................................................. V-2

5.1.3 Implementasi Data Komentar Sementara........................................ V-3

5.2 Implementasi Proses............................................................................ V-4

5.2.1 Implementasi Proses Crawling........................................................ V-5

5.2.2 Implementasi Proses Preprocessing ................................................ V-8

5.2.3 Implementasi Proses Feature Extraction....................................... V-11

5.2.4 Implementasi Proses Klasifikasi ................................................... V-13

BAB VI UJI COBA DAN EVALUASI.............................................................VI-1

6.1 Verifikasi............................................................................................VI-1

6.1.1 Fitur Login .....................................................................................VI-1

xi

6.1.2 Fitur Ambil Data ............................................................................VI-3

6.1.3 Fitur Cek Komentar......................................................................VI-11

6.1.4 Fitur Tampilan Semua Komentar.................................................VI-12

6.1.5 Fitur Validasi................................................................................VI-13

6.1.6 Verifikasi Kebenaran Sistem........................................................VI-15

6.1.7 Verifikasi Perhitungan Metode Support Vector Machine............VI-23

6.2 Validasi ............................................................................................VI-36

6.2.1 Validasi dengan Pihak KEMENDIKBUDRISTEK.....................VI-36

6.2.2 Validasi dengan Pihak Guru, Orang Tua, dan Siswa ...................VI-38

BAB VII KESIMPULAN DAN SARAN........................................................ VII-1

7.1 Kesimpulan ...................................................................................... VII-1

7.2 Saran................................................................................................. VII-2

DAFTAR PUSTAKA ........................................................................................... xx

LAMPIRAN........................................................................................................ xxv

xii

**DAFTAR GAMBAR**

Gambar 2. 1 Alur Tahapan Proses Analisis Sentimen ........................................ II-2

Gambar 2. 2 Alur preprocessing ........................................................................ II-3

Gambar 2. 3 Ilustrasi Algoritma TF-IDF ............................................................ II-8

Gambar 2. 4 Hyperplane Terbaik yang Memisahkan 2 Kelas Data.................. II-11

Gambar 3. 1 Contoh Tweet Negatif ................................................................... III-3

Gambar 3. 2 Contoh Tweet Positif..................................................................... III-4

Gambar 3. 3 Contoh Komentar Negatif pada Youtube...................................... III-5

Gambar 3. 4 Contoh Komentar Positif pada Youtube ....................................... III-6

Gambar 4. 1 Desain Basis Data..........................................................................IV-1

Gambar 4. 2 Desain Proses Sistem Secara Umum.............................................IV-4

Gambar 4. 3 Desain Proses Crawling ................................................................IV-5

Gambar 4. 4 Desain Proses Preprocessing.........................................................IV-6

Gambar 4. 5 Desain Proses Feature Extraction..................................................IV-6

Gambar 4. 6 Desain Proses Klasifikasi..............................................................IV-7

Gambar 4. 7 Desain Tampilan Halaman Login .................................................IV-8

Gambar 4. 8 Desain Tampilan Pesan Pemberitahuan Saat Login......................IV-9

Gambar 4. 9 Desain Tampilan Halaman Beranda Tanpa Login ......................IV-10

Gambar 4. 10 Desain Tampilan Halaman Beranda Admin..............................IV-11

Gambar 4. 11 Desain Tampilan Halaman Ambil Data dari Youtube ..............IV-12

Gambar 4. 12 Desain Tampilan Halaman Ambil Data dari Twitter ................IV-13

Gambar 4. 13 Desain Tampilan Ambil Data dari Unggahan File....................IV-14

Gambar 4. 14 Desain Tampilan Halaman Komentar .......................................IV-15

xiii

Gambar 4. 15 Desain Tampilan Halaman Cek Komentar................................IV-16

Gambar 4. 16 Desain Tampilan Halaman Validasi..........................................IV-17

Gambar 5. 1 Tabel Struktur Data Komentar ....................................................... V-2

Gambar 5. 2 Tabel Struktur Data Users .............................................................. V-3

Gambar 5. 3 Tabel Struktur Data Komentar Sementara ..................................... V-4

Gambar 6. 1 Tampilan Halaman Login..............................................................VI-2

Gambar 6. 2 Tampilan Halaman Beranda Admin..............................................VI-2

Gambar 6. 3 Tampilan Halaman Tidak Berhasil Login.....................................VI-3

Gambar 6. 4 Tampilan Halaman Ambil Data ....................................................VI-4

Gambar 6. 5 Tampilan Berhasil Ambil Data dari Twitter .................................VI-5

Gambar 6. 6 Proses Ambil Twitter Gagal..........................................................VI-5

Gambar 6. 7 Pengumuman Penghentian Akses Gratis API ...............................VI-6

Gambar 6. 8 Penundaan Penghentian Akses Gratis API....................................VI-6

Gambar 6. 9 Pernyataan Akses Gratis API Resmi Dihentikan ..........................VI-7

Gambar 6. 10 Akses API Twitter Berbayar .......................................................VI-7

Gambar 6. 11 Pembatasan Pembacaan Tweet....................................................VI-8

Gambar 6. 12 Tampilan Berhasil Ambil Data dari Youtube .............................VI-9

Gambar 6. 13 Tampilan Tidak Berhasil Ambil Data dari Youtube ...................VI-9

Gambar 6. 14 Tampilan Halaman Upload File ................................................VI-10

Gambar 6. 15 Contoh Template Format Excel ................................................VI-10

Gambar 6. 16 Contoh Tampilan Berhasil Unggah File....................................VI-11

Gambar 6. 17 Tampilan Halaman Cek Komentar............................................VI-11

Gambar 6. 18 Tampilan Berhasil Cek Komentar.............................................VI-12

xiv

Gambar 6. 19 Tampilan Halaman Semua Data Komentar...............................VI-12

Gambar 6. 20 Tampilan Halaman Validasi......................................................VI-13

Gambar 6. 21 Notifikasi Berhasil Validasi ......................................................VI-14

Gambar 6. 22 Notifikasi Berhasil Hapus Data.................................................VI-14

Gambar 6. 23 Notifikasi Berhasil Validasi Seluruh Data ................................VI-15

Gambar 6. 24 Hasil Prediksi Sistem ................................................................VI-36

xv

**DAFTAR TABEL**

Tabel 2. 1 Contoh Input dan Output Tahapan Case Folding.............................. II-4

Tabel 2. 2 Contoh Input dan Output Tahapan Data Cleaning............................. II-4

Tabel 2. 3 Contoh Input dan Output Tahapan Slang Words ............................... II-5

Tabel 2. 4 Contoh Input dan Output Tahapan Negation ..................................... II-5

Tabel 2. 5 Contoh Input dan Output Tahapan Stopword Removal.................... II-6

Tabel 2. 6 Contoh Input dan Output Tahapan Stemming ................................... II-6

Tabel 2. 7 Contoh Perhitungan Pembobotan TF-IDF ......................................... II-9

Tabel 2. 8 Contoh Perhitungan SVM................................................................ II-13

Tabel 2. 9 Confusion Matrix ............................................................................. II-15

Tabel 2. 10 Contoh Data Uji ............................................................................. II-15

Tabel 2. 11 Contoh Perhitungan Presisi............................................................ II-16

Tabel 2. 12 Contoh Perhitungan Recall ............................................................ II-17

Tabel 4. 1 Tabel Data Users...............................................................................IV-2

Tabel 4. 2 Tabel Data Komentar ........................................................................IV-2

Tabel 4. 3 Tabel Data Komentar Sementara ......................................................IV-3

Tabel 6. 1 Tabel Evaluasi Dataset dengan Kamus Slangword Awal...............VI-16

Tabel 6. 2 Hasil Evaluasi Dataset dengan Kamus Slangword Modifikasi.......VI-17

Tabel 6. 3 Contoh 10 Kalimat Uji....................................................................VI-18

Tabel 6. 4 Hasil Evaluasi Pengujian 10 Kalimat..............................................VI-19

Tabel 6. 5 Confussion Matrix 10 Kalimat Uji per Kernel................................VI-22

Tabel 6. 6 Perhitungan Evaluasi 10 Kalimat Uji..............................................VI-23

Tabel 6. 7 Contoh 10 Kalimat Perhitungan......................................................VI-23

xvi

Tabel 6. 8 Perhitungan TF-IDF........................................................................VI-24

Tabel 6. 9 Fungsi Kernel One vs One..............................................................VI-26

Tabel 6. 10 Perhitungan Fungsi Kernel Model 1 .............................................VI-26

Tabel 6. 11 Perhitungan Fungsi Kernel Model 2 .............................................VI-27

Tabel 6. 12 Perhitungan Fungsi Kernel Model 3 .............................................VI-27

Tabel 6. 13 Perhitungan Matrix Hessian untuk Model 1 .................................VI-28

Tabel 6. 14 Perhitungan Matrix Hessian untuk Model 2 .................................VI-29

Tabel 6. 15 Perhitungan Matrix Hessian untuk Model 3 .................................VI-29

Tabel 6. 16 Perhitungan Nilai Eror Model 1....................................................VI-30

Tabel 6. 17 Perhitungan Nilai Eror Model 2....................................................VI-30

Tabel 6. 18 Perhitungan Nilai Eror Model 3....................................................VI-30

Tabel 6. 19 Perhitungan Delta Alpha Model 1 ................................................VI-31

Tabel 6. 20 Perhitungan Delta Alpha Model 2 ................................................VI-31

Tabel 6. 21 Perhitungan Delta Alpha Model 3 ................................................VI-32

Tabel 6. 22 Perhitungan Alpha Baru Model 1 .................................................VI-32

Tabel 6. 23 Perhitungan Alpha Baru Model 2 .................................................VI-32

Tabel 6. 24 Perhitungan Alpha Baru Model 3 .................................................VI-33

Tabel 6. 25 Perhitungan W+, W-, dan Bias Model 1.......................................VI-33

Tabel 6. 26 Perhitungan W+, W-, dan Bias Model 2.......................................VI-34

Tabel 6. 27 Perhitungan W+, W-, dan Bias Model 3.......................................VI-34

Tabel 6. 28 Perhitungan Fungsi Kernel Kalimat 13 Model 1 ..........................VI-34

Tabel 6. 29 Perhitungan Fungsi Kernel Kalimat 13 Model 2 ..........................VI-34

Tabel 6. 30 Perhitungan Fungsi Kernel Kalimat 13 Model 3 ..........................VI-35

xvii

Tabel 6. 31 Hasil Perhitungan Hyperplane Tiap Model ..................................VI-35

Tabel 6. 32 Hasil Validasi Pihak KEMENDIKBUDRISTEK.........................VI-37

Tabel 6. 33 Hasil Wawancara dengan Pihak Guru ..........................................VI-38

Tabel 6. 34 Hasil Wawancara dengan Pihak Orang Tua .................................VI-39

Tabel 6. 35 Hasil Wawancara dengan Siswa ...................................................VI-39

xviii

**DAFTAR LISTING**

Listing 5. 1 Potongan Kode Program Proses Crawling Twitter.......................... V-5

Listing 5. 2 Potongan Kode Program Proses Crawling Youtube........................ V-6

Listing 5. 3 Potongan Kode Program Proses Preprocessing............................... V-8

Listing 5. 4 Potongan Kode Program Proses Feature Extraction...................... V-11

Listing 5. 5 Potongan Kode Program Proses Klasifikasi .................................. V-13

xix

**BAB 5**

**IMPLEMENTASI**

Bab ini menjelaskan mengenai proses implementasi sistem yang telah didesain sebelumnya. Proses-proses implementasi yang dibahas meliputi penjelasan data pada *database* dan implementasi proses pencatatan informasi barang dan transaksi pada program. Implementasi sistem dilakukan dengan memanfaatkan *framework* Laravel dan *database* MySQL.

**5.1. Implementasi *Database***

Bagian ini menjelaskan mengenai implementasi database di MySQL. Database yang digunakan terdiri dari 15 tabel yang berkorelasi satu sama lain. Penjelasan dari masing-masing tabel beserta data dapat dijabarkan seperti berikut:

1. Entitas *users*

Tabel ini digunakan untuk menyimpan data pengguna untuk proses login. Terdapat 3 atribut yang digunakan pada tabel ini. Semua atribut beserta fungsinya dapat dilihat pada tabel di bawah ini.

Tabel 5.1. Entitas *users*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *id* | *bigint (primary key)* | *Id* pengguna |
| *name* | *varchar* | Nama pengguna |
| *password* | *varchar* | *Password* akun pengguna |

1. Entitas *invmaster*

Tabel ini digunakan untuk menyimpan data barang perusahaan. Terdapat 12 atribut pada tabel ini. Semua atribut beserta penjelasan dari tabel ini dapat dilihat pada tabel di bawah ini.

Tabel 5.2. Entitas *invmaster*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *id* | *varchar* | *Id* barang |
| *kode\_brg* | *varchar* | Kode barang |
| *nama\_brg* | *varchar* | Nama barang |
| *kode\_divisi* | *varchar* | Kode divisi barang |
| *kode\_jenis* | *varchar* | Kode jenis barang |
| *kode\_type* | *varchar* | Kode tipe barang |
| *packing* | *varchar* | *Packing* barang |
| *quantity* | *int* | Kuantitas barang |
| *id\_satuan* | *int* | *Id* satuan barang |
| *hrg\_jual* | *float* | Harga jual per pack |
| *kode\_gudang* | *varchar* | kode gudang barang |
| keterangan | *varchar* | keterangan |

1. Entitas *invdivisi*

Tabel ini digunakan untuk menyimpan daftar divisi barang. Terdapat 2 atribut pada tabel ini. Atribut beserta penjelasan dari tabel ini dapat dilihat pada tabel berikut ini.

Tabel 5.3. Entitas *invdivisi*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| kode | *varchar* | Kode divisi |
| divisi | *varchar* | Nama divisi |

1. Entitas *invgudang*

Tabel ini digunakan untuk menyimpan daftar gudang pada perusahaan. Terdapat 4 atribut pada tabel ini. Atribut serta penjelasan dapat dilihat pada tabel berikut ini.

Tabel 5.4. Entitas *invgudang*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| kode | *varchar* | Kode gudang |
| nama | *varchar* | Nama gudang |
| alamat | *varchar* | Alamat gudang |
| keterangan | *varchar* | Keterangan |

1. Entitas *invjenis*

Tabel ini digunakan untuk menyimpan data jenis barang. Terdapat 2 buah atribut pada tabel ini. Atribut serta penjelasan dari tabel ini telah dijabarkan pada tabel di bawah ini.

Tabel 5.5. Entitas *invjenis*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| kode | *varchar* | Kode jenis |
| jenis | *varchar* | Nama jenis |

1. Entitas *invtype*

Tabel ini digunakan untuk menyimpan daftar tipe barang. Terdapat 2 atribut yang digunakan pada tabel ini. Untuk atribut serta penjelasan dapat dilihat pada tabel berikut ini.

Tabel 5.6. Entitas *invtype*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| kode | *varchar* | Kode tipe |
| *type* | *varchar* | Nama tipe |

1. Entitas beli

Tabel ini digunakan untuk menyimpan data transaksi pembelian dari *supplier*. Terdapat 12 atribut yang digunakan pada tabel ini. Atribut serta penjelasan telah dijabarkan pada tabel berikut ini.

Tabel 5.7. Entitas beli

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *no\_bukti* | *varchar* | Nomor nota |
| tanggal | *varchar* | Tanggal nota |
| *jatuh\_tempo* | *varchar* | Tanggal jatuh tempo |
| *kode\_supp* | *varchar* | Kode s*upplier* |
| *sub\_total* | *float* | Harga sub total |
| *persen\_ppn* | *int* | Persentase PPN |
| total | *float* | Harga total |
| lunas | *varchar* | Status pembayaran |
| *tgl\_lunas* | *varchar* | Tanggal pelunasan |
| status | *varchar* | Status pengiriman barang |
| *tgl\_terkirim* | *varchar* | Tanggal terkirimnya barang |
| *author* | *varchar* | Nama pembuat nota |

1. Entitas *beli\_dtl*

Tabel ini digunakan untuk mencatat informasi *detail* masing-masing nota pembelian. Terdapat 9 atribut pada tabel ini. Atribut serta penjelasan dari tabel ini telah dijabarkan pada tabel di bawah ini.

Tabel 5.8. Entitas *beli\_dtl*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *no\_bukti* | *varchar* | Nomor nota |
| *kode\_brg* | *varchar* | Kode barang |
| *nama\_brg* | *varchar* | Nama barang |
| *qty\_order* | *int* | Kuantitas barang yang dipesan |
| *packing* | *varchar* | Packing barang |
| *id\_satuan* | *int* | *Id* satuan barang |
| *hrg\_per\_unit* | *float* | Harga per unit |
| *hrg\_total* | *float* | Harga total |
| *kirim\_gudang* | *varchar* | Gudang tujuan barang |

1. Entitas *customer*

Tabel ini digunakan untuk menyimpan data *customer* perusahaan. Terdapat 11 buah atribut pada tabel ini. Atribut tabel serta penjelasan dapat dilihat pada tabel berikut ini.

Tabel 5.9. Entitas *customer*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *kode\_cust* | *varchar* | Kode perusahaan customer |
| *nama\_cust* | *varchar* | Nama perusahaan c*ustomer* |
| *type\_cust* | *varchar* | Tipe perusahaan c*ustomer* |
| *alm\_1* | *varchar* | Alamat 1 perusahaan c*ustomer* |
| *alm\_2* | *varchar* | Alamat 2 perusahaan c*ustomer* |
| *alm\_3* | *varchar* | Alamat 3 perusahaan c*ustomer* |
| kota | *varchar* | Kota asal c*ustomer* |
| kontak | *varchar* | Nama kontak c*ustomer* |
| *no\_telp* | *varchar* | Nomor telepon kontak |
| email | *varchar* | Email kontak |
| *kode\_sales* | *varchar* | Kode s*ales person* |

1. Entitas jual

Tabel ini digunakan untuk mencatat transaksi penjualan ke *customer*. Terdapat 12 atribut yang disediakan pada tabel ini. Untuk atribut tabel serta penjelasan dapat dilihat pada tabel di bawah ini.

Tabel 5.10. Entitas jual

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *no\_bukti* | *varchar* | Nomor nota |
| tanggal | *varchar* | Tanggal nota |
| *jatuh\_tempo* | *varchar* | Tanggal jatuh tempo |
| *kode\_cust* | *varchar* | Kode c*ustomer* |
| *sub\_total* | *float* | Harga sub total |
| *persen\_ppn* | *int* | Persentase PPN |
| total | *float* | Harga total |
| lunas | *varchar* | Status pembayaran |
| *tgl\_lunas* | *varchar* | Tanggal pelunasan |
| status | *varchar* | Status pengiriman barang |
| *tgl\_terkirim* | *varchar* | Tanggal terkirimnya barang |
| *author* | *varchar* | Nama pembuat nota |

1. Entitas *jual\_dtl*

Tabel ini digunakan untuk mencatat data *detail* masing-masing transaksi penjualan. Terdapat 9 atribut yang disediakan pada tabel ini. Atribut serta penjelasan untuk tabel ini dapat dilihat pada tabel di bawah ini.

Tabel 5.11. Entitas *jual\_dtl*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *no\_bukti* | *varchar* | Nomor nota |
| *id\_brg* | *int* | Id barang |
| *kode\_brg* | *varchar* | Kode barang |
| *nama\_brg* | *varchar* | Nama barang |
| *qty\_order* | *int* | Kuantitas barang pesanan |
| *id\_satuan* | *int* | *Id* satuan barang |
| *hrg\_per\_unit* | *float* | Harga per unit |
| *hrg\_total* | *float* | Harga total |
| *kode\_gudang* | *varchar* | Gudang asal barang |

1. Entitas *sales\_person*

Tabel ini digunakan untuk menyimpan data *sales person* untuk masing-masing *customer*. Terdapat 3 atribut yang digunakan pada tabel ini. Atribut tabel serta penjelasan telah dijabarkan pada tabel berikut ini.

Tabel 5.12. Entitas *sales\_person*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *kode\_sales* | *varchar* | Kode s*ales person* |
| *nama\_sales* | *varchar* | Nama s*ales person* |
| divisi | *varchar* | Divisi s*ales person* |

1. Entitas satuan

Tabel ini digunakan untuk menyimpan data satuan barang. Terdapat 2 buah atribut pada tabel ini. Atribut tabel serta penjelasan dapat dilihat pada tabel di bawah ini.

Tabel 5.13. Entitas satuan

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *id* | *int* | *Id* satuan |
| satuan | *varchar* | Nama satuan |

1. Entitas *supplier*

Tabel ini digunakan untuk menyimpan data *supplier* perusahaan. Terdapat 12 buah atribut pada tabel ini. Atribut tabel serta penjelasan telah dijabarkan pada tabel berikut ini.

Tabel 5.14. Entitas *supplier*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *kode\_supp* | *varchar* | Kode perusahaan s*upplier* |
| *nama\_supp* | *varchar* | Nama perusahaan s*upplier* |
| bank | *varchar* | Nama bank |
| *acc\_bank* | *varchar* | Nomor rekening perusahaan s*upplier* |
| *alm\_1* | *varchar* | Alamat 1 perusahaan s*upplier* |
| *alm\_2* | *varchar* | Alamat 2 perusahaan s*upplier* |
| kota | *varchar* | Kota asal perusahaan s*upplier* |
| negara | *varchar* | Negara asal perusahaan s*upplier* |
| kontak | *varchar* | Nama kontak s*upplier* |
| jabatan | *varchar* | Jabatan kontak |
| *no\_telp* | *varchar* | Nomor telepon kontak |
| email | *varchar* | Email kontak |

1. Entitas *mutasi\_stok*

Tabel ini digunakan untuk menyimpan jumlah barang yang masuk dan keluar di gudang tertentu pada suatu transaksi. Terdapat 12 atribut yang disediakan pada tabel ini. Atribut serta penjelasan untuk entitas ini dapat dilihat pada tabel di bawah ini.

Tabel 5.15. Entitas *mutasi\_stok*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *id* | *int* | *Id* mutasi |
| *no\_bukti* | *varchar* | Nomor nota |
| tanggal | *date* | Tanggal mutasi barang |
| *kode\_brg* | *varchar* | Kode barang |
| *nama\_brg* | *varchar* | Nama barang |
| *id\_satuan* | *int* | *Id* satuan barang |
| *kode\_gudang* | *varchar* | Kode gudang barang |
| *stok\_awal* | *int* | Stok awal barang |
| *qty\_masuk* | *int* | Jumlah kuantitas barang yang masuk |
| *qty\_keluar* | *int* | Jumlah kuantitas barang yang keluar |
| *qty\_rusak\_exp* | *int* | Jumlah kuantitas barang yang rusak/e*xpired* |
| *stok\_akhir* | *int* | Stok akhir barang setelah mutasi |

1. Entitas *opname\_stok*

Tabel ini digunakan untuk menyimpan informasi stok opname yang dilakukan di suatu gudang. Terdapat 10 atribut yang disediakan pada tabel ini. Untuk atribut serta penjelasan dapat dilihat pada tabel berikut ini.

Tabel 5.16. Entitas *opname\_stok*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| *id* | *int* | *Id* data stok opname |
| tanggal | *varchar* | Tanggal stok opname |
| *kode\_brg* | *varchar* | Kode barang |
| *nama\_brg* | *varchar* | Nama barang |
| *id\_satuan* | *int* | *Id* satuan barang |
| *kode\_gudang* | *varchar* | Kode gudang barang |
| *qty\_sistem* | *int* | Kuantitas pada sistem |
| *qty\_fisik* | *int* | Kuantitas barang fisik |
| selisih | *int* | Jumlah selisih |
| keterangan | *varchar* | Penyebab terjadi selisih |

**5.2. Implementasi Program**

**5.2.1.** **Proses pada Halaman *Login***

Pada halaman *login*, pengguna akan menginputkan *username* dan *password* sesuai dengan data yang telah terdaftar di tabel *users* pada *database*. Setelah melakukan *input*, data kemudian dikirim ke *server* untuk dilakukan proses validasi. Jika *username* dan *password* yang diinputkan telah terdaftar pada *database*, maka pengguna akan masuk ke halaman utama aplikasi. Sebaliknya jika tidak terdaftar, maka pengguna diarahkan kembali ke halaman *login* dan akan muncul notifikasi “Detail login tidak valid”. Proses diatas diimplementasikan dalam sebuah fungsi bernama *loginPost* dengan sintaks seperti pada listing di bawah ini.

**Listing 5.1. Proses *Login***

|  |
| --- |
| function loginPost(Request $request){          $request->validate([              'name' => 'required',              'password' => 'required'          ]);          $credentials = $request->only('name','password');            if(Auth::attempt($credentials)){              return redirect()->intended(route('home'));          }  return redirect(route('login'))->with("error","Login details are  not valid");      } |

**5.2.2. Proses pada Halaman *Master***

Pada halaman *master*, ditampilkan daftar barang pada perusahaan. Data yang ditampilkan diambil dari tabel *invmaster*. Informasi dari tabel *master* akan ditampilkan hanya 5 buah barang untuk setiap halamannya. Fitur ini diimplementasikan menggunakan fungsi *paginate* setelah proses *filter* dilakukan. *Filter* berdasarkan gudang, jenis barang, maupun kode atau nama barang akan dilakukan terlebih dahulu jika pengguna menginputkan nama gudang melalui *combo box,* nama jenis barang melalui sebuah *combo box* juga, atau mengisi nama barang pada sebuah *input field*. Untuk informasi divisi, jenis, gudang, dan tipe barang masing-masing akan diambil dari tabel *invdivisi, invjenis, invgudang*, dan *invtype*. Sintaks pengambilan data dari masing-masing tabel tersebut untuk ditampilkan pada halaman *master* dapat dilihat pada listing berikut ini.

**Listing 5.2. Proses Pengambilan Data Halaman *Master***

|  |
| --- |
| public function master(Request $request){          $selectedGudang = $request->get('gudang');  $jenis = $request->get('jenis');          $search = $request->get('search');          $query = Master::query();          if($selectedGudang && $selectedGudang != 'All'){          $query->whereHas('gudang', function ($q)  use ($selectedGudang) {                  $q->where('nama', $selectedGudang);              });          }    if($jenis && $jenis != 'All'){  $query->whereHas('jenis', function ($j) use ($jenis){  $j->where('jenis',$jenis);  });  }          if($search){              $query->where('nama\_brg', 'like', '%'.$search.'%');          }          $master = $query->paginate(5);          $divisi = Divisi::all();          $gudang = Gudang::all();          $jenis = Jenis::all();          $type = Type::all();          $satuan = Satuan::all();    return view('master.master',compact('master','divisi','gudang',  'jenis','type','satuan','selectedGudang','search'));      } |

Halaman ini menyediakan fitur stok opname barang yang dapat digunakan jika jumlah barang pada sistem tidak sesuai dengan jumlah barang fisik pada gudang. Pengguna dapat menekan tombol “Opname” yang terletak pada kolom “Actions” pada tabel *master* yang akan memunculkan sebuah dialog untuk menginputkan jumlah barang fisik serta keterangan yang menandakan sebab dari adanya selisih jumlah barang. Penyebabnya bisa karena rusak, kedaluwarsa, atau salah pencatatan. Setelah menginputkan data, informasi tersebut kemudian dikirim ke fungsi *opnameBarang* untuk dilakukan pencatatan informasi stok opname ke database untuk pembuatan laporan.

Setelah fungsi menerima seluruh informasi, jika penyebab terjadinya selisih adalah karena rusak atau kedaluwarsa, dilakukan proses pencarian apakah barang dan keterangan yang sama telah tercatat pada sistem. Jika sudah tercatat, maka dilakukan penambahan kuantitas khusus untuk baris data tersebut. Jika belum tercatat, maka dilakukan penambahan data barang dengan kuantitas dan keterangan yang diinputkan pengguna untuk membentuk baris data baru. Hal ini dilakukan untuk memisahkan barang yang rusak atau kedaluwarsa dengan barang yang kualitasnya masih baik. Sistem kemudian menyimpan informasi jumlah barang yang rusak atau kedaluwarsa di kartu stok sebagai bagian dari mutasi barang. Selanjutnya, kuantitas barang yang sama dan tidak rusak atau kedaluwarsa akan dikurangi sesuai dengan jumlah barang yang diinputkan pengguna pada dialog.

Selain itu, jika penyebab terjadinya selisih jumlah barang sistem dengan jumlah barang fisik adalah karena salah pencatatan, sistem akan menampilkan fitur koreksi transaksi pembelian dan penjualan agar pengguna dapat mengubah kesalahan pencatatan jumlah barang yang dibeli atau dijual pada suatu transaksi. Setelah melakukan koreksi, sistem secara otomatis akan mengupdate informasi transaksi tersebut serta data mutasi barang terkait. Kuantitas barang sistem kemudian disesuaikan dengan jumlah barang fisik dan dilakukan *update* harga jual untuk barang tersebut jika telah terjadi perubahan di transaksi pembelian. Sintaks dari proses stok opname ini dapat dilihat pada listing berikut ini.

**Listing 5.3. Proses Stok Opname**

|  |
| --- |
| public function opnameBarang(Request $request){          $kode\_brg = $request->input('kode\_brg');          $nama\_brg = $request->input('nama\_brg');          $kode\_divisi = $request->input('kode\_divisi');          $kode\_jenis = $request->input('kode\_jenis');          $kode\_type = $request->input('kode\_type');          $packing = $request->input('packing');          $quantity = $request->input('quantity');          $qty\_awal = $request->input('qty\_awal');          $id\_satuan = $request->input('id\_satuan');          $hrg\_jual = $request->input('hrg\_jual');          $kode\_gudang = $request->input('kode\_gudang');          $keterangan = $request->input('keterangan');          $qty\_fisik = $request->input('qty\_fisik');          $transaction = $request->input('transaction');          $no\_bukti = $request->input('no\_bukti');          $qty\_order = $request->input('qty\_order');          $hrg\_total = $request->input('hrg\_total');          DB::table('opname\_stok')->insert([              'tanggal' => Carbon::now()->format('d-m-Y'),              'kode\_brg' => $kode\_brg,              'nama\_brg' => $nama\_brg,              'id\_satuan' => $id\_satuan,              'kode\_gudang' => $kode\_gudang,              'qty\_sistem' => $qty\_awal,              'qty\_fisik' => $qty\_fisik,              'selisih' => $qty\_fisik - $qty\_awal,              'keterangan' => $keterangan          ]);          $keteranganArray = ["BARANG RUSAK", "BARANG EXPIRED"  ,"SALAH PENCATATAN"];          if($keterangan == "BARANG RUSAK" || $keterangan  == "BARANG EXPIRED"){              $master = Master::where('kode\_brg', $kode\_brg)                  ->where('nama\_brg', $nama\_brg)                  ->where('kode\_gudang', $kode\_gudang)                  ->where('keterangan', $keterangan)                  ->first();              $quantity = abs($quantity);              if($master){                  DB::table('invmaster')                  ->where('kode\_brg', $kode\_brg)                  ->where('nama\_brg', $nama\_brg)                  ->where('kode\_gudang', $kode\_gudang)                  ->where('keterangan', [$keterangan])                  ->increment('quantity', $quantity);                }else{                  DB::table('invmaster')->insert([                      'kode\_brg' => $kode\_brg,                      'nama\_brg' => $nama\_brg,                      'kode\_divisi' => $kode\_divisi,                      'kode\_jenis' => $kode\_jenis,                      'kode\_type' => $kode\_type,                      'packing' => $packing,                      'quantity' => $quantity,                      'id\_satuan' => $id\_satuan,                      'hrg\_jual' => $hrg\_jual,                      'kode\_gudang' => $kode\_gudang,                      'keterangan' => $keterangan,                  ]);              }                DB::table('mutasi\_stok')->insert([                  'no\_bukti' => "-",                  'tanggal' => Carbon::now()->format('Y-m-d'),                  'kode\_brg' => $kode\_brg,                  'nama\_brg' => $nama\_brg,                  'id\_satuan' => $id\_satuan,                  'kode\_gudang' => $kode\_gudang,                  'stok\_awal' => $qty\_awal,                  'qty\_masuk' => 0,                  'qty\_keluar' => 0,                  'qty\_rusak\_exp' => $quantity,                  'stok\_akhir' => $qty\_awal - $quantity              ]);                DB::table('invmaster')                  ->where('kode\_brg', $kode\_brg)                  ->where('nama\_brg', $nama\_brg)                  ->where('kode\_gudang', $kode\_gudang)                  ->whereNotIn('keterangan', $keteranganArray)                  ->decrement('quantity', $quantity);            } else{              if($transaction == 'pembelian'){                  DB::table('beli\_dtl')->where('no\_bukti', $no\_bukti)                      ->where('kode\_brg', $kode\_brg)                      ->where('nama\_brg', $nama\_brg)                      ->where('kirim\_gudang', $kode\_gudang)                      ->update([                          'qty\_order' => $qty\_order,                          'hrg\_total' => $hrg\_total                      ]);                  $sub\_total = DB::table('beli\_dtl')                      ->where('no\_bukti', $no\_bukti)                      ->sum('hrg\_total');                    $beli\_data = DB::table('beli')->where('no\_bukti',  $no\_bukti)->first();                  $sub\_total = floatval($sub\_total);                  $total = $sub\_total + ($sub\_total \* ($beli\_data->  persen\_ppn / 100));                    DB::table('beli')                      ->where('no\_bukti', $no\_bukti)                      ->update([                          'sub\_total' => $sub\_total,                          'total' => $total                      ]);                  DB::table('mutasi\_stok')                      ->where('no\_bukti', $no\_bukti)                      ->where('kode\_brg', $kode\_brg)                      ->where('nama\_brg', $nama\_brg)                      ->where('kode\_gudang', $kode\_gudang)                      ->update([                          'qty\_masuk' => $qty\_order,                          'stok\_akhir' => DB::raw('stok\_awal + ' . $qty\_order)                      ]);              } else{                  DB::table('jual\_dtl')->where('no\_bukti', $no\_bukti)                      ->where('kode\_brg', $kode\_brg)                      ->where('nama\_brg', $nama\_brg)                      ->where('kode\_gudang', $kode\_gudang)                      ->update([                          'qty\_order' => $qty\_order,                          'hrg\_total' => $hrg\_total                      ]);                  $sub\_total = DB::table('jual\_dtl')                      ->where('no\_bukti', $no\_bukti)                      ->sum('hrg\_total');                    $jual\_data = DB::table('jual')->where('no\_bukti',  $no\_bukti)->first();                  $sub\_total = floatval($sub\_total);                  $total = $sub\_total + ($sub\_total \* ($jual\_data->  persen\_ppn / 100));                    DB::table('jual')                      ->where('no\_bukti', $no\_bukti)                      ->update([                          'sub\_total' => $sub\_total,                          'total' => $total                      ]);                    DB::table('mutasi\_stok')                      ->where('no\_bukti', $no\_bukti)                      ->where('kode\_brg', $kode\_brg)                      ->where('nama\_brg', $nama\_brg)                      ->where('kode\_gudang', $kode\_gudang)                      ->update([                          'qty\_keluar' => $qty\_order,                          'stok\_akhir' => DB::raw('stok\_awal - ' . $qty\_order)                      ]);              }              $dataRow = DB::table('mutasi\_stok')                  ->where('no\_bukti', $no\_bukti)                  ->where('kode\_brg', $kode\_brg)                  ->where('nama\_brg', $nama\_brg)                  ->where('kode\_gudang', $kode\_gudang)                  ->get(['id', 'stok\_akhir']);                foreach($dataRow as $row){                  $rowId = $row->id;                  $rowStokAkhir = $row->stok\_akhir;              }                $rowsToUpdate = DB::table('mutasi\_stok')                  ->where('kode\_brg', $kode\_brg)                  ->where('nama\_brg', $nama\_brg)                  ->where('kode\_gudang', $kode\_gudang)                  ->where('id', '>', $rowId)                  ->orderBy('id')                  ->get();              $count = count($rowsToUpdate);              for($i = 0; $i < $count; $i++){                  $currentRow = $rowsToUpdate[$i];                  $stokAwal = $rowStokAkhir;                  $masuk = $currentRow->qty\_masuk;                  $keluar = $currentRow->qty\_keluar;                  $rusakExp = $currentRow->qty\_rusak\_exp;                  $stokAkhir = $stokAwal + $masuk - $keluar - $rusakExp;                  DB::table('mutasi\_stok')                      ->where('id', $currentRow->id)                      ->update([                          'stok\_akhir' => $stokAkhir                      ]);                  if ($i < $count - 1) {                      DB::table('mutasi\_stok')                          ->where('id', $rowsToUpdate[$i + 1]->id)                          ->update([                              'stok\_awal' => $stokAkhir                          ]);                  }                  $rowStokAkhir = $stokAkhir;              }              DB::table('invmaster')                  ->where('kode\_brg', $kode\_brg)                  ->where('nama\_brg', $nama\_brg)                  ->where('kode\_gudang', $kode\_gudang)                  ->whereNotIn('keterangan', $keteranganArray)                  ->update([                      'quantity' => $qty\_fisik                  ]);              $transactions = DB::table('beli\_dtl')                  ->where('kode\_brg', $kode\_brg)                  ->get();              $totalCost = 0;              $currentQuantity = 0;              foreach($transactions as $transaction){                  $totalCost += $transaction->hrg\_total;                  $currentQuantity += $transaction->qty\_order;              }              $minSellPrice = $totalCost / $currentQuantity;              $sellPrice = $minSellPrice + ($minSellPrice \* 0.5);              DB::table('invmaster')                  ->where('kode\_brg', $kode\_brg)                  ->whereNotIn('keterangan', $keteranganArray)                  ->update(['hrg\_jual' => $sellPrice]);          }          return response()->json(['success' => true]);      } |

**5.2.3. Proses pada Halaman Transaksi Pembelian**

Pada halaman ini, ditampilkan daftar transaksi pembelian dari *supplier*. Data-data transaksi pembelian diambil menggunakan fungsi beli yang akan mengambil semua data dari tabel beli pada *database*. Pengguna dapat melakukan *filter* data transaksi berdasarkan tanggal melalui sebuah *date picker*. Saat pengguna memilih tanggal tertentu, data tersebut kemudian dikirim ke fungsi beli untuk dapat memfilter data transaksi berdasarkan tanggal. Setelah melakukan *filter*, dilakukan proses *sorting* secara *descending* sehingga data transaksi dengan tanggal yang terbaru akan ditampilkan terlebih dahulu. Setelah itu, diterapkan juga *paging* sebanyak 5 data ditampilkan untuk setiap halaman. Untuk informasi *supplier* dan gudang di setiap transaksi, dilakukan pengambilan seluruh data pada tabel supplier dan *invgudang*. Pada akhirnya, data transaksi beli, *supplier*, dan gudang akan dikirim ke halaman transaksi beli untuk ditampilkan pada tabel. Sintaks untuk menampilkan data transaksi pembelian dapat dilihat pada listing di bawah ini.

**Listing 5.4. Proses Pengambilan Data Transaksi Pembelian**

|  |
| --- |
| public function beli(Request $request){          $selectedDate = $request->get('selectedDate');          $query = Beli::query();          if($selectedDate){              $query->where('tanggal', $selectedDate);          }          $beli = $query->orderBy('tanggal', 'desc')->paginate(5);          $supplier = Supplier::all();          $gudang = Gudang::all();         return view('transaksi.beli',compact('beli','supplier','gudang',  'selectedDate'));      } |

Pada halaman ini, pengguna dapat mengakses halaman tambah transaksi dengan menekan tombol “Tambah Transaksi”. Pengguna akan menginputkan informasi untuk membuat transaksi baru beserta *detail* transaksi pada *input field* yang disediakan. Setelah mengisi semua *field*, informasi yang diinputkan kemudian dikirim ke fungsi *store* untuk disimpan ke *database*. Jika proses penambahan transaksi dan *detail* berhasil, maka pengguna diarahkan kembali ke halaman transaksi pembelian dan muncul notifikasi bahwa proses penambahan berhasil. Sintaks dari proses tambah transaksi dan *detail* pembelian telah dijabarkan pada listing di bawah ini.

**Listing 5.5. Proses Tambah *Detail* dan Transaksi Pembelian**

|  |
| --- |
| public function store(Request $request)  {  $data = new Beli();  $data->no\_bukti = $request->get('no\_bukti');  $data->tanggal = $request->get('datepicker');  $data->kode\_supp = $request->get('select\_supplier');  $data->sub\_total = $request->get('sub\_total');  $data->persen\_ppn = $request->get('persen\_ppn');  $data->total = $request->get('total');  $data->lunas = 'Belum Lunas';  $data->status = 'Belum Terkirim';  $data->create\_time = Carbon::now()->format('d-m-Y');  $data->author = auth()->user()->name;  $data->jatuh\_tempo = Carbon::parse($data->tanggal)  ->addMonth()->format('d-m-Y');  $data->tgl\_lunas = '-';  $data->tgl\_terkirim = '-';  $data->save();  $kode\_brg = $request->get('kode\_brg');  $nama\_brg = $request->get('nama\_brg');  $qty\_order = $request->get('qty\_order');  $packing = $request->get('packing');  $id\_satuan = $request->get('select\_satuan');  $hrg\_per\_unit = $request->get('hrg\_per\_unit');  $hrg\_total = $request->get('hrg\_total');  $kirim\_gudang = $request->get('select\_gudang');  foreach($kode\_brg as $key => $value) {  $detail = new BeliDetail();  $detail->no\_bukti = $data->no\_bukti;  $detail->kode\_brg = $kode\_brg[$key];  $detail->nama\_brg = $nama\_brg[$key];  $detail->qty\_order = $qty\_order[$key];  $detail->packing = $packing[$key];  $detail->id\_satuan = $id\_satuan[$key];  $detail->hrg\_per\_unit = $hrg\_per\_unit[$key];  $detail->hrg\_total = $hrg\_total[$key];  $detail->kirim\_gudang = $kirim\_gudang[$key];  $detail->save();  }  return redirect()->route('pembelian')->with('status','Hooray!!  Your new transaction is already inserted');  } |

Pengguna dapat melakukan *update* status pembayaran menjadi lunas dengan menekan tombol “Belum Lunas” dan *update* status pengiriman menjadi “Sudah Terkirim” jika menekan tombol “Belum Terkirim”. Proses *update* status pembayaran dilakukan dalam fungsi *updateBayar* yang menerima data *no\_bukti* dan tanggal pembayaran. *no\_bukti* akan menjadi indikator transaksi yang perlu diubah status pembayarannya. Setelah menemukan transaksi yang ingin diupdate, status pembayaran pada transaksi tersebut kemudian diubah dari “Belum Lunas” menjadi “Lunas” dan tanggal pembayaran akan disimpan juga pada transaksi tersebut. Hasil perubahan kemudian disimpan pada *database* dan muncul notifikasi bahwa *update* status pembayaran sukses dilakukan.

Proses *update* status pengiriman dilakukan dalam fungsi *updateKirim* yang juga menerima data *no\_bukti* sebagai acuan untuk menemukan transaksi yang ingin diubah status pengirimannya dari “Belum Terkirim” menjadi “Sudah Terkirim” dan data tanggal terkirimnya barang. Perubahan tersebut dilakukan pada *database* dan berikutnya dilakukan proses penyimpanan informasi barang masuk ke tabel *mutasi\_stok* untuk ditampilkan pada fitur kartu stok. Setelah itu, dilakukan proses penambahan kuantitas barang jika pada sistem telah terdapat informasi mengenai barang yang dibeli. Jika barang belum ada sebelumnya pada gudang, maka dilakukan penambahan data barang baru. Selanjutnya dilakukan proses pembaharuan harga jual untuk barang yang dibeli menggunakan metode *average cost* karena terjadi perubahan jumlah stok barang tersebut pada gudang.

Proses perhitungan harga jual dengan metode *average cost* dilakukan dengan cara menjumlahkan harga total dari semua pembelian barang terkait. Kemudian dilakukan pembagian antara harga total dengan total jumlah barang yang pernah dibeli untuk mendapatkan rata-rata biaya pembelian per kuantitas barang. Harga jual barang ditentukan dengan menggunakan harga rata-rata pembelian barang per unit, kemudian dinaikkan sebesar 50%. Harga jual tersebut kemudian disimpan ke *database*. Setelah itu, pengguna diarahkan kembali ke halaman transaksi pembelian dan dimunculkan notifikasi bahwa status pengiriman berhasil dirubah. Sintaks dari proses pengubahan status pembayaran dan pengiriman dapat dilihat pada listing di bawah ini.

**Listing 5.6. Proses *Update* Status Pembayaran dan Pengiriman pada Transaksi Pembelian**

|  |
| --- |
| public function updateBayar(Request $request)      {          $no\_bukti = $request->input('no\_bukti');          $tgl\_lunas = $request->input('tgl\_lunas');          $beli = Beli::where('no\_bukti', $no\_bukti)->firstOrFail();          $beli->lunas = 'Lunas';          $beli->tgl\_lunas = $tgl\_lunas;          $beli->save();          return response()->json(['success' => true]);      }      public function updateKirim(Request $request)      {          $no\_bukti = $request->input('no\_bukti');          $tgl\_terkirim = $request->input('tgl\_terkirim');          $beli = Beli::where('no\_bukti', $no\_bukti)->firstOrFail();          $beli->status = 'Sudah Terkirim';          $beli->tgl\_terkirim = $tgl\_terkirim;          $beli->save();          $beliDetail = BeliDetail::where('no\_bukti', $no\_bukti)->get();          $keteranganArray = ["BARANG RUSAK",  "BARANG EXPIRED", "BARANG RUSAK & EXPIRED"];          foreach ($beliDetail as $detail) {              $master = DB::table('invmaster')                  ->where('kode\_brg', $detail->kode\_brg)                  ->where('nama\_brg', $detail->nama\_brg)                  ->where('kode\_gudang', $detail->kirim\_gudang)                  ->whereNotIn('keterangan', $keteranganArray)                  ->first();              if ($master) {                  $stok\_awal = $master->quantity;                  DB::table('mutasi\_stok')->insert([                      'no\_bukti' => $no\_bukti,                      'tanggal' => Carbon::parse($tgl\_terkirim)->format('Y-m-d'),                      'kode\_brg' => $detail->kode\_brg,                      'nama\_brg' => $detail->nama\_brg,                      'id\_satuan' => $detail->id\_satuan,                      'kode\_gudang' => $detail->kirim\_gudang,                      'stok\_awal' => $stok\_awal,                      'qty\_masuk' => $detail->qty\_order,                      'qty\_keluar' => 0,                      'qty\_rusak\_exp' => 0,                      'stok\_akhir' => $stok\_awal + $detail->qty\_order                  ]);                  DB::table('invmaster')                      ->where('kode\_brg', $detail->kode\_brg)                      ->where('nama\_brg', $detail->nama\_brg)                      ->where('kode\_gudang', $detail->kirim\_gudang)                      ->whereNotIn('keterangan', $keteranganArray)                      ->increment('quantity', $detail->qty\_order);                    $transactions = DB::table('beli\_dtl')                      ->where('kode\_brg', $detail->kode\_brg)                      ->get();                  $totalCost = 0;                  $currentQuantity = 0;                  foreach($transactions as $transaction){                      $totalCost += $transaction->hrg\_total;                      $currentQuantity += $transaction->qty\_order;                  }                  $minSellPrice = $totalCost / $currentQuantity;                  $sellPrice = $minSellPrice + ($minSellPrice \* 0.5);                  DB::table('invmaster')                      ->where('kode\_brg', $detail->kode\_brg)                      ->whereNotIn('keterangan', $keteranganArray)                      ->update(['hrg\_jual' => $sellPrice]);              } else {                  DB::table('mutasi\_stok')->insert([                      'no\_bukti' => $no\_bukti,                      'tanggal' => Carbon::parse($tgl\_terkirim)->format('Y-m-d'),                      'kode\_brg' => $detail->kode\_brg,                      'nama\_brg' => $detail->nama\_brg,                      'id\_satuan' => $detail->id\_satuan,                      'kode\_gudang' => $detail->kirim\_gudang,                      'stok\_awal' => 0,                      'qty\_masuk' => $detail->qty\_order,                      'qty\_keluar' => 0,                      'qty\_rusak\_exp' => 0,                      'stok\_akhir' => $detail->qty\_order                  ]);                  DB::table('invmaster')->insert([                      'kode\_brg' => $detail->kode\_brg,                      'nama\_brg' => $detail->nama\_brg,                      'packing' => $detail->packing,                      'quantity' => $detail->qty\_order,                      'id\_satuan' => $detail->id\_satuan,                      'kode\_gudang' => $detail->kirim\_gudang,                      'keterangan' => '-',                  ]);                  $transactions = DB::table('beli\_dtl')                      ->where('kode\_brg', $detail->kode\_brg)                      ->get();                  $totalCost = 0;                  $currentQuantity = 0;                  foreach($transactions as $transaction){                      $totalCost += $transaction->hrg\_total;                      $currentQuantity += $transaction->qty\_order;                  }                  $minSellPrice = $totalCost / $currentQuantity;                  $sellPrice = $minSellPrice + ($minSellPrice \* 0.5);                  DB::table('invmaster')                      ->where('kode\_brg', $detail->kode\_brg)                      ->update(['hrg\_jual' => $sellPrice]);              }          }          return response()->json(['success' => true]);      } |

Pengguna dapat melakukan navigasi dari halaman transaksi pembelian ke halaman *detail* pembelian dengan menekan tombol “Details” pada suatu baris data transaksi. Halaman ini akan menampilkan data *detail* dari suatu transaksi dalam bentuk tabel. *Detail* yang ditampilkan adalah data barang, kuantitas pesanan, serta harga barang yang dipesan perusahaan dari *supplier* melalui transaksi tersebut, serta gudang tujuan pengiriman barang. Data *detail* transaksi pembelian diambil menggunakan fungsi *showDetail* berparameter *no\_bukti* yang akan menjadi acuan untuk mencari data *detail* berdasarkan nomor nota. Setelah mendapatkan semua data *detail* dari suatu transaksi, data tersebut kemudian akan dikirim ke halaman *detail* transaksi pembelian untuk ditampilkan dalam bentuk tabel. Untuk data satuan dan gudang barang pada *detail* akan diambil dari tabel satuan dan *invgudang*. Sintaks dari proses diatas dapat dilihat pada listing di bawah ini.

**Listing 5.7. Proses Pengambilan Data *Detail* Transaksi Pembelian**

|  |
| --- |
| public function showDetail($no\_bukti){  $beliDetail = BeliDetail::where('no\_bukti', $no\_bukti)->get();  $satuan = Satuan::all();  $gudang = Gudang::all();  return view('transaksi.belidetail', compact('beliDetail','satuan',  'gudang','no\_bukti'));  } |

Pengguna dapat melihat faktur beli dari masing-masing transaksi pembelian dengan menekan tombol “Lihat Faktur” pada masing-masing baris data transaksi. Jika tombol tersebut ditekan, sistem akan mengirimkan data *no\_bukti* dari baris data tersebut ke fungsi *cetak\_pdf* yang menerima parameter ini agar sistem dapat mengambil data transaksi untuk kemudian ditampilkan dalam sebuah dokumen PDF. Pembuatan dokumen faktur beli melibatkan sebuah *library* Dompdf yang disediakan oleh Laravel. Fungsi pengambilan data faktur beli ini dapat dilihat pada listing di bawah ini.

**Listing 5.8. Proses Pengambilan Data Faktur Beli**

|  |
| --- |
| public function cetak\_pdf($no\_bukti)      {          $beli = Beli::find($no\_bukti);          $beliDetail = BeliDetail::where('no\_bukti', $no\_bukti)->get();          $supplier = Supplier::all();          $gudang = Gudang::all();          $satuan = Satuan::all();          $data = $beli;          $dataDetail = $beliDetail;            $view = View::make('transaksi.belipdf', ['data'=>  $data, 'dataDetail'=>$dataDetail, 'supplier'=>  $supplier, 'gudang'=>$gudang, 'satuan'=>$satuan]);          $pdf = new Dompdf();          $pdf->loadHtml($view->render());          $pdf->setPaper('A4', 'landscape');          $pdf->render();          return response($pdf->output(), 200, [              'Content-Type' => 'application/pdf',              'Content-Disposition' => 'inline;',          ]);      } |

**5.2.4. Proses pada Halaman Transaksi Penjualan**

Halaman transaksi penjualan akan menampilkan data transaksi penjualan ke *customer*. Data transaksi diambil menggunakan fungsi jual yang berparameter *request* untuk menerima data *filter* berdasarkan tanggal yang akan dikirim ke fungsi ini jika pengguna menginputkan suatu tanggal pada *date picker* di halaman transaksi penjualan. Jika tanggal *filter* tersebut didapatkan, maka data tersebut akan disertakan pada *query* tabel jual agar pada proses pengambilan data transaksi, hanya diambil data yang memiliki *value* kolom tanggal yang sama dengan tanggal *filter*. Jika tidak ada *request* dari halaman transaksi penjualan, maka proses *filter* tidak akan dijalankan sehingga fungsi *query* akan mengambil seluruh data transaksi penjualan. Data transaksi akan diurutkan berdasarkan tanggal secara *descending* sehingga transaksi dengan tanggal terbaru akan ditampilkan terlebih dahulu pada tabel. Fitur *paging* juga diterapkan dengan menampilkan 5 data transaksi untuk setiap halaman. Untuk data *customer* akan diambil dari tabel *customer*. Data transaksi penjualan dan *customer* akan dikirim ke halaman transaksi penjualan untuk ditampilkan pada tabel. Sintaks dari proses pengambilan data transaksi ini telah dijabarkan pada listing di bawah ini.

**Listing 5.9. Proses Pengambilan Data Transaksi Penjualan**

|  |
| --- |
| public function jual(Request $request){          $selectedDate = $request->get('selectedDateJual');          $query = Jual::query();          if($selectedDate){              $query->where('tanggal', $selectedDate);          }          $jual = $query->orderBy('tanggal', 'desc')->paginate(5);          $customer = Customer::all();          return view('transaksi.jual',compact('jual','customer','selectedDate'));      } |

Pada halaman ini, disediakan juga tombol “Tambah Transaksi” yang akan mengarahkan pengguna menuju halaman tambah transaksi. Pengguna dapat menginputkan informasi transaksi serta *detail* penjualan baru dengan memasukkan data ke kumpulan *input field*. Setelah proses *input* selesai dilakukan, informasi-informasi tersebut kemudian dikirim ke fungsi *store* yang akan menerima semua data untuk dimasukkan ke *database*. Setelah itu, pengguna akan diarahkan kembali ke halaman transaksi penjualan dan muncul notifikasi bahwa proses penambahan transaksi serta *detail* berhasil dilakukan. Sintaks dari proses diatas dapat dilihat pada listing di bawah ini.

**Listing 5.10. Proses Tambah *Detail* dan Transaksi Penjualan**

|  |
| --- |
| public function store(Request $request)  {  $data = new Jual();  $data->no\_bukti = $request->get('no\_bukti');  $data->tanggal = $request->get('datepicker');  $data->kode\_cust = $request->get('select\_customer');  $data->sub\_total = $request->get('sub\_total');  $data->persen\_ppn = $request->get('persen\_ppn');  $data->total = $request->get('total');  $data->lunas = 'Belum Lunas';  $data->status = 'Belum Terkirim';  $data->create\_time = Carbon::now()->format('d-m-Y');  $data->author = auth()->user()->name;  $data->jatuh\_tempo = Carbon::parse($data->tanggal)  ->addMonth()->format('d-m-Y');  $data->tgl\_lunas = '-';  $data->tgl\_terkirim = '-';  $data->save();  $id\_brg = $request->get('id\_brg');  $kode\_brg = $request->get('kode\_brg');  $nama\_brg = $request->get('nama\_brg');  $qty\_order = $request->get('qty\_order');  $id\_satuan = $request->get('select\_satuan');  $hrg\_per\_unit = $request->get('hrg\_per\_unit');  $hrg\_total = $request->get('hrg\_total');  $kode\_gudang = $request->get('select\_gudang');  foreach($kode\_brg as $key => $value) {  $detail = new JualDetail();  $detail->id\_brg = $id\_brg[$key];  $detail->no\_bukti = $data->no\_bukti;  $detail->kode\_brg = $kode\_brg[$key];  $detail->nama\_brg = $nama\_brg[$key];  $detail->qty\_order = $qty\_order[$key];  $detail->id\_satuan = $id\_satuan[$key];  $detail->hrg\_per\_unit = $hrg\_per\_unit[$key];  $detail->hrg\_total = $hrg\_total[$key];  $detail->kode\_gudang = $kode\_gudang[$key];  $detail->save();  }  return redirect()->route('penjualan')->with('status','Hooray!!  Your new transaction is already inserted');  } |

Halaman ini juga menyediakan fitur *update* status pembayaran dan pengiriman. Proses *update* status pembayaran dilakukan pada fungsi *updateBayar* yang akan merubah data lunas pada suatu transaksi dari yang sebelumnya dinyatakan “Belum Lunas” menjadi “Lunas”. Proses *update* status pengiriman dilakukan pada fungsi *updateKirim* yang akan merubah data status pada suatu transaksi dari “Belum Terkirim” menjadi “Sudah Terkirim”. Kedua fungsi membutuhkan data *no\_bukti* untuk dapat mencari transaksi penjualan yang ingin diupdate statusnya. Setelah mengubah status pengiriman menjadi “Sudah Terkirim”, dilakukan penambahan jumlah barang yang keluar sesuai dengan setiap transaksi pada tabel *mutasi\_stok* untuk ditampilkan pada fitur kartu stok. Berikutnya, dilakukan pengurangan jumlah stok barang yang dijual. Sintaks dari proses diatas secara keseluruhan telah ditampilkan pada listing berikut ini.

**Listing 5.11. Proses *Update* Status Pembayaran dan Pengiriman pada Transaksi Penjualan**

|  |
| --- |
| public function updateBayar(Request $request)      {          $no\_bukti = $request->input('no\_bukti');          $tgl\_lunas = $request->input('tgl\_lunas');          $jual = Jual::where('no\_bukti', $no\_bukti)->firstOrFail();          $jual->lunas = 'Lunas';          $jual->tgl\_lunas = $tgl\_lunas;          $jual->save();          return response()->json(['success' => true]);      }      public function updateKirim(Request $request)      {          $no\_bukti = $request->input('no\_bukti');          $tgl\_terkirim = $request->input('tgl\_terkirim');          $jual = Jual::where('no\_bukti', $no\_bukti)->firstOrFail();          $jual->status = 'Sudah Terkirim';          $jual->tgl\_terkirim = $tgl\_terkirim;          $jual->save();          $jualDetail = JualDetail::where('no\_bukti', $no\_bukti)->get();          foreach ($jualDetail as $detail) {              $master = Master::find($detail->id\_brg);              if ($master) {                  DB::table('mutasi\_stok')->insert([                      'no\_bukti' => $no\_bukti,                      'tanggal' => Carbon::parse($tgl\_terkirim)->format('Y-m-d'),                      'kode\_brg' => $detail->kode\_brg,                      'nama\_brg' => $detail->nama\_brg,                      'id\_satuan' => $detail->id\_satuan,                      'kode\_gudang' => $detail->kode\_gudang,                      'stok\_awal' => $master->quantity,                      'qty\_masuk' => 0,                      'qty\_keluar' => $detail->qty\_order,                      'qty\_rusak\_exp' => 0,                      'stok\_akhir' => $master->quantity - $detail->qty\_order                  ]);                  $master->quantity -= $detail->qty\_order;                  $master->save();              }          }          return response()->json(['success' => true]);      } |

Pengguna dapat mengakses halaman *detail* untuk setiap transaksi penjualan agar dapat melihat data *detail* berupa informasi mengenai barang yang dijual ke *customer*. Sistem akan menjalankan fungsi *showDetail* yang menerima parameter *no\_bukti* yang akan digunakan untuk menemukan semua *detail* yang terkait dengan suatu transaksi. Untuk data satuan dan gudang akan diambil dari tabel satuan dan *invgudang*. Seluruh data *detail* dan satuan kemudian dikirim ke halaman *detail* penjualan untuk ditampilkan pada tabel. Sintaks dari proses ini dapat dilihat pada listing berikut ini.

**Listing 5.12. Proses Pengambilan Data *Detail* Transaksi Penjualan**

|  |
| --- |
| public function showDetail($no\_bukti){  $jualDetail = JualDetail::where('no\_bukti', $no\_bukti)->get();  $satuan = Satuan::all();  $gudang = Gudang::all();  return view('transaksi.jualdetail', compact('jualDetail','satuan',  'gudang','no\_bukti'));  } |

Pada halaman ini disediakan sebuah tombol yang digunakan untuk melihat *invoice* suatu transaksi penjualan. Jika pengguna menekan tombol tersebut, sistem akan mengirimkan data *no\_bukti* ke fungsi *cetak\_pdf*  yang nantinya digunakan untuk mencari dan mengambil data sebuah transaksi. Sistem kemudian mengirim data yang didapat dari hasil pencarian ke sebuah halaman yang menampilkan *invoice* dalam bentuk PDF. Pengguna dapat mengunduh invoice tersebut untuk diberikan ke *customer*. Fungsi proses pengambilan data *invoice* dapat dilihat pada listing di bawah ini.

**Listing 5.13. Proses Pengambilan Data *Invoice***

|  |
| --- |
| public function cetak\_pdf($no\_bukti)      {          $jual = Jual::find($no\_bukti);          $jualDetail = JualDetail::where('no\_bukti', $no\_bukti)->get();          $customer = Customer::all();          $satuan = Satuan::all();            $data = $jual;          $dataDetail = $jualDetail;          $view = View::make('transaksi.jualpdf', ['data'=>  $data, 'dataDetail'=>$dataDetail, 'customer'=>  $customer, 'satuan'=>$satuan]);          $pdf = new Dompdf();          $pdf->loadHtml($view->render());          $pdf->setPaper('A4', 'landscape');          $pdf->render();          return response($pdf->output(), 200, [              'Content-Type' => 'application/pdf',              'Content-Disposition' => 'inline;',          ]);      } |

**5.2.5. Proses pada Halaman Kartu Stok**

Halaman ini menampilkan informasi barang yang masuk dan keluar di semua transaksi. Data pada halaman ini didapat dari tabel *mutasi\_stok* yang dikirim melalui fungsi *kartuStok* yang menerima parameter *request* yang berisikan data-data yang dibutuhkan untuk proses *filter* tabel pada kartu stok. *Filter* yang dapat dilakukan adalah berdasarkan gudang, kode barang, nama barang, jenis transaksi, serta tanggal terjadinya mutasi. Langkah pertama yang dilakukan pada fungsi ini adalah mendapatkan semua data untuk proses *filter*. Kemudian dilakukan proses *query* menggunakan semua data *filter* untuk mendapatkan data dengan klasifikasi yang sesuai. Setelah melakukan *filter*, data mutasi yang didapat kemudian diterapkan *sorting* berdasarkan tanggal mutasi secara *ascending* agar data yang ditampilkan pada tabel terlebih dahulu adalah data dengan tanggal yang terlama. Untuk data gudang dan satuan akan diambil dari tabel *invgudang* dan satuan. Data mutasi hasil *filter* kemudian dikirim ke halaman kartu stok untuk ditampilkan dalam bentuk tabel. Sintaks dari proses diatas secara keseluruhan dapat dilihat pada listing di bawah ini.

**Listing 5.14. Proses Pengambilan Data Kartu Stok**

|  |
| --- |
| public function kartuStok(Request $request){          $selectedGudang = $request->get('gudangKartu');          $search = $request->get('searchKartu');          $selectedTrans = $request->get('selectedTrans');          $tglAwal = $request->get('tglAwal');          $tglAkhir = $request->get('tglAkhir');          $tglAwalFormatted = $tglAwal ? Carbon::createFromFormat(  'd-m-Y', $tglAwal)->startOfDay()->format('d-m-Y') :Carbon::  createFromFormat('Y-m-d', MutasiStok::min('tanggal'))  ->format('d-m-Y');          $tglAkhirFormatted = $tglAkhir ? Carbon::createFromFormat(  'd-m-Y', $tglAkhir)->endOfDay()->format('d-m-Y') : Carbon::  createFromFormat('Y-m-d', MutasiStok::max('tanggal'))  ->format('d-m-Y');          $query = MutasiStok::query();          if($selectedGudang && $selectedGudang != 'All'){              $query->where('kode\_gudang', $selectedGudang);          }            if($search){              $query->where(function ($query) use ($search){                  $query->where('kode\_brg', 'like', '%'.$search.'%')                        ->orWhere('nama\_brg', 'like', '%'.$search.'%');              });          }          if($selectedTrans && $selectedTrans != 'All' && $selectedTrans  != "-"){              $query->where(function ($query) use ($selectedTrans){                  $query->where('no\_bukti', 'like', '%'.$selectedTrans.'%');              });          }          if($selectedTrans == "-"){              $query->where('no\_bukti', '-');          }          if($tglAwal && $tglAkhir){              $tglAwalFlipped = Carbon::createFromFormat(  'd-m-Y', $tglAwalFormatted)->format('Y-m-d');              $tglAkhirFlipped = Carbon::createFromFormat(  'd-m-Y', $tglAkhirFormatted)->format('Y-m-d');              $query->whereBetween('tanggal', [$tglAwalFlipped  , $tglAkhirFlipped]);          } elseif ($tglAwal){              $tglAwalFlipped = Carbon::createFromFormat(  'd-m-Y', $tglAwalFormatted)->format('Y-m-d');              $query->where('tanggal', '>=', $tglAwalFlipped);          } elseif ($tglAkhir){              $tglAkhirFlipped = Carbon::createFromFormat(  'd-m-Y', $tglAkhirFormatted)->format('Y-m-d');              $query->where('tanggal', '<=', $tglAkhirFlipped);          }          $kartuStok = $query->orderBy('tanggal', 'asc')->paginate(10);          $gudang = Gudang::all();          $satuan = Satuan::all();          return view('mutasi.kartustok',compact('kartuStok','gudang',  'satuan'));      } |

Pengguna dapat melihat kartu stok sesuai pilihan *filter* yang disediakan pada halaman kartu stok. Ketika tombol “Lihat Kartu Stok” ditekan, sistem akan menjalankan fungsi *cetak\_pdf* yang digunakan untuk mengambil data mutasi barang sesuai *filter*. Untuk tahapan proses pengambilan data masih sama dengan proses pengambilan data untuk halaman kartu stok. Setelah mendapatkan data mutasi barang hasil *filter* melalui proses *query*, data tersebut kemudian dikirim ke sebuah halaman untuk menampilkan kartu stok dalam bentuk PDF yang dapat diunduh oleh pengguna. Sintaks dari proses pengambilan data untuk dokumen kartu stok dapat dilihat pada listing berikut ini.

**Listing 5.15. Proses Pengambilan Data untuk Dokumen Kartu Stok**

|  |
| --- |
| public function cetak\_pdf(Request $request)      {          $selectedGudang = $request->input('selectedGudang');          $searchText = $request->input('searchText');          $selectedTrans = $request->get('selectedTrans');          $tglAwal = $request->get('tglAwal');          $tglAkhir = $request->get('tglAkhir');          $tglAwalFormatted = $tglAwal ? Carbon::createFromFormat(  'd-m-Y', $tglAwal)->startOfDay()->format('d-m-Y') : Carbon::  createFromFormat('Y-m-d', MutasiStok::min('tanggal'))  ->format('d-m-Y');          $tglAkhirFormatted = $tglAkhir ? Carbon::createFromFormat(  'd-m-Y', $tglAkhir)->endOfDay()->format('d-m-Y') : Carbon::  createFromFormat('Y-m-d', MutasiStok::max('tanggal'))  ->format('d-m-Y');          $query = MutasiStok::query();          if($selectedGudang && $selectedGudang != 'All'){              $query->where('kode\_gudang', $selectedGudang);          }          if($searchText){              $query->where(function ($query) use ($searchText){                  $query->where('kode\_brg', 'like', '%' . $searchText . '%')                      ->orWhere('nama\_brg', 'like', '%' . $searchText . '%');              });          }          if($selectedTrans && $selectedTrans != 'All' && $selectedTrans  != "-"){              $query->where(function ($query) use ($selectedTrans){                  $query->where('no\_bukti', 'like', '%'.$selectedTrans.'%');              });          }          if($selectedTrans == "-"){              $query->where('no\_bukti', $selectedTrans);          }          if($tglAwal || $tglAkhir || ($tglAwal && $tglAkhir)){              $tglAwalFlipped = Carbon::createFromFormat(  'd-m-Y', $tglAwalFormatted)->format('Y-m-d');              $tglAkhirFlipped = Carbon::createFromFormat(  'd-m-Y', $tglAkhirFormatted)->format('Y-m-d');              $query->whereBetween('tanggal', [$tglAwalFlipped  , $tglAkhirFlipped]);          }          $totalQtyMasuk = $query->sum('qty\_masuk');          $totalQtyKeluar = $query->sum('qty\_keluar');          $totalQtyRusak = $query->sum('qty\_rusak\_exp');          $kartuStok = $query->orderBy('tanggal', 'asc')->get();          $data = $kartuStok;          $gudang = Gudang::all();          $satuan = Satuan::all();      $view = View::make('mutasi.kartustokpdf', ['data'=>  $data, 'gudang'=>$gudang, 'satuan'=>$satuan, 'totalMasuk'=>  $totalQtyMasuk, 'totalKeluar'=>$totalQtyKeluar, 'totalRusak'=>  $totalQtyRusak,'tglAwal'=>$tglAwalFormatted, 'tglAkhir'=>  $tglAkhirFormatted, 'selectedGudang'=>$selectedGudang]);          $pdf = new Dompdf();          $pdf->loadHtml($view->render());          $pdf->setPaper('A4', 'portrait');          $pdf->render();          return response($pdf->output(), 200, [              'Content-Type' => 'application/pdf',              'Content-Disposition' => 'inline;',          ]);    } |

**5.2.6. Proses pada Halaman *Supplier***

Halaman ini digunakan untuk menampilkan data *supplier* perusahaan dalam bentuk tabel. Data *supplier* akan dikirim ke halaman ini melalui fungsi *supplier* yang menerima paramater *request* untuk mendapatkan *filter search* berdasarkan nama *supplier* yang akan dikirim ke fungsi ini jika pengguna menginputkan teks di *input field search* pada halaman *supplier*. *Filter search* tersebut akan dimanfaatkan dalam proses pengambilan data *supplier* sehingga data yang dikirim hanya yang sesuai dengan teks *search* yang diinputkan pengguna. Jika tidak menggunakan *filter*, maka fungsi akan mengambil semua data dari tabel *supplier*. *Paging* juga diterapkan dengan menampilkan 5 data per halaman. Data *supplier* kemudian dikirim ke halaman *supplier* untuk ditampilkan pada tabel. Sintaks dari proses diatas dapat dilihat pada listing di bawah ini.

**Listing 5.16. Proses Pengambilan Data *Supplier***

|  |
| --- |
| public function supplier(Request $request){          $search = $request->get('search');          $query = Supplier::query();          if($search){              $query->where('nama\_supp', 'like', '%'.$search.'%');          }          $supplier = $query->paginate(5);          return view('supplier.supplier',compact('supplier','search'));      } |

Pengguna dapat melakukan penambahan data *supplier* melalui tombol “Tambah Supplier”. Pada halaman ini, disediakan beberapa *input field* yang bisa diberi informasi oleh pengguna untuk kemudian dikirim seluruh datanya ke fungsi *store* yang akan menyimpan informasi yang diinputkan pengguna ke *database*. Setelah berhasil *input* ke *database*, sistem akan mengarahkan pengguna kembali ke halaman *supplier* untuk diberi notifikasi bahwa data *supplier* berhasil ditambahkan. Proses ini dapat dilihat pada listing berikut ini.

**Listing 5.17. Proses Tambah *Supplier***

|  |
| --- |
| public function store(Request $request)  {  $data = new Supplier();  $data->kode\_supp = $request->get('kode\_supp');  $data->nama\_supp = $request->get('nama\_supp');  $data->bank = $request->get('bank');  $data->acc\_bank = $request->get('acc\_bank');  $data->alm\_1 = $request->get('alm\_1');  $data->alm\_2 = $request->get('alm\_2');  $data->kota = $request->get('kota');  $data->negara = $request->get('negara');  $data->kontak = $request->get('kontak');  $data->jabatan = $request->get('jabatan');  $data->no\_telp = $request->get('no\_telp');  $data->email = $request->get('email');  $data->save();  return redirect()->route('supplier')->with('status','Hooray!!  Your new item is already inserted');  } |

Halaman *supplier* dapat mengakses halaman *edit* *supplier* melalui tombol “Edit” yang berada di setiap baris data *supplier* pada tabel. Pengguna dapat mengubah informasi *supplier* sebelumnya dengan mengubah data *supplier* yang disajikan di setiap *input field*. Data yang berisikan informasi *supplier* didapat dari proses pengambilan data *supplier* yang dilakukan pada fungsi *edit* berparameter *kode\_supp* yang digunakan untuk mencari *supplier* yang ingin diubah datanya. Data tersebut kemudian dikirim ke halaman *edit supplier* untuk ditampilkan di setiap *input field*. Setelah pengguna mengubah informasi *supplier*, seluruh data kemudian dikirim ke fungsi *update* untuk disimpan ke *database*. Setelah itu, pengguna diarahkan kembali ke halaman *supplier* untuk diberi notifikasi bahwa proses *edit supplier* berhasil dilakukan. Sintaks dari proses ini dapat dilihat pada listing berikut ini.

**Listing 5.18. Proses *Edit* *Supplier***

|  |
| --- |
| public function edit($kode\_supp)  {  $objSupplier = Supplier::find($kode\_supp);  $data = $objSupplier;  return view('supplier.formedit',compact('data'));  }  public function update(Request $request, $kode\_supp)  {  $objSupplier = Supplier::find($kode\_supp);  $objSupplier->kode\_supp = $request->get('kode\_supp');  $objSupplier->nama\_supp = $request->get('nama\_supp');  $objSupplier->bank = $request->get('bank');  $objSupplier->acc\_bank = $request->get('acc\_bank');  $objSupplier->alm\_1 = $request->get('alm\_1');  $objSupplier->alm\_2 = $request->get('alm\_2');  $objSupplier->kota = $request->get('kota');  $objSupplier->negara = $request->get('negara');  $objSupplier->kontak = $request->get('kontak');  $objSupplier->jabatan = $request->get('jabatan');  $objSupplier->no\_telp = $request->get('no\_telp');  $objSupplier->email = $request->get('email');  $objSupplier->save();  return redirect()->route('supplier')->with('status','Your item is  up-to-date');  } |

Pada halaman *supplier* disediakan tombol “Delete” pada setiap baris data untuk menghapus suatu *supplier*. Jika pengguna menekan tombol tersebut, maka data *kode\_supp* dari *supplier* tersebut akan dikirim ke fungsi *destroy* sebagai acuan untuk menemukan data *supplier* tersebut dan kemudian dihapus dari *database*. Jika berhasil, maka pengguna diarahkan kembali ke halaman *supplier* dan muncul notifikasi bahwa proses penghapusan berhasil. Jika gagal, maka muncul pesan gagal hapus *supplier*. Sintaks dari proses ini dapat dilihat pada listing di bawah ini.

**Listing 5.19. Proses Hapus *Supplier***

|  |
| --- |
| public function destroy($kode\_supp)      {          try{              $objSupplier = Supplier::find($kode\_supp);              $objSupplier->delete();              return redirect()->route('supplier')->with('status',  'Deleted Successfully');          }          catch(\PDOException $ex){              $msg = "Delete data failed. Make sure there is no correlated  data before deleting!";              return redirect()->route('supplier')->with('status',$msg);          }      } |

**5.2.7. Proses pada Halaman *Customer***

Halaman ini digunakan untuk menampilkan data *customer* perusahaan dalam bentuk tabel. Data *customer* akan dikirim ke halaman ini melalui fungsi *customer* berparamater *request* untuk mendapatkan data *search* yang digunakan untuk *filter* berdasarkan nama *customer* yang akan dikirim ke fungsi ini jika pengguna menginputkan teks di *input field search* pada halaman *customer*. Data *search* tersebut akan dimanfaatkan dalam proses pengambilan data *customer* sehingga data yang diambil hanya yang sesuai dengan teks *search* yang telah diinputkan pengguna. Jika tidak menggunakan fitur *search* tersebut, maka fungsi akan mengambil semua data dari tabel *customer*. Fitur *paging* akan menampilkan 5 data per halaman. Data *customer* kemudian dikirim ke halaman *customer* untuk ditampilkan pada tabel. Sintaks dari proses diatas dapat dilihat pada listing berikut ini.

**Listing 5.20. Proses Pengambilan Data *Customer***

|  |
| --- |
| public function customer(Request $request){          $search = $request->get('search');          $query = Customer::query();          if($search){              $query->where('nama\_cust', 'like', '%'.$search.'%');          }          $customer = $query->paginate(5);          $salesPerson = SalesPerson::all();          return view('customer.customer',compact('customer','salesPerson',  'search'));  } |

Pengguna dapat melakukan penambahan data *customer* melalui tombol “Tambah Customer”. Pada halaman ini, disediakan beberapa *input field* yang bisa diberi informasi oleh pengguna untuk kemudian dikirim seluruh datanya ke fungsi *store* yang akan menyimpan informasi yang diinputkan pengguna ke *database*. Setelah berhasil *input* data ke *database*, sistem akan mengarahkan pengguna kembali ke halaman *customer* dan ditampilkan notifikasi bahwa data *customer* berhasil ditambahkan. Proses ini dapat dilihat pada listing berikut ini.

**Listing 5.21. Proses Tambah *Customer***

|  |
| --- |
| public function store(Request $request)      {          $data = new Customer();          $data->kode\_cust = $request->get('kode\_cust');          $data->nama\_cust = $request->get('nama\_cust');          $data->type\_cust = $request->get('type\_cust');          $data->alm\_1 = $request->get('alm\_1');          $data->alm\_2 = $request->get('alm\_2');          $data->alm\_3 = $request->get('alm\_3');          $data->kota = $request->get('kota');          $data->kontak = $request->get('kontak');          $data->no\_telp = $request->get('no\_telp');          $data->kode\_sales = $request->get('select\_sales');          $data->save();          return redirect()->route('customer')->with('status','Hooray!!  Your new item is already inserted');      } |

Halaman *customer* juga dapat mengakses halaman *edit customer* melalui tombol “Edit” yang berada di setiap baris data *customer* pada tabel. Pengguna dapat mengubah informasi *customer* dengan mengubah data *customer* yang ditampilkan di setiap *input field*. Data yang berisikan informasi *customer* didapat dari proses pengambilan data *customer* yang dilakukan pada fungsi *edit* berparameter *kode\_cust* yang digunakan untuk mencari *customer* yang ingin diubah datanya pada *database*. Data tersebut kemudian dikirim ke halaman *edit customer* untuk ditampilkan di setiap *input field*. Setelah pengguna mengubah informasi *customer*, seluruh data kemudian dikirim ke fungsi *update* untuk disimpan ke *database*. Setelah itu, pengguna diarahkan kembali ke halaman *customer* dan diberi notifikasi bahwa proses *edit customer* berhasil dilakukan. Sintaks dari proses ini dapat dilihat pada listing di bawah ini.

**Listing 5.22. Proses Edit *Customer***

|  |
| --- |
| public function edit($kode\_cust)      {          $objCustomer = Customer::find($kode\_cust);          $salesPerson = SalesPerson::all();          $data = $objCustomer;          return view('customer.formedit',compact('data','salesPerson'));      }      public function update(Request $request, $kode\_cust)      {          $objCustomer = Customer::find($kode\_cust);          $objCustomer->kode\_cust = $request->get('kode\_cust');          $objCustomer->nama\_cust = $request->get('nama\_cust');          $objCustomer->type\_cust = $request->get('type\_cust');          $objCustomer->alm\_1 = $request->get('alm\_1');          $objCustomer->alm\_2 = $request->get('alm\_2');          $objCustomer->alm\_3 = $request->get('alm\_3');          $objCustomer->kota = $request->get('kota');          $objCustomer->kontak = $request->get('kontak');          $objCustomer->no\_telp = $request->get('no\_telp');          $objCustomer->kode\_sales = $request->get('select\_sales');          $objCustomer->save();          return redirect()->route('customer')->with('status','Your item is  up-to-date');      } |

Pada halaman *customer* juga disediakan tombol “Delete” pada setiap baris data *customer* untuk menghapus suatu data *customer*. Jika pengguna menekan tombol tersebut, maka data *kode\_cust* dari *customer* tersebut akan dikirim ke fungsi *destroy* sebagai acuan untuk menemukan data *customer* berdasarkan kode *customer* agar bisa dihapus dari *database*. Jika berhasil, maka pengguna diarahkan kembali ke halaman *customer* dan muncul notifikasi bahwa proses penghapusan berhasil. Jika gagal, maka muncul notifikasi bahwa hapus *customer* gagal dilakukan. Sintaks dari proses ini dapat dilihat pada listing di bawah ini.

**Listing 5.23. Proses Hapus *Customer***

|  |
| --- |
| public function destroy($kode\_cust)      {          try{              $objCustomer = Customer::find($kode\_cust);              $objCustomer->delete();              return redirect()->route('customer')->with('status',  'Deleted Successfully');          }          catch(\PDOException $ex){              $msg = "Delete data failed. Make sure there is no correlated  data before deleting!";              return redirect()->route('customer')->with('status',$msg);          }      } |

parameter normalisasi L1 dan random state 114. Hasil uji nilai kebenaran menggunakan dataset oversampling dan parameter normalisasi L1

mendapatkan hasil yaitu 29 data dari 40 data bernilai benar. Berdasarkan hasil diatas maka sistem akan menggunakan teknik oversampling dan dengan parameter normalisasi L1.

• Hasil validasi yang telah dilakukan mendapatkan hasil bahwa sistem dapat membantu pemerintah dan masyarakat dalam mengetahui polaritas sentimen masyarakat terhadap PTM pada masa pandemi dan hasil laporan klasifikasi dapat dijadikan bahan dasar evaluasi untuk menentukan kebijakan selanjutnya.

• Fitur-fitur yang terdapat dalam sistem dalam membantu dalam hal mengumpulkan data komentar dan juga mengklasifikasikan komentar tersebut kedalam sentimen negatif, netral, dan positif.

**7.2 Saran**

Pada sub bab ini berisikan saran dan masukan dari hasil verifikasi dan validasi yang telah dilakukan untuk perkembangan sistem dikemudian hari. Beberapa saran yang didapatkan diantaranya:

• Akurasi yang didapatkan terbilang masih kurang dari 80% dikarenakan dataset yang terlalu banyak *noise* karena berasal dari sosial media*.* Maka dari itu dapat dilakukan penambahan kamus *slangword* dan juga *convert* emoji.

• Penambahan kategori komentar seperti komentar tersebut termasuk kedalam kategori fasilitas, pengajar, atau lain sebagainya supaya memudahkan untuk pengkategorian masalah.

• Penambahan fitur melihat komentar berdasarkan sekolah.