



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202216963, 11 Maret 2022

Pencipta

Nama : **Leonard Onnanda Soli, Teguh Sutanto, Anjik Sukmaaji**
Alamat : **Jl. Dukuh Kupang XV, KODYA SURABAYA, KOTA SURABAYA, JAWA TIMUR, 60225**
Kewarganegaraan : **Indonesia**

Pemegang Hak Cipta

Nama : **Universitas Dinamika**
Alamat : **Jl. Raya Kedung Baruk No.98, Kedung Baruk, Kec. Rungkut, Kota SBY, Jawa Timur 60298, KOTA SURABAYA, JAWA TIMUR, 60298**
Kewarganegaraan : **Indonesia**
Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **Aplikasi Lalulintas Komoditas Pertanian Berbasis WEB**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : **15 Februari 2022, di Surabaya**
Jangka waktu perlindungan : **Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.**
Nomor pencatatan : **000332328**

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri


Anggoro Dasananto
NIP.196412081991031002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

Buku Panduan

Aplikasi Lalulintas Komoditas
Pertanian Berbasis WEB

2022



UNIVERSITAS
Dinamika

Teguh Sutanto
Anjik Sukmaaji
Leonard Onnanda Soli

DAFTAR ISI

	Halaman
DAFTAR ISI	1
LATAR BELAKANG	2
TOOLS	3
REFERENSI	4
PANDUAN APLIKASI.....	5
SOURCE CODE	21
• Bagian <i>Login</i> dan <i>Logout</i>	21
• Bagian Kontrol Pengguna.....	22
• Bagian Grafik <i>Dahsboard</i>	25
• Bagian Domestik Masuk Hewan	50
• Bagian Domestik Masuk Tumbuhan	52
• Bagian Domestik Keluar Hewan	54
• Bagian Domestik Keluar Tumbuhan	56
• Bagian <i>Import</i> Hewan	58
• Bagian <i>Import</i> Tumbuhan	60
• Bagian <i>Export</i> Hewan	62
• Bagian <i>Export</i> Tumbuhan	64
• Bagian <i>Import</i> Data Komoditas	66
• Fuction <i>Import</i> Hewan	68
• Fuction <i>Import</i> Tumbuhan	70
• Bagian Rekapitulasi	72
• Bagian Cetak Data.....	87

LATAR BELAKANG

- **Masalah**

Dalam kegiatan pengawasan lalu lintas komoditas pertanian dalam Balai Karantina Pertanian Kelas II Ternate memiliki beberapa kendala yang memperlambat proses perekapan dan pengontrollan data lalu lintas komoditas pertanian. Dimana untuk melakukan rekapitulasi data lalu lintas komoditas pertanian, Balai Karantina Pertanian Kelas II Ternate membutuhkan waktu kurang lebih 1-2 minggu untuk menyelesaikannya. Selain itu, belum adanya sistem aplikasi yang menampilkan grafik perkembangan lalu lintas pertanian membuat Balai Karantina Pertanian Kelas II Ternate sulit memantau tingkat laju perkembangan lalu lintas komoditas pertanian yang terjadi di wilayah Maluku Utara. Akibat adanya beberapa kendala tersebut, sering terjadi *crash* dan duplikasi data pada pergerakan data lalu lintas komoditas pertanian. Selain itu ditemukan pula beberapa permasalahan pada pemenuhan permintaan data rekapitulasi, yang terbilang lambat dari beberapa instansi pemerintahan seperti *Badan Pusat Statistik* (BPS), Kantor Layanan Pajak Maluku Utara, serta *Pusat Data dan Informasi Kemetrian Pertanian* (Pusdatin).

- **Solusi**

Berdasarkan permasalahan di atas maka melalui penelitian ini dilakukanlah perancangan aplikasi yang dimana dengan itu dapat menghasilkan Aplikasi Lalu Lintas Komoditas Pertanian Berbasis *Website* Pada Balai Karantina Pertanian Kelas II Ternate yang dapat memonitoring dan dapat melakukan merekapitulasi data pergerakan lalu lintas komoditas pertanian dalam wilayah Maluku Utara dengan lebih cepat dan akurat.

TOOLS

Dalam perancangannya dibutuhkan beberapa tools yang digunakan untuk membangun aplikasi ini yang diantaranya:



Laravel

Fungsi Laravel adalah untuk memaksimalkan penggunaan PHP dalam perancangan dan pengembangan website. Konsep Laravel menggunakan struktur MVC (*Model, View dan Controller*) yang memisahkan aplikasi logika dari presentasi.



Visual Studio Code

Visual Studio Code merupakan editor kode sumber yang di buat sebagai media pencatat code dalam perancangan aplikasi yang didalamnya dapat dilakukan debugging, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, pemfaktoran ulang kode, dan Git yang disematkan.



PHP

Hypertext Preprocessor merupakan sebuah bahasa pemrograman server side scripting yang bersifat *open source*. Bahasa ini digunakan guna membantu pembangunan aplikasi lalu lintas komoditas pertanian pada Balai Karantina Pertanian Kelas II Ternate



MySQL

MySQL adalah sebuah sistem manajemen database yang berguna untuk mengelola database di dalam website. Bahasa pemrograman ini digunakan untuk mengelolah database dalam aplikasi ini.



Javascript

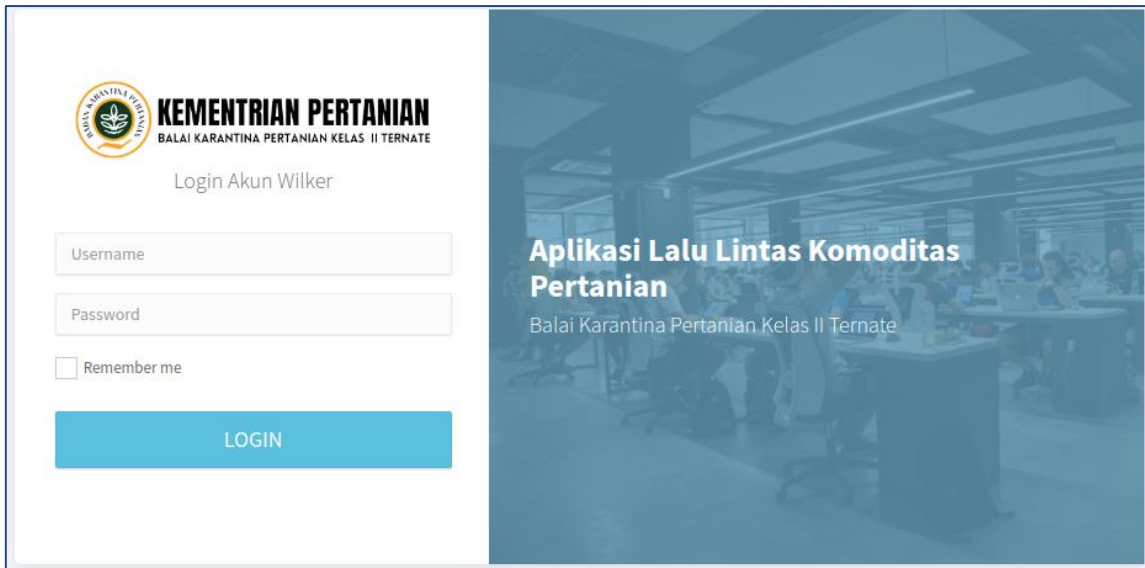
JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan *website*. Javascript digunakan pada pembangunan aplikasi ini dengan tujuan agar *website* yang dihasilkan lebih dinamis dan interaktif.

REFERENSI

- Ade Suryadi, Y. S. (2019). Rancang Bangun Sistem Pengelolaan Arsip Surat Berbasis Web Menggunakan Metode Waterfall. *JURNAL KHATULISTIWA INFORMATIKA*, V OL. VII, NO. 1, 13-21.
- Badan Karantina Pertanian. (2017). *Profil Organisasi*. Retrieved from Badan Karantina Pertanian: <https://karantina.pertanian.go.id/page-8-profil-organisasi.html>
- Balai Karantina Pertanian Kelas II Ternate. (2015). *Rencana Strategis BKP Ternate*. Retrieved from <http://sakip.pertanian.go.id/>: <http://sakip.pertanian.go.id/admin/file/Rencana%20Strategis%20BKP%20Ternate.pdf>
- Balai Karantina Pertanian Kelas II Ternate. (2021). *Profil Balai*. Retrieved from Balai Karantina Pertanian Kelas II Ternate: <https://ternate.karantina.pertanian.go.id/profil-balai-karantina-pertanian-kelas-ii-ternate/>
- Balai Karantina Pertanian Kelas II Ternate. (2021). *Wilayah Kerja*. Retrieved from Balai Karantina Pertanian Kelas II Ternate: <https://ternate.karantina.pertanian.go.id/wilayah-kerja/>
- Faisal, M. (2019). Rancang Bangun Sistem Informasi Housekeeping Inventory. *Jurnal Infotech*, Vol 1 No.1, 28-34.
- Fredisen. (2021). Rancang Bangun Sistem Informasi Produk Dan Pemasaran Pertanian Berbasis Web. *Journal Innovation Informatics*, 1-9.
- Imroatus Sholikhah, M. S. (2017). Aplikasi Pembelian dan Penjualan Barang. *Jurnal Teknik Komputer AMIK BSI*, 16-23.
- Juniardi Dermawan, S. H. (2017). Implementasi Model Waterfall Pada Pengembangan Sistem Informasi Perhitungan Nilai Mata Pelajaran Berbasis Web Pada Sekolah Dasar Al-Azhar Syifa Budi Jatibening. *Paradigma Vol. 19*, No. 2, 142-147.
- Maulani, J. (2020). Penerapan Metode Waterfall Pada Pengembangan Aplikasi. *Technologia Vol 11*, No. 2,, 64-70.
- Muhamad Tabrani, E. P. (2017). Penerapan Metode Waterfall. *Jurnal Inkofar*, 31-38.
- Sania, H. P. (2020). Sistem Informasi Lalu Lintas Ternak (Studi Kasus Dinas Pertanian dan Peternakan Kayon Utara) . *Jurnal Sistem dan Teknologi Informasi* , 58-67.

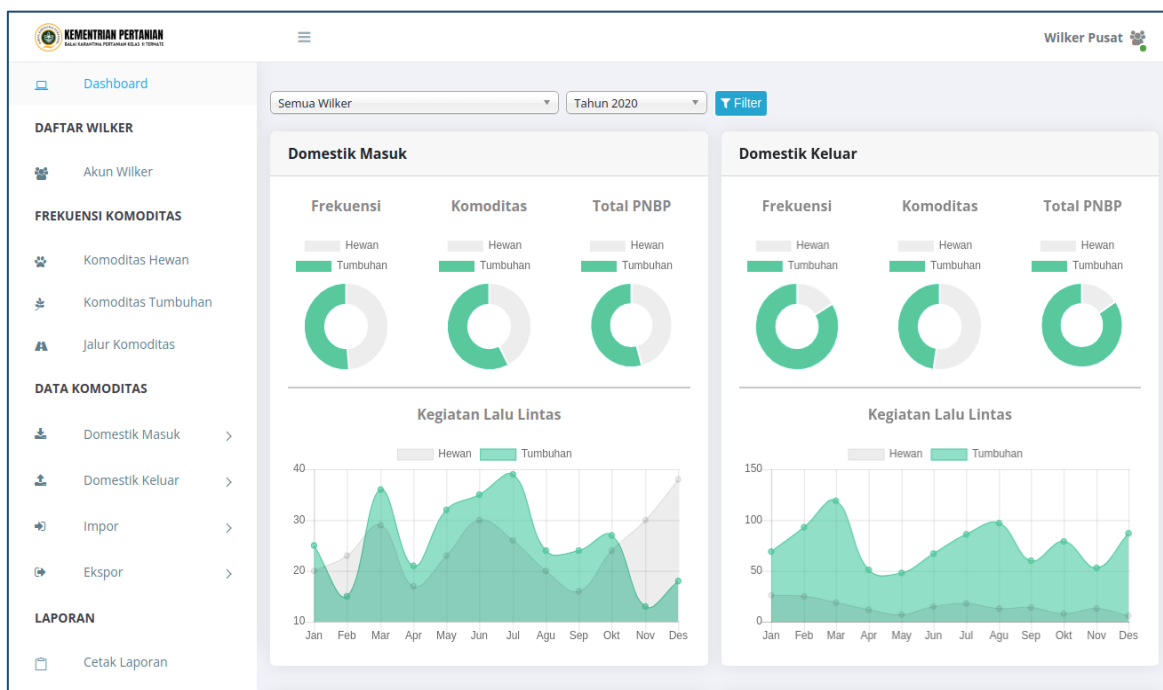
PANDUAN APLIKASI

1. Buka aplikasi Lalulintas Komoditas Pertanian
2. Halaman pertama aplikasi yang akan ditampilkan halaman *login*. Masukkan *username* dan *password* sesuai dengan lokasi *wilker* dan jabatan yakni adminstrator *wilker* atau *supervisor*.



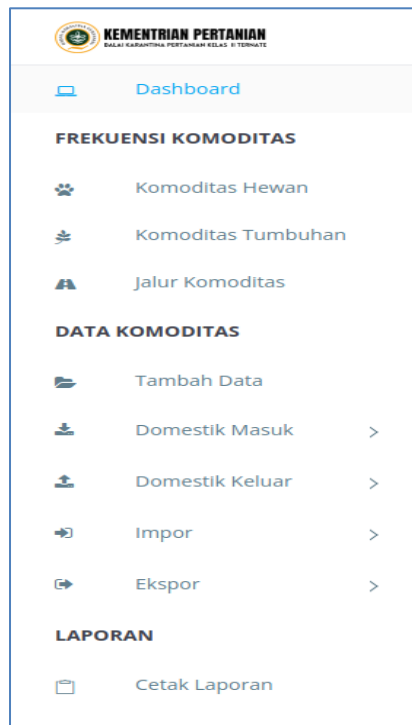
Gambar 1. Halaman Login

3. Setelah melakukan login dan melewati proses validasi, *user* akan diarah pada halaman *dashboard* yang menampilkan gambaran grafik pergerakan kegiatan lalulintas komoditas pertanian seperti yang ada pada gambar 2.













Gambar 2. Halaman Dashboard

4. Pada bagian kiri halaman *dashboard* terdapat beberapa menu dan sub menu seperti pada gambar 3.



Gambar 3. Rincian Menu Dalam Aplikasi

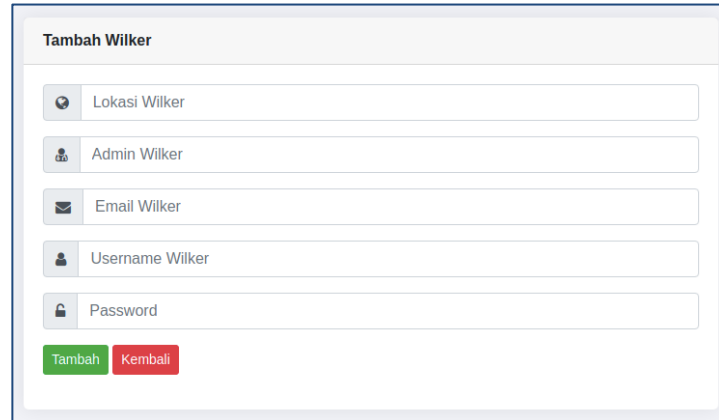
5. Pada menu *dashboard* akan menampilkan gambar pergerakan lalu lintas komoditas pertanian dalam bentuk grafik seperti yang ada pada gambar 2.
6. Pada menu akun *wilker* akan menampilkan daftar akun dari setiap *wilker* yang ada dalam lingkungan Balai Karantina Pertanian Kelas II Ternate seperti pada gambar 4. Namun halaman ini hanya dapat diakses oleh bagian *supervisor*.

Nama Wilker	Penanggung Jawab	Email	Aksi
Wilker Bandara Sultan Babullah	M. Rusli Samma	karantinabandara@gmail.com	 
Wilker Kantor Pos Ternate	Artini	karantinapos@gmail.com	 
Wilker Pelabuhan Laut Ahmad Yani, Ternate	R. Danang Setyo Pambudi	karantinapelabuhan@gmail.com	 
Wilker Pelabuhan Laut Bacan	Apirman	karantinabacan@gmail.com	 
Wilker Pelabuhan Laut Morotai	M. Taufik Adjam	karantinamorotai@gmail.com	 

Gambar 4. Daftar Akun *Wilker*

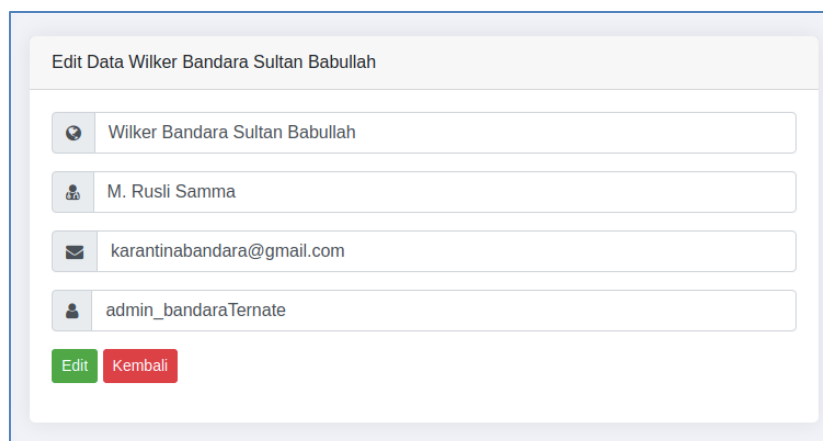
Jika ingin melakukan penambahan data pengurus *wilker*, *user* dapat mengklik tombol Tambah Wilker warna biru yang terletak sebelah kanan atas. Setelah *user* akan diarahkan

pada halaman tambah administrator *wilker* seperti pada gambar 5, selanjutnya *user* wajib mengisi beberapa *form* seperti lokasi *wilker*, nama administrator *wilker*, email *wilker*, *username* dan *password* setelah itu *user* dapat menekan tombol tambah dan jika ingin membatalkan inputan dapat menekan tombol kembali.



Gambar 5. Form Tambah Adminstrator *Wilker*

7. Pada daftar akun *wilker* akan menampilkan daftar *wilker* yang ada di Balai Karantina Pertanian Kelas II Ternate. Pada bagian ini akan menampilkan rincian nama penanggung jawab *wilker*, nama *wilker*, email *wilker*, *username* dan *password*.
8. Pada bagian kanan setiap akun *wilker* terdapat beberapa tombol aksi dimana tombol tersebut terdiri dari tombol edit (pensil), detail (mata) untuk melihat detail keterangan dari akun, dan tombol hapus (tempat sampah) untuk menghapus akun *wilker*
9. Ketika *supervisor* menekan tombol edit (pensil) maka *user* akan diarahkan pada halaman editing seperti pada gambar 8, dimana pada halaman ini *supervisor* dapat merubah beberapa data pada setiap akun *wilker*. Setelah melakukan editing *supervisor* dapat mengklik tombol edit untuk menyimpan hasil editing, namun jika ingin kembali *supervisor* dapat menekan tombol kembali.



Gambar 8. Edit Data *Wilker*

10. Pada menu komoditas hewan seperti pada gambar 9 akan menampilkan daftar hasil rekapitulasi komoditas hewan seperti nama dan jumlah komoditas yang dimana hasil

rekapitulasi terbagi menjadi empat bagian diantaranya rekapitulasi hewan, rekapitulasi bahan asal hewan (BAH), hasil bahan asal hewan (HBAH), dan Benda lainnya. Selain itu pada bagian ini *user* dapat memfilter data rekapitulasi berdasarkan lokasi *wilker*, dan tahun kegiatan yang terletak pada bagian atas tabel.

The screenshot shows a web interface for 'Frekuensi Hewan'. At the top, there are filters for 'Semua Wilker' and 'Tahun 2020', along with a 'Filter' button. Below the filters, there is a 'Show 10 entries' control and a search box. The main table displays the following data:

Nama Hewan	Jumlah
SAPI POTONG (LOKAL)	817 ekor
AYAM BANGKOK	8 ekor
ANJING	8 ekor
AYAM PHILIPIN	7 ekor

Gambar 9. Rekapitulasi Komoditas Hewan

11. Pada menu komoditas tumbuhan seperti pada gambar 10 akan menampilkan daftar hasil rekapitulasi komoditas tumbuhan seperti nama dan jumlah komoditas yang dimana hasil rekapitulasi terbagi menjadi empat bagian diantaranya rekapitulasi bibit tumbuhan, rekapitulasi hasil tanaman hidup (HTH), hasil tanaman mati (HTM), dan Benda lainnya. Selain itu pada bagian ini *user* dapat memfilter data rekapitulasi berdasarkan lokasi *wilker*, dan tahun kegiatan yang terletak pada bagian atas tabel.

The screenshot shows a web interface for 'Frekuensi Bibit Tumbuhan'. At the top, there are filters for 'Semua Wilker' and 'Tahun 2020', along with a 'Filter' button. Below the filters, there is a 'Show 10 entries' control and a search box. The main table displays the following data:

Nama Tumbuhan	Jumlah
BIBIT TANAMAN HIAS	9 buah/butir/batang
BIBIT PALA	9 buah/butir/batang
BIBIT DURIAN	500 buah/butir/batang
LANGSAT	5 buah/butir/batang
BIBIT SUKUN	4 buah/butir/batang
BIBIT PURING IET	30 buah/butir/batang

Gambar 10. Rekapitulasi Komoditas Tumbuhan

12. Pada menu jalur komoditas seperti pada gambar 11 akan menampilkan daftar hasil rekapitulasi jalur komoditas seperti nama daerah dan jumlah kegiatan yang dimana hasil rekapitulasi terbagi menjadi empat jalur diantaranya jalur domestik masuk (DOMAS), domestik keluar (DOKEL), *Import*, dan *Eksport*. Empat lajur tersebut akan dibagi menjadi dua jenis komoditas yakni komoditas hewan dan tumbuhan. Selain itu pada bagian ini *user* dapat memfilter data rekapitulasi berdasarkan lokasi *wilker*, dan tahun kegiatan yang terletak pada bagian atas tabel.

Frekuensi Domestik Masuk

Frekuensi Hewan

Show 7 entries Search:

Kota Asal	Kegiatan
KABUPATEN BONE	8 Kegiatan
KOTA SURABAYA	77 Kegiatan
KABUPATEN MOJOKERTO	6 Kegiatan
KABUPATEN JOMBANG	21 Kegiatan
KOTA BITUNG	20 Kegiatan
KOTA BAUBAU	2 Kegiatan

Gambar 11. Rekapitulasi Jalur Komoditas

13. Pada menu tambah data seperti gambar 12 dapat dilakukan proses *import* data lalu lintas komoditas dimana data yang dimasukkan harus berformat .xls atau .xlsx dimana data tersebut harus berasal dari aplikasi iQfast dan data yang dimasukkan harus sesuai dengan jenis komoditas yakni hewan dan tumbuhan. Setelah memilih data yang sesuai maka *user* dapat menekan tombol tambah yang terletak pada bagian bawah, dan jika data berhasil dimasukkan maka akan muncul konfirmasi data berhasil *diimport*.

Tambah Data Hewan

Wilker Pelabuhan Laut Tobelo

Choose File No file chosen

+ Tambah

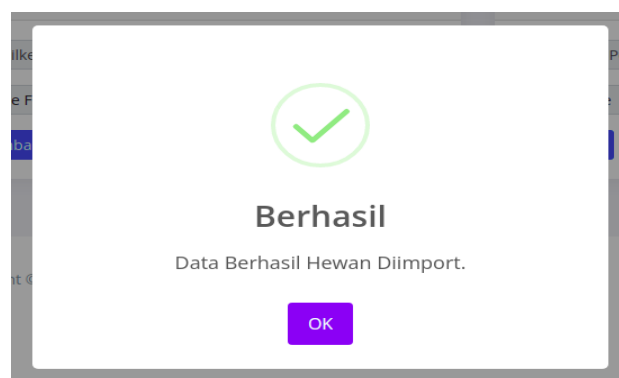
Tambah Data Tumbuhan

Wilker Pelabuhan Laut Tobelo

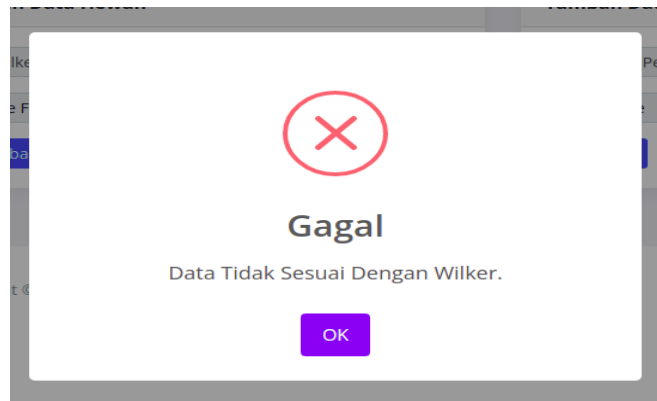
Choose File No file chosen

+ Tambah

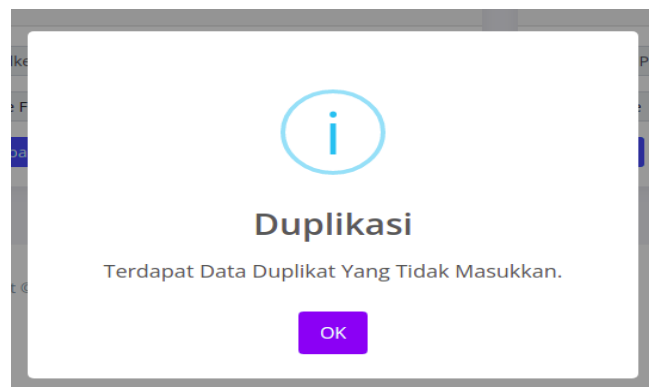
Gambar 12. *Import* Data Lalulintas komoditas



Gambar 13. Konfirmasi Keberhasilan



Gambar 14. Konfirmasi Kegagalan



Gambar 15. Konfirmasi Duplikasi Data

14. Pada menu domestik masuk terdapat dua submenu yakni hewan dan tumbuhan. Pada menu ini menampilkan rincian dari data komoditas hewan dan tumbuhan yang berasal dari jalur domestik masuk.

Daftar DOMAS Hewan

Edit Data Terpilih Hapus Data Terpilih

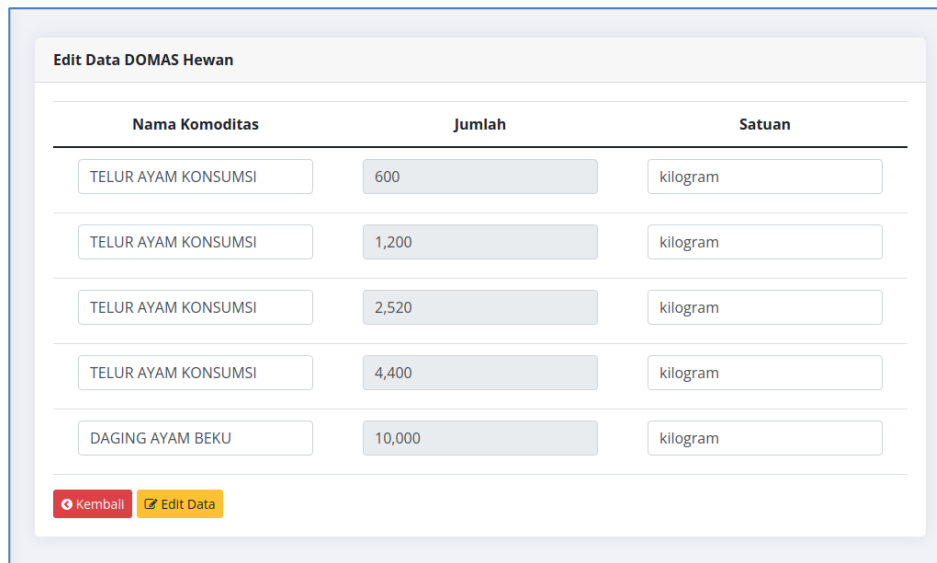
Show 10 entries Search:

<input type="checkbox"/>	Nama Wilker	Tanggal	Kota Asal	Kc
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	10 January 2020	KOTA MANADO	Kc
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	10 January 2020	KOTA MANADO	Kc
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	10 January 2020	KOTA MANADO	Kc
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	10 January 2020	KOTA MANADO	Kc

Gambar 16. Daftar Domestik Masuk Hewan

Pada gambar 16 akan menampilkan informasi domestik masuk komoditas hewan dimana informasi tersebut berasal dari hasil import data lalu lintas komoditas hewan yang dilakukan pada menu tambah data. Pada bagian pojok atas terdapat tombol edit data dan hapus data, untuk mengedit data *user* dapat memilih beberapa data yang ingin diedit dengan mengklik tanda centang pada bagian kiri dari setiap data setelah itu *user* dapat

menekan tombol edit data terpilih. Setelah itu *user* akan diarahkan pada halaman editing seperti pada gambar 17.

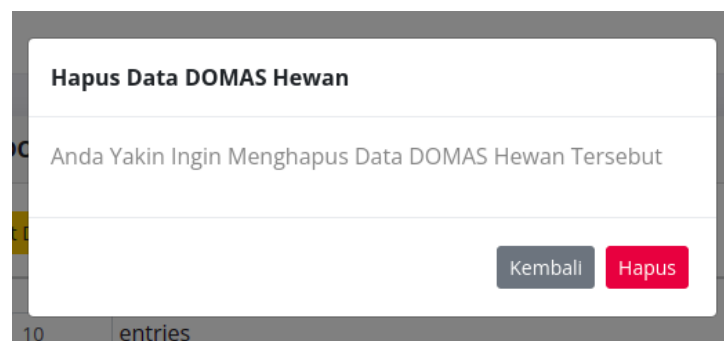


Nama Komoditas	Jumlah	Satuan
TELUR AYAM KONSUMSI	600	kilogram
TELUR AYAM KONSUMSI	1,200	kilogram
TELUR AYAM KONSUMSI	2,520	kilogram
TELUR AYAM KONSUMSI	4,400	kilogram
DAGING AYAM BEKU	10,000	kilogram

Kembali Edit Data

Gambar 17. Edit Domestik Masuk Hewan

Pada halaman ini *user* dapat mengedit lebih dari satu data domestik masuk dari komoditas hewan. Bagian yang dapat diedit *user* hanya pada bagian nama komoditas dan satuan komoditas. Setelah selesai melakukan perubahan *user* dapat menekan tombol edit data dan *user* akan diarahkan kembali ke halaman data domestik masuk komoditas hewan dengan konfirmasi keberhasilan proses editing.



Hapus Data DOMAS Hewan

Anda Yakin Ingin Menghapus Data DOMAS Hewan Tersebut

Kembali Hapus

10 entries

Gambar 18. Konfirmasi Hapus Domestik Masuk Hewan

Seperti pada kegiatan editing, setiap *user* pun dapat melakukan penghapusan beberapa data dengan cara mencentang beberapa data dan menekan tombol hapus data setelah itu akan muncul menu konfirmasi dan jika ingin benar-benar menghapus data maka *user* dapat menekan tombol hapus dan data yang dipilih akan terhapus.

Daftar DOMAS Tumbuhan

Edit Data Terpilih Hapus Data Terpilih

Show 10 entries Search:

<input type="checkbox"/>	Nama Wilker	Tanggal	Kota Asal	Kota Tujuan
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	02 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	02 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	08 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	08 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	08 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	08 January 2020	KOTA MANADO	KABUPATEN

Gambar 19. Daftar Domestik Masuk Tumbuhan

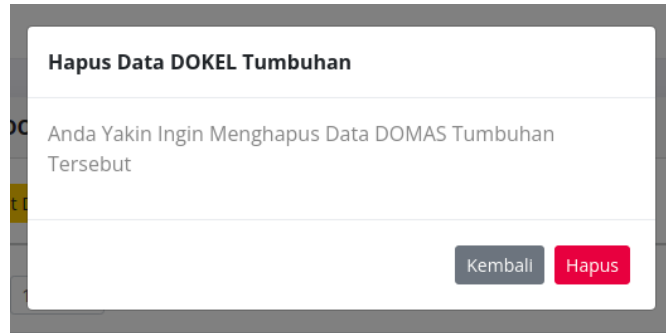
Pada gambar 19 akan menampilkan informasi domestik masuk komoditas tumbuhan dimana informasi tersebut berasal dari hasil import data lalu lintas komoditas tumbuhan yang dilakukan pada menu tambah data. Pada bagian pojok atas terdapat tombol edit data dan hapus data, untuk mengedit data *user* dapat memilih beberapa data yang ingin diedit dengan mengklik tanda centang pada bagian kiri dari setiap data setelah itu *user* dapat menekan tombol edit data terpilih. Setelah itu *user* akan diarahkan pada halaman editing seperti pada gambar 20.

Edit Data DOMAS Tumbuhan

Nama Komoditas	Jumlah	Satuan
SAYURAN KUBIS	1,200	kilogram
SAYURAN KUBIS	1,200	kilogram
SAYURAN KUBIS	1,000	kilogram
BAWANG MERAH	750	kilogram
BAWANG MERAH	750	kilogram

Gambar 20. Edit Domestik Masuk Tumbuhan

Pada halaman ini *user* dapat mengedit lebih dari satu data domestik masuk dari komoditas tumbuhan. Bagian yang dapat diedit *user* hanya pada bagian nama komoditas dan satuan komoditas. Setelah selesai melakukan perubahan *user* dapat menekan tombol edit data dan *user* akan diarahkan kembali ke halaman data domestik masuk komoditas tumbuhan dengan konfirmasi keberhasilan proses editing.



Gambar 21. Konfirmasi Hapus Domestik Masuk Tumbuhan

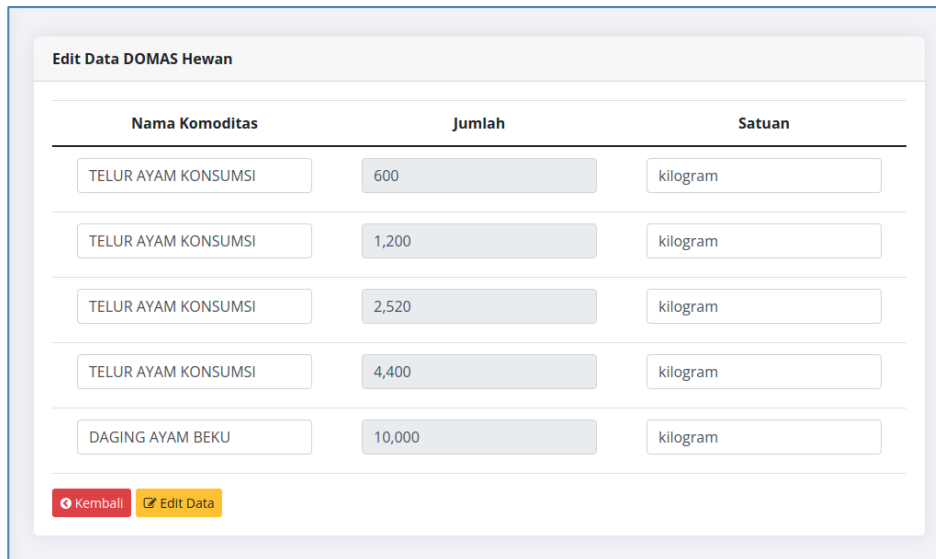
Seperti pada kegiatan editing, setiap *user* pun dapat melakukan penghapusan beberapa data komoditas tumbuhan dengan cara mencentang beberapa data dan menekan tombol hapus data setelah itu akan muncul menu konfirmasi dan jika ingin benar-benar menghapus data maka *user* dapat menekan tombol hapus dan data yang dipilih akan terhapus

15. Pada menu domestik keluar terdapat dua submenu yakni hewan dan tumbuhan. Pada menu ini menampilkan rincian dari data komoditas hewan dan tumbuhan yang berasal dari jalur domestik keluar.

<input type="checkbox"/>	Nama Wilker	Tanggal	Kota Asal	Kc
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	10 January 2020	KOTA MANADO	Kc
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	10 January 2020	KOTA MANADO	Kc
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	10 January 2020	KOTA MANADO	Kc
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	10 January 2020	KOTA MANADO	Kc

Gambar 22. Daftar Domestik Keluar Hewan

Pada gambar 22 akan menampilkan informasi domestik keluar komoditas hewan dimana informasi tersebut berasal dari hasil *import* data lalu lintas komoditas hewan yang dilakukan pada menu tambah data. Pada bagian pojok atas terdapat tombol edit data dan hapus data, untuk mengedit data *user* dapat memilih beberapa data yang ingin diedit dengan mengklik tanda centang pada bagian kiri dari setiap data setelah itu *user* dapat menekan tombol edit data terpilih. Setelah itu *user* akan diarahkan pada halaman editing seperti pada gambar 23.

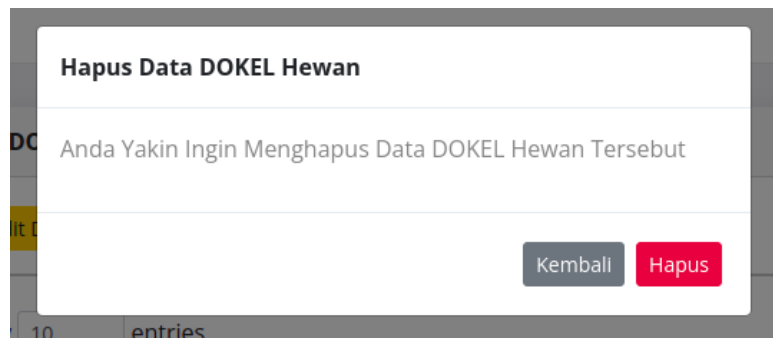


Nama Komoditas	Jumlah	Satuan
TELUR AYAM KONSUMSI	600	kilogram
TELUR AYAM KONSUMSI	1,200	kilogram
TELUR AYAM KONSUMSI	2,520	kilogram
TELUR AYAM KONSUMSI	4,400	kilogram
DAGING AYAM BEKU	10,000	kilogram

Kembali Edit Data

Gambar 23. Edit Domestik Keluar Hewan

Pada halaman ini *user* dapat mengedit lebih dari satu data domestik keluar dari komoditas hewan. Bagian yang dapat diedit *user* hanya pada bagian nama komoditas dan satuan komoditas. Setelah selesai melakukan perubahan *user* dapat menekan tombol edit data dan *user* akan diarahkan kembali ke halaman data domestik keluar komoditas hewan dengan konfirmasi keberhasilan proses editing.



Hapus Data DO KEL Hewan

Anda Yakin Ingin Menghapus Data DO KEL Hewan Tersebut

Kembali Hapus

Gambar 24. Konfirmasi Hapus Domestik Keluar Hewan

Seperti pada kegiatan editing, setiap *user* pun dapat melakukan penghapusan beberapa data dengan cara mencentang beberapa data dan menekan tombol hapus data setelah itu akan muncul menu konfirmasi dan jika ingin benar-benar menghapus data maka *user* dapat menekan tombol hapus dan data yang dipilih akan terhapus.

Daftar DOMAS Tumbuhan

Edit Data Terpilih Hapus Data Terpilih

Show entries Search:

<input type="checkbox"/>	Nama Wilker	Tanggal	Kota Asal	Kota Tujuan
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	02 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	02 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	08 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	08 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	08 January 2020	KOTA MANADO	KABUPATEN
<input type="checkbox"/>	Wilker Pelabuhan Laut Tobelo	08 January 2020	KOTA MANADO	KABUPATEN

Gambar 25. Daftar Domestik Keluar Tumbuhan

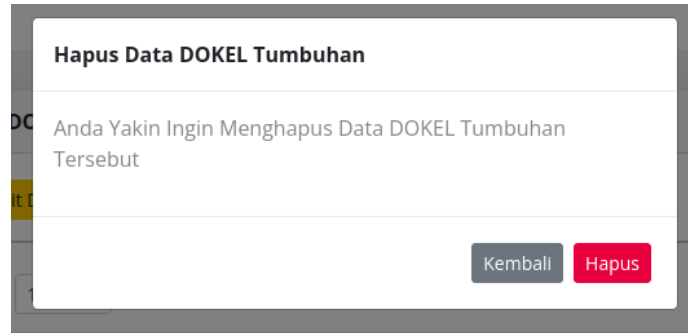
Pada gambar 25 akan menampilkan informasi domestik keluar komoditas tumbuhan dimana informasi tersebut berasal dari hasil import data lalu lintas komoditas tumbuhan yang dilakukan pada menu tambah data. Pada bagian pojok atas terdapat tombol edit data dan hapus data, untuk mengedit data *user* dapat memilih beberapa data yang ingin diedit dengan mengklik tanda centang pada bagian kiri dari setiap data setelah itu *user* dapat menekan tombol edit data terpilih. Setelah itu *user* akan diarahkan pada halaman editing seperti pada gambar 26.

Edit Data DOMAS Tumbuhan

Nama Komoditas	Jumlah	Satuan
<input type="text" value="SAYURAN KUBIS"/>	<input type="text" value="1,200"/>	<input type="text" value="kilogram"/>
<input type="text" value="SAYURAN KUBIS"/>	<input type="text" value="1,200"/>	<input type="text" value="kilogram"/>
<input type="text" value="SAYURAN KUBIS"/>	<input type="text" value="1,000"/>	<input type="text" value="kilogram"/>
<input type="text" value="BAWANG MERAH"/>	<input type="text" value="750"/>	<input type="text" value="kilogram"/>
<input type="text" value="BAWANG MERAH"/>	<input type="text" value="750"/>	<input type="text" value="kilogram"/>

Gambar 26. Edit Domestik Keluar Tumbuhan

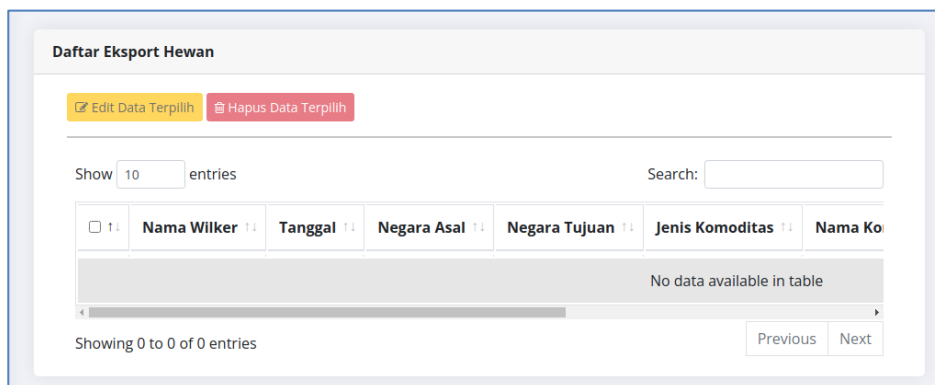
Pada halaman ini *user* dapat mengedit lebih dari satu data domestik keluar dari komoditas tumbuhan. Bagian yang dapat diedit *user* hanya pada bagian nama komoditas dan satuan komoditas. Setelah selesai melakukan perubahan *user* dapat menekan tombol edit data dan *user* akan diarahkan kembali ke halaman data domestik keluar komoditas tumbuhan dengan konfirmasi keberhasilan proses editing.



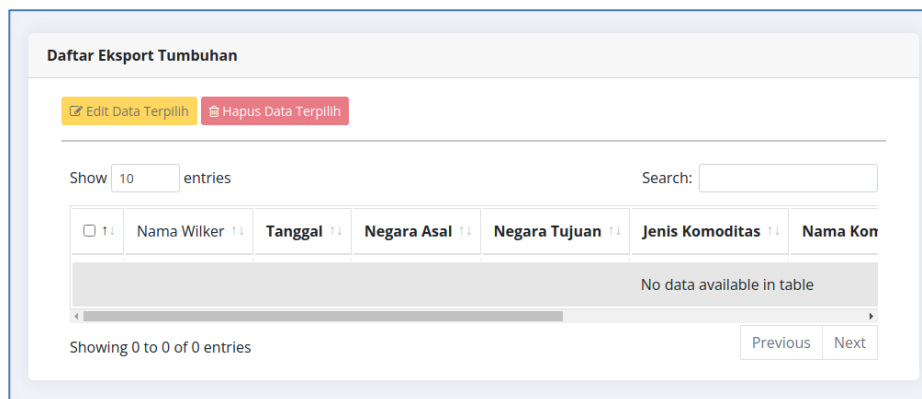
Gambar 27. Konfirmasi Hapus Domestik Keluar Tumbuhan

Seperti pada kegiatan editing, setiap *user* pun dapat melakukan penghapusan beberapa data komoditas tumbuhan dengan cara mencentang beberapa data dan menekan tombol hapus data setelah itu akan muncul menu konformasi dan jika ingin benar-benar menghapus data maka *user* dapat menekan tombol hapus dan data yang dipilih akan terhapus

16. Pada bagian menu ekspor juga terdapat dua submenu yakni hewan dan tumbuhan dan kegiatan dan setiap kegiatan pada dua menu ini dapat dilakukan sama seperti yang ada pada menu domestik keluar.



Gambar 28. Daftar *Ekspor* Hewan



Gambar 29. Daftar *Ekspor* Tumbuhan

17. Pada bagian menu *import* juga terdapat dua submenu yakni hewan dan tumbuhan dan kegiatan dan setiap kegiatan pada dua menu ini dapat dilakukan sama seperti yang ada pada menu domestik masuk.

Gambar 30. Daftar *Import* Hewan

Gambar 31. Daftar *Import* Tumbuhan

18. Pada menu cetak data seperti pada gambar 32, setiap *user* dapat melakukan percetakan data lalu lintas komoditas

Gambar 32. Halaman Cetak Data Rekapitulasi (*Supervisor*)

Gambar 33. Halaman Cetak Data Rekapitulasi (Administrator *Wilker*)

Pada halaman ini setiap *user* diarahkan untuk memilih rentang tanggal, asal wilker (*hanya supervisor*), jalur komoditas yang terdiri dari domestik masuk (DOMAS), domestik keluar (DOKEL), impor dan ekspor, serta pemilihan jenis komoditas yang terdiri dari komoditas hewan dan tumbuhan. Setelah selanjutnya *user* dapat menekan tombol cetak laporan yang terletak pada bagian pojok kiri bawah. Selanjutnya data akan *terdownload* secara otomatis dalam format .pdf.

No	Nama Komoditas	Jumlah	PNBP
1	DAGING AYAM BEKU	189,000 kilogram	Rp 95,000
2	TELUR AYAM KONSUMSI	30,910 kilogram	Rp 90,000
3	DAGING BABI BEKU	1,500 kilogram	Rp 10,000
4	TELUR KONSUMSI	1,225 kilogram	Rp 15,000
5	DAGING ITIK BEKU	1,000 kilogram	Rp 5,000
6	KAMBING KACANG	322 ekor	Rp 935,000
7	SAPI POTONG	1 ekor	Rp 1,000,000

Gambar 34. Hasil Cetak Laporan Komoditas Hewan

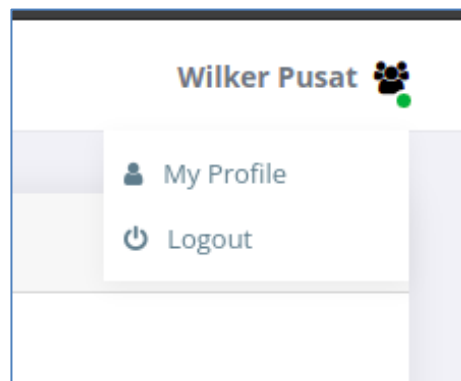
Pada gambar 34 merupakan hasil cetak laporan komoditas hewan yang dimana menampilkan rincian nama komoditas, jumlah komoditas dan total PNBP yang didapat.

No	Nama Komoditas	Jumlah	PNBP
1	KOPRA	1,152,208 kilogram	Rp 2,414,416
2	GETAH DAMAR	20,000 kilogram	Rp 10,000
3	PALA BIJI	10,075 kilogram	Rp 10,470
4	KELAPA BULAT	10,000 kilogram	Rp 55,000
5	CENGKEH	6,175 kilogram	Rp 19,100
6	BUAH PISANG	4,850 kilogram	Rp 34,250
7	SAYURAN KUBIS	1,800 kilogram	Rp 20,800

Gambar 35. Hasil Cetak Laporan Komoditas Tumbuhan

Pada gambar 35 merupakan hasil cetak laporan komoditas tumbuhan yang dimana menampilkan rincian nama komoditas, jumlah komoditas dan total PNBP yang didapat.

19. Pada pojok kanan halaman aplikasi terdapat icon *user* dimana ketika *user* mengklik icon tersebut akan ditampilkan dua menu diantaranya menu my profil dan logout.



Gambar 36. Menu Profil dan Logout

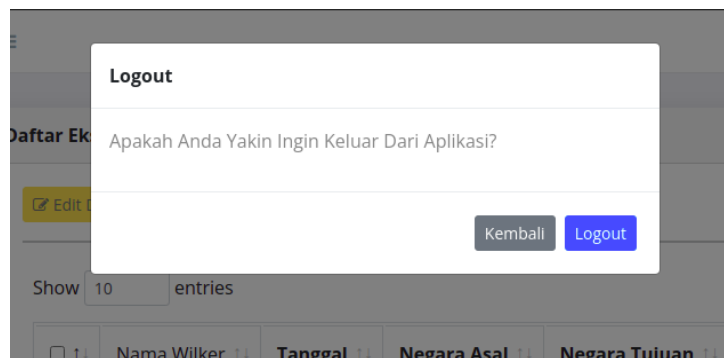
Ketika *user* menekan menu my profil maka setiap *user* akan diarahkan pada halaman profil dari masing-masing *user*, seperti pada gambar 37.

The screenshot shows a user profile page with two main sections. The first section, titled 'Profil Wilker Pusat', contains four input fields: 'Wilker Pusat' (with a user icon), 'Leonard Onnanda Soli' (with a person icon), 'karantinapusat@gmail.com' (with an email icon), and 'supervisor' (with a person icon). Below these fields is a green 'Edit Profil' button. The second section, titled 'Ganti Password', contains three input fields: 'Password Lama', 'Password Baru', and 'Konfirmasi Password', each with a password icon. Below these fields are two buttons: a green 'Ganti Password' button and a red 'Kembali' button.

Gambar 37. Halaman Profil User

Pada halaman ini *user* dapat melakukan editing profil dengan cara merubah beberapa item seperti nama *wilker*, administrator *wilker*, email *wilker* setelah melakukan perubahan *user* dapat menekan tombol edit profil. Jika *user* ingin melakukan penggantian *password* *user* dapat melakukannya pada bagian form ganti *password* dengan mengisi *password* lama setelah itu mengisi *password* baru dan mengisi konfirmasi *password* baru. Setiap *password* minimal memiliki panjang 6 karakter didalamnya. Setelah merubahnya *user* dapat mengklik tombol ganti *password* dan jika berhasil maka akan muncul konfirmasi keberhasilan dan jika gagal maka akan muncul konfirmasi kegagalan.

20. Ketika *user* mengklik tombol logout maka akan muncul konfirmasi keseriusan *user* untuk logout seperti pada gambar 38.



Gambar 38. Konfirmasi Logout

Ketika *user* benar ingin berniat *logout* maka *user* dapat mengklik tombol *logout* yang berwarna biru dan *user* akan diarahkan kembali ke halaman *login*. Selain itu, *user* pun dapat membatalkan proses *logout* dengan mengklik tombol kembali.

SOURCE CODE

- **Bagian Login dan Logout**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    public function index()
    {
        return view("auth.login");
    }

    public function store(Request $request)
    {
        $dataLogin = $request->validate([
            'username' => 'required',
            'password' => 'required'
        ]);

        if (Auth::attempt($dataLogin)) {
            $request->session()->regenerate();
            return redirect()->intended('/dashboard');
        } else {
            toast('Username Tidak Tersedia','error');
            return back();
        }
    }

    public function logout()
    {
        Auth::logout();

        request()->session()->invalidate();

        request()->session()->regenerateToken();

        toast('Berhasil Logout','success');
        return redirect('/');
    }
}
```

• Bagian Kontrol Pengguna

```

<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Auth\Events\Validated;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use RealRashid\SweetAlert\Facades\Alert;

class UsersController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        if (auth()->user()->username != 'supervisor') {
            return abort(403, 'Anda tidak punya akses ke halaman ini');
        }
        $wilker = User::where('username', '!=', 'supervisor')->get();
        return view("Users.users", compact('wilker'));
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        if (auth()->user()->username != 'supervisor') {
            return abort(403, 'Anda tidak punya akses ke halaman ini');
        }
        $dataUser = $request->validate([
            'lokasi' => 'required|max:255',
            'email' => 'required|email:dns|unique:users',
            'admin_wilker' => 'required|min:5',
            'username' => 'required|min:5|max:255',
            'password' => 'required|min:5|max:255'
        ]);

        $dataUser['password'] = bcrypt($dataUser['password']);

        User::create($dataUser);
        toast('Data Berhasil Ditambahkan', 'success');
        // Alert::success('Data Berhasil Ditambahkan');
        return back();
    }

    /**
     * Display the specified resource.
     *

```



```

    * @param int $id
    * @return \Illuminate\Http\Response
    */
    public function show($id)
    {
        $profil = User::where('id_user', $id)->get();
        return view('Users.profil', compact('profil'));
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $wilker = User::where('id_user', $id)->get();
        return view('Users.edtUser', compact('wilker'));
    }

    public function password(Request $request, $id)
    {
        $request->validate([
            'old_password' => 'required',
            'password' => 'required|min:5|confirmed'
        ]);

        $cekPassword = auth()->user()->password;

        $old_password = $request->old_password;

        if (Hash::check($old_password, $cekPassword)) {
            User::where('id_user', $id)->update([
                'password' => bcrypt($request->get('password'))
            ]);

            alert()->success('Berhasil', 'Password Berhasil Diperbaharui.');
```

```

            return back();
        } else {
            alert()->error('Gagal', 'Password Gagal Diperbaharui.');
```

```

            return back();
        }
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $user = User::where('id_user', $id)->get();

        $rules = [
            'lokasi' => 'required|min:3|max:255',
            'admin_wilker' => 'required|min:5',
            'username' => 'required|min:5|max:255'
        ];
    }
}

```

```

];

if ($request->email != $user[0]->email) {
    $rules['email'] = 'required|email:dns|unique:users';
}

$validate = $request->validate($rules);

User::where('id_user', $id)->update($validate);

alert()->success('Berhasil','Data Berhasil Diperbaharui.');
```

```

return back();
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    if (auth()->user()->username != 'supervisor') {
        return abort(403, 'Anda tidak punya akses ke halaman ini');
    }

    DB::table('users')->where('id_user', $id)->delete();
    alert()->success('Berhasil','Data Berhasil Terhapus.');
```

```

return redirect('/user');
}
}

```

• Bagian Grafik *Dahsboard*

S<?php

```
namespace App\Http\Controllers;

use App\Models\KomoditasHewan;
use App\Models\KomoditasTumbuhan;
use App\Models>NamaKomoditasHewan;
use App\Models>NamaKomoditasTumbuhan;
use App\Models\User;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class DashboardController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $tahun = Carbon::now()->isoFormat('Y');
        $filter = request('tahun');
        $wilker = request('nama_wilker');
        $nama_wilker = User::where('username', '!=', 'supervisor')->get();

        if (request('tahun')) {
            if (auth()->user()->username == 'supervisor') {
                if (request('nama_wilker') != 'Semua Wilker') {
                    $f_hewan_domas =
KomoditasHewan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>count('asal_wilker');
                    $f_tumbuhan_domas =
KomoditasTumbuhan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>count('asal_wilker');
                    $nama_hewan_domas =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
                    ->where('jalur_komoditas', 'DOMAS')
                    ->where('asal_wilker', $wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy(DB::raw("nama_komoditas"))
                    ->pluck('nama');
                    $r_hewan_domas = count($nama_hewan_domas);
                    $nama_tumbuhan_domas =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
                    ->where('jalur_komoditas', 'DOMAS')
                    ->where('asal_wilker', $wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy(DB::raw("nama_komoditas"))
                    ->pluck('nama');
                    $r_tumbuhan_domas = count($nama_tumbuhan_domas);
                    $pnbp_hewan_domas =
KomoditasHewan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>sum('tot_pnbp');
```

```

        $pnbp_tumbuhan_domas =
KomoditasTumbuhan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>sum('tot_pnbp');

        // dokel
        $f_hewan_dokel =
KomoditasHewan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>count('asal_wilker');
        $f_tumbuhan_dokel =
KomoditasTumbuhan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>count('asal_wilker');
        $nama_hewan_dokel =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
                    ->where('jalur_komoditas', 'DOKEL')
                    ->whereYear('tgl_kegiatan', $filter)-
>where('asal_wilker', $wilker)
                    ->groupBy(DB::raw("nama_komoditas"))
                    ->pluck('nama');
        $r_hewan_dokel = count($nama_hewan_dokel);
        $nama_tumbuhan_dokel =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
                    ->where('jalur_komoditas', 'DOKEL')
                    ->whereYear('tgl_kegiatan', $filter)-
>where('asal_wilker', $wilker)
                    ->groupBy(DB::raw("nama_komoditas"))
                    ->pluck('nama');
        $r_tumbuhan_dokel = count($nama_tumbuhan_dokel);
        $pnbp_hewan_dokel =
KomoditasHewan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>sum('tot_pnbp');
        $pnbp_tumbuhan_dokel =
KomoditasTumbuhan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>sum('tot_pnbp');

        // impor
        $f_hewan_impор =
KomoditasHewan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>count('asal_wilker');
        $f_tumbuhan_impор =
KomoditasTumbuhan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>count('asal_wilker');
        $nama_hewan_impор =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
                    ->where('jalur_komoditas', 'Impor')
                    ->whereYear('tgl_kegiatan', $filter)-
>where('asal_wilker', $wilker)
                    ->groupBy(DB::raw("nama_komoditas"))
                    ->pluck('nama');
        $r_hewan_impор = count($nama_hewan_impор);
        $nama_tumbuhan_impор =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
                    ->where('jalur_komoditas', 'Impor')
                    ->whereYear('tgl_kegiatan', $filter)-
>where('asal_wilker', $wilker)

```

```

        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_impор = count($nama_tumbuhan_impор);
        $pnbp_hewan_impор =
KomoditasHewan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>sum('tot_pnbp');
        $pnbp_tumbuhan_impор =
KomoditasTumbuhan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>sum('tot_pnbp');

// ekspor
        $f_hewan_ekspor =
KomoditasHewan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>count('asal_wilker');
        $f_tumbuhan_ekspor =
KomoditasTumbuhan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>count('asal_wilker');
        $nama_hewan_ekspor =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $filter)-
>where('asal_wilker', $wilker)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_ekspor = count($nama_hewan_ekspor);
        $nama_tumbuhan_ekspor =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $filter)-
>where('asal_wilker', $wilker)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_ekspor = count($nama_tumbuhan_ekspor);
        $pnbp_hewan_ekspor =
KomoditasHewan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>sum('tot_pnbp');
        $pnbp_tumbuhan_ekspor =
KomoditasTumbuhan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $filter)->where('asal_wilker', $wilker)-
>sum('tot_pnbp');

// Line Chart
        // Domas
        $kegiatan_domas_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_domas_h"))
        ->where('jalur_komoditas',
'DOMAS')
        ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
        -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('jml_domas_h');

        $kegiatan_domas_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_domas_t"))

```

```

                                ->where('jalur_komoditas',
'DOMAS')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_domas_t');

                                $domas_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'DOMAS')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');
$domas_bulan_t = KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as
bulan"))
                                ->where('jalur_komoditas',
'DOMAS')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

                                // Dokel
                                $kegiatan_dokel_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_dokel_h"))
                                ->where('jalur_komoditas',
'DOKEL')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_dokel_h');

                                $kegiatan_dokel_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_dokel_t"))
                                ->where('jalur_komoditas',
'DOKEL')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_dokel_t');

                                $dokel_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'DOKEL')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');
$dokel_bulan_t = KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as
bulan"))
                                ->where('jalur_komoditas',
'DOKEL')

```

```

                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

// Impor
                                $kegiatan_impор_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_impор_h"))
                                ->where('jalur_komoditas',
'Impor')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_impор_h');

                                $kegiatan_impор_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_impор_t"))
                                ->where('jalur_komoditas',
'Impor')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_impор_t');

                                $impор_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'Impor')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

                                $impор_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'Impor')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

// Ekspor
                                $kegiatan_ekspor_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_ekspor_h"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_ekspor_h');

                                $kegiatan_ekspor_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_ekspor_t"))
                                ->where('jalur_komoditas',
'ekspor')

```

```

->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('jml_ekspor_t');

    $ekspor_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas',
'ekspor')
->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

    $ekspor_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas',
'ekspor')
->whereYear('tgl_kegiatan',
$filter)->where('asal_wilker', $wilker)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');
} else {
    $f_hewan_domas =
KomoditasHewan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $filter)->count('asal_wilker');
    $f_tumbuhan_domas =
KomoditasTumbuhan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $filter)->count('asal_wilker');
    $nama_hewan_domas =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
->where('jalur_komoditas', 'DOMAS')
->whereYear('tgl_kegiatan', $filter)
->groupBy(DB::raw("nama_komoditas"))
->pluck('nama');
    $r_hewan_domas = count($nama_hewan_domas);
    $nama_tumbuhan_domas =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
->where('jalur_komoditas', 'DOMAS')
->whereYear('tgl_kegiatan', $filter)
->groupBy(DB::raw("nama_komoditas"))
->pluck('nama');
    $r_tumbuhan_domas = count($nama_tumbuhan_domas);
    $pnbp_hewan_domas =
KomoditasHewan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');
    $pnbp_tumbuhan_domas =
KomoditasTumbuhan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');

    // dokel
    $f_hewan_dokel =
KomoditasHewan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $filter)->count('asal_wilker');

```



```

        $f_tumbuhan_dokel =
KomoditasTumbuhan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $filter)->count('asal_wilker');
        $nama_hewan_dokel =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_dokel = count($nama_hewan_dokel);
        $nama_tumbuhan_dokel =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_dokel = count($nama_tumbuhan_dokel);
        $pnbp_hewan_dokel =
KomoditasHewan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');
        $pnbp_tumbuhan_dokel =
KomoditasTumbuhan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');
// impor
        $f_hewan_impор =
KomoditasHewan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $filter)->count('asal_wilker');
        $f_tumbuhan_impор =
KomoditasTumbuhan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $filter)->count('asal_wilker');
        $nama_hewan_impор =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_impор = count($nama_hewan_impор);
        $nama_tumbuhan_impор =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_impор = count($nama_tumbuhan_impор);
        $pnbp_hewan_impор =
KomoditasHewan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');
        $pnbp_tumbuhan_impор =
KomoditasTumbuhan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');

// ekspor
        $f_hewan_ekspor =
KomoditasHewan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $filter)->count('asal_wilker');
        $f_tumbuhan_ekspor =
KomoditasTumbuhan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $filter)->count('asal_wilker');
        $nama_hewan_ekspor =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Ekspor')

```

```

        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
    $r_hewan_ekspor = count($nama_hewan_ekspor);
    $nama_tumbuhan_ekspor =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
    $r_tumbuhan_ekspor = count($nama_tumbuhan_ekspor);
    $pnbp_hewan_ekspor =
KomoditasHewan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');
    $pnbp_tumbuhan_ekspor =
KomoditasTumbuhan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');

// Line Chart
// Domas
    $kegiatan_domas_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_domas_h"))
        ->where('jalur_komoditas',
'DOMAS')
        ->whereYear('tgl_kegiatan',
$filter)
    -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('jml_domas_h');

    $kegiatan_domas_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_domas_t"))
        ->where('jalur_komoditas',
'DOMAS')
        ->whereYear('tgl_kegiatan',
$filter)
    -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('jml_domas_t');

    $domas_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
        ->where('jalur_komoditas',
'DOMAS')
        ->whereYear('tgl_kegiatan',
$filter)
    -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('bulan');

    $domas_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
        ->where('jalur_komoditas',
'DOMAS')
        ->whereYear('tgl_kegiatan',
$filter)
    -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('bulan');

```

```

// Dokel
$kegiatan_dokel_hewan =
KomoditasHewan::select (DB::raw("COUNT(*) as jml_dokel_h"))
->where ('jalur_komoditas',
'DOKEL')
->whereYear ('tgl_kegiatan',
$filter)
-
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
->pluck ('jml_dokel_h');

$kegiatan_dokel_tumbuhan =
KomoditasTumbuhan::select (DB::raw ("COUNT(*) as jml_dokel_t"))
->where ('jalur_komoditas',
'DOKEL')
->whereYear ('tgl_kegiatan',
$filter)
-
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
->pluck ('jml_dokel_t');

$dokel_bulan_h =
KomoditasHewan::select (DB::raw ("Month (tgl_kegiatan) as bulan"))
->where ('jalur_komoditas',
'DOKEL')
->whereYear ('tgl_kegiatan',
$filter)
-
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
->pluck ('bulan');

$dokel_bulan_t =
KomoditasTumbuhan::select (DB::raw ("Month (tgl_kegiatan) as bulan"))
->where ('jalur_komoditas',
'DOKEL')
->whereYear ('tgl_kegiatan',
$filter)
-
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
->pluck ('bulan');

// Impor
$kegiatan_impор_hewan =
KomoditasHewan::select (DB::raw ("COUNT(*) as jml_impор_h"))
->where ('jalur_komoditas',
'Impor')
->whereYear ('tgl_kegiatan',
$filter)
-
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
->pluck ('jml_impор_h');

$kegiatan_impор_tumbuhan =
KomoditasTumbuhan::select (DB::raw ("COUNT(*) as jml_impор_t"))
->where ('jalur_komoditas',
'Impor')
->whereYear ('tgl_kegiatan',
$filter)
-
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
->pluck ('jml_impор_t');

```

```

        $impor_bulan_h =
KomoditasHewan::select (DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'Impor')
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```

```

        $impor_bulan_t =
KomoditasTumbuhan::select (DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'Impor')
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```

```

        // Ekspor
        $kegiatan_ekspor_hewan =
KomoditasHewan::select (DB::raw("COUNT(*) as jml_ekspor_h"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_ekspor_h');

```

```

        $kegiatan_ekspor_tumbuhan =
KomoditasTumbuhan::select (DB::raw("COUNT(*) as jml_ekspor_t"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_ekspor_t');

```

```

        $ekspor_bulan_h =
KomoditasHewan::select (DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```

```

        $ekspor_bulan_t =
KomoditasTumbuhan::select (DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```

```

// batas
    }
}

else {
    // Donut Chart
    // domas
    $lokasi = auth()->user()->lokasi;
    // dd($user);

    $f_hewan_domas = KomoditasHewan::where('asal_wilker',
    $lokasi)->where('jalur_komoditas', 'DOMAS')->whereYear('tgl_kegiatan',
    $filter)->count('asal_wilker');
    $f_tumbuhan_domas = KomoditasTumbuhan::where('asal_wilker',
    $lokasi)->where('jalur_komoditas', 'DOMAS')->whereYear('tgl_kegiatan',
    $filter)->count('asal_wilker');
    $nama_hewan_domas =
    KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
    ->where('asal_wilker', $lokasi)-
    >where('jalur_komoditas', 'DOMAS')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy(DB::raw("nama_komoditas"))
    ->pluck('nama');
    $r_hewan_domas = count($nama_hewan_domas);
    $nama_tumbuhan_domas =
    KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
    ->where('asal_wilker', $lokasi)-
    >where('jalur_komoditas', 'DOMAS')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy(DB::raw("nama_komoditas"))
    ->pluck('nama');
    $r_tumbuhan_domas = count($nama_tumbuhan_domas);
    $pnbp_hewan_domas = KomoditasHewan::where('asal_wilker',
    $lokasi)->where('jalur_komoditas', 'DOMAS')->whereYear('tgl_kegiatan',
    $filter)->sum('tot_pnbp');
    $pnbp_tumbuhan_domas =
    KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
    'DOMAS')->whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');

    // dokel
    $f_hewan_dokel = KomoditasHewan::where('asal_wilker',
    $lokasi)->where('jalur_komoditas', 'DOKEL')->whereYear('tgl_kegiatan',
    $filter)->count('asal_wilker');
    $f_tumbuhan_dokel = KomoditasTumbuhan::where('asal_wilker',
    $lokasi)->where('jalur_komoditas', 'DOKEL')->whereYear('tgl_kegiatan',
    $filter)->count('asal_wilker');
    $nama_hewan_dokel =
    KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
    ->where('asal_wilker', $lokasi)-
    >where('jalur_komoditas', 'DOKEL')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy(DB::raw("nama_komoditas"))
    ->pluck('nama');
    $r_hewan_dokel = count($nama_hewan_dokel);
    $nama_tumbuhan_dokel =
    KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
    ->where('asal_wilker', $lokasi)-
    >where('jalur_komoditas', 'DOKEL')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy(DB::raw("nama_komoditas"))

```

```

        ->pluck('nama');
        $r_tumbuhan_dokel = count($nama_tumbuhan_dokel);
        $pnbp_hewan_dokel = KomoditasHewan::where('asal_wilker',
$lokasi)->where('jalur_komoditas', 'DOKEL')->whereYear('tgl_kegiatan',
$filter)->sum('tot_pnbp');
        $pnbp_tumbuhan_dokel =
KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
'DOKEL')->whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');

        // impor
        $f_hewan_impор = KomoditasHewan::where('asal_wilker',
$lokasi)->where('jalur_komoditas', 'Impor')->whereYear('tgl_kegiatan',
$filter)->count('asal_wilker');
        $f_tumbuhan_impор = KomoditasTumbuhan::where('asal_wilker',
$lokasi)->where('jalur_komoditas', 'Impor')->whereYear('tgl_kegiatan',
$filter)->count('asal_wilker');
        $nama_hewan_impор =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
>where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_impор = count($nama_hewan_impор);
        $nama_tumbuhan_impор =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
>where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_impор = count($nama_tumbuhan_impор);
        $pnbp_hewan_impор = KomoditasHewan::where('asal_wilker',
$lokasi)->where('jalur_komoditas', 'Impor')->whereYear('tgl_kegiatan',
$filter)->sum('tot_pnbp');
        $pnbp_tumbuhan_impор =
KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
'Impor')->whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');

        // ekspor
        $f_hewan_ekspor = KomoditasHewan::where('asal_wilker',
$lokasi)->where('jalur_komoditas', 'Ekspor')->whereYear('tgl_kegiatan',
$filter)->count('asal_wilker');
        $f_tumbuhan_ekspor =
KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
'Ekspor')->whereYear('tgl_kegiatan', $filter)->count('asal_wilker');
        $nama_hewan_ekspor =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
>where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_ekspor = count($nama_hewan_ekspor);
        $nama_tumbuhan_ekspor =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
>where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');

```

```

        $r_tumbuhan_ekspor = count($nama_tumbuhan_ekspor);
        $pnbp_hewan_ekspor = KomoditasHewan::where('asal_wilker',
$lokasi)->where('jalur_komoditas', 'Ekspor')->whereYear('tgl_kegiatan',
$filter)->sum('tot_pnbp');
        $pnbp_tumbuhan_ekspor =
KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
'Ekspor')->whereYear('tgl_kegiatan', $filter)->sum('tot_pnbp');

        // Line Chart
        // Domas
        $kegiatan_domas_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_domas_h"))
                                ->where('jalur_komoditas', 'DOMAS')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_domas_h');

        $kegiatan_domas_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_domas_t"))
                                ->where('jalur_komoditas', 'DOMAS')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_domas_t');

        $domas_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas', 'DOMAS')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

        $domas_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas', 'DOMAS')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

        // Dokel
        $kegiatan_dokel_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_dokel_h"))
                                ->where('jalur_komoditas', 'DOKEL')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_dokel_h');

```

```

    $kegiatan_dokel_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_dokel_t"))
                                ->where('jalur_komoditas', 'DOKEL')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_dokel_t');

```

```

    $dokel_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas', 'DOKEL')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```

```

    $dokel_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas', 'DOKEL')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```

```

    // Impor
    $kegiatan_impор_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_impор_h"))
                                ->where('jalur_komoditas', 'Impor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_impор_h');

```

```

    $kegiatan_impор_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_impор_t"))
                                ->where('jalur_komoditas', 'Impor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_impор_t');

```

```

    $impор_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas', 'Impor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```



```

        $impor_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas', 'Impor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

        // Ekspor
        $kegiatan_ekspor_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_ekspor_h"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_ekspor_h');

        $kegiatan_ekspor_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_ekspor_t"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_ekspor_t');

        $ekspor_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

        $ekspor_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan',
$filter)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

    }
    } else {
        // if no filter
        if (auth()->user()->username == 'superviser') {

```

```

        $f_hewan_domas = KomoditasHewan::where('jalur_komoditas',
'DOMAS')->whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
        $f_tumbuhan_domas =
KomoditasTumbuhan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
        $nama_hewan_domas =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_domas = count($nama_hewan_domas);
        $nama_tumbuhan_domas =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_domas = count($nama_tumbuhan_domas);
        $pnbp_hewan_domas =
KomoditasHewan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');
        $pnbp_tumbuhan_domas =
KomoditasTumbuhan::where('jalur_komoditas', 'DOMAS')-
>whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');

        // dokel
        $f_hewan_dokel = KomoditasHewan::where('jalur_komoditas',
'DOKEL')->whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
        $f_tumbuhan_dokel =
KomoditasTumbuhan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
        $nama_hewan_dokel =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_dokel = count($nama_hewan_dokel);
        $nama_tumbuhan_dokel =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_dokel = count($nama_tumbuhan_dokel);
        $pnbp_hewan_dokel =
KomoditasHewan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');
        $pnbp_tumbuhan_dokel =
KomoditasTumbuhan::where('jalur_komoditas', 'DOKEL')-
>whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');

        // impor
        $f_hewan_impor = KomoditasHewan::where('jalur_komoditas',
'Impor')->whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
        $f_tumbuhan_impor =
KomoditasTumbuhan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
        $nama_hewan_impor =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))

```

```

        ->where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
    $r_hewan_impор = count($nama_hewan_impор);
    $nama_tumbuhan_impор =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
    $r_tumbuhan_impор = count($nama_tumbuhan_impор);
    $pnbp_hewan_impор =
KomoditasHewan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');
    $pnbp_tumbuhan_impор =
KomoditasTumbuhan::where('jalur_komoditas', 'Impor')-
>whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');

    // ekspor
    $f_hewan_ekspor = KomoditasHewan::where('jalur_komoditas',
'Ekspor')->whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
    $f_tumbuhan_ekspor =
KomoditasTumbuhan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
    $nama_hewan_ekspor =
KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
    $r_hewan_ekspor = count($nama_hewan_ekspor);
    $nama_tumbuhan_ekspor =
KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
    $r_tumbuhan_ekspor = count($nama_tumbuhan_ekspor);
    $pnbp_hewan_ekspor =
KomoditasHewan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');
    $pnbp_tumbuhan_ekspor =
KomoditasTumbuhan::where('jalur_komoditas', 'Ekspor')-
>whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');

    // Line Chart
    // Domas
    $kegiatan_domas_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_domas_h"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)
        -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('jml_domas_h');

    $kegiatan_domas_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_domas_t"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)

```

```

->groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('jml_domas_t');

$domas_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas', 'DOMAS')
->whereYear('tgl_kegiatan', $tahun)
->groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

$domas_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas', 'DOMAS')
->whereYear('tgl_kegiatan', $tahun)
->groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

// Dokel
$kegiatan_dokel_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_dokel_h"))
->where('jalur_komoditas', 'DOKEL')
->whereYear('tgl_kegiatan', $tahun)
->groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('jml_dokel_h');

$kegiatan_dokel_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_dokel_t"))
->where('jalur_komoditas', 'DOKEL')
->whereYear('tgl_kegiatan', $tahun)
->groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('jml_dokel_t');

$dokel_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas', 'DOKEL')
->whereYear('tgl_kegiatan', $tahun)
->groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

$dokel_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas', 'DOKEL')
->whereYear('tgl_kegiatan', $tahun)
->groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

// Impor
$kegiatan_impор_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_impор_h"))
->where('jalur_komoditas', 'Impor')
->whereYear('tgl_kegiatan', $tahun)
->groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('jml_impор_h');

```

```

    $kegiatan_impор_tumbuhan =
KomoditasTumbuhan::select (DB::raw ("COUNT (*) as jml_impор_t"))
    ->where ('jalur_komoditas', 'Impor')
    ->whereYear ('tgl_kegiatan', $tahun)
    -
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
    ->pluck ('jml_impор_t');

    $impор_bulan_h =
KomoditasHewan::select (DB::raw ("Month (tgl_kegiatan) as bulan"))
    ->where ('jalur_komoditas', 'Impor')
    ->whereYear ('tgl_kegiatan', $tahun)
    -
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
    ->pluck ('bulan');

    $impор_bulan_t =
KomoditasTumbuhan::select (DB::raw ("Month (tgl_kegiatan) as bulan"))
    ->where ('jalur_komoditas', 'Impor')
    ->whereYear ('tgl_kegiatan', $tahun)
    -
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
    ->pluck ('bulan');

    // Ekspor
    $kegiatan_ekspor_hewan =
KomoditasHewan::select (DB::raw ("COUNT (*) as jml_ekspor_h"))
    ->where ('jalur_komoditas',
'ekspor')
    ->whereYear ('tgl_kegiatan', $tahun)
    -
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
    ->pluck ('jml_ekspor_h');

    $kegiatan_ekspor_tumbuhan =
KomoditasTumbuhan::select (DB::raw ("COUNT (*) as jml_ekspor_t"))
    ->where ('jalur_komoditas',
'ekspor')
    ->whereYear ('tgl_kegiatan', $tahun)
    -
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
    ->pluck ('jml_ekspor_t');

    $ekspor_bulan_h =
KomoditasHewan::select (DB::raw ("Month (tgl_kegiatan) as bulan"))
    ->where ('jalur_komoditas',
'ekspor')
    ->whereYear ('tgl_kegiatan', $tahun)
    -
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
    ->pluck ('bulan');

    $ekspor_bulan_t =
KomoditasTumbuhan::select (DB::raw ("Month (tgl_kegiatan) as bulan"))
    ->where ('jalur_komoditas',
'ekspor')
    ->whereYear ('tgl_kegiatan', $tahun)
    -
>groupBy (DB::raw ("Month (tgl_kegiatan)"))
    ->pluck ('bulan');

```

```

    }
    else {
        // Donut Chart
        // domas
        $lokasi = auth()->user()->lokasi;
        // dd($user);

        $f_hewan_domas = KomoditasHewan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'DOMAS')->whereYear('tgl_kegiatan',
        $tahun)->count('asal_wilker');
        $f_tumbuhan_domas = KomoditasTumbuhan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'DOMAS')->whereYear('tgl_kegiatan',
        $tahun)->count('asal_wilker');
        $nama_hewan_domas =
        KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
        >where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_domas = count($nama_hewan_domas);
        $nama_tumbuhan_domas =
        KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
        >where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_domas = count($nama_tumbuhan_domas);
        $pnbp_hewan_domas = KomoditasHewan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'DOMAS')->whereYear('tgl_kegiatan',
        $tahun)->sum('tot_pnbp');
        $pnbp_tumbuhan_domas =
        KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
        'DOMAS')->whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');

        // dokel
        $f_hewan_dokel = KomoditasHewan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'DOKEL')->whereYear('tgl_kegiatan',
        $tahun)->count('asal_wilker');
        $f_tumbuhan_dokel = KomoditasTumbuhan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'DOKEL')->whereYear('tgl_kegiatan',
        $tahun)->count('asal_wilker');
        $nama_hewan_dokel =
        KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
        >where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_dokel = count($nama_hewan_dokel);
        $nama_tumbuhan_dokel =
        KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
        >where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_dokel = count($nama_tumbuhan_dokel);
    }
}

```

```

        $pnbp_hewan_dokel = KomoditasHewan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'DOKEL')->whereYear('tgl_kegiatan',
        $tahun)->sum('tot_pnbp');
        $pnbp_tumbuhan_dokel =
        KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
        'DOKEL')->whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');

        // impor
        $f_hewan_impор = KomoditasHewan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'Impor')->whereYear('tgl_kegiatan',
        $tahun)->count('asal_wilker');
        $f_tumbuhan_impор = KomoditasTumbuhan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'Impor')->whereYear('tgl_kegiatan',
        $tahun)->count('asal_wilker');
        $nama_hewan_impор =
        KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
        >where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_impор = count($nama_hewan_impор);
        $nama_tumbuhan_impор =
        KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
        >where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_impор = count($nama_tumbuhan_impор);
        $pnbp_hewan_impор = KomoditasHewan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'Impor')->whereYear('tgl_kegiatan',
        $tahun)->sum('tot_pnbp');
        $pnbp_tumbuhan_impор =
        KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
        'Impor')->whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');

        // ekspor
        $f_hewan_ekspor = KomoditasHewan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'Ekspor')->whereYear('tgl_kegiatan',
        $tahun)->count('asal_wilker');
        $f_tumbuhan_ekspor =
        KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
        'Ekspor')->whereYear('tgl_kegiatan', $tahun)->count('asal_wilker');
        $nama_hewan_ekspor =
        KomoditasHewan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
        >where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_hewan_ekspor = count($nama_hewan_ekspor);
        $nama_tumbuhan_ekspor =
        KomoditasTumbuhan::select(DB::raw("nama_komoditas as nama"))
        ->where('asal_wilker', $lokasi)-
        >where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy(DB::raw("nama_komoditas"))
        ->pluck('nama');
        $r_tumbuhan_ekspor = count($nama_tumbuhan_ekspor);
    
```

```

        $pnbp_hewan_ekspor = KomoditasHewan::where('asal_wilker',
        $lokasi)->where('jalur_komoditas', 'Ekspor')->whereYear('tgl_kegiatan',
        $tahun)->sum('tot_pnbp');
        $pnbp_tumbuhan_ekspor =
        KomoditasTumbuhan::where('asal_wilker', $lokasi)->where('jalur_komoditas',
        'Ekspor')->whereYear('tgl_kegiatan', $tahun)->sum('tot_pnbp');

        // Line Chart
        // Domas
        $kegiatan_domas_hewan =
        KomoditasHewan::select(DB::raw("COUNT(*) as jml_domas_h"))
        ->where('jalur_komoditas', 'DOMAS')
        ->where('asal_wilker', $lokasi)
        ->whereYear('tgl_kegiatan', $tahun)
        -
        >groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('jml_domas_h');

        $kegiatan_domas_tumbuhan =
        KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_domas_t"))
        ->where('jalur_komoditas', 'DOMAS')
        ->where('asal_wilker', $lokasi)
        ->whereYear('tgl_kegiatan', $tahun)
        -
        >groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('jml_domas_t');

        $domas_bulan_h =
        KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
        ->where('jalur_komoditas', 'DOMAS')
        ->where('asal_wilker', $lokasi)
        ->whereYear('tgl_kegiatan', $tahun)
        -
        >groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('bulan');

        $domas_bulan_t =
        KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
        ->where('jalur_komoditas', 'DOMAS')
        ->where('asal_wilker', $lokasi)
        ->whereYear('tgl_kegiatan', $tahun)
        -
        >groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('bulan');

        // Dokel
        $kegiatan_dokel_hewan =
        KomoditasHewan::select(DB::raw("COUNT(*) as jml_dokel_h"))
        ->where('jalur_komoditas', 'DOKEL')
        ->where('asal_wilker', $lokasi)
        ->whereYear('tgl_kegiatan', $tahun)
        -
        >groupBy(DB::raw("Month(tgl_kegiatan)"))
        ->pluck('jml_dokel_h');

        $kegiatan_dokel_tumbuhan =
        KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_dokel_t"))
        ->where('jalur_komoditas', 'DOKEL')
        ->where('asal_wilker', $lokasi)
        ->whereYear('tgl_kegiatan', $tahun)

```



```

-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('jml_dokel_t');

    $dokel_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas', 'DOKEL')
->where('asal_wilker', $lokasi)
->whereYear('tgl_kegiatan', $tahun)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

    $dokel_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas', 'DOKEL')
->where('asal_wilker', $lokasi)
->whereYear('tgl_kegiatan', $tahun)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

    // Impor
    $kegiatan_impор_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_impор_h"))
->where('jalur_komoditas', 'Impor')
->where('asal_wilker', $lokasi)
->whereYear('tgl_kegiatan', $tahun)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('jml_impор_h');

    $kegiatan_impор_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_impор_t"))
->where('jalur_komoditas', 'Impor')
->where('asal_wilker', $lokasi)
->whereYear('tgl_kegiatan', $tahun)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('jml_impор_t');

    $impор_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas', 'Impor')
->where('asal_wilker', $lokasi)
->whereYear('tgl_kegiatan', $tahun)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

    $impор_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
->where('jalur_komoditas', 'Impor')
->where('asal_wilker', $lokasi)
->whereYear('tgl_kegiatan', $tahun)
-
>groupBy(DB::raw("Month(tgl_kegiatan)"))
->pluck('bulan');

    // Ekspor

```

```

        $kegiatan_ekspor_hewan =
KomoditasHewan::select(DB::raw("COUNT(*) as jml_ekspor_h"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan', $tahun)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_ekspor_h');

```

```

        $kegiatan_ekspor_tumbuhan =
KomoditasTumbuhan::select(DB::raw("COUNT(*) as jml_ekspor_t"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan', $tahun)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('jml_ekspor_t');

```

```

        $ekspor_bulan_h =
KomoditasHewan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan', $tahun)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```

```

        $ekspor_bulan_t =
KomoditasTumbuhan::select(DB::raw("Month(tgl_kegiatan) as bulan"))
                                ->where('jalur_komoditas',
'ekspor')
                                ->where('asal_wilker', $lokasi)
                                ->whereYear('tgl_kegiatan', $tahun)
                                -
>groupBy(DB::raw("Month(tgl_kegiatan)"))
                                ->pluck('bulan');

```

```

    }
}

```

```

// Domas
$data_domas_hewan = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
foreach ($domas_bulan_h as $index => $bulan_domas) {
    $data_domas_hewan[$bulan_domas - 1 ] =
$kegiatan_domas_hewan[$index];
}

```

```

    $data_domas_tumbuhan = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
    foreach ($domas_bulan_t as $index => $bulan_domas) {
        $data_domas_tumbuhan[$bulan_domas - 1 ] =
$kegiatan_domas_tumbuhan[$index];
    }

```

```

// Dokel
$data_dokel_hewan = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
foreach ($dokel_bulan_h as $index => $bulan_dokel) {
    $data_dokel_hewan[$bulan_dokel - 1 ] =
$kegiatan_dokel_hewan[$index];
}

```

```

    }

    $data_dokel_tumbuhan = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
    foreach ($dokel_bulan_t as $index => $bulan_dokel) {
        $data_dokel_tumbuhan[$bulan_dokel - 1 ] =
$kegiatan_dokel_tumbuhan[$index];
    }

    // Impor
    $data_impор_hewan = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
    foreach ($impор_bulan_h as $index => $bulan_impор) {
        $data_impор_hewan[$bulan_impор - 1 ] =
$kegiatan_impор_hewan[$index];
    }

    $data_impор_tumbuhan = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
    foreach ($impор_bulan_t as $index => $bulan_impор) {
        $data_impор_tumbuhan[$bulan_impор - 1 ] =
$kegiatan_impор_tumbuhan[$index];
    }

    // Ekspor
    $data_ekspor_hewan = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
    foreach ($ekspor_bulan_h as $index => $bulan_ekspor) {
        $data_ekspor_hewan[$bulan_ekspor - 1 ] =
$kegiatan_ekspor_hewan[$index];
    }

    $data_ekspor_tumbuhan = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
    foreach ($ekspor_bulan_t as $index => $bulan_ekspor) {
        $data_ekspor_tumbuhan[$bulan_ekspor - 1 ] =
$kegiatan_ekspor_tumbuhan[$index];
    }

    return view("dashboard", compact('f_hewan_domas',
'f_tumbuhan_domas', 'r_hewan_domas', 'r_tumbuhan_domas',
'pnbp_hewan_domas', 'pnbp_tumbuhan_domas',
        'f_hewan_dokel', 'f_tumbuhan_dokel', 'r_hewan_dokel',
'r_tumbuhan_dokel', 'pnbp_hewan_dokel', 'pnbp_tumbuhan_dokel',
        'f_hewan_impор', 'f_tumbuhan_impор', 'r_hewan_impор',
'r_tumbuhan_impор', 'pnbp_hewan_impор', 'pnbp_tumbuhan_impор',
        'f_hewan_ekspor', 'f_tumbuhan_ekspor', 'r_hewan_ekspor',
'r_tumbuhan_ekspor', 'pnbp_hewan_ekspor', 'pnbp_tumbuhan_ekspor',
        'data_domas_hewan', 'data_domas_tumbuhan',
'data_dokel_hewan', 'data_dokel_tumbuhan', 'data_impор_hewan',
'data_impор_tumbuhan',
        'data_ekspor_hewan', 'data_ekspor_tumbuhan', 'tahun',
'wilker', 'nama_wilker'));
    }
}

```

• Bagian Domestik Masuk Hewan

```
<?php
```

```
namespace App\Http\Controllers;

use App\Models\KategoriKomoditas;
use App\Models\KomoditasHewan;
use App\Models\User;
use Illuminate\Http\Request;

class DomasHewanController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = auth()->user()->username;
        $lokasi = auth()->user()->lokasi;
        if ($user == 'supervisor') {
            $domashewan = KomoditasHewan::where('jalur_komoditas',
'DOMAS')->get();

            } elseif ($user != 'supervisor') {
                $domashewan = KomoditasHewan::where('jalur_komoditas',
'DOMAS')->where('asal_wilker', $lokasi)->get();
            }
            // dd($domashewan);
            return view("Domas.domasHewan", compact('domashewan'));
        }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Request $request, $id)
    {
        $jmlDomas = explode(",", $id);

        $editDomasHewan = KomoditasHewan::where('jalur_komoditas', 'DOMAS')
            ->whereIn('id_komoditas_hewan', $jmlDomas)
            ->get();

        // dd($editDomasHewan);
        return view('Domas.editDomasHewan', compact('editDomasHewan'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
```

```

    {
        $request->validate([
            'nama_komoditas' => 'required',
            'jumlah' => 'required',
            'satuan' => 'required']);

        // dd($request->all());

        $jml_edit = count($request->id);

        $wilker = auth()->user()->lokasi;

        for ($i=0; $i < $jml_edit; $i++) {
            $form_edit = array(
                'id_komoditas_hewan' => $request->id[$i],
                'nama_komoditas' => $request->nama_komoditas[$i],
                'satuan_komoditas' => $request->satuan[$i]
            );
            KomoditasHewan::where('id_komoditas_hewan', $request->id[$i])
                ->where('asal_wilker', $wilker)-
>update($form_edit);
        }

        alert()->success('Berhasil','Data Berhasil Diedit.');
```

```

        return back();
    }

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function delete(Request $request)
{
    KomoditasHewan::where('jalur_komoditas', 'DOMAS')
        ->whereIn('id_komoditas_hewan', $request->id)
        ->delete();

    alert()->success('Berhasil','Data Berhasil Terhapus.');
```

```

    return response()->json(true);
}
}

```

• Bagian Domestik Masuk Tumbuhan

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\KomoditasTumbuhan;
```

```
use App\Models\User;
```

```
use Illuminate\Http\Request;
```

```
class DomasTumbuhanController extends Controller
```

```
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = auth()->user()->username;
        $lokasi = auth()->user()->lokasi;
        if ($user == 'supervisor') {
            $domastumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'DOMAS')->get();
        } elseif ($user != 'supervisor') {
            $domastumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'DOMAS')->where('asal_wilker', $lokasi)->get();
        }
        // dd($domastumbuhan);
        return view('Domas.domasTumbuhan', compact('domastumbuhan'));
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Request $request, $id)
    {
        $jmlDomasTbh = explode(",", $id);

        $editDomasTumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'DOMAS')
            ->whereIn('id_komoditas_tumbuhan', $jmlDomasTbh)
            ->get();
        // dd($editDomasHewan);
        return view('Domas.editDomasTumbuhan',
compact('editDomasTumbuhan'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
    {
```

```

$request->validate([
    'nama_komoditas' => 'required|min:3',
    'jumlah' => 'required',
    'satuan' => 'required']);

// dd($request->all());

$jmlDomasTbh = count($request->id);

$wilker = auth()->user()->lokasi;

for ($i=0; $i < $jmlDomasTbh; $i++) {
    $form_edit = array(
        'id_komoditas_tumbuhan' => $request->id[$i],
        'nama_komoditas' => $request->nama_komoditas[$i],
        'satuan_komoditas' => $request->satuan[$i]
    );
    KomoditasTumbuhan::where('id_komoditas_tumbuhan', $request-
>id[$i])
        ->where('asal_wilker', $wilker)-
>update($form_edit);
}

alert()->success('Berhasil', 'Data Berhasil Diedit.');
```

```

return back();
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function delete(Request $request)
{
    KomoditasTumbuhan::where('jalur_komoditas', 'DOMAS')
        ->whereIn('id_komoditas_tumbuhan', $request->id)
        ->delete();

    alert()->success('Berhasil', 'Data Berhasil Terhapus.');
```

```

return response()->json(true);
}
}

```

• Bagian Domestik Keluar Hewan

```
S<?php
```

```
namespace App\Http\Controllers;

use App\Models\KomoditasHewan;
use App\Models\User;
use Illuminate\Http\Request;

class DokelHewanController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = auth()->user()->username;
        $lokasi = auth()->user()->lokasi;
        if ($user == 'supervisor') {
            $dokelhewan = KomoditasHewan::where('jalur_komoditas',
'DOKEL')->get();
        } elseif ($user != 'supervisor') {
            $dokelhewan = KomoditasHewan::where('jalur_komoditas',
'DOKEL')->where('asal_wilker', $lokasi)->get();
        }
        return view('Dokel.dokelHewan', compact('dokelhewan'));
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Request $request , $id)
    {
        $jmlDokelH = explode(",",$id);

        $editDokelHewan = KomoditasHewan::where('jalur_komoditas', 'DOKEL')
            ->whereIn('id_komoditas_hewan', $jmlDokelH)
            ->get();
        // dd($editDomasHewan);
        return view('Dokel.editDokelHewan', compact('editDokelHewan'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
    {
        $request->validate([
            'nama_komoditas' => 'required',
            'jumlah' => 'required',

```



```

        'satuan' => 'required']);

    // dd($request->all());

    $jml_edit = count($request->id);

    $wilker = auth()->user()->lokasi;

    for ($i=0; $i < $jml_edit; $i++) {
        $form_edit = array(
            'id_komoditas_hewan' => $request->id[$i],
            'nama_komoditas' => $request->nama_komoditas[$i],
            'satuan_komoditas' => $request->satuan[$i]
        );
        KomoditasHewan::where('id_komoditas_hewan', $request->id[$i])
            ->where('asal_wilker', $wilker)-
>update($form_edit);
    }

    alert()->success('Berhasil','Data Berhasil Diedit.');
```

```

    return back();
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function delete(Request $request)
{
    KomoditasHewan::where('jalur_komoditas', 'DOKEL')
        ->whereIn('id_komoditas_hewan', $request->id)
        ->delete();

    alert()->success('Berhasil','Data Berhasil Terhapus.');
```

```

    return response()->json(true);
}
}

```

• Bagian Domestik Keluar Tumbuhan

```

<?php

namespace App\Http\Controllers;

use App\Models\KomoditasTumbuhan;
use App\Models\User;
use Illuminate\Http\Request;

class DokelTumbuhanController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = auth()->user()->username;
        $lokasi = auth()->user()->lokasi;
        if ($user == 'supervisor') {
            $dokeltumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'DOKEKEL')->get();
        } elseif ($user != 'supervisor') {
            $dokeltumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'DOKEKEL')->where('asal_wilker', $lokasi)->get();
        }
        return view('Dokel.dokelTumbuhan', compact('dokeltumbuhan'));
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Request $request, $id)
    {
        $jmlDokelTbh = explode(",", $id);

        $editDokelTumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'DOKEKEL')
            ->whereIn('id_komoditas_tumbuhan', $jmlDokelTbh)
            ->get();
        // dd($editDomasHewan);
        return view('Dokel.editDokelTumbuhan',
compact('editDokelTumbuhan'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
    {
        $request->validate([

```

```

        'nama_komoditas' => 'required',
        'jumlah' => 'required',
        'satuan' => 'required']);

    // dd($request->all());

    $jmlDomasTbh = count($request->id);

    $wilker = auth()->user()->lokasi;

    for ($i=0; $i < $jmlDomasTbh; $i++) {
        dd($request->id[$i]);
        $form_edit = array(
            'id_komoditas_tumbuhan' => $request->id[$i],
            'nama_komoditas' => $request->nama_komoditas[$i],
            'satuan_komoditas' => $request->satuan[$i]
        );
        KomoditasTumbuhan::where('id_komoditas_tumbuhan', $request-
>id[$i])
            ->where('asal_wilker', $wilker)->update($form_edit);
    }

    alert()->success('Berhasil', 'Data Berhasil Diedit.');
```

```

    return back();
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function delete(Request $request)
{
    KomoditasTumbuhan::where('jalur_komoditas', 'DOKEL')
        ->whereIn('id_komoditas_tumbuhan', $request->id)
        ->delete();

    alert()->success('Berhasil', 'Data Berhasil Terhapus.');
```

```

    return response()->json(true);
}
}

```

• Bagian *Import* Hewan

```
<?php
```

```
namespace App\Http\Controllers;

use App\Models\KomoditasHewan;
use App\Models\User;
use Illuminate\Http\Request;

class ImporHewanController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = auth()->user()->username;
        $lokasi = auth()->user()->lokasi;
        if ($user == 'supervisor') {
            $importhewan = KomoditasHewan::where('jalur_komoditas',
'Impor')->get();
        } elseif ($user != 'supervisor') {
            $importhewan = KomoditasHewan::where('jalur_komoditas',
'Impor')->where('asal_wilker', $lokasi)->get();
        }
        return view('Impor.importhewan', compact('importhewan'));
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Request $request, $id)
    {
        $jmlImporH = explode(",", $id);
        $editImporHewan = KomoditasHewan::where('jalur_komoditas', 'Impor')
            ->whereIn('id_komoditas_hewan', $jmlImporH)
            ->get();

        // dd($editDomasHewan);
        return view('Impor.editImporHewan', compact('editImporHewan'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
    {
        $request->validate([
            'nama_komoditas' => 'required',
            'jumlah' => 'required',
            'satuan' => 'required']);
    }
}
```

```

        // dd($request->all());

        $jml_edit = count($request->id);

        $wilker = auth()->user()->lokasi;

        for ($i=0; $i < $jml_edit; $i++) {
            $form_edit = array(
                'id_komoditas_hewan' => $request->id[$i],
                'nama_komoditas' => $request->nama_komoditas[$i],
                'satuan_komoditas' => $request->satuan[$i]
            );
            KomoditasHewan::where('id_komoditas_hewan', $request->id[$i])
                ->where('asal_wilker', $wilker)-
>update($form_edit);
        }

        alert()->success('Berhasil','Data Berhasil Diedit.');
```

```

        return back();
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function delete(Request $request)
    {
        KomoditasHewan::where('jalur_komoditas', 'Impor')
            ->whereIn('id_komoditas_hewan', $request->id)
            ->delete();

        alert()->success('Berhasil','Data Berhasil Terhapus.');
```

```

        return response()->json(true);
    }
}

```

• Bagian *Import* Tumbuhan

```
<?php
```

```
namespace App\Http\Controllers;

use App\Models\KomoditasTumbuhan;
use App\Models\User;
use Illuminate\Http\Request;

class ImportTumbuhanController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = auth()->user()->username;
        $lokasi = auth()->user()->lokasi;
        if ($user == 'supervisor') {
            $importtumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'Impor')->get();
        } elseif ($user != 'supervisor') {
            $importtumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'Impor')->where('asal_wilker', $lokasi)->get();
        }
        return view('Impor.importTumbuhan', compact('importtumbuhan'));
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Request $request, $id)
    {
        $jmlImporTbh = explode(",", $id);
        $editImporTumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'Impor')
            ->whereIn('id_komoditas_tumbuhan', $jmlImporTbh)
            ->get();
        // dd($editDomasHewan);
        return view('Impor.editImporTumbuhan',
compact('editImporTumbuhan'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
    {
        $request->validate([
            'nama_komoditas' => 'required',

```

```

        'jumlah' => 'required',
        'satuan' => 'required']);

    // dd($request->all());

    $jmlImporTbh = count($request->id);

    $wilker = auth()->user()->lokasi;

    for ($i=0; $i < $jmlImporTbh; $i++) {
        $form_edit = array(
            'id_komoditas_tumbuhan' => $request->id[$i],
            'nama_komoditas' => $request->nama_komoditas[$i],
            'satuan_komoditas' => $request->satuan[$i]
        );
        KomoditasTumbuhan::where('id_komoditas_tumbuhan', $request-
>id[$i])
                                ->where('asal_wilker', $wilker)-
>update($form_edit);
    }

    alert()->success('Berhasil', 'Data Berhasil Diedit.');
```

```

    return back();
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function delete(Request $request)
{
    KomoditasTumbuhan::where('jalur_komoditas', 'Impor')
        ->whereIn('id_komoditas_tumbuhan', $request->id)
        ->delete();

    alert()->success('Berhasil', 'Data Berhasil Terhapus.');
```

```

    return response()->json(true);
}
}

```

• Bagian *Export* Hewan

```
<?php
```

```
namespace App\Http\Controllers;

use App\Models\KomoditasHewan;
use App\Models\User;
use Illuminate\Http\Request;

class EksporHewanController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = auth()->user()->username;
        $lokasi = auth()->user()->lokasi;
        if ($user == 'supervisor') {
            $eksporhewan = KomoditasHewan::where('jalur_komoditas',
'Ekspor')->get();
        } elseif ($user != 'supervisor') {
            $eksporhewan = KomoditasHewan::where('jalur_komoditas',
'Ekspor')->where('asal_wilker', $lokasi)->get();
        }
        return view('Ekspor.eksporHewan', compact('eksporhewan'));
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Request $request, $id)
    {
        $jmlEksporH = explode(",", $id);

        $editEksporHewan = KomoditasHewan::where('jalur_komoditas',
'Ekspor')
            ->whereIn('id_komoditas_hewan', $jmlEksporH)
            ->get();
        // dd($editDomasHewan);
        return view('Ekspor.editEksporHewan', compact('editEksporHewan'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
    {
        $request->validate([
            'nama_komoditas' => 'required',

```



```

        'jumlah' => 'required',
        'satuan' => 'required']);

    // dd($request->all());

    $jml_edit = count($request->id);

    $wilker = auth()->user()->lokasi;

    for ($i=0; $i < $jml_edit; $i++) {
        $form_edit = array(
            'id_komoditas_hewan' => $request->id[$i],
            'nama_komoditas' => $request->nama_komoditas[$i],
            'satuan_komoditas' => $request->satuan[$i]
        );
        KomoditasHewan::where('id_komoditas_hewan', $request->id[$i])
            ->where('asal_wilker', $wilker)-
>update($form_edit);
    }

    alert()->success('Berhasil','Data Berhasil Diedit.');
```

```

    return back();
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function delete(Request $request)
{
    KomoditasHewan::where('jalur_komoditas', 'Ekspor')
        ->whereIn('id_komoditas_hewan', $request->id)
        ->delete();

    alert()->success('Berhasil','Data Berhasil Terhapus.');
```

```

    return response()->json(true);
}
}

```

• Bagian *Export* Tumbuhan

```
<?php
```

```
namespace App\Http\Controllers;

use App\Models\KomoditasTumbuhan;
use App\Models>NamaKomoditasTumbuhan;
use App\Models\NegaraTujuan;
use App\Models\User;
use Illuminate\Http\Request;

class EksporTumbuhanController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $user = auth()->user()->username;
        $lokasi = auth()->user()->lokasi;
        if ($user == 'supervisor') {
            $eksportumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'Ekspor')->get();
        } elseif ($user != 'supervisor') {
            $eksportumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'Ekspor')->where('asal_wilker', $lokasi)->get();
        }
        return view('Ekspor.eksporTumbuhan', compact('eksportumbuhan'));
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Request $request, $id)
    {
        $jmlEksporTbh = explode(",", $id);
        $editEksporTumbuhan = KomoditasTumbuhan::where('jalur_komoditas',
'Ekspor')
            ->whereIn('id_komoditas_tumbuhan', $jmlEksporTbh)
            ->get();
        // dd($editEksporTumbuhan);
        return view('Ekspor.editEksporTumbuhan',
compact('editEksporTumbuhan'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request)
    {

```

```

$request->validate([
    'nama_komoditas' => 'required|min:3',
    'jumlah' => 'required',
    'satuan' => 'required']);

// dd($request->all());

$jmlEksporTbh = count($request->id);

$wilker = auth()->user()->lokasi;

for ($i=0; $i < $jmlEksporTbh; $i++) {
    $form_edit = array(
        'id_komoditas_tumbuhan' => $request->id[$i],
        'nama_komoditas' => $request->nama_komoditas[$i],
        'satuan_komoditas' => $request->satuan[$i]
    );
    KomoditasTumbuhan::where('id_komoditas_tumbuhan', $request-
>id[$i])
        ->where('asal_wilker', $wilker)-
>update($form_edit);
}

alert()->success('Berhasil', 'Data Berhasil Diedit.');
```

```

return back();
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function delete(Request $request)
{
    KomoditasTumbuhan::where('jalur_komoditas', 'Ekspor')
        ->whereIn('id_komoditas_tumbuhan', $request->id)
        ->delete();

    alert()->success('Berhasil', 'Data Berhasil Terhapus.');
```

```

return response()->json(true);
}
}

```

• Bagian *Import* Data Komoditas

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Imports\HewanImport;
use App\Imports\TumbuhanImport;
use App\Models\JalurKomoditas;
use Illuminate\Http\Request;
use Maatwebsite\Excel\Facades\Excel;
```

```
class LaluLintasController extends Controller
```

```
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        if (auth()->user()->username == 'supervisor') {
            return abort(403, 'Anda tidak punya akses ke halaman ini');
        }
        return view('tbhLaluLintas');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        if (auth()->user()->username == 'supervisor') {
            return abort(403, 'Anda tidak punya akses ke halaman ini');
        }
        $request->validate(["file_hewan" => "required|mimes:xls,xlsx"]);

        $file = $request->file('file_hewan');
        $namaFile = $file->getClientOriginalName();

        $importHewan = new HewanImport;
        $importHewan->import($file);
        $file->move(public_path() . '/Import', $namaFile);

        $error = $importHewan->getError();
        // dd($error);
        foreach ($error as $key => $value) {
            if (strpos($value, 'tidak') !== FALSE) {
                alert()->error('Gagal', 'Data Tidak Sesuai Dengan Wilker.');
```

```
                return back();
            }
        }

        // dd(count($importHewan->failures()));
        if (count($importHewan->failures()) > 0) {
            alert()->info('Duplikasi', 'Terdapat Data Duplikat Yang Tidak
Masukkan.');
```

```

        return back();
    }
    alert()->success('Berhasil','Data Berhasil Hewan Diimport.');
```

```

    return back();
}

public function store_t(Request $request)
{
    if (auth()->user()->username == 'supervisor') {
        return abort(403,'A nda tidak punya akses ke halaman ini');
    }

    $request->validate(["file_tumbuhan" => "required|mimes:xls,xlsx"]);

    $fileT = $request->file('file_tumbuhan');
    $namaFile = $fileT->getClientOriginalName();

    $importTumbuhan = new TumbuhanImport;
    $importTumbuhan->import($fileT);
    $fileT->move(public_path() . '/Import', $namaFile);

    $error = $importTumbuhan->getErrorT();

    foreach ($error as $key => $value) {
        if (strpos($value,'tidak') !== FALSE) {
            alert()->error('Gagal','Data Tidak Sesuai Dengan Wilker.');
```

```

            return back();
        }
    }

    // dd(count($importTumbuhan->failures()));
    if (count($importTumbuhan->failures()) > 0) {
        alert()->info('Duplikasi','Terdapat Data Duplikat Yang Tidak
Masukkan.');
```

```

        return back();
    }

    alert()->success('Berhasil','Data Berhasil Tumbuhan Diimport.');
```

```

    return back();
}
}

```

• Fuction *Import Hewan*

```
<?php
```

```
namespace App\Imports;

use App\Models\KomoditasHewan;
use Maatwebsite\Excel\Concerns\Importable;
use Maatwebsite\Excel\Concerns\SkipsErrors;
use Maatwebsite\Excel\Concerns\SkipsOnError;
use Maatwebsite\Excel\Concerns\WithHeadingRow;
use Maatwebsite\Excel\Concerns\SkipsFailures;
use Maatwebsite\Excel\Concerns\SkipsOnFailure;
use Maatwebsite\Excel\Concerns\ToModel;
use Maatwebsite\Excel\Concerns\WithValidation;
use PhpOffice\PhpSpreadsheet\Shared\Date;
use Throwable;

class HewanImport implements
    ToModel,
    WithHeadingRow,
    SkipsOnError,
    WithValidation,
    SkipsOnFailure
{
    use Importable, SkipsErrors, SkipsFailures;
    /**
     * @param array $row
     *
     * @return \Illuminate\Database\Eloquent\Model|null
     */
    public $error = [];
    public function model(array $row)
    {
        // dd($row['nama_wilker']);

        // dd($lokasi);
        $lokasi = auth()->user()->lokasi;
        if (!empty($row['nama_wilker'])) {
            if (strpos($row['nama_wilker'],$lokasi) !== FALSE) {
                return new KomoditasHewan([
                    'kode_kegiatan' => $row['no_aju'],
                    'tgl_kegiatan' =>
Date::excelToDateTimeObject($row['tanggal_permohonan']),
                    'jalur_komoditas' => $row['jenis_permohonan'],
                    'asal_wilker' => $row['nama_wilker'],
                    'kota_asal' => $row['kota_asal'],
                    'asal' => $row['asal'],
                    'kota_tujuan' => $row['kota_tuju'],
                    'tujuan' => $row['tujuan'],
                    'jenis_komoditas' => $row['jenis_mp'],
                    'nama_komoditas' => $row['nama_mp'],
                    'jml_komoditas' => $row['jumlah'],
                    'satuan_komoditas' => $row['satuan'],
                    'harga_komoditas' => $row['harga'],
                    'tot_pnbp' => $row['total_pnbp']
                ]);
            } else {
                $this->error = ['tidak '. $row['nama_wilker']];
            }
        }
    }
}
```

```
    }  
  
    public function rules(): array  
    {  
        return [  
            '*no_aju' => ['unique:komoditas_hewan,kode_kegiatan'],  
        ];  
    }  
  
    public function getError()  
    {  
        return $this->error;  
    }  
}
```

• Fuction *Import* Tumbuhan

```
<?php
```

```
namespace App\Imports;

use App\Models\KomoditasTumbuhan;
use Maatwebsite\Excel\Concerns\Importable;
use Maatwebsite\Excel\Concerns\SkipsErrors;
use Maatwebsite\Excel\Concerns\SkipsFailures;
use Maatwebsite\Excel\Concerns\SkipsOnError;
use Maatwebsite\Excel\Concerns\SkipsOnFailure;
use Maatwebsite\Excel\Concerns\ToModel;
use Maatwebsite\Excel\Concerns\WithHeadingRow;
use Maatwebsite\Excel\Concerns\WithValidation;
use PhpOffice\PhpSpreadsheet\Shared\Date;
use Throwable;

class TumbuhanImport implements
ToModel,
WithHeadingRow,
SkipsOnError,
WithValidation,
SkipsOnFailure
{
    use Importable, SkipsErrors, SkipsFailures;
    /**
     * @param array $row
     *
     * @return \Illuminate\Database\Eloquent\Model|null
     */
    public $error = [];
    public function model(array $row)
    {
        // dd($row);
        // ddd($jmp);
        $lokasi = auth()->user()->lokasi;
        if (!empty($row['nama_wilker'])) {
            if (strpos($row['nama_wilker'],$lokasi) !== FALSE ) {
                return new KomoditasTumbuhan([
                    'kode_kegiatan' => $row['no_aju'],
                    'tgl_kegiatan' =>
Date::excelToDateTimeObject($row['tanggal_permohonan']),
                    'jalur_komoditas' => $row['jenis_permohonan'],
                    'asal_wilker' => $lokasi,
                    'kota_asal' => $row['kota_asal'],
                    'asal' => $row['asal'],
                    'kota_tujuan' => $row['kota_tuju'],
                    'tujuan' => $row['tujuan'],
                    'jenis_komoditas' => $row['golongan'],
                    'nama_komoditas' => $row['nama_komoditas'],
                    'jml_komoditas' => $row['volume_netto'],
                    'satuan_komoditas' => $row['sat_netto'],
                    'harga_komoditas' => $row['harga'],
                    'tot_pnbp' => $row['total_pnbp']
                ]);
            } else {
                $this->error = ['tidak'. $row['nama_wilker']];
            }
        }
    }
}
```



```
public function rules(): array
{
    return [
        '*no_aju' => ['unique:komoditas_tumbuhan,kode_kegiatan'],
    ];
}

public function getErrorT()
{
    return $this->error;
}
}
```

• Bagian Rekapitulasi

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\KomoditasHewan;
```

```
use App\Models\KomoditasTumbuhan;
```

```
use App\Models\User;
```

```
use Carbon\Carbon;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\DB;
```

```
class RincianController extends Controller
```

```
{
```

```
    /**
```

```
     * Display a listing of the resource.
```

```
     *
```

```
     * @return \Illuminate\Http\Response
```

```
     */
```

```
    public function index()
```

```
    {
```

```
        $tahun = Carbon::now()->isoFormat('Y');
```

```
        $user = auth()->user()->username;
```

```
        $wilker = auth()->user()->lokasi;
```

```
        $filter = request('tahun');
```

```
        $nama_wilker = request('nama_wilker');
```

```
        $nama_wilker = User::where('username', '!=', 'supervisor')->get();
```

```
        if (request('tahun')) {
```

```
            if ($user == 'supervisor') {
```

```
                if ($nama_wilker != 'Semua Wilker') {
```

```
                    $rincian_h = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
```

```
                        ->where('jenis_komoditas', 'Hewan')
```

```
                        ->where('asal_wilker', $nama_wilker)
```

```
                        ->whereYear('tgl_kegiatan', $filter)
```

```
                        ->groupBy('nama_komoditas',
```

```
'satuan_komoditas')
```

```
                        ->orderBy('jml_hewan', 'DESC')
```

```
                        ->get();
```

```
                    $rincian_bah = KomoditasHewan::select("nama_komoditas"
, DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
```

```
                        ->where('jenis_komoditas', 'B.A.H.')
```

```
                        ->where('asal_wilker', $nama_wilker)
```

```
                        ->whereYear('tgl_kegiatan', $filter)
```

```
                        ->groupBy('nama_komoditas',
```

```
'satuan_komoditas')
```

```
                        ->orderBy('jml_hewan', 'DESC')
```

```
                        ->get();
```

```
                    $rincian_hbah = KomoditasHewan::select("nama_komoditas"
, DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
```

```
                        ->where('jenis_komoditas', 'H.B.A.H.')
```

```
                        ->where('asal_wilker', $nama_wilker)
```

```
                        ->whereYear('tgl_kegiatan', $filter)
```

```
                        ->groupBy('nama_komoditas',
```

```
'satuan_komoditas')
```

```
                        ->orderBy('jml_hewan', 'DESC')
```

```

->get ();

    $rincian_bl = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
    ->where('jenis_komoditas', 'Benda
Lain')
    ->where('asal_wilker', $nama_wilker)
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy('nama_komoditas',
'satuan_komoditas')
    ->orderBy('jml_hewan', 'DESC')
    ->get ();
} else {
    $rincian_h = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
    ->where('jenis_komoditas', 'Hewan')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy('nama_komoditas',
'satuan_komoditas')
    ->orderBy('jml_hewan', 'DESC')
    ->get ();

    $rincian_bah = KomoditasHewan::select("nama_komoditas"
, DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
    ->where('jenis_komoditas', 'B.A.H.')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy('nama_komoditas',
'satuan_komoditas')
    ->orderBy('jml_hewan', 'DESC')
    ->get ();

    $rincian_hbah = KomoditasHewan::select("nama_komoditas"
, DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
    ->where('jenis_komoditas', 'H.B.A.H.')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy('nama_komoditas',
'satuan_komoditas')
    ->orderBy('jml_hewan', 'DESC')
    ->get ();

    $rincian_bl = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
    ->where('jenis_komoditas', 'Benda
Lain')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy('nama_komoditas',
'satuan_komoditas')
    ->orderBy('jml_hewan', 'DESC')
    ->get ();
}
} else {
    $rincian_h = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
    ->where('jenis_komoditas', 'Hewan')
    ->whereYear('tgl_kegiatan', $filter)
    ->groupBy('nama_komoditas',
'satuan_komoditas')
    ->orderBy('jml_hewan', 'DESC')
    ->get ();
}
}

```

```

        $rincian_bah = KomoditasHewan::select("nama_komoditas"
, DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'B.A.H.')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_hewan', 'DESC')
        ->get();

        $rincian_hbah = KomoditasHewan::select("nama_komoditas"
, DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'H.B.A.H.')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_hewan', 'DESC')
        ->get();

        $rincian_bl = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'Benda
Lain')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_hewan', 'DESC')
        ->get();
    }
    } else {
        if ($user == 'supervisor') {
            $rincian_h = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
            ->where('jenis_komoditas', 'Hewan')
            ->whereYear('tgl_kegiatan', $tahun)
            ->groupBy('nama_komoditas',
'satuan_komoditas')
            ->orderBy('jml_hewan', 'DESC')
            ->get();

            $rincian_bah = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
            ->where('jenis_komoditas', 'B.A.H.')
            ->whereYear('tgl_kegiatan', $tahun)
            ->groupBy('nama_komoditas',
'satuan_komoditas')
            ->orderBy('jml_hewan', 'DESC')
            ->get();

            $rincian_hbah = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
            ->where('jenis_komoditas', 'H.B.A.H.')
            ->whereYear('tgl_kegiatan', $tahun)
            ->groupBy('nama_komoditas',
'satuan_komoditas')
            ->orderBy('jml_hewan', 'DESC')
            ->get();

            $rincian_bl = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
            ->where('jenis_komoditas', 'Benda Lain')
            ->whereYear('tgl_kegiatan', $tahun)

```

```

        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_hewan', 'DESC')
        ->get();
    } else {
        $rincian_h = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'Hewan')
        ->whereYear('tgl_kegiatan', $tahun)
        ->where('asal_wilker', $wilker)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_hewan', 'DESC')
        ->get();

        $rincian_bah = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'B.A.H.')
        ->whereYear('tgl_kegiatan', $tahun)
        ->where('asal_wilker', $wilker)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_hewan', 'DESC')
        ->get();

        $rincian_hbah = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'H.B.A.H.')
        ->whereYear('tgl_kegiatan', $tahun)
        ->where('asal_wilker', $wilker)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_hewan', 'DESC')
        ->get();

        $rincian_bl = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'Benda Lain')
        ->whereYear('tgl_kegiatan', $tahun)
        ->where('asal_wilker', $wilker)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_hewan', 'DESC')
        ->get();
    }
}
// dd($rincian_h);

return view('Rincian.rincianHewan', compact('rincian_h',
'rincian_bah', 'rincian_hbah', 'rincian_bl', 'nama_wilker'));
}

public function index_t()
{
    $tahun = Carbon::now()->isoFormat('Y');
    $user = auth()->user()->username;
    $wilker = auth()->user()->lokasi;
    $filter = request('tahun');
    $nama_wilker = request('nama_wilker');
    $nama_wilker = User::where('username', '!=', 'supervisor')->get();

```

```

        if (request('tahun')) {
            if ($user == 'supervisor') {
                if ($nma_wilker != 'Semua Wilker') {
                    $rincian_bt =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                    ->where('jenis_komoditas', 'Bibit Tanaman')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_tumbuhan', 'DESC')
                    ->get();

                    $rincian_htm =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                    ->where('jenis_komoditas', 'Hasil
Tanaman Mati')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_tumbuhan', 'DESC')
                    ->get();

                    $rincian_hth =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                    ->where('jenis_komoditas', 'Hasil
Tanaman Hidup')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_tumbuhan', 'DESC')
                    ->get();

                    $rincian_bl =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                    ->where('jenis_komoditas', 'Benda
Lain')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_tumbuhan', 'DESC')
                    ->get();

                } else {
                    $rincian_bt =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                    ->where('jenis_komoditas', 'Bibit Tanaman')
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_tumbuhan', 'DESC')
                    ->get();
                }
            }
        }
    
```

```

        $rincian_htm =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                                ->where('jenis_komoditas', 'Hasil
Tanaman Mati')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->get();

```

```

        $rincian_hth =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                                ->where('jenis_komoditas', 'Hasil
Tanaman Hidup')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->get();

```

```

        $rincian_bl =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                                ->where('jenis_komoditas', 'Benda
Lain')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->get();

```

```

    }
    } else {
        $rincian_bt =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                                ->where('jenis_komoditas', 'Bibit Tanaman')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->get();

```

```

        $rincian_htm =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                                ->where('jenis_komoditas', 'Hasil
Tanaman Mati')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->get();

```

```

        $rincian_hth =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
                                ->where('jenis_komoditas', 'Hasil
Tanaman Hidup')
                                ->whereYear('tgl_kegiatan', $filter)

```

```

->groupBy('nama_komoditas',
'satuan_komoditas')
->orderBy('jml_tumbuhan', 'DESC')
->get();

    $rincian_bl =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
->where('jenis_komoditas', 'Benda
Lain')
->whereYear('tgl_kegiatan', $filter)
->groupBy('nama_komoditas',
'satuan_komoditas')
->orderBy('jml_tumbuhan', 'DESC')
->get();
    }
} else {
    if ($user == 'supervisor') {
        $rincian_bt =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
->where('jenis_komoditas', 'Bibit Tanaman')
->whereYear('tgl_kegiatan', $tahun)
->groupBy('nama_komoditas',
'satuan_komoditas')
->orderBy('jml_tumbuhan', 'DESC')
->get();

        $rincian_htm =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
->where('jenis_komoditas', 'Hasil
Tanaman Mati')
->whereYear('tgl_kegiatan', $tahun)
->groupBy('nama_komoditas',
'satuan_komoditas')
->orderBy('jml_tumbuhan', 'DESC')
->get();

        $rincian_hth =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
->where('jenis_komoditas', 'Hasil
Tanaman Hidup')
->whereYear('tgl_kegiatan', $tahun)
->groupBy('nama_komoditas',
'satuan_komoditas')
->orderBy('jml_tumbuhan', 'DESC')
->get();

        $rincian_bl =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
->where('jenis_komoditas', 'Benda Lain')
->whereYear('tgl_kegiatan', $tahun)
->groupBy('nama_komoditas',
'satuan_komoditas')
->orderBy('jml_tumbuhan', 'DESC')
->get();
    } else {

```



```

        $rincian_bt =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'Bibit Tanaman')
        ->where('asal_wilker', $wilker)
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_tumbuhan', 'DESC')
        ->get();

        $rincian_htm =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'Hasil
Tanaman Mati')
        ->where('asal_wilker', $wilker)
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_tumbuhan', 'DESC')
        ->get();

        $rincian_hth =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'Hasil
Tanaman Hidup')
        ->where('asal_wilker', $wilker)
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_tumbuhan', 'DESC')
        ->get();

        $rincian_bl =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"), "satuan_komoditas")
        ->where('jenis_komoditas', 'Benda Lain')
        ->where('asal_wilker', $wilker)
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_tumbuhan', 'DESC')
        ->get();
    }
}
return view('Rincian.rincianTumbuhan', compact('rincian_bt',
'rincian_htm', 'rincian_hth', 'rincian_bl', 'nama_wilker'));
}

public function jalur()
{
    $tahun = Carbon::now()->isoFormat('Y');
    $user = auth()->user()->username;
    $wilker = auth()->user()->lokasi;
    $filter = request('tahun');
    $nama_wilker = request('nama_wilker');
    $nama_wilker = User::where('username', '!=', 'supervisor')->get();

    if (request('tahun')) {

```

```

        if ($user == 'supervisor') {
            if ($nma_wilker != 'Semua Wilker') {
                // domas
                $rincian_domas_h = KomoditasHewan::select("kota_asal",
DB::raw("COUNT(kota_asal) as domas_hewan"))
                    ->where('jalur_komoditas', 'DOMAS')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('kota_asal')
                    ->orderBy('domas_hewan', 'DESC')
                    ->get();

                $rincian_domas_t =
KomoditasTumbuhan::select("kota_asal", DB::raw("COUNT(kota_asal) as
domas_tumbuhan"))
                    ->where('jalur_komoditas', 'DOMAS')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('kota_asal')
                    ->orderBy('domas_tumbuhan', 'DESC')
                    ->get();

                // dokel
                $rincian_dokel_h =
KomoditasHewan::select("kota_tujuan", DB::raw("COUNT(kota_tujuan) as
dokel_hewan"))
                    ->where('jalur_komoditas', 'DOKEL')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('kota_tujuan')
                    ->orderBy('dokel_hewan', 'DESC')
                    ->get();

                $rincian_dokel_t =
KomoditasTumbuhan::select("kota_tujuan", DB::raw("COUNT(kota_tujuan) as
dokel_tumbuhan"))
                    ->where('jalur_komoditas', 'DOKEL')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('kota_tujuan')
                    ->orderBy('dokel_tumbuhan', 'DESC')
                    ->get();

                // import
                $rincian_impор_h = KomoditasHewan::select("asal",
DB::raw("COUNT(asal) as impor_hewan"))
                    ->where('jalur_komoditas', 'Impor')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('asal')
                    ->orderBy('impor_hewan', 'DESC')
                    ->get();

                $rincian_impор_t = KomoditasTumbuhan::select("asal",
DB::raw("COUNT(asal) as impor_tumbuhan"))
                    ->where('jalur_komoditas', 'Impor')
                    ->where('asal_wilker', $nma_wilker)
                    ->whereYear('tgl_kegiatan', $filter)
                    ->groupBy('asal')
                    ->orderBy('impor_tumbuhan', 'DESC')
                    ->get();

```

```

// ekspor
    $rincian_ekspor_h = KomoditasHewan::select("tujuan",
DB::raw("COUNT(tujuan) as ekspor_hewan"))
        ->where('jalur_komoditas', 'Ekspor')
        ->where('asal_wilker', $nama_wilker)
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('tujuan')
        ->orderBy('ekspor_hewan', 'DESC')
        ->get();

    $rincian_ekspor_t = KomoditasTumbuhan::select("tujuan",
DB::raw("COUNT(tujuan) as ekspor_tumbuhan"))
        ->where('jalur_komoditas', 'Ekspor')
        ->where('asal_wilker', $nama_wilker)
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('tujuan')
        ->orderBy('ekspor_tumbuhan', 'DESC')
        ->get();

} else {
    // domas
    $rincian_domas_h = KomoditasHewan::select("kota_asal",
DB::raw("COUNT(kota_asal) as domas_hewan"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('kota_asal')
        ->orderBy('domas_hewan', 'DESC')
        ->get();

    $rincian_domas_t =
KomoditasTumbuhan::select("kota_asal", DB::raw("COUNT(kota_asal) as
domas_tumbuhan"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('kota_asal')
        ->orderBy('domas_tumbuhan', 'DESC')
        ->get();

// dokel
    $rincian_dokel_h =
KomoditasHewan::select("kota_tujuan", DB::raw("COUNT(kota_tujuan) as
dokel_hewan"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('kota_tujuan')
        ->orderBy('dokel_hewan', 'DESC')
        ->get();

    $rincian_dokel_t =
KomoditasTumbuhan::select("kota_tujuan", DB::raw("COUNT(kota_tujuan) as
dokel_tumbuhan"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('kota_tujuan')
        ->orderBy('dokel_tumbuhan', 'DESC')
        ->get();

// import
    $rincian_impор_h = KomoditasHewan::select("asal",
DB::raw("COUNT(asal) as impор_hewan"))

```

```

                                ->where('jalur_komoditas', 'Impor')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('asal')
                                ->orderBy('impor_hewan', 'DESC')
                                ->get();

                                $rincian_impор_t = KomoditasTumbuhan::select("asal",
DB::raw("COUNT(asal) as impor_tumbuhan"))
                                ->where('jalur_komoditas', 'Impor')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('asal')
                                ->orderBy('impor_tumbuhan', 'DESC')
                                ->get();

                                // ekspor
                                $rincian_ekspor_h = KomoditasHewan::select("tujuan",
DB::raw("COUNT(tujuan) as ekspor_hewan"))
                                ->where('jalur_komoditas', 'Ekspor')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('tujuan')
                                ->orderBy('ekspor_hewan', 'DESC')
                                ->get();

                                $rincian_ekspor_t = KomoditasTumbuhan::select("tujuan",
DB::raw("COUNT(tujuan) as ekspor_tumbuhan"))
                                ->where('jalur_komoditas', 'Ekspor')
                                ->whereYear('tgl_kegiatan', $filter)
                                ->groupBy('tujuan')
                                ->orderBy('ekspor_tumbuhan', 'DESC')
                                ->get();

                                }

                                } else {
                                    // domas
                                    $rincian_domas_h = KomoditasHewan::select("kota_asal",
DB::raw("COUNT(kota_asal) as domas_hewan"))
                                    ->where('jalur_komoditas', 'DOMAS')
                                    ->whereYear('tgl_kegiatan', $filter)
                                    ->groupBy('kota_asal')
                                    ->orderBy('domas_hewan', 'DESC')
                                    ->get();

                                    $rincian_domas_t = KomoditasTumbuhan::select("kota_asal",
DB::raw("COUNT(kota_asal) as domas_tumbuhan"))
                                    ->where('jalur_komoditas', 'DOMAS')
                                    ->whereYear('tgl_kegiatan', $filter)
                                    ->groupBy('kota_asal')
                                    ->orderBy('domas_tumbuhan', 'DESC')
                                    ->get();

                                    // dokel
                                    $rincian_dokel_h = KomoditasHewan::select("kota_tujuan",
DB::raw("COUNT(kota_tujuan) as dokel_hewan"))
                                    ->where('jalur_komoditas', 'DOKEL')
                                    ->whereYear('tgl_kegiatan', $filter)
                                    ->groupBy('kota_tujuan')
                                    ->orderBy('dokel_hewan', 'DESC')
                                    ->get();

```

```

        $rincian_dokel_t = KomoditasTumbuhan::select("kota_tujuan",
DB::raw("COUNT(kota_tujuan) as dokel_tumbuhan"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('kota_tujuan')
        ->orderBy('dokel_tumbuhan', 'DESC')
        ->get();

        // import
        $rincian_impор_h = KomoditasHewan::select("asal",
DB::raw("COUNT(asal) as impor_hewan"))
        ->where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('asal')
        ->orderBy('impor_hewan', 'DESC')
        ->get();

        $rincian_impор_t = KomoditasTumbuhan::select("asal",
DB::raw("COUNT(asal) as impor_tumbuhan"))
        ->where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('asal')
        ->orderBy('impor_tumbuhan', 'DESC')
        ->get();

        // ekspor
        $rincian_ekspor_h = KomoditasHewan::select("tujuan",
DB::raw("COUNT(tujuan) as ekspor_hewan"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('tujuan')
        ->orderBy('ekspor_hewan', 'DESC')
        ->get();

        $rincian_ekspor_t = KomoditasTumbuhan::select("tujuan",
DB::raw("COUNT(tujuan) as ekspor_tumbuhan"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $filter)
        ->groupBy('tujuan')
        ->orderBy('ekspor_tumbuhan', 'DESC')
        ->get();

    }
    // batas
} else {
    if ($user == 'supervisor') {
        // domas
        $rincian_domas_h = KomoditasHewan::select("kota_asal",
DB::raw("COUNT(kota_asal) as domas_hewan"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy('kota_asal')
        ->orderBy('domas_hewan', 'DESC')
        ->get();

        $rincian_domas_t = KomoditasTumbuhan::select("kota_asal",
DB::raw("COUNT(kota_asal) as domas_tumbuhan"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy('kota_asal')
        ->orderBy('domas_tumbuhan', 'DESC')
    }
}

```

```

->get ();

        // dokel
        $rincian_dokel_h = KomoditasHewan::select("kota_tujuan",
DB::raw("COUNT(kota_tujuan) as dokel_hewan"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy('kota_tujuan')
        ->orderBy('dokel_hewan', 'DESC')
        ->get ();

        $rincian_dokel_t = KomoditasTumbuhan::select("kota_tujuan",
DB::raw("COUNT(kota_tujuan) as dokel_tumbuhan"))
        ->where('jalur_komoditas', 'DOKEL')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy('kota_tujuan')
        ->orderBy('dokel_tumbuhan', 'DESC')
        ->get ();

        // import
        $rincian_impор_h = KomoditasHewan::select("asal",
DB::raw("COUNT(asal) as impor_hewan"))
        ->where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy('asal')
        ->orderBy('impor_hewan', 'DESC')
        ->get ();

        $rincian_impор_t = KomoditasTumbuhan::select("asal",
DB::raw("COUNT(asal) as impor_tumbuhan"))
        ->where('jalur_komoditas', 'Impor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy('asal')
        ->orderBy('impor_tumbuhan', 'DESC')
        ->get ();

        // ekspor
        $rincian_ekspor_h = KomoditasHewan::select("tujuan",
DB::raw("COUNT(tujuan) as ekspor_hewan"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy('tujuan')
        ->orderBy('ekspor_hewan', 'DESC')
        ->get ();

        $rincian_ekspor_t = KomoditasTumbuhan::select("tujuan",
DB::raw("COUNT(tujuan) as ekspor_tumbuhan"))
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->groupBy('tujuan')
        ->orderBy('ekspor_tumbuhan', 'DESC')
        ->get ();
    } else {
        // domas
        $rincian_domas_h = KomoditasHewan::select("kota_asal",
DB::raw("COUNT(kota_asal) as domas_hewan"))
        ->where('jalur_komoditas', 'DOMAS')
        ->whereYear('tgl_kegiatan', $tahun)
        ->where('asal_wilker', $wilker)
        ->groupBy('kota_asal')
        ->orderBy('domas_hewan', 'DESC')
    }
}

```

```

->get ();

    $rincian_domas_t = KomoditasTumbuhan::select ("kota_asal",
DB::raw ("COUNT(kota_asal) as domas_tumbuhan"))
    ->where ('jalur_komoditas', 'DOMAS')
    ->whereYear ('tgl_kegiatan', $tahun)
    ->where ('asal_wilker', $wilker)
    ->groupBy ('kota_asal')
    ->orderBy ('domas_tumbuhan', 'DESC')
    ->get ();

    // dokel
    $rincian_dokel_h = KomoditasHewan::select ("kota_tujuan",
DB::raw ("COUNT(kota_tujuan) as dokel_hewan"))
    ->where ('jalur_komoditas', 'DOKEL')
    ->whereYear ('tgl_kegiatan', $tahun)
    ->where ('asal_wilker', $wilker)
    ->groupBy ('kota_tujuan')
    ->orderBy ('dokel_hewan', 'DESC')
    ->get ();

    $rincian_dokel_t = KomoditasTumbuhan::select ("kota_tujuan",
DB::raw ("COUNT(kota_tujuan) as dokel_tumbuhan"))
    ->where ('jalur_komoditas', 'DOKEL')
    ->whereYear ('tgl_kegiatan', $tahun)
    ->where ('asal_wilker', $wilker)
    ->groupBy ('kota_tujuan')
    ->orderBy ('dokel_tumbuhan', 'DESC')
    ->get ();

    // impor
    $rincian_impор_h = KomoditasHewan::select ("asal",
DB::raw ("COUNT(asal) as impor_hewan"))
    ->where ('jalur_komoditas', 'Impor')
    ->whereYear ('tgl_kegiatan', $tahun)
    ->where ('asal_wilker', $wilker)
    ->groupBy ('asal')
    ->orderBy ('impor_hewan', 'DESC')
    ->get ();

    $rincian_impор_t = KomoditasTumbuhan::select ("asal",
DB::raw ("COUNT(asal) as impor_tumbuhan"))
    ->where ('jalur_komoditas', 'Impor')
    ->whereYear ('tgl_kegiatan', $tahun)
    ->where ('asal_wilker', $wilker)
    ->groupBy ('asal')
    ->orderBy ('impor_tumbuhan', 'DESC')
    ->get ();

    // ekspor
    $rincian_ekspor_h = KomoditasHewan::select ("tujuan",
DB::raw ("COUNT(tujuan) as ekspor_hewan"))
    ->where ('jalur_komoditas', 'Ekspor')
    ->whereYear ('tgl_kegiatan', $tahun)
    ->where ('asal_wilker', $wilker)
    ->groupBy ('tujuan')
    ->orderBy ('ekspor_hewan', 'DESC')
    ->get ();

    $rincian_ekspor_t = KomoditasTumbuhan::select ("tujuan",
DB::raw ("COUNT(tujuan) as ekspor_tumbuhan"))

```

```
        ->where('jalur_komoditas', 'Ekspor')
        ->whereYear('tgl_kegiatan', $tahun)
        ->where('asal_wilker', $wilker)
        ->groupBy('tujuan')
        ->orderBy('ekspor_tumbuhan', 'DESC')
        ->get();
    }
}

return view('Rincian.rincianJalur', compact('rincian_domas_h',
'rincian_domas_t', 'rincian_dokel_h', 'rincian_dokel_t', 'rincian_impор_h',
'rincian_impор_t', 'rincian_ekspor_h', 'rincian_ekspor_t', 'nama_wilker'));
}
```


• Bagian Cetak Data

```
<?php
```

```
namespace App\Http\Controllers;

use App\Models\KategoriKomoditas;
use App\Models\KomoditasHewan;
use App\Models\KomoditasTumbuhan;
use App\Models\KotaAsal;
use App\Models\User;
use Barryvdh\DomPDF\Facade as PDF;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class LaporanController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $wilker = User::where('lokasi', '!=', 'Wilker Pusat')->get();
        return view('Laporan.Laporan', compact('wilker'));
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        if (auth()->user()->username == 'supervisor') {
            $request->validate([
                'dari' => 'required',
                'sampai' => 'required',
                'asal_wilker' => 'required',
                'jalur_komoditas' => 'required',
                'jenis_komoditas' => 'required'
            ]);
        } else {
            $request->validate([
                'dari' => 'required',
                'sampai' => 'required',
                'jalur_komoditas' => 'required',
                'jenis_komoditas' => 'required'
            ]);
        }

        $wilker = auth()->user()->lokasi;
        $dari = $request->dari;
        $sampai = $request->sampai;
        $asal_wilker = $request->asal_wilker;
        $jenis_komoditas = $request->jenis_komoditas;
        $jalur_komoditas = $request->jalur_komoditas;
    }
}
```

```

    $cek = Carbon::today();
    $hari_ini = $cek->toDateString();

    // dd($hari_ini);

    if ($dari > $sampai) {
        alert()->error('Data Gagal Dicetak','Tanggal Dari Melebihi
Tanggal Sampai.');
```

return back();

```
    }

    if ($dari > $hari_ini) {
        alert()->error('Data Gagal Dicetak.','Tanggal Dari Melebihi
Hari Ini.');
```

return back();

```
    }

    if ( $sampai > $hari_ini) {
        alert()->error('Data Gagal Dicetak.','Tanggal Sampai Melebihi
Hari Ini.');
```

return back();

```
    }

    if (auth()->user()->username == 'supervisor') {
        if ($request->asal_wilker == "Semua") {
            if ($request->jalur_komoditas == "Semua") {
                if ($request->jenis_komoditas == "Semua") {
                    $dataHewan =
KomoditasHewan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_hewan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
                                "satuan_komoditas")
                                ->groupBy('nama_komoditas',
                                'satuan_komoditas')
                                ->orderBy('jml_hewan', 'DESC')
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();

                    $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
                                "satuan_komoditas")
                                ->groupBy('nama_komoditas',
                                'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();

                } elseif ($request->jenis_komoditas == "Hewan") {
                    $dataHewan =
KomoditasHewan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_hewan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
                                "satuan_komoditas")
                                ->groupBy('nama_komoditas',
                                'satuan_komoditas')
                                ->orderBy('jml_hewan', 'DESC')
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();
                }
            }
        }
    }

```

```

        } elseif ($request->jenis_komoditas == "Tumbuhan") {
            $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();
        }
    } elseif ($request->jalur_komoditas != "Semua") {
        if ($request->jenis_komoditas == "Semua") {
            $dataHewan =
KomoditasHewan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_hewan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_hewan', 'DESC')
                                ->where('jalur_komoditas',
$jalur_komoditas)
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();
        }
        $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->where('jalur_komoditas',
$jalur_komoditas)
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();
    } elseif ($request->jenis_komoditas == "Hewan") {
        $dataHewan =
KomoditasHewan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_hewan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_hewan', 'DESC')
                                ->where('jalur_komoditas',
$jalur_komoditas)
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();
    } elseif ($request->jenis_komoditas == "Tumbuhan") {
        $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")

```

```

        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_tumbuhan', 'DESC')
        ->where('jalur_komoditas',
$jalur_komoditas)
        ->where('tgl_kegiatan', '>=', $dari)
        ->where('tgl_kegiatan', '<=', $sampai)-
>get();
    }
} else {
    // where asal wilker != Semua
    if ($request->jalur_komoditas == "Semua") {
        if ($request->jenis_komoditas == "Semua") {
            $dataHewan =
KomoditasHewan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_hewan"),
                    DB::raw("SUM(tot_pnbp) as pnbp"),
'satuan_komoditas")
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_hewan', 'DESC')
                    ->where('asal_wilker', $asal_wilker)
                    ->where('tgl_kegiatan', '>=', $dari)
                    ->where('tgl_kegiatan', '<=', $sampai)-
>get();

            $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                    DB::raw("SUM(tot_pnbp) as pnbp"),
'satuan_komoditas")
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_tumbuhan', 'DESC')
                    ->where('asal_wilker',
$asal_wilker)
                    ->where('tgl_kegiatan', '>=',
$dari)
                    ->where('tgl_kegiatan', '<=',
$sampai)->get();
        } elseif ($request->jenis_komoditas == "Hewan") {
            $dataHewan =
KomoditasHewan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_hewan"),
                    DB::raw("SUM(tot_pnbp) as pnbp"),
'satuan_komoditas")
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_hewan', 'DESC')
                    ->where('asal_wilker', $asal_wilker)
                    ->where('tgl_kegiatan', '>=', $dari)
                    ->where('tgl_kegiatan', '<=', $sampai)-
>get();
        } elseif ($request->jenis_komoditas == "Tumbuhan") {
            $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                    DB::raw("SUM(tot_pnbp) as pnbp"),
'satuan_komoditas")

```

```

        ->groupBy('nama_komoditas',
'satuan_komoditas')
        ->orderBy('jml_tumbuhan', 'DESC')
        ->where('asal_wilker',
$asal_wilker)
        ->where('tgl_kegiatan', '>=',
$dari)
        ->where('tgl_kegiatan', '<=',
$sampai)->get();
    }
    } elseif ($request->jalur_komoditas != "Semua") {
        if ($request->jenis_komoditas == "Semua") {
            $dataHewan =
KomoditasHewan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_hewan"),
                    DB::raw("SUM(tot_pnbp) as pnbp"),
'satuan_komoditas")
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_hewan', 'DESC')
                    ->where('asal_wilker', $asal_wilker)
                    ->where('jalur_komoditas',
$jalur_komoditas)
                    ->where('tgl_kegiatan', '>=', $dari)
                    ->where('tgl_kegiatan', '<=', $sampai)-
>get();

            $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                    DB::raw("SUM(tot_pnbp) as pnbp"),
'satuan_komoditas")
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_tumbuhan', 'DESC')
                    ->where('asal_wilker',
$asal_wilker)
                    ->where('jalur_komoditas',
$jalur_komoditas)
                    ->where('tgl_kegiatan', '>=',
$dari)
                    ->where('tgl_kegiatan', '<=',
$sampai)->get();
        } elseif ($request->jenis_komoditas == "Hewan") {
            $dataHewan =
KomoditasHewan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_hewan"),
                    DB::raw("SUM(tot_pnbp) as pnbp"),
'satuan_komoditas")
                    ->groupBy('nama_komoditas',
'satuan_komoditas')
                    ->orderBy('jml_hewan', 'DESC')
                    ->where('asal_wilker', $asal_wilker)
                    ->where('jalur_komoditas',
$jalur_komoditas)
                    ->where('tgl_kegiatan', '>=', $dari)
                    ->where('tgl_kegiatan', '<=', $sampai)-
>get();
        } elseif ($request->jenis_komoditas == "Tumbuhan") {

```

```

        $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->where('asal_wilker',
$asal_wilker)
                                ->where('jalur_komoditas',
$jalur_komoditas)
                                ->where('tgl_kegiatan', '>=',
$dari)
                                ->where('tgl_kegiatan', '<=',
$sampai)->get();
    }
}
} else {
    // where bukan supervisor
    if ($request->jalur_komoditas == "Semua") {
        if ($request->jenis_komoditas == "Semua") {
            $dataHewan = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_hewan', 'DESC')
                                ->where('asal_wilker', $wilker)
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();

        $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->where('asal_wilker', $wilker)
                                ->where('tgl_kegiatan', '>=',
$dari)
                                ->where('tgl_kegiatan', '<=',
$sampai)->get();
        } elseif ($request->jenis_komoditas == "Hewan") {
            $dataHewan = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_hewan', 'DESC')
                                ->where('asal_wilker', $wilker)
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();
        } elseif ($request->jenis_komoditas == "Tumbuhan") {

```

```

        $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->where('asal_wilker', $wilker)
                                ->where('tgl_kegiatan', '>=',
$dari)
                                ->where('tgl_kegiatan', '<=',
$sampai)->get();
    }
    } elseif ($request->jalur_komoditas != "Semua") {
        if ($request->jenis_komoditas == "Semua") {
            $dataHewan = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_hewan', 'DESC')
                                ->where('asal_wilker', $wilker)
                                ->where('jalur_komoditas',
$jalur_komoditas)
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();

```

```

        $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->where('asal_wilker', $wilker)
                                ->where('jalur_komoditas',
$jalur_komoditas)
                                ->where('tgl_kegiatan', '>=',
$dari)
                                ->where('tgl_kegiatan', '<=',
$sampai)->get();
    } elseif ($request->jenis_komoditas == "Hewan") {
        $dataHewan = KomoditasHewan::select("nama_komoditas" ,
DB::raw("SUM(jml_komoditas) as jml_hewan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_hewan', 'DESC')
                                ->where('asal_wilker', $wilker)
                                ->where('jalur_komoditas',
$jalur_komoditas)
                                ->where('tgl_kegiatan', '>=', $dari)
                                ->where('tgl_kegiatan', '<=', $sampai)-
>get();
    } elseif ($request->jenis_komoditas == "Tumbuhan") {

```

```

        $dataTumbuhan =
KomoditasTumbuhan::select("nama_komoditas" , DB::raw("SUM(jml_komoditas) as
jml_tumbuhan"),
                                DB::raw("SUM(tot_pnbp) as pnbp"),
"satuan_komoditas")
                                ->groupBy('nama_komoditas',
'satuan_komoditas')
                                ->orderBy('jml_tumbuhan', 'DESC')
                                ->where('asal_wilker', $wilker)
                                ->where('jalur_komoditas',
$jalur_komoditas)
                                ->where('tgl_kegiatan', '>=',
$dari)
                                ->where('tgl_kegiatan', '<=',
$sampai)->get();
    }
}

// dd([$dataHewan->SUM('pnbp'), $dataTumbuhan->SUM('pnbp')]);

    if ($jenis komoditas == 'Hewan') {
        $pdf = PDF::loadView('Laporan.HasilHewan', compact('dataHewan',
'dari', 'sampai', 'asal_wilker'))->setPaper('A4', 'potrait');
        return $pdf->stream('Laporan.pdf');
    } elseif ($jenis_komoditas == 'Tumbuhan') {
        $pdf = PDF::loadView('Laporan.HasilTumbuhan',
compact('dataTumbuhan', 'dari', 'sampai', 'asal_wilker'))->setPaper('A4',
'potrait');
        return $pdf->stream('Laporan.pdf');
    }
}
}

```