

LAMPIRAN

Lampiran 1. Surat Izin Observasi

	POLITEKNIK HARAPAN BERSAMA <small>FOR THE BRIGHTER FUTURE</small>	Sarjana Terapan Teknik Informatika
Nomor	: 46.03/TL.PHB/V/2025	
Lampiran	: -	
Hal	: Permohonan Observasi	
Kepada	: Pemilik TOKO ITO	
Yth.	Di Tempat	
Dengan hormat, mahasiswa dengan identitas berikut ini:		
nama	: YESAYA WIDIYA	
NIM	: 21090034	
prodi	: Sarjana Terapan Teknik Informatika	
Bermaksud melakukan penelitian untuk keperluan Skripsi dengan judul "Rancang Bangun Sistem Manajemen Inventori Berbasis Website". Kami memohon Bapak/Ibu memberikan izin kepada mahasiswa yang bersangkutan agar dapat memperoleh data, keterangan, dan bahan yang diperlukan.		
Demikian permohonan ini disampaikan, atas perhatian kami ucapkan terima kasih.		
<p>Tegal, 26 Mei 2025 Kampus Sarjana Terapan Teknik Informatika,  Dyah Apriyanti, S.T., M.Kom NIPY : 09.015.225</p>		
<hr/>		
 Jl. Mataram No. 9 Kota Tegal 52143, Jawa Tengah, Indonesia.	 informatika@politektegal.ac.id	
 (0283)352000	 politektegal.ac.id	

Lampiran 2. Surat Balasan Izin Observasi

Kepada : Politeknik Harapan Bersama Tegal, Kaprodi S.Tr Teknik Informatika
Alamat : Jl. Mataram No.9, Pesurungan Lor, Kec. Margadana, Kota Tegal
Perihal : Surat Balasan Izin Penelitian

Dengan Hormat,

Mengacu pada surat no. 46.03/TL.PHB/V/2025 perihal permohonan izin observasi.

Dengan ini saya sampaikan bahwa izin observasi untuk mahasiswa,

Nama : YESAYA WIDIYA

NIM : 21090034

Di Toko ITO KAMI SETUJUL.

Demikian surat ini kami sampaikan, terimakasih atas kepercayaan yang diberikan kepada kami.

Brebes, 26 Mei 2025

Kepala Toko ITO



Lampiran 3. Surat Kesepakatan Pembimbing I

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : YESAYA WIDIYA
NIM : 21090034
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Slamet Wiyono, S.Pd., M.Eng.
Status : Dosen
NIDN : 0626059001
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata Tk.I


Pada hari ini Rabu, 28 Mei 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan Skripsi minimal 8 kali kepada Pihak Kedua. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Pihak Pertama,


YESAYA WIDIYA

Pihak Kedua,


Slamet Wiyono, S.Pd., M.Eng.

Tegal, 28 Mei 2025

Mengetahui,
Ketua Program Studi Sarjana Terapan
Teknik Informatika


Dyah Apriani, S.T., M.Kom.
NIPY. 09.015.225

Lampiran 4. Lembar Bimbingan Pembimbing I



D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

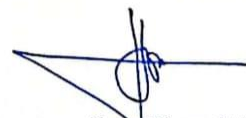
Nama : YESAYA WIDIYA
NIM : 21090034
No. Ponsel : 083808840257
Judul TA : RANCANG BANGUN
SISTEM MANAJEMEN INVENTORI
PADA TOKO ITO BERBASIS WEBSITE
Dosen Pembimbing I : Slamet Wiyono, S.Pd., M.Eng.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	2/5 -2025	Observasi	- Analisis fitur	HP
2.	26/5 -2025	User Interface	- Menyesuaikan fitur & mengidentifikasi kebutuhan mitra	HP
3.	27/5 2025		- meminta Surat Persetujuan fitur	HP
4.	08/7 2025	Project	- Lanjutkan aplikasi - Meminta Surat Pernyataan telah diisi mitra	HP

5.	15/7 2025	Silahkan HKE		HP
6.	31/7 2025	Produk	- pengujian fungsi - usability testing	HP
7.	1/8 2025	Laporan	- Latar Belakang - Tujuan - Produk - Kesimpulan	HP
8.	9/8 2025	Laporan	User Acceptance Testing.	HP

Tegal, Juli 2025

Dosen Pembimbing I



Slamet Wiyono, S.Pd., M.Eng.
NIPY. 08 015.222

Lampiran 5. Surat Kesepakatan Pembimbing II

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : YESAYA WIDIYA
NIM : 21090034
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Dega Surono Wibowo, S.T., M.Kom.
Status : Dosen
NIDN : 0607108202
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata /III/C

Pada hari ini Rabu, 28 Mei 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib melakukan bimbingan 1 kali dalam seminggu atau setidaknya 2 kali bimbingan dalam satu bulan (dengan progress), apabila saya tidak memenuhi syarat tersebut maka saya tidak berhak meminta surat rekomendasi mengikuti sidang skripsi. Dan saya juga berjanji akan memenuhi persyaratan tersebut dan menyelesaikan skripsi baik produk maupun laporan tepat waktu.


Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 28 Mei 2025

Pihak Pertama,


YESAYA WIDIYA

Pihak Kedua,


Dega Surono Wibowo, S.T., M.Kom.

Mengetahui,
Ketua Program Studi Sarjana Terapan
Teknik Informatika


Dyah A. Priatama, S.T., M.Kom.
NIPY. 060118 2025

Lampiran 6. Lembar Bimbingan Pembimbing II

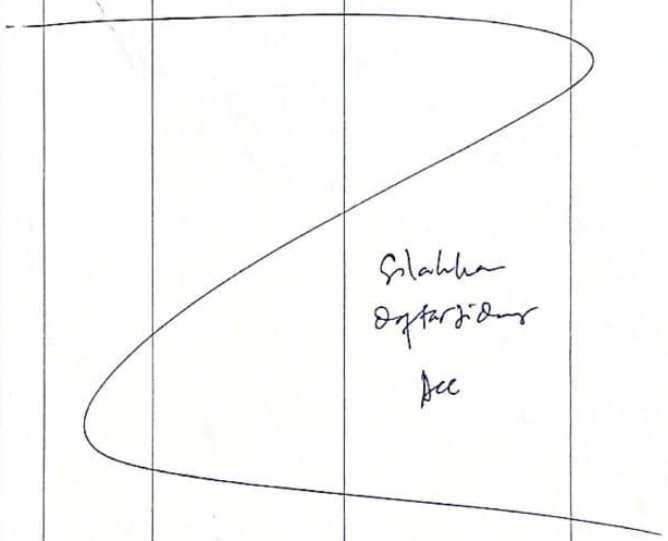


**D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : YESAYA WIDIYA
NIM : 21090034
No. Ponsel : 083808840257
Judul TA : Rancang Bangun Sistem Manajemen Inventori Berbasis Website pada Toko ITO
Dosen Pembimbing II : Dega Surono Wibowo, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	4/6 2015	* perbaiki use case. Lihat referensi dari roni satrio wahono.		<i>[Signature]</i>
2.	13/6 2015	* proses bisnis & perbaiki.		<i>[Signature]</i>
3.	24/6 2015	* lanjut develop aplikasi.		<i>[Signature]</i>
4	8/7 2015	* email masih palsu. * masih banyak form yang lokes uji		<i>[Signature]</i>
5.	18/7 2015	Silahkan lanjut HKL		<i>[Signature]</i>
7.	29/7 2015	Class Diagram belum ada. + usability testing. Lengkap gambar		<i>[Signature]</i>

8.	31/7 2025	Laporan	aku	5
				
Silahkan dipertanggung jawab				

Tegal, 31/7 2025
 Dosen Pembimbing II



Dega Suncho Wibowo, S.T., M.Kom.
 NIPY. 06.014.183

Lampiran 7. Surat Kelayakan Aplikasi

Kepada : Politeknik Harapan Bersama Tegal, Kaprodi S.Tr Teknik Informatika
Alamat : Jl. Mataram No.9, Pesurungan Lor, Kec. Margadana, Kota Tegal
Perihal : Surat Keterangan Kelayakan Aplikasi

Yang bertanda tangan dibawah ini :

Nama : JUNAENI

Jabatan : Kepala Toko

Mitra : Toko ITO

Dengan ini menyatakan bahwa, aplikasi iStock berbasis website yang dirancang oleh :

Nama : YESAYA WIDIYA

NIM : 21090034

Sebagai bagian dari tugas akhir dengan judul :

**"RANCANG BANGUN SISTEM MANAJEMEN INVENTORI PADA TOKO ITO
BERBASIS WEBSITE"**

Telah selesai dikembangkan dan diuji coba. Berdasarkan hasil pengujian dan evaluasi internal, aplikasi tersebut telah memenuhi kriteria fungsional dan kebutuhan pengguna, serta layak untuk digunakan secara resmi dalam mendukung sistem manajemen inventori atau persediaan barang di Toko ITO.

Aplikasi ini diharapkan dapat memberikan kemudahan dalam proses mengelola barang yang lebih terintegrasi dan efisien.

Demikian surat ini kami sampaikan, terimakasih atas kepercayaan yang diberikan kepada kami.

Brebes, 10 Juli 2025

Kepala Toko ITO




Lampiran 8. Surat Hak Cipta

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : YESAYA WIDIYA
Kewarganegaraan : Indonesia
Alamat : Desa Dukuh Jeruk, RT 005 / RW 004 Kecamatan Banjarharjo,
Kabupaten Brebes, Jawa Tengah, 52265
2. Nama : Slamet Wiyono, S.Pd., M.Eng
Kewarganegaraan : Indonesia
Alamat : Perumahan Mutiara Vantavin I Pacul, Kecamatan Talang,
Kabupaten Tegal, Jawa Tengah, 52193
3. Nama : Dega Suroño Wibowo, S.T., M.Kom
Kewarganegaraan : Indonesia
Alamat : Perumahan Sapphire Regency Blok H, No.1, RT.004, RW.001,
Kelurahan Pulosari, Kecamatan Brebes, 52213

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : RANCANG BANGUN SISTEM MANAJEMEN INVENTORI PADA TOKO ITO
BERBASIS WEBSITE
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
 - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.

- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 15 Juli 2025



(YESAYA WIDIYA)
Pemegang Hak Cipta *

(Slamet Wiyono, S.Pd., M.Eng.)
Pemegang Hak Cipta *

(Dega Surono Wibowo, S.T., M.Kom.)
Pemegang Hak Cipta *

REPUBLIK INDONESIA
KEMENTERIAN HUKUM

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 : EC002025101511, 30 Juli 2025

Pencipta

Nama : YESAYA WIDIYA, Slamet Wiyono, S.Pd., M.Eng dkk

Alamat : Desa Dukuh Jeruk, Rt 005 / Rw 004, Banjarharjo, Kab. Brebes, Jawa Tengah, 52265

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama

Alamat : Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142

Kewarganegaraan : Indonesia

Jenis Ciptaan : Program Komputer

Judul Ciptaan : Rancang Bangun Sistem Manajemen Inventori Pada Toko ITO Berbasis Website

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 30 Juli 2025, di Kota Tegal

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor Pencatatan : 000941772

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
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri

Agung Damarsasongko, SH., MH.
NIP. 196912261994031001

No	Nama	Alamat
1	YESAYA WIDIYA	Desa DukuhJeruk, Rt 005 / Rw 004 Banjarharjo, Kab. Brebes
2	Slamet Wiyono, S.Pd., M.Eng	Perumahan Mutiara Vantavin I Pacul Talang, Kab. Tegal
3	Dega Surono Wibowo, S.T., M.Kom	Perumahan Shapphire Regency Blok H, No. 1, RT. 004, RW. 001, Kelurahan Pulosari Brebes, Kab. Brebes





Manual Book

Rancang Bangun Sistem Manajemen Inventori pada Toko ITO Berbasis Website

Disusun Oleh :

YESAYA WIDIYA

Slamet Wiyono, S.Pd., M.Eng.

Dega Surono Wibowo, S.T., M.Kom.

1. PENDAHULUAN

Di tengah perkembangan teknologi digital yang pesat, pengelolaan inventori secara manual mulai ditinggalkan karena dinilai kurang efisien dan rawan kesalahan. Untuk menjawab tantangan tersebut, sistem manajemen inventori berbasis website hadir sebagai solusi modern yang dapat membantu pelaku usaha dalam mengelola stok barang secara lebih akurat, cepat, dan terorganisir. Sistem ini dirancang untuk memudahkan pemantauan keluar masuknya barang, pencatatan transaksi, hingga pelaporan data inventori secara real-time. Melalui platform ini, pengguna dapat dengan mudah mengakses informasi stok, melakukan pembaruan data, serta menghindari kekurangan maupun kelebihan barang secara lebih efektif. Dengan demikian, sistem ini memberikan dukungan penting dalam meningkatkan efisiensi operasional dan pengambilan keputusan yang lebih tepat dalam kegiatan usaha sehari-hari.

2. PERANGKAT YANG DIBUTUHKAN

Perangkat lunak yang dibutuhkan sebagai berikut :

- a. Sistem operasi seperti windows untuk menjalankan seluruh pengembangan.
- b. Web Browser digunakan untuk mengakses dan menguji sistem.
- c. Visual Studio Code berfungsi sebagai editor utama dalam mengembangkan kode proyek.
- d. Flask sebagai framework Python untuk membangun dan mengelola sistem backend.
- e. MySQL sebagai sistem manajemen bisnis data untuk menyimpan dan mengelola data aplikasi.
- f. Laragon digunakan sebagai lingkungan pengembangan server lokal yang lengkap dan mudah digunakan.

3. PENGGUNA APLIKASI

Pengguna aplikasi yang akan menggunakan Aplikasi iStock adalah :

- a. Admin : Pengguna ini memiliki fitur untuk mengelola data supplier, data produk, data pembelian, data penjualan, dan laporan stok.

Admin juga dapat menonaktifkan supplier jika nantinya sudah tidak bekerja sama lagi dengan Toko ITO.

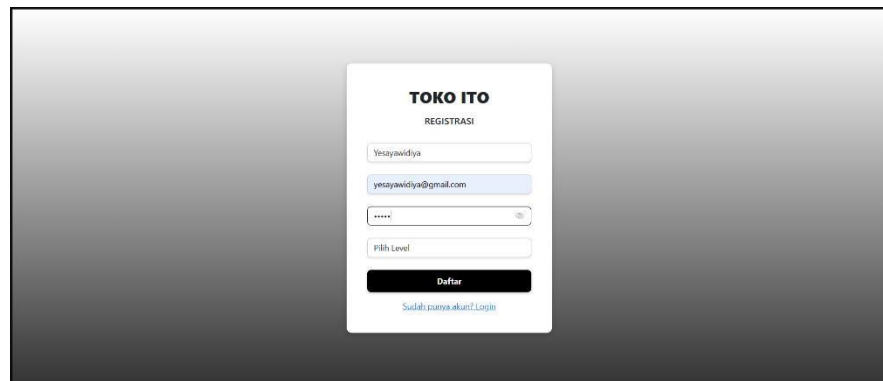
- b. Pemilik Toko ITO (Owner) : Pengguna ini memiliki fitur untuk melihat data supplier, data produk, data pembelian sekaligus menyetujui data pembelian yang sudah diinput oleh admin sehingga stok barang otomatis ditambahkan, melihat data penjualan.

4. CARA PENGGUNAAN APLIKASI

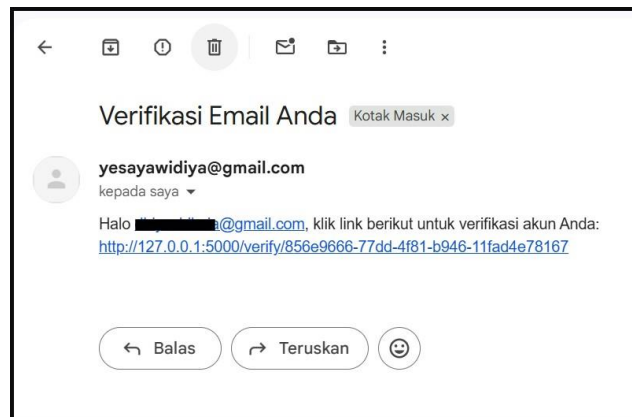
a. Bagi Admin

Pada bagian ini akan dijelaskan mengenai menu-menu yang dapat diakses oleh admin iStock serta cara penggunaannya.

1. Melakukan Register

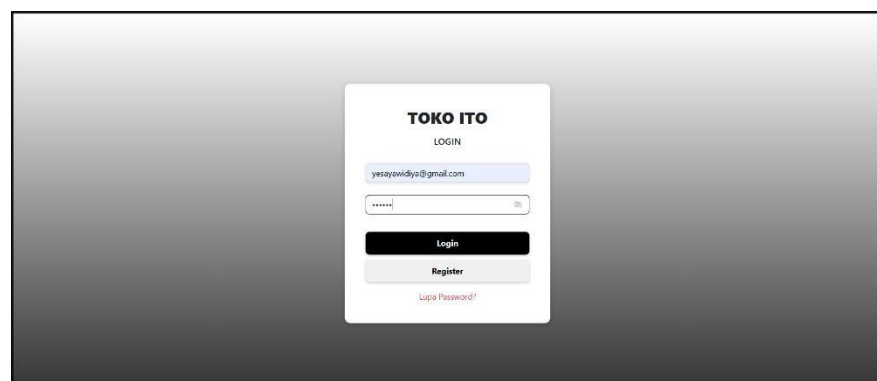
The image shows a registration form titled 'TOKO ITO REGISTRASI'. It contains several input fields: a text field for 'Username' with the value 'Yesyawidiya', an email field with 'yesyawidiya@gmail.com', a password field with masked characters '*****', and a dropdown menu for 'Pilih Level'. Below these fields is a black button labeled 'Daftar' and a blue link that says 'Sudah punya akun? Login'.

- Tampilan halaman register, untuk mempunyai akun, setiap user harus melakukan registrasi akun menggunakan email yang aktif, mengisi username yang akan ditampilkan di aplikasi iStock, masukkan password baru dan memilih role sesuai jabatan (contoh: Admin) Kemudian Klik Register.



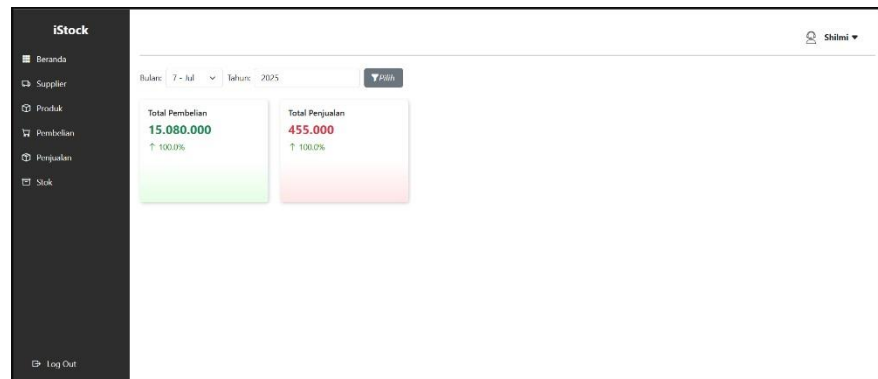
- Tampilan halaman verifikasi, setelah mengisi form registrasi pengguna harus melakukan verifikasi dengan cara mengklik token yang dikirimkan melalui email yang dicantumkan (pastikan alamat email aktif).

2. Melakukan Login



- Tampilan login, setelah verifikasi user akan otomatis masuk ke halaman login. Kemudian masukkan email dan password yang sudah dibuat.

3. Halaman Beranda



- Tampilan halaman beranda admin.

4. Halaman Supplier

The screenshot shows the 'Daftar Supplier' (Supplier List) page. It includes a search bar and a table with the following columns: No., Nama Supplier, No. Telepon, Email, Alamat, Status, and Aksi. The table lists 10 suppliers.

No.	Nama Supplier	No. Telepon	Email	Alamat	Status	Aksi
001	CV. Indo Primera	+62852-9331-8583	indoprimer@gmail.com	Jl. Perakatan No. 24, Purnokerto Selatan, Purnokerto	Aktif	[Edit] [Hapus]
002	CV. Graha Maslita Grogol	+62859-6733-2055	grahamaslita@gmail.com	Jl. Dr. Angka Gang Tulip No. 2, Karangloban, Purnokerto Timur	Aktif	[Edit] [Hapus]
003	CV. Sentosa Jaya Mandiri	+628767676	sentosajaya@gmail.com	Jl. Arteri Yoi Sudarto No. 2, Semarang Barat, Semarang	Aktif	[Edit] [Hapus]
004	Indogrosir Semarang	+6276928282	indogrosirng@gmail.com	Jl. Kaligawe Raya KM 5.1 No. 38, Genuk, Semarang	Aktif	[Edit] [Hapus]
005	PT. Cipta Niaga Semesta	+628767678	uptaniaga@gmail.com	Jl. Pemuda, Rte. 11, Pesisir Selatan, Kec. Wanasari, Kabupaten Ende, Jawa Tengah	Aktif	[Edit] [Hapus]
006	PT. Indomarco Adi Prima	+62822-4203-1680	indomarco@gmail.com	Jalan Raya Cirebon Tegul No. KM 10, Endar, Kec. Pangeman, Kabupaten Cirebon, Jawa Barat 45182	Aktif	[Edit] [Hapus]
007	PT. Intiniaga Teras Segelberta	+62836767676	intiniagatu@gmail.com	Jl. Raya Maribaya No. 99A, Pengasih, Maribaya, Kec. Kiamat, Kabupaten Tegay, Jawa Tengah 52181	Aktif	[Edit] [Hapus]
008	PT. Mitra Sarana Purnama	+6287600337	mibasaran@gmail.com	Jl. Tambak Aji 15, Tambakaji, Semarang	Aktif	[Edit] [Hapus]
009	PT. Olagafood Sukses Perkasa	+6287612333	olagaperkasa@gmail.com	Jl. Dr. Sunarto No. 212-214, Semarang	Aktif	[Edit] [Hapus]
010	PT. Trimira Usaha Niaga	+6286717605	trimiraniaga@gmail.com	Jl. Walter Monginoldi No. 1, Semarang	Aktif	[Edit] [Hapus]

- Tampilan halaman supplier, admin dapat mengelola data supplier, seperti menambahkan supplier dan menonaktifkan supplier yang sudah tidak lagi bekerja sama dengan Toko ITO.

This image shows a close-up of the search functionality in the supplier list. A text input field contains the name 'Intiniaga', and a magnifying glass icon is visible to its right. Below the input field, there are two buttons labeled 'Status' and 'Aksi'.

- Pengguna bisa melakukan pencarian menggunakan fitur pencarian

- Tampilan halaman formulir tambah supplier.

5. Halaman Produk

No.	Nama Produk	Supplier	Harga Beli	Harga Jual	Aksi
001	Aqua Botol 600ml	CV. Sempurna Jaya Mandiri	Rp. 50.000	Rp. 55.000	✓
002	Duaa Bubuk 1kg	CV. Indo Perdana	Rp. 180.000	Rp. 195.000	✓
003	Emperan coklat	PT. Trimikra Usaha Niaga	Rp. 88.000	Rp. 95.000	✓
004	Gula Pasir 1kg	Indogrosir Semarang	Rp. 147.000	Rp. 155.000	✓
005	Indomie Goreng HypeAbis	PT. Intinaga I Lanes Sejahtera	Rp. 130.000	Rp. 165.000	✓
006	Indomie Goreng	PT. Intinaga I Lanes Sejahtera	Rp. 123.000	Rp. 145.000	✓
007	Kacap Mendo ABC 620ml	PT. Mitra Sarana Purnama	Rp. 155.000	Rp. 165.000	✓
008	Muko Pewangi Pakelan 900ml	PT. Olagelood Sukses Perkasa	Rp. 180.000	Rp. 190.000	✓
009	Rissui Detergen Bubuk	PT. Olagelood Sukses Perkasa	Rp. 238.000	Rp. 245.000	✓
010	SOM Indomie Sachet 40g	PT. Indomercu Adi Prima	Rp. 57.000	Rp. 65.000	✓

- Tampilan halaman produk, admin dapat mengelola data produk, seperti menambahkan produk apa saja yang dijual dari setiap supplier.

- Tampilan halaman tambah produk.

- Sebelum menambahkan produk, pilih supplier terlebih dahulu. Karena produk yang akan ditampilkan adalah produk yang sesuai dengan supplier yang dipilih.

6. Halaman Pembelian

No.	No. Pembelian	Tanggal	Supplier	Total (Rp)	Status	Nota	Aksi
001	PO-18-07-2025-8780E440	18-07-2025	PT. Mitra Sarana Purnama	Rp 775.000	Menunggu	Ubat Gambar	
002	PO-18-07-2025-FE528834	17-07-2025	PT. Indinaga Tunas Segitama	Rp 2.800.000	Terpenuhi	Ubat Gambar	
003	PO-18-07-2025-7208C8F5	17-07-2025	Indogrosik Semarang	Rp 2.115.000	Terpenuhi	Ubat Gambar	
004	PO-18-07-2025-60841148	17-07-2025	PT. Timbra Usaha Niaga	Rp 440.000	Terpenuhi	Ubat Gambar	
005	PO-18-07-2025-85960873	17-07-2025	CV. Graha Masatika Grosir	Rp 750.000	Terpenuhi	Ubat Gambar	
006	PO-18-07-2025-67D337C0	17-07-2025	CV. Indo Primera	Rp 8.400.000	Terpenuhi	Ubat Gambar	

- Tampilan halaman pembelian, admin dapat mengelola data pembelian, seperti menambahkan data transaksi pembelian apa saja dari supplier. Data yang disimpan akan menunggu untuk diverifikasi atau disetujui oleh owner sehingga jumlah stok dalam sistem bisa bertambah.

- Tampilan tambah data pembelian, data produk tidak akan semuanya muncul. Yang akan ditampilkan sesuai dengan data produk dari supplier yang dipilih oleh admin.

- Pada halaman tambah data pembelian, admin harus mengupload nota pembelian pada form upload nota pembelian sebagai dokumentasi arsip digital.

7. Halaman Penjualan

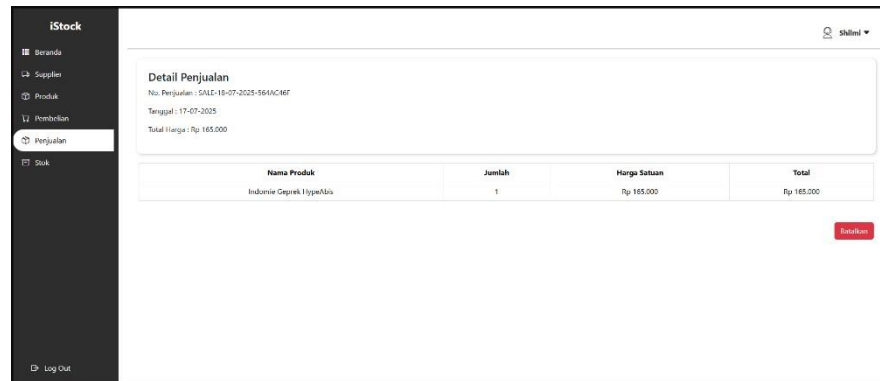
No.	No. Penjualan	Tanggal	Total (Rp)	Aksi
001	SALE-18-07-2025-56A4C46F	17-07-2025	Rp 163.000	
002	SALE-18-07-2025-1DEA9A2E	17-07-2025	Rp 290.000	

- Tampilan halaman penjualan, admin dapat mengelola data penjualan, seperti menambahkan data transaksi penjualan apa saja dari customer.

Produk	Harga	Jumlah	total
Sunlight Jeruk Nipis 750ml	Rp180.000	1	Rp180.000
Molto Pemanis Putahan 900ml	Rp190.000	1	Rp190.000

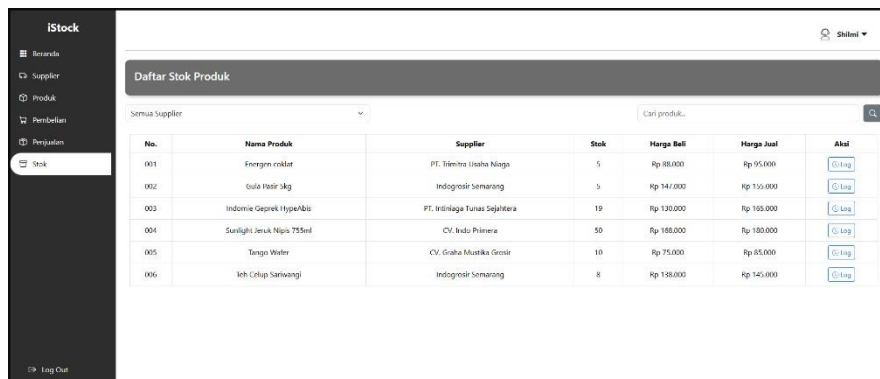
Total Harga Seluruh Produk: Rp3.030.000

- Tampilan halaman tambah data penjualan, Data yang disimpan akan otomatis diverifikasi sehingga jumlah stok dalam sistem bisa berkurang.



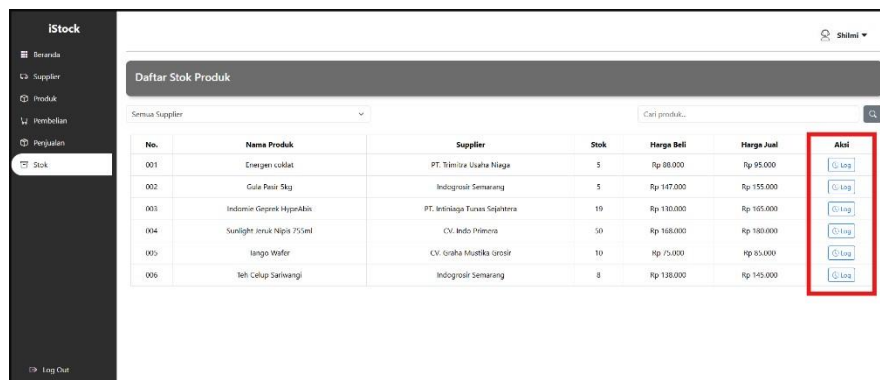
- Tampilan detail penjualan, pengguna bisa mengklik aksi detail untuk melihat aksi penjualan.

8. Halaman Stok

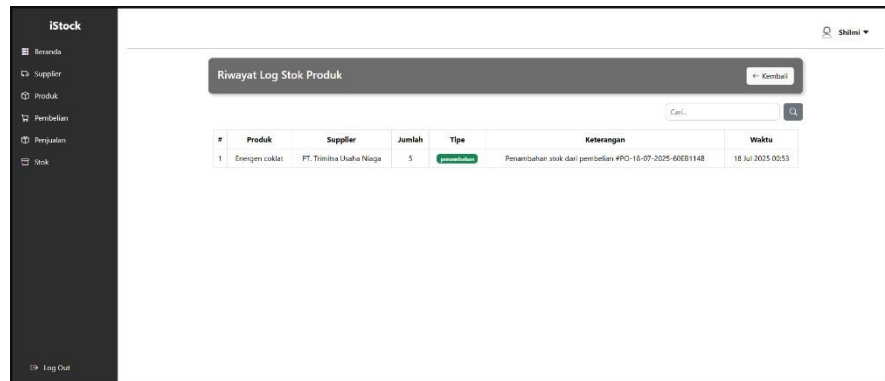


- Pada halaman ini, admin dapat melihat jumlah stok produk yang tersedia saat ini.

9. Halaman Log Riwayat Stok



- Tampilan halaman log riwayat stok, pada halaman ini sistem akan mencatat riwayat transaksi yang dilakukan pada setiap produk (contoh: penambahan)

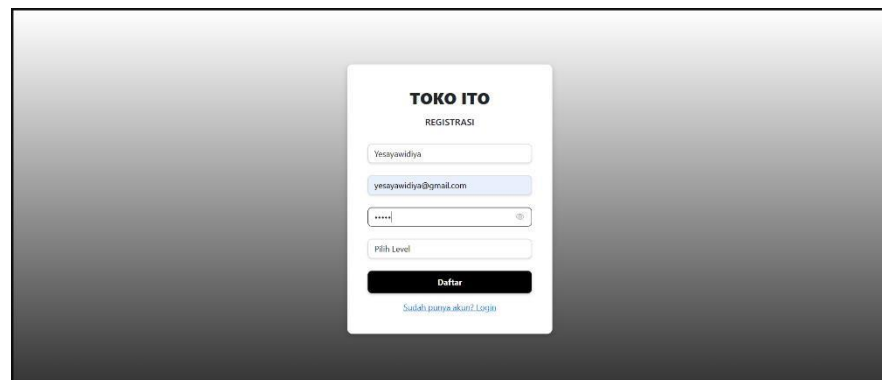


No	Produk	Supplier	Jumlah	Tipe	Keterangan	Waktu
1	Energen coklat	PT. Trimitra Usaha Negera	5	Penambahan	Penambahan stok dari pembelian WPD-18-07-2025-60EB114B	18 Jul 2025 00:53

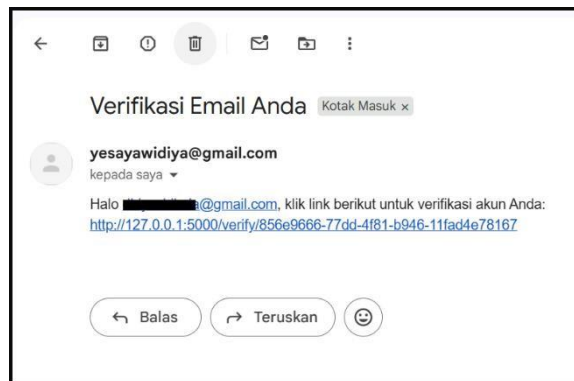
b. Bagi Pemilik Toko (Owner)

Pada bagian ini akan dijelaskan mengenai menu-menu yang dapat diakses oleh Pemilik Toko (Owner) iStock serta cara penggunaannya.

1. Melakukan Register

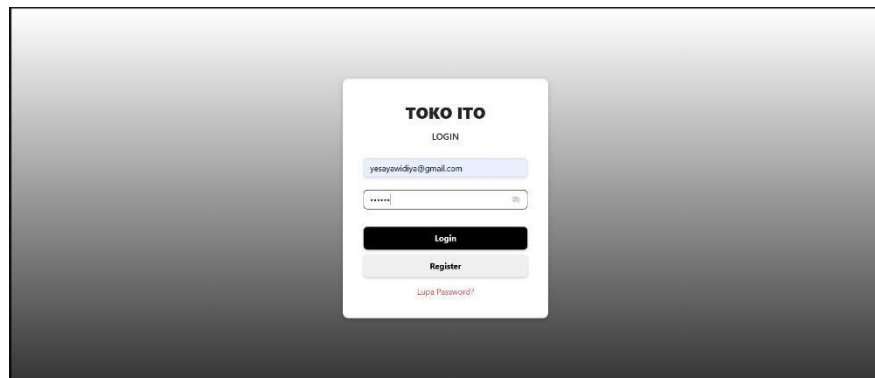


- Tampilan halaman register, untuk mempunyai akun, setiap user harus melakukan registrasi akun menggunakan email yang aktif, mengisi username yang akan ditampilkan di aplikasi iStock, masukkan password baru dan memilih role sesuai jabatan (contoh: Owner) Kemudian Klik Register.



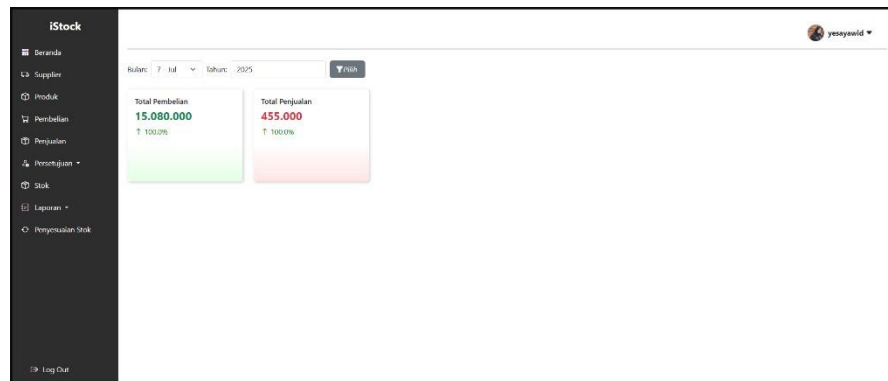
- Tampilan halaman verifikasi, setelah mengisi form registrasi pengguna harus melakukan verifikasi dengan cara mengklik token yang dikirimkan melalui email yang dicantumkan (pastikan alamat email aktif).

2. Melakukan Login



- Tampilan login, setelah verifikasi user akan otomatis masuk ke halaman login. Kemudian masukkan email dan password yang sudah dibuat.

3. Halaman Beranda



- Tampilan halaman beranda owner.

4. Halaman Supplier

No.	Nama Supplier	No. Telp	Email	Alamat	Status
001	CV. Indo Prima	+62852-9334-9562	indoprimer@gmail.com	Jl. Pertabaran No. 24, Purwokerto Selatan, Purwokerto	Aktif
002	CV. Graha Mustika Gendit	+62858-6733-2055	grahamustika@gmail.com	Jl. Dr. Angka Gang Tulip No. 7, Karangasem, Purwokerto Timur	Aktif
003	CV. Sentosa Jaya Mandiri	+628767676	sentosajaya@gmail.com	Jl. Arteri Yos Sudarso No. 2, Semarang Barat, Semarang	Aktif
004	Indogrosir Semarang	+6276626262	indogrosirsem@gmail.com	Jl. Kaligawe Raya KM 5,1 No. 38, Genuk, Semarang	Aktif
005	PT. Cipta Niaga Semesta	+628767676	ciptaniaga@gmail.com	Jl. Pemuda, Riw. 11, Pesantunan, Kec. Wanasari, Kabupaten Brebes, Jawa Tengah	Aktif
006	PT. Indomarco Adi Prima	+62822-4203-4480	indomarco@gmail.com	Jalan Raya Cirebon Tegel No. KM 18, Endre, Km. Pengaman, Kabupaten Cirebon, Jawa Barat 45182	Aktif
007	PT. Intinaga Tunas Sejati	+6288867676	intinagat@gmail.com	Jl. Raya Marbaya No. 99a, Pengasinan, Marbaya, Kec. Kramat, Kabupaten Jegal, Jawa Tengah 52181	Aktif
008	PT. Mira Sarana Panama	+6287603337	mira sarana@gmail.com	Jl. Tambak Aji 15, Tembakaji, Semarang	Aktif
009	PT. Olagafood Sukses Perkasa	+6287612333	olagaperkasa@gmail.com	Jl. Dr. Sutomo No. 312-314, Semarang	Aktif
010	PT. Teleria Usaha Niaga	+6286717605	teloriana@gmail.com	Jl. Walier Manghidil No. 1, Semarang	Aktif

- Tampilan halaman supplier, pada halaman ini owner hanya bisa melihat data supplier yang sudah diinputkan oleh admin

5. Halaman Produk

No.	Nama Produk	Supplier	Harga Beli	Harga Jual
001	Aqua Botol 600ml	CV. Sentosa Jaya Mandiri	Rp 50.000	Rp 55.000
002	Dua Bubuk 1kg	CV. Indo Prima	Rp 180.000	Rp 185.000
003	Intergen coklat	PT. Intinaga Tunas Sejati	Rp 88.000	Rp 95.000
004	Gula Pasir 5kg	Indogrosir Semarang	Rp 147.000	Rp 155.000
005	Indonir Gekrek Hygienis	PT. Intinaga Tunas Sejati	Rp 130.000	Rp 165.000
006	Indonir Goring	PT. Intinaga Tunas Sejati	Rp 125.000	Rp 140.000
007	Kecap Mantis ABC 620ml	PT. Mira Sarana Panama	Rp 155.000	Rp 165.000
008	Melon Pwargi Pakisan 100ml	PT. Olagafood Sukses Perkasa	Rp 180.000	Rp 190.000
009	Rinco Detergen Bubuk	PT. Olagafood Sukses Perkasa	Rp 228.000	Rp 245.000
010	SAM Indomali Sachet 40g	PT. Indomarco Adi Prima	Rp 57.000	Rp 65.000

- Tampilan halaman produk, pada halaman ini owner hanya bisa melihat data produk yang sudah diinputkan oleh admin.

6. Halaman Pembelian

No.	No. Pembelian	Tanggal	Supplier	Total (Rp)	Status	Nota	Aksi
001	PO-18-07-2025-6700E44D	18-07-2025	PT. Mitra Sarana Purnama	Rp 775.000	Menunggu	Lihat	[Edit] [Hapus]
002	PO-18-07-2025-72526334	17-07-2025	PT. Indinaga Tunas Segelitera	Rp 2.600.000	Disetujui	Lihat	[Edit] [Hapus]
003	PO-18-07-2025-7208C8F5	17-07-2025	Indogrosir Semarang	Rp 2.115.000	Disetujui	Lihat	[Edit] [Hapus]
004	PO-18-07-2025-60E81148	17-07-2025	PT. Silemra Usaha Niaga	Rp 440.000	Disetujui	Lihat	[Edit] [Hapus]
005	PO-18-07-2025-3F960873	17-07-2025	CV. Graha Mustika Groser	Rp 750.000	Disetujui	Lihat	[Edit] [Hapus]
006	PO-18-07-2025-67D381CD	17-07-2025	CV. Indo Primera	Rp 8.400.000	Disetujui	Lihat	[Edit] [Hapus]

- Tampilan halaman pembelian.

7. Halaman Penjualan

No.	No. Penjualan	Tanggal	Total (Rp)	Aksi
001	SALE-18-07-2025-5644C46F	17-07-2025	Rp 165.000	[Edit] [Hapus]
002	SALE-18-07-2025-1DEAB82E	17-07-2025	Rp 290.000	[Edit] [Hapus]

- Tampilan halaman penjualan.

8. Halaman Persetujuan

No.	No. Pembelian	Tanggal	Supplier	Total (Rp)	Status	Nota	Aksi
001	PO-18-07-2025-6700E44D	18-07-2025	PT. Mitra Sarana Purnama	Rp 775.000	Menunggu	Lihat	[Edit] [Hapus]
002	PO-18-07-2025-72526334	17-07-2025	PT. Indinaga Tunas Segelitera	Rp 2.600.000	Disetujui	Lihat	[Edit] [Hapus]
003	PO-18-07-2025-7208C8F5	17-07-2025	Indogrosir Semarang	Rp 2.115.000	Disetujui	Lihat	[Edit] [Hapus]
004	PO-18-07-2025-60E81148	17-07-2025	PT. Silemra Usaha Niaga	Rp 440.000	Disetujui	Lihat	[Edit] [Hapus]
005	PO-18-07-2025-3F960873	17-07-2025	CV. Graha Mustika Groser	Rp 750.000	Disetujui	Lihat	[Edit] [Hapus]
006	PO-18-07-2025-67D381CD	17-07-2025	CV. Indo Primera	Rp 8.400.000	Disetujui	Lihat	[Edit] [Hapus]

- Tampilan halaman data persetujuan, Owner dapat melakukan verifikasi terhadap data pembelian yang telah diinputkan oleh

admin, jika data itu benar maka akan disetujui, jika tidak benar maka owner punya wewenang untuk menghapus data tersebut.

No.	No. Pembelian	Tanggal	Supplier	Total (Rp)	Status	Nota	Aksi
001	PO-18-07-2025-6780440	18-07-2025	PT. Mitra Sarana Purnama	Rp 775.000	Menunggu	Lihat	Detail, Hapus, Setujui
002	PO-18-07-2025-44520844	17-07-2025	PT. Indinaga Tunas Sejahtera	Rp 2.600.000	Setujui	Lihat	Detail, Hapus, Setujui
003	PO-18-07-2025-7080895	17-07-2025	Indogrosir Semarang	Rp 2.115.000	Setujui	Lihat	Detail, Hapus, Setujui
004	PO-18-07-2025-69881148	17-07-2025	PT. Wimbis Usaha Niaga	Rp 440.000	Setujui	Lihat	Detail, Hapus, Setujui
005	PO-18-07-2025-2090873	17-07-2025	CV. Graha Mustika Grosir	Rp 750.000	Setujui	Lihat	Detail, Hapus, Setujui
006	PO-18-07-2025-67033700	17-07-2025	CV. Indo Primera	Rp 8.400.000	Setujui	Lihat	Detail, Hapus, Setujui

- Tampilan halaman detail pesanan, untuk menyetujui data pembelian owner bisa memverifikasi melalui tombol detail pembelian.

No. Pembelian	Tanggal	Nama Produk	Jumlah	Total Harga	Arsip Nota	Supplier
PO-18-07-2025-6780440	18-07-2025	Kecap Manis ABC 620ml	5	Rp 775.000	Lihat	PT. Mitra Sarana Purnama

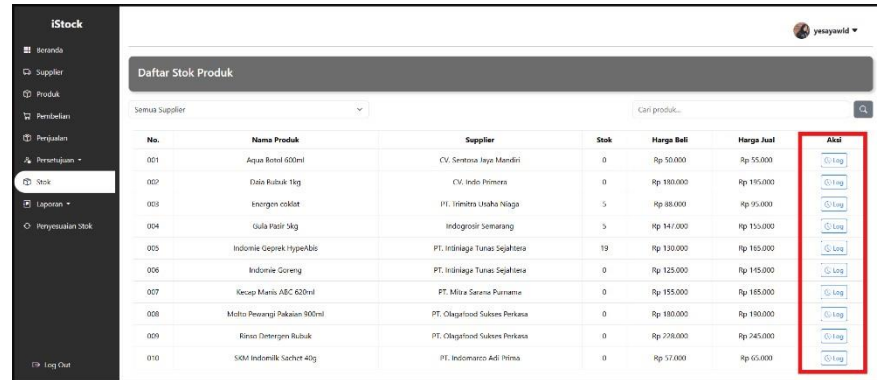
- Tampilan halaman detail pesanan, klik setuju untuk menyetujui data.

9. Halaman Stok

No.	Nama Produk	Supplier	Stok	Harga Beli	Harga Jual	Aksi
001	Aqua Botol 600ml	CV. Sentosa Jaya Mandiri	0	Rp 50.000	Rp 55.000	Log
002	Dada Putih 1kg	CV. Indo Primera	0	Rp 110.000	Rp 115.000	Log
003	Empon coklat	PT. Wimbis Usaha Niaga	5	Rp 88.000	Rp 95.000	Log
004	Gula Putih 5kg	Indogrosir Semarang	5	Rp 147.000	Rp 155.000	Log
005	Indomie Geprek HypekBis	PT. Indinaga Tunas Sejahtera	19	Rp 130.000	Rp 165.000	Log
006	Indomie Goreng	PT. Indinaga Tunas Sejahtera	0	Rp 125.000	Rp 115.000	Log
007	Kecap Manis ABC 620ml	PT. Mitra Sarana Purnama	0	Rp 155.000	Rp 185.000	Log
008	Melita Peeringi Pakan 900ml	PT. Olegood Sukem Perkasa	0	Rp 180.000	Rp 190.000	Log
009	Renes Detergen Bubuk	PT. Olegood Sukem Perkasa	0	Rp 298.000	Rp 245.000	Log
010	SUM Indomilb Sarchet 40g	PT. Indomilb A&B Prima	0	Rp 57.000	Rp 65.000	Log

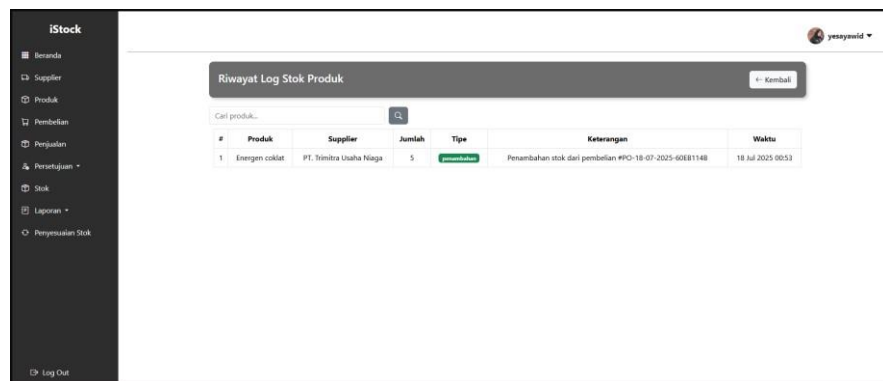
- Tampilan halaman stok, pada halaman ini owner dapat melihat jumlah stok produk yang tersedia saat ini.

10. Halaman Log Riwayat Stok



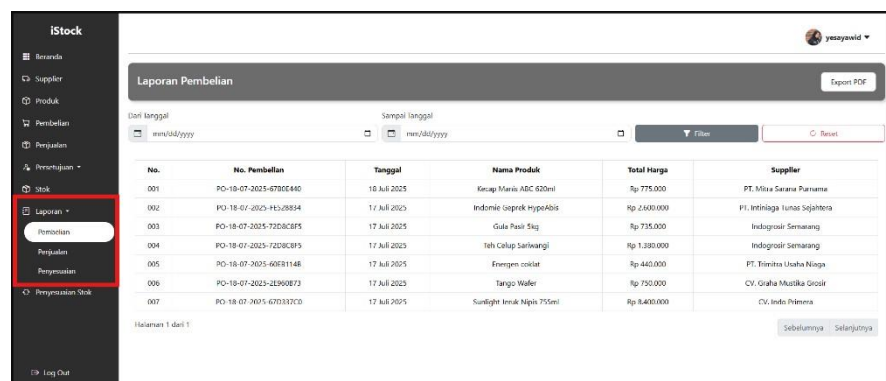
No.	Nama Produk	Supplier	Stok	Harga Beli	Harga Jual	Aksi
001	Aqua Botol 600ml	CV. Semesta Jaya Mandiri	0	Rp 55.000	Rp 55.000	Log
002	Dada Bubuk 1kg	CV. Indo Primera	0	Rp 180.000	Rp 195.000	Log
003	Energen coklat	PT. Trimitra Usaha Niaga	5	Rp 88.000	Rp 95.000	Log
004	Gula Pasir 5kg	Indogrosir Semarang	5	Rp 147.000	Rp 150.000	Log
005	Indomie Geprek MypeAbis	PT. Indinaga Tunas Sejahtera	19	Rp 130.000	Rp 165.000	Log
006	Indomie Goreng	PT. Indinaga Tunas Sejahtera	0	Rp 125.000	Rp 195.000	Log
007	Kecap Manis ABC 620ml	PT. Misa Sarana Purnama	0	Rp 155.000	Rp 185.000	Log
008	Molto Pemanis Pakan 900ml	PT. Olegrosid Sukono Perkasa	0	Rp 180.000	Rp 190.000	Log
009	Kirana Detergen Bubuk	PT. Olegrosid Sukono Perkasa	0	Rp 238.000	Rp 245.000	Log
010	SOM Indomilk Sachet 40g	PT. Indomarco Adi Prima	0	Rp 57.000	Rp 65.000	Log

- Tampilan halaman log riwayat stok, pada halaman ini sistem akan mencatat riwayat transaksi yang dilakukan pada setiap produk (contoh: penambahan).



#	Produk	Supplier	Jumlah	Tipe	Keterangan	Waktu
1	Energen coklat	PT. Trimitra Usaha Niaga	5	penambahan	Penambahan stok dari pembelian #PO-18-07-2025-608B1148	18 Jul 2025 00:53

11. Halaman Laporan



No.	No. Pembelian	Tanggal	Nama Produk	Total Harga	Supplier
001	PO-18-07-2025-6700C440	16 Juli 2025	Kecap Manis ABC 620ml	Rp 775.000	PT. Misa Sarana Purnama
002	PO-18-07-2025-4432B934	17 Juli 2025	Indomie Geprek MypeAbis	Rp 2.600.000	PT. Indinaga Tunas Sejahtera
003	PO-18-07-2025-720BC8F5	17 Juli 2025	Gula Pasir 5kg	Rp 735.000	Indogrosir Semarang
004	PO-18-07-2025-720BC8F5	17 Juli 2025	Teh Celup Sariwangi	Rp 1.380.000	Indogrosir Semarang
005	PO-18-07-2025-608B1148	17 Juli 2025	Energen coklat	Rp 440.000	PT. Trimitra Usaha Niaga
006	PO-18-07-2025-2E90E873	17 Juli 2025	Tango Waler	Rp 750.000	CV. Graha Mustika Grosir
007	PO-18-07-2025-6700C440	17 Juli 2025	Swirlight Isotak Niplo 755ml	Rp 8.400.000	CV. Indo Primera

- Tampilan halaman laporan, menampilkan seluruh transaksi pembelian, penjualan, dan penyesuaian stok yang terjadi. Jika owner ingin melihat laporan transaksi dari rentang waktu tertentu bisa menggunakan filter.

No. Pembelian	Tanggal	Nama Produk	Jumlah	Total Harga	Supplier	Status
PO-18-07-2025-47902148	18 Jul 2025	Kecap Manis ABC 650ml	5	Rp 775.000	PT. Mitra Sarana Purnama	menunggu
PO-18-07-2025-PT528824	17 Jul 2025	Indomie Goreng Hyponablis	30	Rp 2.850.000	PT. Indinaga Tamas Sejauhena	diambil
PO-18-07-2025-72582395	17 Jul 2025	Gula Pasir Skg	5	Rp 750.000	Indogrofit Semarang	diambil
PO-18-07-2025-72582395	17 Jul 2025	Teh Gula Kawang	10	Rp 1.300.000	Indogrofit Semarang	diambil
PO-18-07-2025-80051145	17 Jul 2025	Emergen cabai	5	Rp 450.000	PT. Trimika Usaha Niaga	diambil
PO-18-07-2025-81903873	17 Jul 2025	Tengg Volder	10	Rp 750.000	CV. Graha Maslita Group	diambil
PO-18-07-2025-87033702	17 Jul 2025	Santitas Jero Nabo 750ml	30	Rp 8.400.000	CV. Ineh Prima	diambil

- Tampilan halaman export to PDF, fitur ini bisa digunakan jika owner butuh mencetak laporan kedalam bentuk PDF.

12. Halaman Penyesuaian Stok

No.	Nama Produk	Supplier	Jumlah Stok	Aksi
1	Aqua Botol 600ml	CV. Sentosa Jaya Mandiri	0	Penyesuaian
2	Dalia Bubuk 1kg	CV. Indo Prima	0	Penyesuaian
3	Emergen cabai	PT. Trimika Usaha Niaga	5	Penyesuaian
4	Gula Pasir Skg	Indogrofit Semarang	5	Penyesuaian
5	Indomie Goreng Hyponablis	PT. Indinaga Tamas Sejauhena	19	Penyesuaian
6	Indomie Goreng	PT. Indinaga Tamas Sejauhena	0	Penyesuaian
7	Kecap Manis ABC 650ml	PT. Mitra Sarana Purnama	0	Penyesuaian
8	Molto (Newang) Pakan 900ml	PT. Olagafood Sukses Perkasa	0	Penyesuaian
9	Rinso Detergen Bubuk	PT. Olagafood Sukses Perkasa	0	Penyesuaian
10	SDM Indomilk Sachet 1kg	PT. Indomaco Adi Prima	0	Penyesuaian

- Tampilan halaman penyesuaian stok, fitur ini digunakan untuk menyesuaikan jumlah stok barang secara manual apabila terjadi ketidaksesuaian antara catatan sistem dengan kondisi stok fisik di gudang.

iStock

Tindakan Penyesuaian

Nama Produk: Airas Botol 600ml

Supplier: CV. Sentosa Jaya Mandiri

Tanggal: 07/08/2025

Jumlah Fisik: 0

Jumlah Sistem: 0

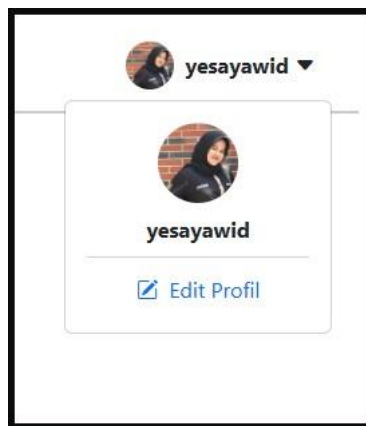
Setor

Simpan

Log Out

- Tampilan formulir penyesuaian stok.

13. Halaman Edit Profil



- Tampilan halaman edit profil, fitur ini digunakan pada saat pengguna ingin mengubah foto dalam sistem.

iStock

Edit Foto Profil

yesayawid

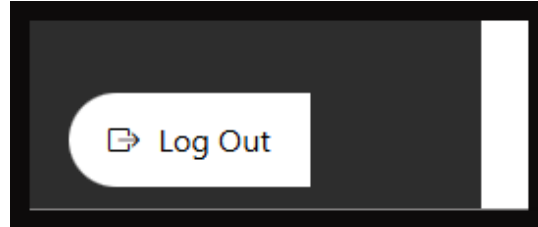
Pilih Foto Baru:

Choose File No file chosen

Upload & Simpan

Log Out

14. LOGOUT



- Untuk keluar dari sistem pengguna bisa mengklik menu Log Out.

The background features a blue header bar at the top. On the left, a large yellow parallelogram is partially visible. On the right, several thin, wavy yellow lines flow downwards. The main title 'Technical Document' is in a large, bold, yellow sans-serif font.

Technical Document

Rancang Bangun Sistem Manajemen Inventori pada Toko ITO Berbasis Website

Disusun Oleh :

YESAYA WIDIYA

Slamet Wiyono, S.Pd., M.Eng.

Dega Surono Wibowo, S.T., M.Kom.

1. PENDAHULUAN

Di tengah perkembangan teknologi digital yang pesat, pengelolaan inventori secara manual mulai ditinggalkan karena dinilai kurang efisien dan rawan kesalahan. Untuk menjawab tantangan tersebut, sistem manajemen inventori berbasis website hadir sebagai solusi modern yang dapat membantu pelaku usaha dalam mengelola stok barang secara lebih akurat, cepat, dan terorganisir. Sistem ini dirancang untuk memudahkan pemantauan keluar masuknya barang, pencatatan transaksi, hingga pelaporan data inventori secara real-time. Melalui platform ini, pengguna dapat dengan mudah mengakses informasi stok, melakukan pembaruan data, serta menghindari kekurangan maupun kelebihan barang secara lebih efektif. Dengan demikian, sistem ini memberikan dukungan penting dalam meningkatkan efisiensi operasional dan pengambilan keputusan yang lebih tepat dalam kegiatan usaha sehari-hari.

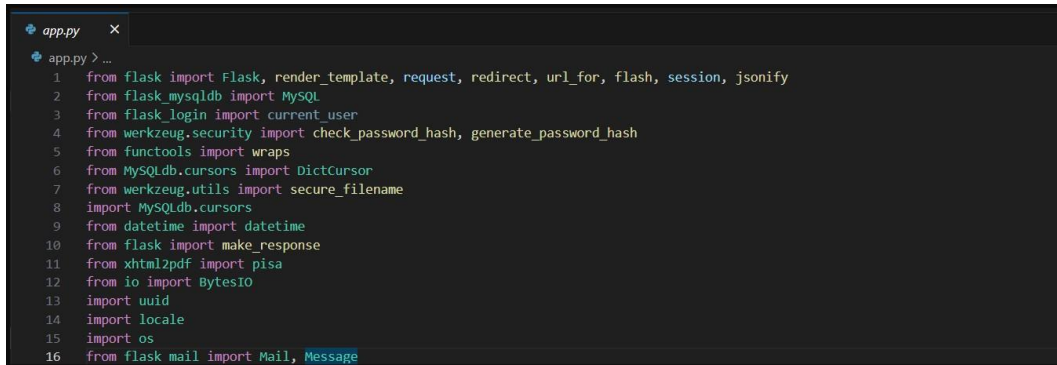
2. PERANGKAT YANG DIBUTUHKAN

Perangkat lunak yang dibutuhkan sebagai berikut :

- a. Sistem operasi seperti windows untuk menjalankan seluruh pengembangan.
- b. Web Browser digunakan untuk mengakses dan menguji sistem.
- c. Visual Studio Code berfungsi sebagai editor utama dalam mengembangkan kode proyek.
- d. Flask sebagai framework Python untuk membangun dan mengelola sistem backend.
- e. MySQL sebagai sistem manajemen bisnis data untuk menyimpan dan mengelola data aplikasi.
- f. Laragon digunakan sebagai lingkungan database.

3. SPESIFIKASI TEKNIS

a. app.py

A screenshot of a code editor window titled 'app.py'. The editor shows a list of Python imports for a Flask application. The imports include Flask, render_template, request, redirect, url_for, flash, session, jsonify from flask; MySQL from flask_mysqlldb; current_user from flask_login; check_password_hash and generate_password_hash from werkzeug.security; wraps from functools; DictCursor from MySQLdb.cursors; secure_filename from werkzeug.utils; MySQLdb.cursors; datetime from datetime; make_response from flask; pisa from xhtml2pdf; BytesIO from io; uuid; locale; os; and Mail, Message from flask_mail.

```
1 from flask import Flask, render_template, request, redirect, url_for, flash, session, jsonify
2 from flask_mysqlldb import MySQL
3 from flask_login import current_user
4 from werkzeug.security import check_password_hash, generate_password_hash
5 from functools import wraps
6 from MySQLdb.cursors import DictCursor
7 from werkzeug.utils import secure_filename
8 import MySQLdb.cursors
9 from datetime import datetime
10 from flask import make_response
11 from xhtml2pdf import pisa
12 from io import BytesIO
13 import uuid
14 import locale
15 import os
16 from flask_mail import Mail, Message
```

Pada bagian ini, dilakukan proses *import* berbagai modul dan pustaka yang berfungsi untuk mendukung operasional aplikasi berbasis web. Modul Flask, flask_mysqlldb, dan flask_login digunakan untuk membangun struktur aplikasi web, melakukan koneksi dengan basis data MySQL, serta mengelola proses autentikasi pengguna. Pustaka werkzeug.security dimanfaatkan untuk melakukan enkripsi kata sandi guna menjaga keamanan data pengguna, sedangkan werkzeug.utils digunakan dalam pengelolaan berkas agar lebih aman. Modul tambahan seperti datetime, uuid, locale, dan os berperan dalam pengaturan waktu, pembuatan identitas unik, pengaturan lokal, serta interaksi dengan sistem operasi. Selain itu, pustaka xhtml2pdf digunakan untuk menghasilkan dokumen dalam format PDF, dan flask_mail digunakan untuk mengirimkan pesan melalui surat elektronik (*email*). Dengan demikian, potongan kode ini berfungsi sebagai inisialisasi komponen-komponen utama yang diperlukan dalam pengembangan aplikasi web berbasis Flask yang memiliki fitur autentikasi, pengelolaan data, pembuatan dokumen, serta pengiriman notifikasi melalui email.

```
# Koneksi database
app.secret_key = 'cobacoba'
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'istock'
mysql = MySQL(app)

app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USERNAME'] = 'yesayawidiya@gmail.com' # Ganti dengan email kamu
app.config['MAIL_PASSWORD'] = 'vcou hiav ryak hozf' # Gunakan App Password

mail = Mail(app)
```

Kode di atas mengatur koneksi aplikasi Flask ke database MySQL dengan host localhost, user root, dan database bernama istock. Selain itu, juga dikonfigurasi pengiriman email menggunakan Gmail SMTP dengan TLS pada port 587, menggunakan App Password untuk keamanan. Kedua konfigurasi ini memungkinkan aplikasi untuk mengakses database dan mengirim email otomatis.

```
# Middleware untuk membatasi akses berdasarkan level pengguna
def login_required(levels=[]):
    def decorator(f):
        @wraps(f) # Menyimpan metadata dari fungsi asli
        def wrapper(*args, **kwargs):
            if 'logged_in' not in session:
                flash('Silakan login terlebih dahulu', 'danger')
                return redirect(url_for('login'))
            if session.get('level') not in levels:
                flash('Anda tidak memiliki akses ke halaman ini', 'danger')
                return redirect(url_for('index'))
            return f(*args, **kwargs)
        return wrapper
    return decorator
```

Kode di atas adalah middleware Flask yang membatasi akses ke halaman tertentu berdasarkan status login dan level pengguna. Fungsi `login_required(level)` digunakan sebagai dekorator. Jika pengguna belum login ('logged_in' not in session), mereka akan diarahkan ke halaman login. Jika pengguna login tapi levelnya tidak sesuai (`session.get('level') not in levels`), mereka akan diarahkan ke halaman utama dengan pesan peringatan. Jika semua syarat terpenuhi, fungsi aslinya dijalankan.

Middleware ini berguna untuk mengamankan halaman berdasarkan hak akses.

```

# Registrasi
@app.route('/registrasi', methods=('GET', 'POST'))
def registrasi():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        level = request.form['level']
        token = str(uuid.uuid4()) # Token unik

        cursor = mysql.connection.cursor()
        cursor.execute('SELECT * FROM users WHERE username=%s OR email=%s', (username, email))
        akun = cursor.fetchone()
        if akun is None:
            hashed_password = generate_password_hash(password)
            cursor.execute(
                'INSERT INTO users (username, email, password, level, is_verified, verification_token) VALUES (%s, %s, %s, %s, %s, %s)'
                (username, email, hashed_password, level, False, token)
            )
            mysql.connection.commit()

```

Kode di atas adalah fungsi registrasi pengguna dalam aplikasi Flask. Saat metode POST dikirim dari form, data seperti username, email, password, dan level diambil. Kemudian dibuat token unik menggunakan `uuid.uuid4()` untuk keperluan verifikasi. Kode memeriksa apakah username atau email sudah terdaftar. Jika belum, password akan dienkripsi dengan `generate_password_hash`, lalu data pengguna disimpan ke tabel `users` dengan status belum diverifikasi (`is_verified=False`) dan token verifikasi. Proses ini diakhiri dengan menyimpan perubahan ke database menggunakan `commit()`.

```

# Kirim email verifikasi
verification_link = url_for('verify_email', token=token, _external=True)
msg = Message('Verifikasi Email Anda', sender='youreemail@gmail.com', recipients=[email])
msg.body = f'Halo {email}, klik link berikut untuk verifikasi akun Anda:\n{verification_link}'
mail.send(msg)

flash('Registrasi berhasil! Cek email untuk verifikasi.', 'success')
return redirect(url_for('login')) # Langsung redirect ke halaman login

else:
    flash('Username atau email sudah ada', 'danger')
return render_template('auth/registrasi.html')

@app.route('/verify/<token>')
def verify_email(token):
    cursor = mysql.connection.cursor()
    cursor.execute('SELECT * FROM users WHERE verification_token=%s', (token,))
    akun = cursor.fetchone()
    if akun:
        cursor.execute('UPDATE users SET is_verified=TRUE, verification_token=NULL WHERE verification_token=%s', (token,))
        mysql.connection.commit()
        flash('Email berhasil diverifikasi. Silakan login.', 'success')
    else:
        flash('Token tidak valid atau sudah digunakan.', 'danger')
    return redirect(url_for('login'))

```

Kode di atas berfungsi untuk mengirim email verifikasi dan memverifikasi akun pengguna. Setelah registrasi berhasil, sistem membuat link verifikasi menggunakan token unik dan mengirimkannya ke email pengguna. Email tersebut berisi pesan dan link untuk mengaktifkan akun. Ketika pengguna mengklik link

verifikasi (/verify/<token>), sistem akan mencari token di database. Jika ditemukan, status akun akan diubah menjadi terverifikasi (is_verified=TRUE) dan token dihapus. Jika token tidak valid atau sudah digunakan, akan ditampilkan pesan error. Setelah proses selesai, pengguna diarahkan ke halaman login.

```
# Login
@app.route('/', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        cursor = mysql.connection.cursor(DictCursor)
        cursor.execute('SELECT * FROM users WHERE email=%s', (email,))
        akun = cursor.fetchone()
        cursor.close()

        if akun is None:
            flash('Login Gagal, Email tidak ditemukan', 'danger')
        elif not check_password_hash(akun['password'], password):
            flash('Login Gagal, Password salah', 'danger')
        elif not akun['is_verified']:
            flash('Email Anda belum diverifikasi. Silakan cek email Anda.', 'warning')
        else:
            # Simpan sesi login
            session['logged_in'] = True
            session['username'] = akun['username']
            session['level'] = akun['level']
            session['foto'] = akun['foto'] if akun['foto'] else 'profile.png'

            # Redirect sesuai level
            if akun['level'] == 'admin':
                return redirect(url_for('admin_page'))
            elif akun['level'] == 'owner':
                return redirect(url_for('owner_page'))
            else:
                return redirect(url_for('index'))

    return render_template("auth/login.html")
```

Kode di atas merupakan fungsi login pada aplikasi Flask. Saat pengguna mengirim form login (POST), sistem mengambil email dan password, lalu mencocokkannya dengan data di database. Jika email tidak ditemukan, password salah, atau akun belum diverifikasi, akan muncul pesan error. Jika berhasil, data pengguna disimpan di session dan diarahkan ke halaman sesuai level-nya (admin_page, owner_page, atau index). Jika halaman diakses dengan metode GET, maka akan menampilkan form login.

```

@app.route('/reset-password/<token>', methods=['GET', 'POST'])
def reset_password(token):
    cursor = mysql.connection.cursor()
    cursor.execute('SELECT * FROM users WHERE verification_token=%s', (token,))
    user = cursor.fetchone()

    if not user:
        flash('Token tidak valid atau sudah kedaluwarsa.', 'danger')
        return redirect(url_for('login'))

    if request.method == 'POST':
        new_password = request.form['password']
        hashed_password = generate_password_hash(new_password)

        # Simpan password baru dan kosongkan token
        cursor.execute('UPDATE users SET password=%s, verification_token=NULL WHERE id=%s', (hashed_password, user[0]))
        mysql.connection.commit()
        flash('Password berhasil diubah. Silakan login kembali.', 'success')
        return redirect(url_for('login'))

    return render_template('auth/reset_password.html', token=token)

```

Kode di atas adalah fungsi reset password menggunakan token verifikasi. Saat pengguna mengakses link `/reset-password/<token>`, sistem mengecek apakah token valid. Jika tidak, pengguna diarahkan ke halaman login dengan pesan error. Jika token valid dan pengguna mengirim form (POST), password baru akan dienkripsi dan disimpan ke database, lalu token dikosongkan. Setelah berhasil, pengguna diarahkan ke halaman login dengan pesan sukses. Jika metode GET, maka akan ditampilkan halaman untuk mengisi password baru.

```

@app.route('/logout')
def logout():
    session.clear()
    return redirect(url_for('login'))

```

Kode di atas adalah fungsi logout dalam aplikasi Flask. Saat pengguna mengakses route `/logout`, semua data sesi (session) akan dihapus menggunakan `session.clear()`, yang berarti pengguna akan keluar dari akun. Setelah itu, pengguna langsung diarahkan ke halaman login menggunakan `redirect(url_for('login'))`.

Fungsi ini memastikan bahwa sesi pengguna benar-benar diakhiri.

```

@app.route('/edit_foto', methods=['GET', 'POST'])
@login_required(['admin', 'owner'])
def edit_foto():
    if request.method == 'POST':
        if 'photo' not in request.files:
            flash('Tidak ada file foto yang diunggah.', 'danger')
            return redirect(request.url)

        photo = request.files['photo']
        if photo.filename == '':
            flash('Tidak ada file yang dipilih.', 'warning')
            return redirect(request.url)

        if photo and allowed_file(photo.filename):
            filename = secure_filename(photo.filename)
            filepath = os.path.join(app.config['PROFILE_UPLOAD_FOLDER'], filename)
            photo.save(filepath)

            # Simpan ke database
            username = session.get('username') # gunakan username yang login
            cursor = mysql.connection.cursor()
            cursor.execute('UPDATE users SET foto=%s WHERE username=%s', (filename, username))
            mysql.connection.commit()
            cursor.close()

            # Update session agar langsung pakai foto baru
            session['foto'] = filename

            flash('Foto profil berhasil diperbarui.', 'success')
            return redirect(url_for('edit_foto'))

        else:
            flash('Format file tidak diizinkan. Gunakan png, jpg, jpeg, atau gif.', 'danger')
            return redirect(request.url)

    return render_template('auth/edit_foto.html', current_path=request.path,)

```

Kode di atas adalah fungsi untuk mengedit foto profil pengguna yang sudah login dengan level 'admin' atau 'owner'. Jika pengguna mengunggah file foto dan formatnya valid (seperti .png, .jpg, dll), maka foto akan disimpan ke folder yang ditentukan, dan nama file disimpan ke database berdasarkan username dari sesi. Setelah itu, sesi juga diperbarui agar langsung menampilkan foto baru. Jika tidak ada file atau format tidak sesuai, akan muncul pesan error. Jika metode GET, akan ditampilkan halaman edit foto.

```

# Route utama untuk halaman supplier
@app.route('/supplier')
@login_required(['admin'])
def supplier():
    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    # Ambil parameter pencarian dan halaman
    search = request.args.get('search', '').strip()
    page = request.args.get('page', default=1, type=int)
    per_page = 10
    offset = (page - 1) * per_page
    search_term = f"%{search}%"

    # Hitung total data
    count_query = """
        SELECT COUNT(*) AS total
        FROM supplier
        WHERE nama LIKE %s
    """
    cur.execute(count_query, (search_term,))
    total_rows = cur.fetchone()[0]
    total_pages = (total_rows + per_page - 1) // per_page

    # Ambil data supplier sesuai halaman dan pencarian
    query = """
        SELECT * FROM supplier
        WHERE nama LIKE %s
        ORDER BY nama ASC
        LIMIT %s OFFSET %s
    """
    cur.execute(query, (search_term, per_page, offset))
    suppliers = cur.fetchall()
    cur.close()

    return render_template('admin/supplier/supplier.html',
        suppliers=suppliers,
        search=search,
        page=page,
        total_pages=total_pages,
        current_path=request.path)

```

Fungsi di atas menampilkan halaman data supplier untuk admin. Data ditampilkan per halaman (pagination) dan bisa dicari berdasarkan nama. Sistem menghitung total data, mengambil data sesuai halaman dan kata kunci pencarian, lalu menampilkannya ke template supplier.html.

```

# Route untuk halaman form tambah supplier
@app.route('/supplier/tambah', methods=['GET', 'POST'])
@login_required(['admin'])
def tambah_supplier():
    if request.method == 'POST':
        nama = request.form['nama']
        telepon = request.form['telepon']
        email = request.form['email']
        alamat = request.form['alamat']

        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO supplier (nama, telepon, email, alamat) VALUES (%s, %s, %s, %s)",
            (nama, telepon, email, alamat))
        mysql.connection.commit()
        cur.close()
        flash('Data supplier berhasil disimpan!', 'success')
        return redirect(url_for('supplier'))

    # Kode otomatis (hanya tampil, bukan id asli)
    cur = mysql.connection.cursor()
    cur.execute("SELECT COUNT(*) as total FROM supplier")
    count = cur.fetchone()[0]
    kode_supplier = f"{count + 1:03d}"
    cur.close()

    return render_template('admin/supplier/tambah_supplier.html', kode_supplier=kode_supplier, current_path=request.path)

```

Fungsi ini menampilkan dan memproses form tambah supplier untuk admin. Jika form dikirim (POST), data supplier disimpan ke database dan ditampilkan notifikasi sukses. Saat halaman dibuka (GET), sistem menghitung total supplier untuk menampilkan kode otomatis, lalu merender halaman tambah_supplier.html.

```

#edit supplier
@app.route('/supplier/edit/<int:id>', methods=['GET', 'POST'])
@login_required(['admin'])
def edit_supplier(id):
    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    if request.method == 'POST':
        nama = request.form['nama']
        telepon = request.form['telepon']
        email = request.form['email']
        alamat = request.form['alamat']

        cur.execute("""
            UPDATE supplier
            SET nama=%s, telepon=%s, email=%s, alamat=%s
            WHERE id=%s
            """, (nama, telepon, email, alamat, id))
        mysql.connection.commit()
        cur.close()
        flash('Data supplier berhasil diperbarui!', 'success')
        return redirect(url_for('supplier'))

    # GET request
    cur.execute("SELECT * FROM supplier WHERE id = %s", (id,))
    supplier = cur.fetchone()
    cur.close()

    if supplier:
        return render_template('admin/supplier/edit_supplier.html', supplier=supplier, current_path=request.path)
    else:
        flash('Supplier tidak ditemukan', 'danger')
        return redirect(url_for('supplier'))

```

Fungsi di atas digunakan untuk mengedit data supplier berdasarkan id. Jika method-nya POST, data dari form akan diambil dan diperbarui di database, lalu ditampilkan pesan sukses dan diarahkan kembali ke halaman supplier. Jika methodnya GET, data supplier yang sesuai id akan diambil dan ditampilkan pada halaman edit_supplier.html. Jika id tidak ditemukan, muncul pesan error dan dialihkan ke halaman supplier. Fungsi ini hanya bisa diakses oleh pengguna dengan level 'admin'.

```

# Mengaktifkan / menonaktifkan supplier
@app.route('/supplier/status/<int:id>', methods=['GET'])
@login_required(['admin'])
def toggle_supplier_status(id):
    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cur.execute("SELECT is_active FROM supplier WHERE id = %s", (id,))
    supplier = cur.fetchone()

    if not supplier:
        flash("Supplier tidak ditemukan.", "danger")
    else:
        new_status = not supplier['is_active']
        cur.execute("UPDATE supplier SET is_active = %s WHERE id = %s", (new_status, id))
        mysql.connection.commit()
        status_msg = "diaktifkan" if new_status else "dinonaktifkan"
        flash(f"Supplier berhasil {status_msg}.", "success")

    cur.close()
    return redirect(url_for('supplier'))

```

Fungsi ini digunakan untuk mengaktifkan atau menonaktifkan supplier berdasarkan id. Sistem mengambil status aktif saat ini (is_active), lalu membalikinya (aktif jadi nonaktif atau sebaliknya), menyimpannya kembali ke database, dan menampilkan pesan berhasil. Hanya admin yang bisa mengakses fungsi ini. Setelah itu, pengguna diarahkan kembali ke halaman supplier.

```

415 #produk
416 @app.route('/produk')
417 @login_required(['admin'])
418 def produk():
419     cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
420
421     # Ambil parameter pencarian dan halaman
422     search = request.args.get('search', '').strip()
423     page = request.args.get('page', default=1, type=int)
424     per_page = 10
425     offset = (page - 1) * per_page
426     search_term = f"%{search}%"
427
428     # Hitung total data untuk pagination (hanya supplier aktif)
429     count_query = """
430     SELECT COUNT(*) AS total
431     FROM produk
432     JOIN supplier ON produk.supplier_id = supplier.id
433     WHERE (produk.nama LIKE %s OR supplier.nama LIKE %s)
434     AND supplier.is_active = TRUE
435     """
436     cur.execute(count_query, (search_term, search_term))
437     total_rows = cur.fetchone()['total']
438     total_pages = (total_rows + per_page - 1) // per_page
439
440     # Ambil data produk (hanya yang supplier-nya aktif)
441     query = """
442     SELECT produk.id, produk.nama, produk.harga_beli, produk.harga_jual, supplier.nama AS supplier_nama
443     FROM produk
444     JOIN supplier ON produk.supplier_id = supplier.id
445     WHERE (produk.nama LIKE %s OR supplier.nama LIKE %s)
446     AND supplier.is_active = TRUE
447     ORDER BY produk.nama ASC
448     LIMIT %s OFFSET %s
449     """
450     cur.execute(query, (search_term, search_term, per_page, offset))
451     products = cur.fetchall()
452     cur.close()
453
454     return render_template('admin/produk/produk.html',
455                           products=products,
456                           search=search,
457                           page=page,
458                           total_pages=total_pages,

```

Fungsi ini menampilkan halaman daftar produk untuk admin dengan fitur pencarian dan pagination. Data produk diambil dari database, hanya yang berasal dari supplier yang aktif, dan bisa dicari berdasarkan nama produk atau nama supplier. Hasilnya ditampilkan ke halaman produk.html bersama info halaman dan kata kunci pencarian.

```

#Tambah Produk
@app.route('/produk/tambah', methods=['GET', 'POST'])
@login_required(['admin'])
def tambah_produk():
    if request.method == 'POST':
        nama = request.form['nama']
        supplier_id = request.form['supplier_id']
        harga_beli = request.form['harga_beli']
        harga_jual = request.form['harga_jual']

        # Simpan ke database
        cur = mysql.connection.cursor()
        cur.execute("""
        INSERT INTO produk (nama, supplier_id, harga_beli, harga_jual)
        VALUES (%s, %s, %s, %s)
        """, (nama, supplier_id, harga_beli, harga_jual))
        mysql.connection.commit()
        cur.close()

        flash('Produk berhasil ditambahkan', 'success')
        return redirect(url_for('produk')) # arahkan ke daftar produk

```

Fungsi ini menangani penambahan produk baru oleh admin. Jika form dikirim (POST), data produk disimpan ke database, lalu pengguna diarahkan ke halaman daftar produk dengan pesan sukses. Saat halaman dibuka (GET), sistem

menampilkan kode produk otomatis dan daftar supplier yang aktif untuk dipilih, lalu merender halaman tambah.html.

```
#edit_produk
@app.route('/produk/edit/<int:produk_id>', methods=['GET', 'POST'])
@login_required(['admin'])
def edit_produk(produk_id):
    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    if request.method == 'POST':
        nama = request.form['nama']
        supplier_id = request.form['supplier_id']
        harga_beli = int(request.form['harga_beli'])
        harga_jual = int(request.form['harga_jual'])

        cur.execute("""
            UPDATE produk SET nama=%s, supplier_id=%s, harga_beli=%s, harga_jual=%s WHERE id=%s
            """, (nama, supplier_id, harga_beli, harga_jual, produk_id))
        mysql.connection.commit()
        flash('Produk berhasil diperbarui.', 'success')
        return redirect(url_for('produk'))

    # Ambil data produk dan supplier untuk ditampilkan di form edit
    cur.execute("SELECT * FROM produk WHERE id = %s", (produk_id,))
    produk = cur.fetchone()
    cur.execute("SELECT id, nama FROM supplier WHERE is_active = TRUE")
    suppliers = cur.fetchall()

    # Kode otomatis produk (untuk tampilan saja)
    cur.execute("SELECT COUNT(*) AS total FROM produk")
    count = cur.fetchone()['total']
    kode_produk = f"{count + 1:03d}"

    cur.close()

    return render_template('admin/produk/edit.html', produk=produk, suppliers=suppliers, kode_produk=kode_produk, current_path=request.path)
```

Fungsi ini digunakan untuk mengedit data produk berdasarkan produk_id, dan hanya bisa diakses oleh admin. Jika form disubmit (POST), data produk akan diperbarui di database dan ditampilkan pesan sukses. Jika halaman dibuka (GET), sistem mengambil data produk dan daftar supplier aktif untuk ditampilkan di form edit. Kode produk otomatis juga dibuat untuk tampilan. Semua data dikirim ke template edit.html.

```

#pembelian
@app.route('/pembelian')
@login_required(['admin'])
def daftar_pembelian():
    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    # Ambil parameter pencarian dan halaman
    search = request.args.get('search', '').strip()
    page = request.args.get('page', default=1, type=int)
    per_page = 10
    offset = (page - 1) * per_page

    # Hitung total data untuk pagination
    count_query = """
        SELECT COUNT(*) AS total
        FROM pembelian p
        JOIN supplier s ON p.supplier_id = s.id
        WHERE (p.no_pembelian LIKE %s OR s.nama LIKE %s)
        AND s.is_active = TRUE
    """

    search_term = f"%{search}%"
    cur.execute(count_query, (search_term, search_term))
    total_rows = cur.fetchone()['total']
    total_pages = (total_rows + per_page - 1) // per_page

    # Ambil data sesuai halaman dan pencarian
    query = """
        SELECT p.id, p.no_pembelian, p.tanggal, p.total, p.nota, p.status, s.nama AS nama_supplier
        FROM pembelian p
        JOIN supplier s ON p.supplier_id = s.id
        WHERE (p.no_pembelian LIKE %s OR s.nama LIKE %s)
        AND s.is_active = TRUE
        ORDER BY p.tanggal DESC
        LIMIT %s OFFSET %s
    """

    cur.execute(query, (search_term, search_term, per_page, offset))
    pembeliangs = cur.fetchall()
    cur.close()

    for p in pembeliangs:
        if isinstance(p['tanggal'], str):
            p['tanggal'] = datetime.datetime.strptime(p['tanggal'], '%Y-%m-%d')

    return render_template("admin/pembelian/daftar_pembelian.html",
        pembeliangs=pembeliangs,
        current_path=request.path,
        search=search,
        page=page,
        total_pages=total_pages)

```

Fungsi ini menampilkan daftar pembelian untuk admin dengan fitur pencarian dan pagination. Data pembelian diambil dari database berdasarkan nomor pembelian atau nama supplier, hanya jika supplier-nya aktif. Data diformat dan ditampilkan di halaman `daftar_pembelian.html` bersama info pencarian, halaman aktif, dan total halaman.

```

app.py > ...
596 #tambah pembelian
597 @app.route('/pembelian/tambah', methods=['GET', 'POST'])
598 @login_required(['admin'])
599 def tambah_pembelian():
600     if request.method == 'POST':
601         no_pembelian = request.form['no_pembelian']
602         tanggal = request.form['tanggal']
603         supplier_id = request.form['supplier_id']
604
605         # Ambil nilai total dalam format decimal asli
606         total_harga = float(request.form.get('total_harga_real', 0))
607
608         nota = request.files['nota']
609         nota_filename = None
610         if nota and nota.filename != '':
611             nota_filename = str(uuid.uuid4()) + "." + nota.filename
612             nota.save(os.path.join(app.config['UPLOAD_FOLDER'], nota_filename))
613
614         cur = mysql.connection.cursor()
615         cur.execute("""
616             INSERT INTO pembelian (no_pembelian, tanggal, supplier_id, total, nota)
617             VALUES (%s, %s, %s, %s, %s)
618             """, (no_pembelian, tanggal, supplier_id, total_harga, nota_filename))
619         pembelian_id = cur.lastrowid
620
621         produk_ids = request.form.getlist('produk_id[]')
622         hargas = request.form.getlist('harga_real[]')
623         jumlabs = request.form.getlist('jumlah[]')
624         totals = request.form.getlist('total_real[]')
625
626         for i in range(len(produk_ids)):
627             cur.execute("""
628                 INSERT INTO pembelian_detail (pembelian_id, produk_id, harga, jumlah, total)
629                 VALUES (%s, %s, %s, %s, %s)
630                 """, (
631                     pembelian_id,
632                     produk_ids[i],
633                     float(hargas[i]),
634                     int(jumlabs[i]),
635                     float(totals[i])
636                 ))

```

Fungsi ini menangani penambahan data pembelian oleh admin. Saat form dikirim (POST), data seperti nomor pembelian, tanggal, supplier, total harga, dan

file nota disimpan ke tabel pembelian. Lalu, detail setiap produk (ID, harga, jumlah, total) disimpan ke tabel pembelian_detail. Jika ada file nota, file tersebut diunggah ke folder yang telah dikonfigurasi.

```
#detail pembelian
@app.route('/pembelian/<no_pembelian>')
@login_required(['admin'])
def detail_pembelian(no_pembelian):
    cur = mysql.connection.cursor(DictCursor)

    # Ambil data pembelian utama
    cur.execute("""
        SELECT p.no_pembelian, p.tanggal, s.nama AS nama_supplier, p.nota
        FROM pembelian p
        JOIN supplier s ON p.supplier_id = s.id
        WHERE p.no_pembelian = %s
    """, (no_pembelian,))
    pembelian = cur.fetchone()

    # Ambil detail produk yang dibeli
    cur.execute("""
        SELECT pr.nama AS nama_produk, dp.jumlah, dp.harga AS harga_satuan, dp.total
        FROM pembelian_detail dp
        JOIN produk pr ON dp.produk_id = pr.id
        JOIN pembelian p ON dp.pembelian_id = p.id
        WHERE p.no_pembelian = %s
    """, (no_pembelian,))
    detail_items = cur.fetchall()

    cur.close()

    return render_template("admin/pembelian/pembelian_detail.html", pembelian=pembelian, detail_items=detail_items, current_path=request.path)
```

Fungsi detail_pembelian bertujuan untuk menampilkan informasi detail dari suatu transaksi pembelian berdasarkan no_pembelian. Pertama, fungsi ini mengambil data pembelian utama seperti nomor pembelian, tanggal pembelian, nama supplier, dan file nota dengan melakukan join antara tabel pembelian dan supplier. Setelah itu, fungsi mengambil daftar produk yang dibeli dalam transaksi tersebut, termasuk nama produk, jumlah yang dibeli, harga satuan, dan total harga, dengan melakukan join antara tabel pembelian_detail, produk, dan pembelian. Setelah seluruh data berhasil diambil, koneksi database ditutup, dan data tersebut dikirim ke template pembelian_detail.html untuk ditampilkan kepada pengguna.

```

# 1. Validasi stok terlebih dahulu
for i in range(len(produk_ids)):
    produk_id = int(produk_ids[i])
    jumlah = int(jumlahs[i])

    cur.execute("SELECT jumlah FROM stok WHERE produk_id = %s", (produk_id,))
    stok = cur.fetchone()
    if not stok or stok[0] < jumlah:
        nama_produk = None
        cur.execute("SELECT nama FROM produk WHERE id = %s", (produk_id,))
        result = cur.fetchone()
        if result:
            nama_produk = result[0]
        mysql.connection.rollback()
        flash("Stok tidak mencukupi untuk produk: (nama_produk or 'ID ' + str(produk_id))", 'danger')
        return redirect(url_for('tambah_penjualan'))

# 2. Jika stok cukup, simpan ke penjualan
cur.execute("""
INSERT INTO penjualan (no_penjualan, tanggal, total)
VALUES (%s, %s, %s)
""", (no_penjualan, tanggal, total_harga))
penjualan_id = cur.lastrowid

# 3. Simpan detail & update stok
for i in range(len(produk_ids)):
    produk_id = int(produk_ids[i])
    harga = float(hargas[i])
    jumlah = int(jumlahs[i])
    total = float(totals[i])

    cur.execute("""
INSERT INTO penjualan_detail (penjualan_id, produk_id, harga, jumlah, total)
VALUES (%s, %s, %s, %s, %s)
""", (penjualan_id, produk_id, harga, jumlah, total))

# Kurangi stok
cur.execute("""
UPDATE stok
SET jumlah = jumlah - %s
WHERE produk_id = %s
""", (jumlah, produk_id))

# Ambil supplier_id dari produk
cur.execute("SELECT supplier_id FROM produk WHERE id = %s", (produk_id,))
supplier = cur.fetchone()
supplier_id = supplier[0] if supplier else None

# Simpan ke log_stok
cur.execute("""
INSERT INTO log_stok (produk_id, supplier_id, jumlah, tipe, keterangan)
VALUES (%s, %s, %s, 'pengurangan', %s)

```

Kode ini menangani proses penyimpanan data penjualan dengan tiga tahap utama:

1. Validasi Stok: Mengecek apakah stok produk mencukupi sebelum penjualan. Jika stok tidak cukup, transaksi dibatalkan, rollback dilakukan, dan pengguna diberi peringatan.
2. Simpan Penjualan: Jika stok mencukupi, data penjualan utama (nomor, tanggal, total harga) disimpan ke tabel penjualan.
3. Simpan Detail & Update Stok: Setiap item produk penjualan disimpan ke tabel penjualan_detail, stok produk dikurangi sesuai jumlah yang dijual, dan catatan pengurangan stok dicatat ke tabel log_stok sebagai log aktivitas stok.

```

# Route untuk menyetujui pembelian dan menambahkan stok serta mencatat log
@app.route('/pembelian/setujui/<no_pembelian>', methods=['POST'])
@login_required(['owner'])
def setujui_pembelian(no_pembelian):
    cur = mysql.connection.cursor(DictCursor)

    # Ambil data pembelian
    cur.execute("SELECT id, supplier_id FROM pembelian WHERE no_pembelian = %s", (no_pembelian,))
    pembelian = cur.fetchone()
    if not pembelian:
        flash("Pembelian tidak ditemukan.", "danger")
        return redirect(url_for('daftar_pembelian'))

    pembelian_id = pembelian['id']
    supplier_id = pembelian['supplier_id']

    # Ambil detail pembelian
    cur.execute("""
        SELECT produk_id, jumlah FROM pembelian_detail
        WHERE pembelian_id = %s
        """, (pembelian_id,))
    detail_items = cur.fetchall()

    for item in detail_items:
        produk_id = item['produk_id']
        jumlah = item['jumlah']

        # Cek apakah stok dengan produk_id dan supplier_id sudah ada
        cur.execute("""
            SELECT id, jumlah FROM stok
            WHERE produk_id = %s AND supplier_id = %s
            """, (produk_id, supplier_id))
        stok = cur.fetchone()

```

Kode ini digunakan untuk menyetujui pembelian oleh pengguna dengan level owner, prosesnya mencakup :

1. Mengambil Data Pembelian: Mencari pembelian berdasarkan no_pembelian. Jika tidak ditemukan, pengguna diarahkan kembali dengan pesan kesalahan.
2. Mengambil Detail Pembelian: Mengambil semua produk dan jumlahnya dari tabel pembelian_detail.
3. Update atau Tambah Stok: Untuk setiap produk, dicek apakah stok dengan produk_id dan supplier_id sudah ada. Jika ada, jumlah stok ditambah sesuai jumlah pembelian.

```

@app.route('/laporan/pembelian', methods=['GET'])
@login_required(['owner'])
def laporan_pembelian():
    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    dari_tanggal = request.args.get('dari_tanggal')
    sampai_tanggal = request.args.get('sampai_tanggal')
    page = int(request.args.get('page', 1))
    per_page = 10
    offset = (page - 1) * per_page

    query_conditions = " AND s.is_active = 1"
    params = []

    if dari_tanggal and sampai_tanggal:
        query_conditions += " AND p.tanggal BETWEEN %s AND %s"
        params.extend([dari_tanggal, sampai_tanggal])

    # Hitung total data
    count_query = f"""
        SELECT COUNT(*) as total
        FROM pembelian p
        JOIN pembelian_detail pd ON p.id = pd.pembelian_id
        JOIN produk pr ON pd.produk_id = pr.id
        LEFT JOIN supplier s ON p.supplier_id = s.id
        WHERE 1=1 {query_conditions}
    """

    cur.execute(count_query, params)
    total_data = cur.fetchone()['total']
    total_pages = (total_data + per_page - 1) // per_page

```

Fungsi `laporan_pembelian()` adalah route yang digunakan untuk menampilkan laporan pembelian kepada pengguna dengan level akses "owner".

Fungsi ini dimulai dengan mengambil parameter filter berupa `dari_tanggal` dan `sampai_tanggal` dari URL, yang berfungsi untuk menyaring data pembelian berdasarkan rentang tanggal tertentu. Selain itu, fungsi ini juga menerapkan sistem paginasi, dengan jumlah data per halaman sebanyak 10 item, dan menghitung offset berdasarkan halaman yang sedang dibuka. Kemudian, fungsi menyusun query SQL dengan kondisi tambahan seperti tanggal dan status supplier yang aktif, serta menghitung total jumlah data pembelian yang sesuai dengan filter tersebut. Proses ini penting untuk menghasilkan tampilan laporan pembelian yang terstruktur dan mudah dibaca sesuai dengan rentang waktu yang diinginkan pengguna.

```
@app.route('/laporan/pembelian/export')
@login_required(['owner'])
def export_pdf_pembelian():
    dari_tanggal = request.args.get('dari_tanggal')
    sampai_tanggal = request.args.get('sampai_tanggal')

    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    query = """
        SELECT
            p.no_pembelian,
            p.tanggal,
            pr.nama AS nama_produk,
            pd.jumlah,
            pd.total AS jumlah_total,
            s.nama AS nama_supplier,
            p.status
        FROM pembelian p
        JOIN pembelian_detail pd ON p.id = pd.pembelian_id
        JOIN produk pr ON pd.produk_id = pr.id
        JOIN supplier s ON p.supplier_id = s.id
        WHERE s.is_active = 1
    """

    params = []

    if dari_tanggal and sampai_tanggal:
        query += " AND p.tanggal BETWEEN %s AND %s"
        params.extend([dari_tanggal, sampai_tanggal])

    query += " ORDER BY p.tanggal DESC"
    cur.execute(query, params)
    pembelian = cur.fetchall()
```

Fungsi `export_pdf_pembelian()` digunakan untuk mengekspor data pembelian ke dalam format PDF berdasarkan rentang tanggal yang dipilih. Fungsi ini hanya dapat diakses oleh pengguna dengan level "owner". Pertama, fungsi mengambil parameter tanggal awal (`dari_tanggal`) dan akhir (`sampai_tanggal`) dari URL, lalu menyiapkan query SQL untuk mengambil data pembelian yang aktif dari

database, termasuk informasi seperti nomor pembelian, tanggal, produk, jumlah, total harga, supplier, dan status. Jika filter tanggal diberikan, query akan membatasi data berdasarkan rentang tersebut. Data yang diambil akan digunakan untuk membuat laporan pembelian dalam bentuk PDF.

```
@app.route('/penyesuaian_stok', methods=['GET'])
@login_required(['owner'])
def penyesuaian_stok():
    supplier_id = request.args.get('supplier_id')
    search_query = request.args.get('search', '')
    page = request.args.get('page', default=1, type=int)
    per_page = 10
    offset = (page - 1) * per_page

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    # Ambil semua supplier yang aktif saja
    cursor.execute("SELECT id, nama FROM supplier WHERE is_active = 1")
    suppliers = cursor.fetchall()

    # Hitung total data stok
    count_query = """
        SELECT COUNT(DISTINCT produk.id) AS total
        FROM produk
        LEFT JOIN stok ON produk.id = stok.produk_id
        LEFT JOIN supplier ON produk.supplier_id = supplier.id
        WHERE supplier.is_active = 1
    """
    count_params = []

    if supplier_id and supplier_id.isdigit():
        count_query += " AND produk.supplier_id = %s"
        count_params.append(supplier_id)

    if search_query:
        count_query += " AND produk.nama LIKE %s"
        count_params.append(f"%{search_query}%")

    cursor.execute(count_query, count_params)
    total_data = cursor.fetchone()['total']
    total_pages = (total_data + per_page - 1) // per_page
```

Fungsi `penyesuaian_stok()` menampilkan halaman penyesuaian stok untuk pengguna owner. Fungsi ini mendukung filter berdasarkan supplier dan pencarian nama produk, serta menggunakan paginasi. Sistem menghitung jumlah total produk yang stoknya akan ditampilkan, hanya dari supplier yang aktif.