

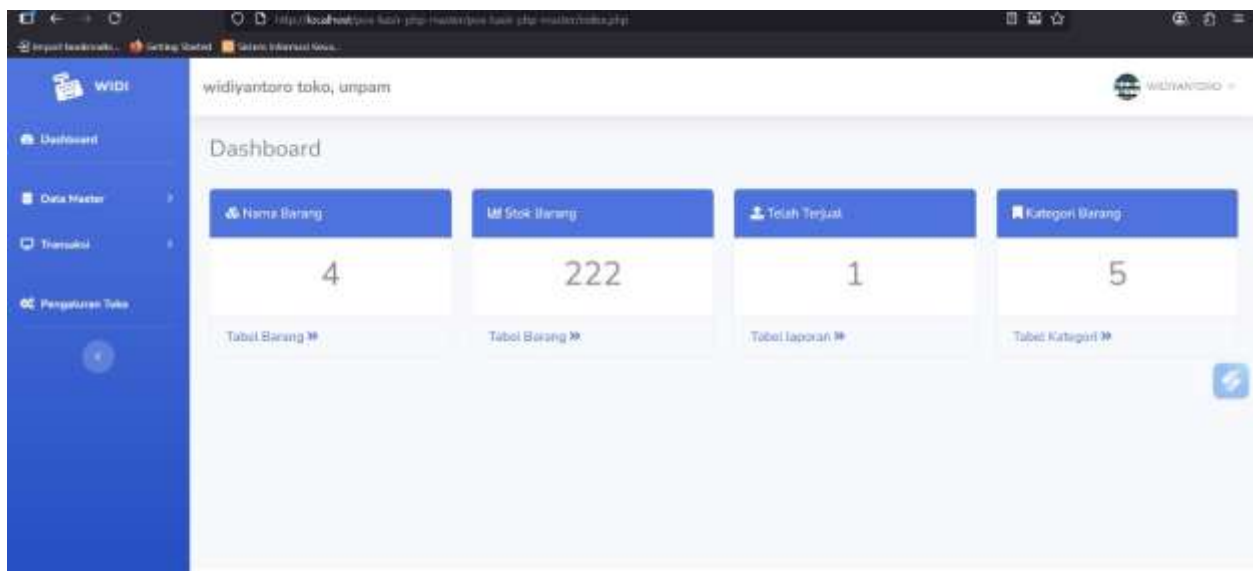
Nama : WIDI YANTORO
NIM : 221011403044
no. peserta : 250917070169

link : <https://github.com/BOYWIDI/website-penjualan-user-addmin-.git>

USER LOGIN



DASHBOARD



INPUT BARANG

+ Tambah Barang

ID Barang: BR005

Kategori: ATK

Nama Barang: aki

Merk Barang: gs. astra

Harga Beli: 12000

Harga Jual: 13000

Satuan Barang: Pilih Satuan

Stock: 2

Tanggal Input: 21 September 2025, 18:43

+ Insert Data **Close**

No.	ID Barang	Kategori	Nama Barang	Merk	Stock	Harga Beli	Harga Jual	Satuan	Aksi
1.	BR004	Sabun	Sabun	detto	18	Rp.1.200,-	Rp.1.300,-	PCS	Detail Hapus
2.	BR005	ATK	aki	gs. astra	2	Rp.12.000,-	Rp.13.000,-	#	Detail Hapus
3.	BR003	ATK	Pulpen	Standard	70	Rp.1.500,-	Rp.2.000,-	PCS	Detail Hapus
4.	BR002	Sabun	Sabun	Lifebuoy	38	Rp.1.800,-	Rp.3.000,-	PCS	Detail Hapus
5.	BR001	ATK	Pensil	Faber Castell	88	Rp.1.500,-	Rp.3.000,-	PCS	Detail Hapus
Total					224	Rp.363.600,-	Rp.594.600,-		

HASIL DATA BARANG YANG SUDAH DI INPUT

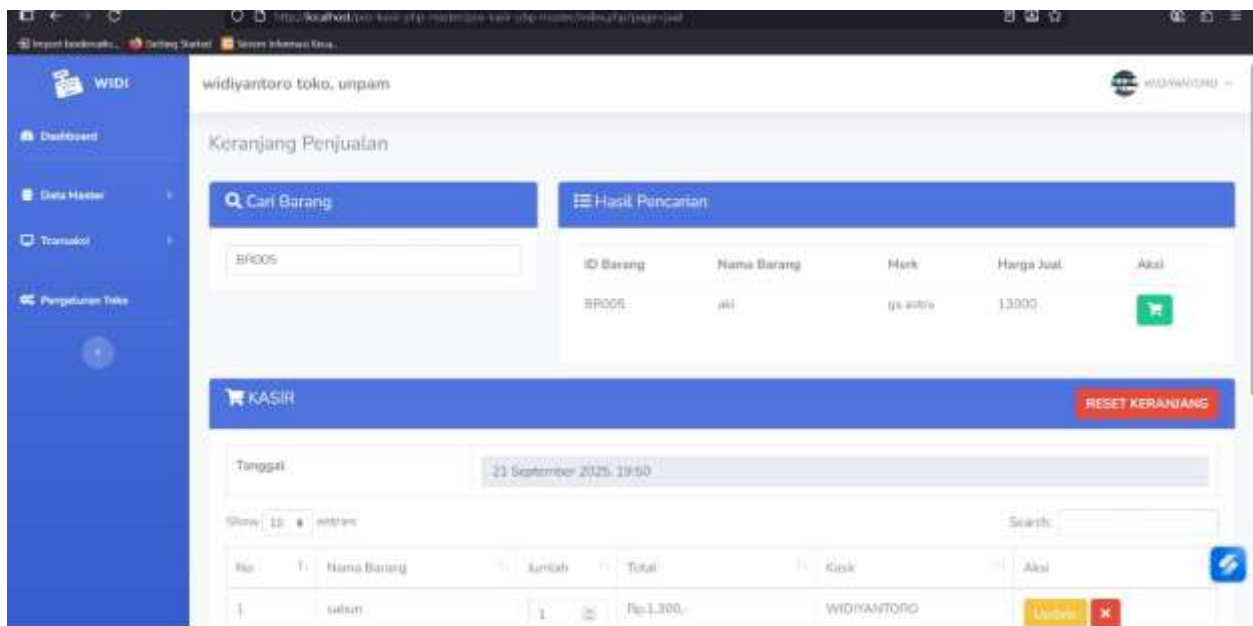
Ada 1 barang yang Stok terendah sudah kurang dari 3 item, silahkan pesan lagi! **Cek Barang**

+ Insert Data **+ Refresh Data** **Refresh Data**

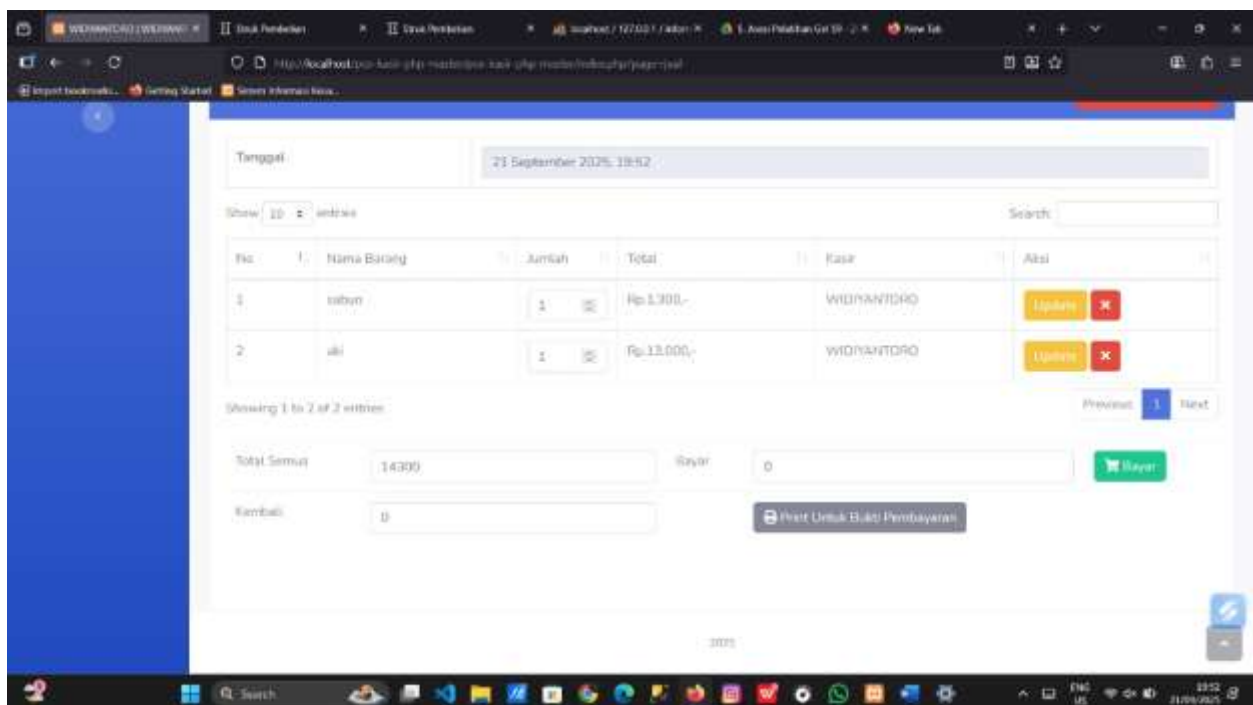
Show: 10 entries

No.	ID Barang	Kategori	Nama Barang	Merk	Stock	Harga Beli	Harga Jual	Satuan	Aksi
1.	BR005	ATK	aki	gs. astra	2	Rp.12.000,-	Rp.13.000,-	#	Detail Hapus
2.	BR004	Sabun	Sabun	detto	18	Rp.1.200,-	Rp.1.300,-	PCS	Detail Hapus
3.	BR003	ATK	Pulpen	Standard	70	Rp.1.500,-	Rp.2.000,-	PCS	Detail Hapus
4.	BR002	Sabun	Sabun	Lifebuoy	38	Rp.1.800,-	Rp.3.000,-	PCS	Detail Hapus
5.	BR001	ATK	Pensil	Faber Castell	88	Rp.1.500,-	Rp.3.000,-	PCS	Detail Hapus
Total					224	Rp.363.600,-	Rp.594.600,-		

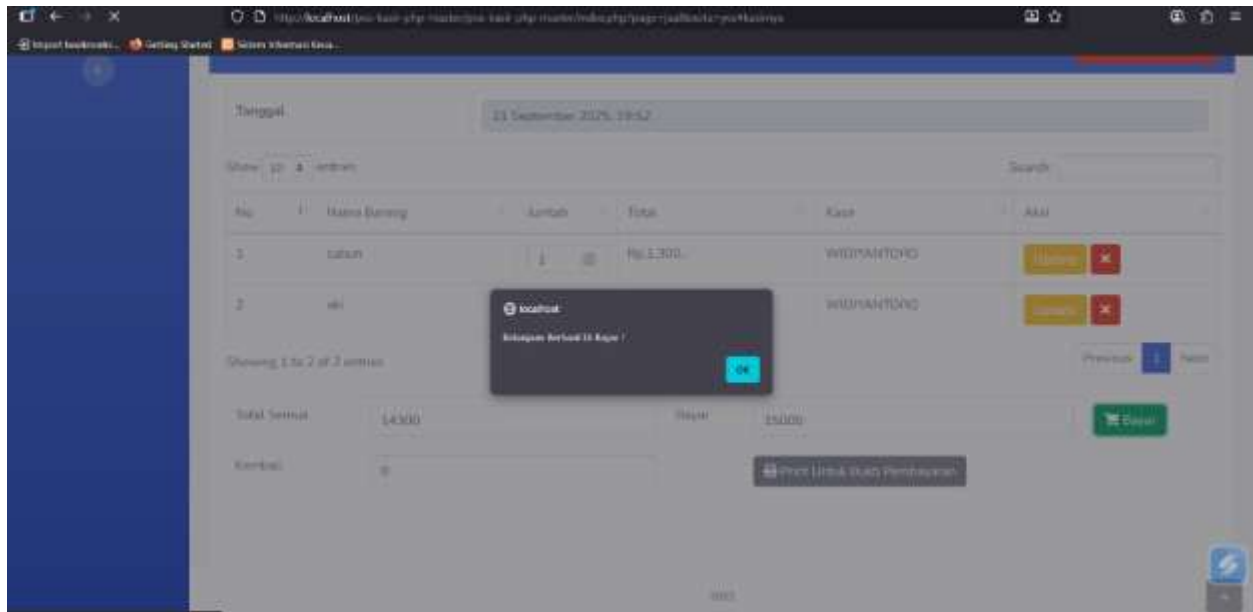
MASUKKAN BARANG KE KERANJANG



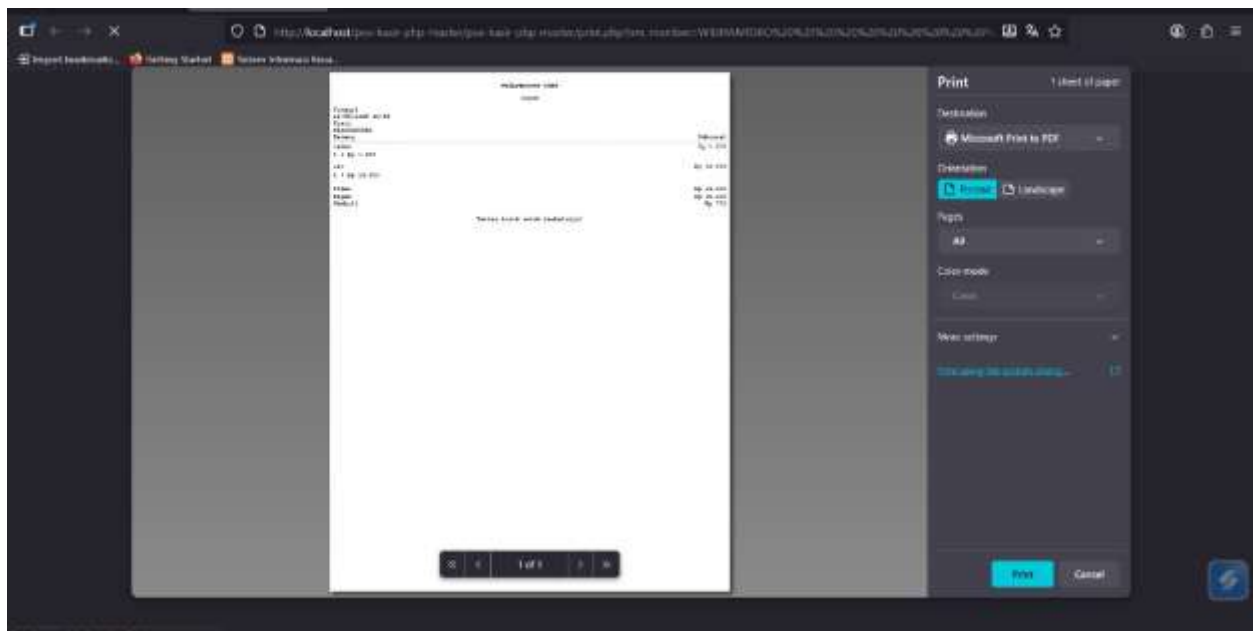
GAMBAR SETELAH BARANG DI LETAKKAN DI KERANJANG



PEMBAYARAN BERHASIL



STRUK BELANJA



TAMPILAN LAPORAN PEMBELIAN

Cari Laporan Per Bulan

Pilih Bulan: Pilih Tahun: Aksi:

Pilih Hari: Aksi:

Showing 1 to 3 of 3 entries

No	ID Barang	Nama Barang	Jumlah	Modal	Total	Kasir	Tanggal Input
1	BR005	sabun	1	Rp.12,000,-	Rp.13,000,-	WIDRYANTORO	21 September 2025, 18:52
2	BR004	sabun	1	Rp.1,300,-	Rp.1,300,-	WIDRYANTORO	21 September 2025, 18:01
3	BR004	sabun	1	Rp.1,300,-	Rp.1,300,-	WIDRYANTORO	21 September 2025, 18:01
Total Terjual			3	Rp.14,400,-	Rp.15,600,-	Keuntungan	Rp.1,200,-

EXCEL DATA LAPORAN HARIAN

Halaman: 1 dari 1

Data Laporan Penjualan 2025-09-21

No	ID Barang	Nama Barang	Jumlah	Modal	Total	Kasir	Tanggal Input
1	BR004	sabun	1	Rp.1,300,-	Rp.		

Black Box Testing

Nama: WIDIYANTORO

NIM: 221011403044

No. Peserta: 250917070169

No .	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1.1	Login ke sistem.	Masukkan username "admin" dan password "..." , lalu klik tombol "Sign In" .	Berhasil masuk ke halaman Dashboard dengan nama pengguna WIDIYANTORO ditampilkan.	Berhasil masuk ke halaman Dashboard, nama pengguna WIDIYANTORO terlihat di kanan atas.	Diterima
1.2	Cek Dashboard.	Navigasi ke halaman Dashboard .	Dashboard menampilkan informasi ringkas berupa jumlah Nama Barang, Stok Barang, Barang Terjual , dan Kategori Barang .	Dashboard menampilkan jumlah Nama Barang (4), Stok Barang (222), Barang Terjual (1) , dan Kategori Barang (5) .	Diterima
1.3	Tambah Barang.	1. Navigasi ke Data Master	Data barang baru berhasil	Data barang dengan ID	Diterima

No .	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
		<p>> Data Barang. 2. Klik tombol " +Insert Data". 3. Isi semua form (ID Barang, Kategori, Nama Barang, dll.) dengan data baru. 4. Klik tombol " +Insert Data" pada pop-up.</p>	<p>ditambahkan dan muncul di daftar Data Barang.</p>	<p>BR005 dan Nama Barang "aki" berhasil ditambahkan ke dalam daftar.</p>	
1.4	Memperbarui Stok Barang.	<p>1. Navigasi ke Data Master > Data Barang. 2. Cari barang dengan stok rendah, contohnya "aki". 3. Klik tombol "Restok".</p>	<p>Stok barang berhasil diperbarui sesuai dengan jumlah yang dimasukkan.</p>	<p>Terdapat notifikasi bahwa ada 1 barang dengan stok kurang dari 3 item, yaitu "aki".</p>	Diterima
1.5	Tambah ke Keranjang Penjualan.	<p>1. Navigasi ke Transaksi > Struk Penjualan. 2. Masukkan ID</p>	<p>Barang berhasil ditambahkan ke dalam</p>	<p>Barang "aki" dengan ID BR005 berhasil</p>	Diterima

No .	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
		Barang "BR005" ke kolom "Cari Barang" . 3. Klik tombol keranjang hijau.	keranjang penjualan.	masuk ke keranjang.	
1.6	Melakukan Pembayaran.	1. Navigasi ke Transaksi > Struk Penjualan . 2. Masukkan jumlah bayar di kolom "Bayar" . 3. Klik tombol "Bayar" .	Muncul pop-up konfirmasi "Belanjaan Berhasil Di Bayar!" . Jumlah kembalian terhitung dengan benar.	Muncul pop-up "Belanjaan Berhasil Di Bayar!" . Total belanja Rp. 14.300 dan bayar Rp. 15.000 , kembalian terhitung Rp. 700 .	Diterima
1.7	Cetak Bukti Pembayaran.	1. Setelah transaksi berhasil, klik tombol "Print Untuk Bukti Pembayaran" .	Muncul halaman pratinjau cetak struk pembayaran.	Halaman pratinjau struk pembayaran muncul.	Diterima
1.8	Cek Laporan Penjualan.	1. Navigasi ke Transaksi > Laporan . 2. Pilih tanggal	Transaksi yang baru dilakukan muncul dalam laporan	Transaksi "aki" dan "sabun" muncul dalam	Diterima

No .	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
		dan klik tombol "Cari" .	penjualan harian. Total modal dan total terjual terhitung dengan benar.	laporan. Total Terjual 3 item, Total Modal Rp. 14.400 , Keuntungan Rp. 1.200 .	
1.9	Ekspor Laporan ke Excel.	1. Navigasi ke Transaksi > Laporan . 2. Pilih tanggal dan klik tombol "Cari" . 3. Klik tombol "Excel" .	File laporan penjualan harian dalam format Excel berhasil diunduh.	File Excel berjudul "data-laporan-penjualan-2025-09-21.xlsx" berhasil diunduh dan menampilkan data penjualan yang relevan.	Diterima

Blackbox Testing (Lanjutan)

No.	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
2.3	Mengonfirmasi Transaksi.	Klik tombol "Confirm" .	1. Status transaksi berubah menjadi	Status transaksi ter-update. Cek di	Diterima

No.	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
			"Confirmed". 2. Stok Laptop Stand berkurang menjadi 15 . 3. Stok Kursi Kantor berkurang menjadi 8 .	halaman item, stok berkurang.	
2.4	Mencoba Menghapus Pelanggan.	Pelanggan: Maju Jaya .	Aksi berhasil. Data pelanggan terhapus.	Data pelanggan terhapus.	Diterima
2.5	Cek Dashboard.	Navigasi ke halaman Dashboard .	Transaksi oleh operator muncul di " Transaksi Terbaru ". Transaksi dengan No. Transaksi yang relevan muncul di bagian atas.	Transaksi dengan No. Transaksi yang relevan muncul di bagian atas.	Diterima
2.6	Hapus Transaksi.	Hapus transaksi yang dibuat sebelumnya.	1. Transaksi berhasil dihapus. 2. Stok Laptop Stand kembali menjadi 20 . 3. Stok Kursi	Transaksi hilang dari daftar. Cek di halaman item, stok kembali ke semula.	Diterima

No.	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
			Kantor kembali menjadi 10 .		
2.7	Logout.	Klik tombol "Logout" .	Berhasil keluar. Sesi terhapus dan halaman dialihkan ke login.	Berhasil, sesi terhapus dan halaman dialihkan ke login.	Diterima

TES TERTULIS

1: Pentingnya Spesifikasi Program

Spesifikasi program adalah blueprint atau cetak biru yang sangat penting dalam pengembangan perangkat lunak karena memastikan semua pihak, mulai dari pengembang, manajer proyek, hingga klien, memiliki pemahaman yang sama mengenai tujuan, fungsionalitas, dan batasan dari perangkat lunak yang akan dibangun. Tanpa spesifikasi yang jelas, proyek bisa kehilangan arah, menghasilkan produk yang tidak sesuai harapan, dan menyebabkan pemborosan waktu serta sumber daya.

Jawaban 2: Tiga Prinsip Utama Penulisan Kode yang Baik

Tiga prinsip utama yang sering dijadikan acuan dalam penulisan kode yang baik adalah DRY, KISS, dan YAGNI.

DRY (Don't Repeat Yourself): Prinsip ini menekankan untuk tidak mengulang-ulang kode yang sama. Jika Anda menemukan blok kode yang digunakan lebih dari sekali, sebaiknya kode tersebut dijadikan fungsi atau modul yang dapat dipanggil kembali. Ini membuat kode lebih mudah dirawat dan diubah.

KISS (Keep It Simple, Stupid): Prinsip ini menganjurkan untuk menjaga kode tetap sederhana dan mudah dipahami. Hindari solusi yang terlalu rumit jika ada cara yang lebih sederhana untuk menyelesaikan masalah yang sama. Kode yang sederhana cenderung memiliki lebih sedikit bug dan lebih mudah di-debug.

YAGNI (You Aren't Gonna Need It): Prinsip ini berarti jangan membuat fungsionalitas atau fitur yang belum Anda butuhkan saat ini. Fokuslah pada apa yang diperlukan sekarang. Menambahkan fitur "untuk masa depan" seringkali menghabiskan waktu dan membuat kode menjadi lebih kompleks tanpa alasan yang jelas.

3: Perbedaan Pemrograman Terstruktur dan Berorientasi Objek

Perbedaan utama antara pemrograman terstruktur dan pemrograman berorientasi objek (OOP) terletak pada cara mereka mengorganisir kode.

Pemrograman Terstruktur : Paradigma ini berfokus pada pemecahan masalah menjadi sub-program atau fungsi yang lebih kecil dan terpisah. Alur eksekusi program mengikuti urutan (sekuensial), percabangan (kondisional), dan perulangan (looping). Data dan fungsi yang memanipulasinya seringkali terpisah.

Pemrograman Berorientasi Objek (OOP): Paradigma ini mengorganisir kode berdasarkan objek, yang merupakan gabungan dari data (disebut properti atau atribut) dan fungsi (disebut

metode). OOP meniru dunia nyata di mana objek-objek berinteraksi satu sama lain. Contohnya, objek `Mobil` dapat memiliki atribut `warna` dan `kecepatan` serta metode `maju()` dan `mundur()`.

4: Konsep Utama dalam Pemrograman Berorientasi Objek (OOP)

Empat pilar utama dalam OOP adalah Enkapsulasi Abstraksi Inheritansi, dan Polimorfisme.

Enkapsulasi (Encapsulation): Konsep ini menggabungkan data dan metode yang beroperasi pada data tersebut ke dalam satu unit, yaitu objek. Enkapsulasi juga menyembunyikan detail implementasi dari luar, hanya mengekspos antarmuka yang diperlukan.

Abstraksi (Abstraction): Abstraksi fokus pada menampilkan hanya informasi yang relevan** dan menyembunyikan detail yang tidak perlu. Ini memungkinkan pengembang untuk membuat kelas atau objek yang lebih sederhana dan fokus pada fungsionalitas inti.

Inheritansi (Inheritance): Konsep ini memungkinkan sebuah kelas (kelas anak atau `subclass`) untuk mewarisi properti dan metode dari kelas lain (kelas induk atau `superclass`). Ini mempromosikan penggunaan kembali kode (`code reuse`).

Polimorfisme (Polymorphism): Polimorfisme berarti "banyak bentuk". Konsep ini memungkinkan objek dari kelas yang berbeda untuk merespons metode dengan nama yang sama secara berbeda, tergantung pada tipe objeknya.

5: Keuntungan Menggunakan Library atau Komponen Pre-existing

Menggunakan library atau komponen yang sudah ada (pre-existing) memiliki beberapa keuntungan signifikan:

Efisiensi Waktu dan Biaya Anda tidak perlu menulis kode dari nol untuk fungsionalitas umum, seperti koneksi database, validasi data, atau antarmuka pengguna. Ini mempercepat proses pengembangan secara drastis.

Kualitas dan Stabilitas: Library yang populer biasanya telah diuji secara luas oleh komunitas pengembang, sehingga lebih stabil dan memiliki bug yang lebih sedikit dibandingkan kode yang ditulis dari awal.

Dukungan Komunitas: Sebagian besar library memiliki dokumentasi yang baik dan komunitas yang aktif, sehingga memudahkan Anda menemukan solusi jika menghadapi masalah.

Jawaban 6: Langkah Dasar Mengakses Database dengan PHP dan MySQL

Berikut adalah langkah-langkah dasar untuk mengakses database MySQL menggunakan PHP:

- 1 Koneksi ke Database: Gunakan fungsi seperti ``mysqli_connect()`` untuk membuat koneksi ke server database. Anda perlu menyediakan informasi seperti host, username, password, dan nama database.
2. Menulis Query SQL: Tulis perintah SQL (Structured Query Language) yang akan Anda jalankan, seperti ``SELECT`` (untuk mengambil data), ``INSERT`` (untuk menambah data), ``UPDATE`` (untuk mengubah data), atau ``DELETE`` (untuk menghapus data).
3. Menjalankan Query: Gunakan fungsi seperti ``mysqli_query()`` untuk mengirimkan query SQL ke database.
4. Mengambil Hasil (jika ada): Jika query Anda adalah ``SELECT``, Anda perlu mengambil hasilnya menggunakan fungsi seperti ``mysqli_fetch_assoc()`` untuk memproses data baris demi baris.
5. Menutup Koneksi: Setelah selesai, penting untuk menutup koneksi database menggunakan ``mysqli_close()`` untuk membebaskan sumber daya.

7: Pentingnya Dokumentasi Kode Program

Dokumentasi kode program sangat penting karena:

Pemahaman dan Pemeliharaan: Dokumentasi yang baik membantu pengembang, termasuk pengembang di masa depan, untuk memahami bagaimana kode bekerja, apa tujuannya, dan bagaimana cara memeliharanya.

Kolaborasi Tim: Dalam tim, dokumentasi memastikan semua anggota tim memiliki pemahaman yang sama tentang arsitektur dan fungsionalitas kode, mengurangi miskomunikasi dan kesalahan.

Transisi Pengembang: Ketika seorang pengembang meninggalkan proyek, dokumentasi yang baik memastikan pengembang baru dapat mengambil alih pekerjaan dengan cepat dan efektif.

8: Teknik untuk Debugging

Debugging adalah proses menemukan dan memperbaiki bug (kesalahan) dalam kode. Beberapa teknik umum yang digunakan adalah:

Debugging dengan Print Statement: Menambahkan perintah ``print`` atau ``echo`` di berbagai bagian kode untuk melihat nilai variabel pada titik tertentu.

Menggunakan Debugger: Menggunakan alat khusus (``debugger``) yang memungkinkan pengembang untuk menjalankan kode baris demi baris, melihat nilai variabel secara real-time, dan mengatur titik henti (``breakpoint``).

Logging: Merekam informasi penting tentang jalannya program, seperti kesalahan atau status variabel, ke dalam file log untuk dianalisis nanti.

Mengisolasi Masalah: Mencegah dan menyisihkan kode yang tidak terkait untuk menemukan bagian kode mana yang menyebabkan masalah.

9: Tujuan Utama Pengujian Unit

Tujuan utama dari pengujian unit (unit testing) adalah untuk memverifikasi dan menguji bagian terkecil dari kode (biasanya sebuah fungsi atau metode) secara individual dan terisolasi.

Pengujian ini dilakukan oleh pengembang sendiri untuk memastikan bahwa setiap unit kode berfungsi dengan benar sesuai dengan yang diharapkan sebelum diintegrasikan dengan bagian lain dari perangkat lunak. Ini membantu menangkap bug di tahap awal pengembangan, yang jauh lebih mudah dan lebih murah untuk diperbaiki.