

Spesifikasi Tugas Besar Milestone 1

IF3110 Web Based Development:

Nimonspedia

Tim Asisten Lab Pemrograman 2022

Versi : 1

Tgl. Revisi Terakhir : 31 Oktober 2025

Deadline : 4 November 2025, 23.59 WIB

Revisi:

1. Perubahan deadline Tugas Besar Milestone 1

Daftar Isi

Daftar Isi	2
Deskripsi Persoalan	5
Tujuan Pembelajaran	8
Spesifikasi Sistem	9
Spesifikasi Umum	9
Tabel User	10
Tabel Store	11
Tabel Product	11
Tabel Cart_Item	12
Tabel Category	12
Tabel Category_Item	12
Tabel Order	12
Penjelasan Status	13
Tabel Order_Items	14
User Story Buyer	16
1. Autentikasi dan Profil	16
2. Balance Management	16
3. Product Discovery	16
4. Shopping Cart	16
User Story Seller	18
1. Autentikasi dan Profil	18
2. Product Management	18
3. Order Management	18
Spesifikasi Fitur	19
1. Autentikasi pengguna	19
2. Navigation bar	19
3. Penyimpanan File Binary	20
4. Pagination	20
5. Rich Text Editor	21
6. Keamanan	21
7. Error handling dan State Indicator	21
8. Responsivitas	22
9. Docker	22

Spesifikasi Tiap Halaman	23
1. Halaman Login (Buyer & Seller)	23
Acceptance Criteria:	23
Things to Explore:	23
2. Halaman Register (Buyer & Seller)	23
Acceptance Criteria:	24
Things to Explore:	24
3. Halaman Product Discovery / Home (Buyer)	24
Acceptance Criteria:	25
Things to Explore:	25
4. Halaman Detail Produk (Buyer)	25
Acceptance Criteria:	26
Things to Explore:	26
5. Halaman Detail store (Buyer)	26
6. Halaman Shopping Cart (Buyer)	26
Acceptance Criteria:	27
Things to Explore:	27
7. Halaman Checkout (Buyer)	27
Acceptance Criteria:	28
8. Halaman Order History (Buyer)	28
Acceptance Criteria:	28
9. Halaman Profile (Buyer)	29
Acceptance Criteria:	29
Things to Explore:	29
10. Halaman Dashboard (Seller)	30
Acceptance Criteria:	30
11. Halaman Product Management (Seller)	30
Acceptance Criteria:	30
Things to Explore:	31
12. Halaman Add Product (Seller)	31
Acceptance Criteria:	31
Things to Explore:	31
13. Halaman Edit Product (Seller)	32
Things to Explore:	32
14. Halaman Order Management (Seller)	32
Acceptance Criteria:	32

Things to Explore:	33
Spesifikasi Bonus	34
1. All Responsive Web Design	34
2. UI/UX Seperti Tokopedia	34
3. Data Export	34
4. Advanced Search	34
5. Google Lighthouse	35
Keyword & Tips	36
Lain-Lain	38
Responsi	38
Daftar Pertanyaan	38
Deliverables	39
Pengumpulan Tugas	39
Pembagian Kelompok	40
Pembagian Tugas	40
Catatan Tambahan	41

Deskripsi Persoalan



"Nimonspedia, marketplace baru pengganti si yang tidak berorientasi pada profit, tetapi pada kebahagiaan seluruh umat Nimons di bumi"

Di balik bayangan kota metropolitan yang penuh warna dan kadang-kadang sedikit kacau balau, hiduplah Gro, sang dalang genius yang kini bertekad untuk membawa inovasi digital ke dunia para Nimons. Gro, bersama dengan tangan kanannya yang setia, Kebin, Stewart, Pop, dan Toto yang lincah, selalu memiliki ide-ide besar yang bisa mengubah segalanya. Kali ini, idenya lebih besar dari sekadar mencuri bulan – ini tentang membangun sebuah *e-commerce*!

Gro melihat potensi yang tak terbatas dalam semangat para Nimons untuk berkreasi dan bertukar barang. Dari pisang paling segar yang dipetik Pop, hingga penemuan-penemuan unik Dr. Neroifa yang kadang-kadang berfungsi, Nimons selalu memiliki sesuatu untuk ditawarkan. Namun, dunia Nimons masih sangat tradisional. Mereka masih bergantung pada pasar fisik yang ramai, di mana Kebin harus berebut tempat dengan Nimons lain untuk menjual jubah supermini, atau Stewart kesulitan menemukan gitar listrik bekas yang langka. Luyi, agen rahasia yang kini fokus pada logistik, sering kebingungan melacak pengiriman di tengah kekacauan.

Inilah mengapa Gro mencetuskan **Nimonspedia** – sebuah platform *e-commerce* ambisius yang dirancang untuk menjadi pusat perdagangan digital di seluruh dunia Nimons. Terinspirasi dari kesuksesan marketplace hijau yang telah menjadi bagian dari kehidupan sehari-hari di dunia nyata, Gro ingin menghadirkan pengalaman serupa untuk para Nimons. Bayangkan sebuah tempat di mana setiap Nimons, dari penjual pisang keliling hingga penemu gila, bisa memiliki toko digital mereka sendiri. Sebuah tempat di mana Toto bisa dengan mudah memesan mainan yang sedang tren, dan Pop bisa menemukan topi koboi baru tanpa harus tersesat di keramaian pasar.

"Kalau manusia bisa belanja dengan mudah dari toko-toko online favorit mereka," kata Gro sambil menunjuk ke layar yang menampilkan berbagai marketplace terkenal, "mengapa para Nimons tidak bisa punya 'encyclopedia' belanja mereka sendiri?" Dan lahirlah nama **Nimonspedia** – ensiklopedia belanja untuk para Nimons yang akan menjadi jembatan digital menghubungkan setiap Nimons dengan barang yang mereka butuhkan dan impian yang ingin mereka wujudkan.

Namun, membangun sebuah platform *e-commerce* yang disesuaikan dengan kebutuhan unik para Nimons, bukanlah pekerjaan mudah. Gro dan timnya telah memulai dengan visi, namun mereka membutuhkan bantuan yang cerdas, inovatif, dan bersemangat. Mereka membutuhkan para pemikir muda dengan keahlian teknologi untuk merancang, mengembangkan, dan mewujudkan **Nimonspedia** ini menjadi kenyataan.

Melihat ke luar jendela labnya yang penuh dengan cetak biru dan skema yang rumit, Gro tahu bahwa kunci kesuksesan ada di tangan generasi penerus. Dia berharap bisa bekerja sama dengan kalian untuk merancang dan mengembangkan aplikasi web yang memiliki fitur-fitur penting – mulai dari sistem toko untuk penjual, keranjang belanja yang pintar, hingga sistem pembayaran yang aman – semua yang membuat proses jual beli di **Nimonspedia** menjadi mudah dan menyenangkan.

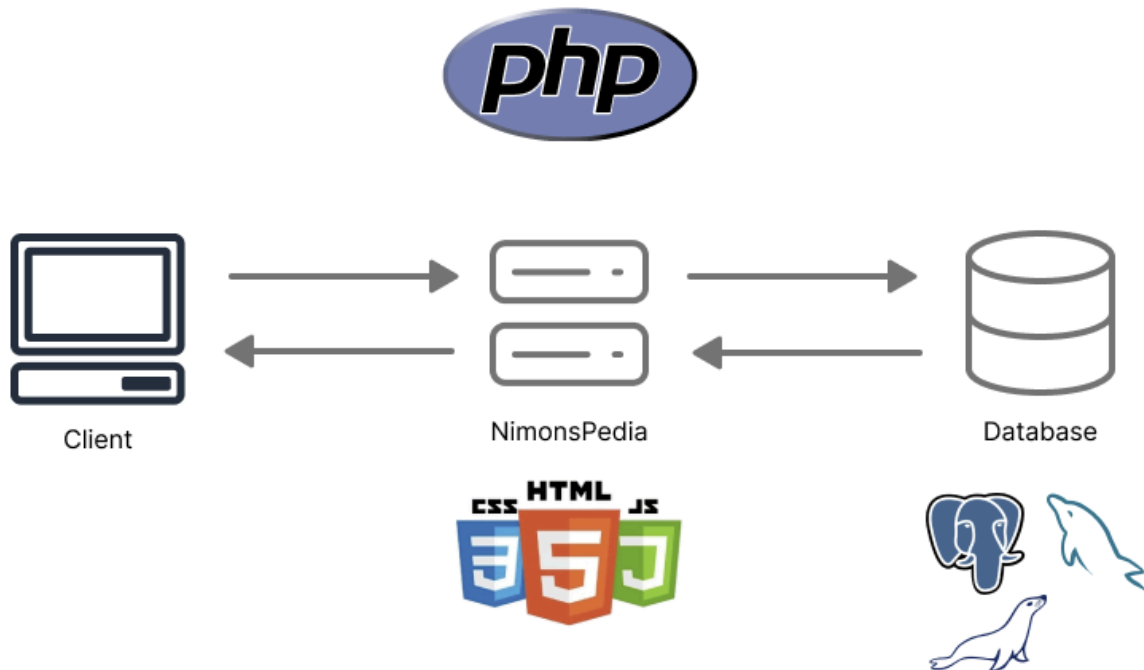
Selama proses pengembangan, kalian akan menghadapi berbagai tantangan unik, seperti merancang antarmuka pengguna yang ramah Nimons dengan warna-warna cerah dan ikon-ikon lucu, memastikan keamanan data para Nimons (terutama resep kue pisang rahasia Pop!), dan mengoptimalkan kinerja aplikasi agar dapat digunakan dengan lancar oleh ribuan Nimons yang antusias secara bersamaan. Dengan kerja sama dan dedikasi, kalian semua dapat membantu Gro dan para Nimons lainnya menemukan dan menawarkan produk yang mereka inginkan, mewujudkan impian belanja digital mereka, serta memperkaya kehidupan seluruh komunitas Nimons.

"Nimonspedia, because buying bananas shouldn't be a Gru-eling task!"

Tujuan Pembelajaran

- Mahasiswa mampu membuat sebuah aplikasi web dengan menggunakan HTML, CSS, dan JS.
- Mahasiswa mampu membuat sebuah layanan web dengan menggunakan PHP.
- Mahasiswa mampu menggunakan AJAX untuk melakukan pemanggilan yang bersifat *asynchronous*.
- Mahasiswa memahami penanganan *state* dan perbedaan HTTP *method*.

Spesifikasi Sistem



Gambaran Umum untuk Milestone 1

Spesifikasi Umum

Berikut ada beberapa ketentuan umum yang harus dipenuhi oleh sistem.

1. Untuk *client-side*, wajib menggunakan **JavaScript**, **HTML**, dan **CSS** secara **murni**. **Tidak diperbolehkan** menggunakan *framework* CSS atau *framework* JS (e.g. jQuery, lodash, Bootstrap, atau Tailwind). CSS harus berada di berkas yang berbeda dengan HTML (gunakan *CSS selector*). Tidak diperbolehkan **memanipulasi** CSS (mengubah isi file CSS programmatically atau dengan JS).
2. Untuk *server-side*, wajib menggunakan **PHP murni** tanpa kakas apapun (seperti Laravel, Codeigniter). Akan tetapi, kalian bisa mengambil inspirasi dari struktur *project* pada Laravel dan sebagainya. Kalian harus mengimplementasikan fitur menggunakan **HTTP method** yang tepat dan sesuai *best practices*.

3. Secara default, **tidak boleh** menggunakan library apapun selain untuk query database dan **quill.js** (untuk rich text editor). Jika ada library yang ingin digunakan, silahkan tanyakan melalui QnA untuk dipertimbangkan.
4. Dalam melakukan request ke *server*, kalian dapat menggunakan [basic form handling](#) ataupun AJAX, dengan **setidaknya tiga** penggunaan pada masing-masing tipe.
5. Untuk basis data, wajib menggunakan MySQL, MariaDB, atau PostgreSQL (*Relational Database*). Tidak diperbolehkan menggunakan ORM atau sejenisnya dan **tidak boleh menyimpan blob atau binary file pada basis data**. Skema yang diberikan di bawah ini dapat dijadikan panduan (**tidak wajib sama**). Kalian dapat menyesuaikan dan mengoptimalkan skema tersebut sesuai kebutuhan, misal jika ingin membuat *multiple image* untuk produk.

Tabel User

Nama	Tipe	Keterangan
user_id	int	Primary key, auto increment
email	string	email address, unique
password	string	Hashed password
role	enum	"BUYER" dan "SELLER"
name	string	Full name pengguna
address	string	Alamat Pengguna
balance	int	Default 0 (hanya berguna untuk role BUYER)
created_at	datetime	Registration timestamp
updated_at	datetime	Last Update timestamp

Tabel Store

Nama	Tipe	Keterangan
store_id	int	Primary key, auto increment
user_id	int	Foreign key User, role harus seller dan, unique
store_name	string	Nama toko (unique)
store_description	string	Deskripsi toko, rich text
store_logo_path	string	Path ke gambar store
balance	int	Balance yang didapat toko
created_at	datetime	Store creation timestamp (default now)
updated_at	datetime	Last Update timestamp (default now)

Tabel Product

Nama	Tipe	Keterangan
product_id	int	Primary key, auto increment
store_id	int	Foreign key to Store
product_name	string	Nama produk
description	text	Deskripsi produk, rich text
price	int	Harga produk
stock	int	Jumlah stok tersedia
main_image_path	string	Path ke main product image
created_at	datetime	Product creation timestamp
updated_at	datetime	Last Update timestamp

deleted_at	datetime	Default null, tanggal ketika barang dihapus (soft delete)
------------	----------	---

Tabel Cart_Item

Nama	Tipe	Keterangan
cart_item_id	int	Primary key, auto increment
buyer_id	int	Foreign key to Users (buyer only)
product_id	int	Foreign key to Products
quantity	int	Item quantity in cart
created_at	datetime	Added to cart timestamp (default now)
updated_at	datetime	Added to cart timestamp (default now)

Tabel Category

Note: untuk category perlu diseed dari awal karena tidak ada CRUD dari aplikasi kita

Nama	Tipe	Keterangan
category_id	int	Primary key, auto increment
name	string	Kategori Item

Tabel Category_Item

Nama	Tipe	Keterangan
category_id	int	Primary key, FK ke tabel Category
product_id	int	Primary key, FK ke tabel Product

Tabel Order

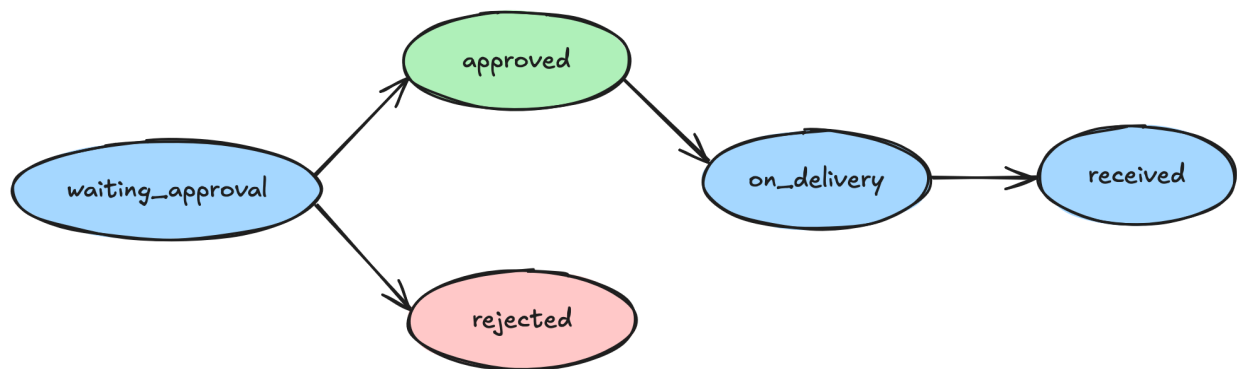
Nama	Tipe	Keterangan
------	------	------------

order_id	int	Primary key, auto increment
buyer_id	int	Foreign key ke User (buyer)
store_id	int	Foreign key ke Stores (toko yang menerima order)
total_price	int	Total harga seluruh item dalam order
shipping_address	string	Alamat pengiriman
status	enum	waiting_approval, approved, rejected, on_delivery, received (default waiting_approval)
reject_reason	text	Alasan jika order ditolak, nullable
confirmed_at	datetime	Waktu ketika order di reject/accept oleh seller, nullable
delivery_time	datetime	Target waktu pengiriman sampai, nullable
received_at	datetime	Waktu ketika order di terima oleh Buyer, nullable
created_at	datetime	Order creation timestamp

Penjelasan Status

Status	Deskripsi	Balance Buyer	Stok Produk	Balance Seller	Next Action
waiting_approval	Menunggu konfirmasi	Di-hold (-)	Dikurangi (-)	Belum masuk	Approve/Reject

approved	Dikonfirmasi, dikemas	Tetap hold	Tetap kurang	Belum masuk	Set Delivery Time, status berubah ke on_delivery
rejected	Ditolak oleh seller	Refund (+)	Kembali seperti semula	Tidak masuk	Masukkan alasan penolakan
on_delivery	Sedang dikirim	Tetap hold	Tetap kurang	Belum masuk	Buyer mengkonfirmasi bahwa barang telah sampai (pastikan bahwa user hanya bisa mengkonfirmasi bila tanggal sekarang >= delivery_time)
received	Sudah sampai	Hilang permanen	Kurang permanen	Masuk (+)	-



State Diagram untuk Status Order

Tabel Order_Items

Nama	Tipe	Keterangan
order_item_id	int	Primary key, auto increment

order_id	int	Foreign key ke Orders
product_id	int	Foreign key ke Products
quantity	int	Jumlah item yang dipesan
price_at_order	int	Harga produk saat order dibuat (tidak terpengaruh oleh perubahan harga barang)
subtotal	int	Subtotal order

User Story Buyer

1. Autentikasi dan Profil

- a. User dapat mendaftar akun baru sebagai buyer dengan nama, email, password, dan address.
- b. Buyer dapat login menggunakan email dan password
- c. Buyer dapat logout dari sistem
- d. Buyer dapat mengupdate profil miliknya. Seluruh data kecuali email dapat diubah.

2. Balance Management

- a. Buyer dapat melihat balance di navbar
- b. Buyer dapat melakukan top-up balance dengan jumlah tertentu melalui interface yang dibebaskan, misalnya ada popup untuk menambah balance.

3. Product Discovery

- a. Buyer dapat melihat daftar produk dengan foto, nama, kategori, harga, produk dan nama toko
- b. Buyer dapat mencari produk berdasarkan nama produk
- c. Buyer dapat memfilter produk berdasarkan kategori dan range harga
- d. Buyer dapat melihat detail produk lengkap dengan deskripsi dan informasi stok
- e. Buyer dapat melihat informasi toko penjual pada halaman detail produk
- f. Buyer dapat melihat detail toko dan semua produk dari toko tertentu"

4. Shopping Cart

- a. Buyer dapat menambahkan produk ke *shopping cart* dengan memilih *quantity* yang diinginkan
- b. Buyer dapat mengakses cart yang sama dari berbagai device (persisten)
- c. Buyer dapat melihat semua barang di *shopping cart* dengan total harga

- d. Buyer dapat melihat cart badge counter di navbar yang menunjukkan jumlah barang **unik** di cart
- e. Buyer dapat mengubah quantity barang di *cart*
- f. Buyer dapat menghapus barang dari *cart*
- g. Buyer dapat menambah *quantity* otomatis jika produk yang sama ditambahkan ke *cart*

5. Checkout & Orders

- a. Buyer dapat melakukan *checkout* barang (penjelasan lengkap ada di spesifikasi)
- b. Buyer dapat melihat riwayat pembelian yang pernah dibuatnya
- c. Buyer akan menerima refund balance jika order *rejected* oleh penjual
- d. Item akan otomatis hilang dari *cart* ketika barang sudah di-*checkout* oleh Buyer
- e. Buyer dapat mengkonfirmasi apabila barang sudah mereka terima

User Story Seller

1. Autentikasi dan Profil

- a. User dapat mendaftar sebagai *seller* dengan nama, email, password, serta alamat, nama, dan deskripsi toko
- b. Seller dapat login menggunakan email dan password untuk masuk ke aplikasi
- c. Seller dapat mengupdate informasi toko (nama toko, logo dan deskripsi)
- d. Seller dapat logout dari sistem

2. Product Management

- a. Seller dapat menambahkan produk baru ke tokonya
- b. Seller dapat melihat daftar semua produk dari tokonya
- c. Seller dapat mengedit informasi produk di tokonya
- d. Seller dapat menghapus produk dari tokonya (*soft delete*)

3. Order Management

- a. Seller dapat melihat daftar order untuk toko dengan status *waiting_approval*
- b. Seller dapat approve atau reject order
- c. Seller dapat mengatur *delivery_time* dan mengubah status ke *on_delivery* untuk order yang sudah ia approve
- d. Balance toko yang ada akan bertambah ketika order dari toko tersebut sudah diterima oleh pengguna

Spesifikasi Fitur

Berikut ada beberapa ketentuan terkait fitur-fitur apa saja yang ada di dalam sistem.

1. Autentikasi pengguna

Dalam melakukan autentikasi, pengguna perlu melakukan aksi login terlebih dahulu. Setelah sukses melakukan login, identitas pengguna akan disimpan dalam browser. Bentuk penyimpanan identitas pengguna dibebaskan kepada mahasiswa.

- Untuk mengetahui pengguna mana yang sedang login, identitas ini dapat dicek di basis data.
- Identitas ini tidak boleh disimpan sebagai parameter HTTP GET.
- Jika identitas ini tidak ada, pengguna dianggap belum login dan diarahkan kembali ke halaman login dari halaman yang sedang dibukanya.
- Ketentuan sebelumnya tidak berlaku untuk halaman yang dapat diakses tanpa autentikasi, yaitu [Halaman Product Discovery](#), [Halaman Detail Produk \(Buyer\)](#), dan [Halaman Detail store \(Buyer\)](#).
- Masa berlaku identitas pengguna dibebaskan.
- Sangat disarankan menggunakan SESSION yang disediakan oleh php untuk melakukan autentikasi

2. Navigation bar

Setiap halaman memiliki navigation bar kecuali halaman login dan register. Isi navbar disesuaikan dengan role pengguna:

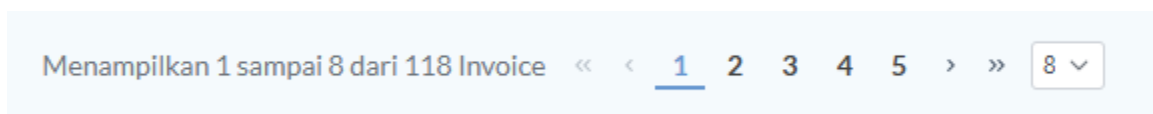
- Guest (belum login):
 - Logo (link ke Product Discovery)
 - Link "[Login](#)"
 - Link "[Daftar](#)"
- Buyer (sudah login):
 - Logo/Home (link ke [Product Discovery](#))

- Cart icon dengan badge counter item unik didalamnya - clickable ke [Halaman Shopping Cart \(Buyer\)](#)
- Balance display (format: "Balance: Rp 1.250.000") - clickable untuk buka modal top-up
- Dropdown user menu:
 1. [Profile](#)
 2. [Order History](#)
 3. Logout
- Seller (sudah login):
 - Logo/Home (link ke Dashboard)
 - Navigation menu:
 1. "Dashboard" (link ke [Dashboard](#))
 2. "Produk" (link ke [Product Management](#))
 3. "Orders" (link ke [Order Management](#))
 - Store balance display (format: "Balance: Rp 5.000.000") - read-only, tidak clickable
 - Logout

3. Penyimpanan File Binary

Tempat penyimpanan file binary (foto produk, logo toko) pada local file system, tidak diperkenankan menyimpan binary file di database. Database hanya menyimpan path menuju lokasi file. Pastikan untuk memperhatikan keamanan dari cara penyimpanan yang kalian pilih.

4. Pagination



- Di [Halaman Product Discovery / Home \(Buyer\)](#), dan [Halaman Order Management \(Seller\)](#), wajib terdapat *pagination*. **Tampilkan sejumlah produk** yang sesuai dengan batasan tertentu per halaman.

- Tampilkan navigasi (halaman berikutnya dan sebelumnya) atau tautan angka halaman untuk membantu pengguna pindah ke halaman lain.
- *Pagination* **wajib dilakukan secara server-side**.
- Pastikan pula pengguna dapat memilih berapa banyak hasil yang dapat ditampilkan di tiap halamannya. Sebagai referensi, gunakan angka 4, 8, 12, dan 20 item setiap halamannya.
- Selain dengan membagi *page*, kalian boleh mengimplementasikannya dengan *infinite scroll*. Tetapi, pastikan bahwa salah satu halaman tetap mengimplementasikan *pagination* secara manual.

5. Rich Text Editor

- Untuk mengimplementasikan *rich text editor*, gunakan kakas [quill.js](https://quilljs.com/).
- Apabila menggunakan *basic form handling event*, kalian dapat mengikuti [guide berikut](#)..
- **Pastikan keamanan** dalam mengimplementasikan *rich text editor* terutama ketika menampilkan hasil teks.

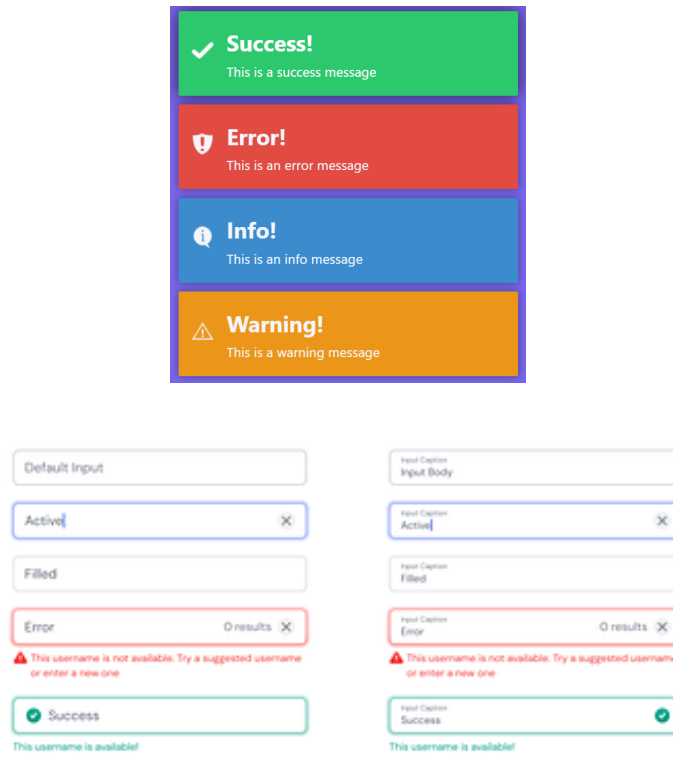
6. Keamanan

- *Client-side* wajib melakukan **validasi masukan** sederhana dari pengguna (misalnya *required field*, regex untuk nomor, etc).
- *Server-side* wajib melakukan **validasi data** dari pengguna untuk memastikan integritas data (data unik, keberadaan data, logika, etc).
- *Password* pengguna wajib di-**hash** pada basis data, algoritma yang digunakan dibebaskan.
- Pastikan pula pengguna hanya dapat mengakses halaman yang bisa diakses oleh *role*-nya

7. Error handling dan State Indicator

Implementasikan **error handling** untuk memberikan umpan balik yang informatif kepada pengguna. Error handling mencakup *server-side* dan juga *client-side*, misalnya error untuk validasi input dan kegagalan request.

Selain untuk *error*, berikan pula umpan balik kepada pengguna terkait dengan situasi / kondisi dari *request* yang telah diberikan. Misal memberikan *loading indicator* dan juga *success notification* yang jelas kepada pengguna.



8. Responsivitas

Implementasikan minimal [Halaman Product](#) dan [Halaman Detail Produk \(Buyer\)](#) (termasuk *navigation bar*) yang **responsif** (minimal untuk ukuran 1280 x 768 dan 400 x 800). Artinya, tampilan mungkin berubah menyesuaikan ukuran layar.

9. Docker

Pada template dari repository yang ada di Github Classroom, sudah terdapat Dockerfile dan docker-compose.yml yang belum dapat dijalankan. Silahkan diimplementasikan sehingga docker dapat dijalankan pada saat demo. Pastikan aplikasi dapat berjalan tanpa kakas apapun selain docker. Kalian juga boleh menggunakan docker untuk development, tetapi tidak diwajibkan dan bukan bagian dari penilaian.

Spesifikasi Tiap Halaman

Spesifikasi disini bersifat sebagai *guide* dari halaman yang perlu dibuat. Kalian dibebaskan untuk menyesuaikan tampilan dari halaman yang ada, selama seluruh **Acceptance Criteria** dipenuhi. Perhatikan pula *Things to Explore* yang mungkin berguna untuk membantu kalian membuat halaman yang baik (atau ketika demo).

1. Halaman Login (Buyer & Seller)

Halaman login berisi form autentikasi untuk **buyer** dan **seller** yang ingin masuk ke sistem menggunakan kredensial **email** dan **password**. Setelah berhasil login, sistem akan mengarahkan pengguna sesuai role mereka. Ketentuan *redirect* ada di bagian AC.

Acceptance Criteria:

- Terdapat *form input* **email** dan **password** (*required*)
- Validasi format **email** dan **password** (Minimal 8 karakter, harus mengandung huruf besar, huruf kecil, angka, dan simbol)
- Tampilkan *password visibility toggle*
- Tampilkan pesan error yang readable untuk user. Jangan ekspos pesan error mentah (detail keamanan)
- Terdapat pranala ke [Halaman Register \(Buyer & Seller\)](#)
- Ketika proses autentikasi berjalan, *disable* tombol dan munculkan *loading indicator*
- *Redirect* setelah login/register sesuai role: Buyer ke [Halaman Product Discover](#), Seller ke [dashboard](#)

Things to Explore:

1. Password hashing
2. Session management

2. Halaman Register (Buyer & Seller)

Halaman register berisi form pendaftaran untuk pengguna baru yang ingin membuat akun sebagai **Buyer** atau **Seller**. Pengguna pertama-tama memilih role mereka, kemudian mengisi form sesuai role yang dipilih. Untuk **Buyer**, form hanya meminta informasi personal dasar. Untuk **Seller**, form meminta

informasi personal ditambah informasi toko (nama toko dan deskripsi toko) yang disimpan di tabel [store](#). Setelah berhasil mendaftar, pengguna akan langsung terautentikasi dan diarahkan ke halaman yang sesuai dengan role mereka.

Acceptance Criteria:

- Pilihan role: **Buyer** atau **Seller** (radio button/toggle)
- Form field untuk semua: nama, email, password, confirm password, alamat
- Validasi password
 - Minimal 8 karakter, harus mengandung huruf besar, huruf kecil, angka, dan simbol
 - Validasi password match dan field tidak kosong
- Form tambahan untuk seller: nama toko (max 100 karakter), deskripsi toko ([rich text](#)), Logo (image)
- Validasi email format, ekstensi file (image type) dan password match
- Lakukan verifikasi bahwa email yang terdaftar harus unique
- Tampilkan password visibility toggle
- Terdapat pranala ke halaman [login](#)
- Ketika proses autentikasi berjalan, disable tombol dan munculkan loading indicator
- Redirect setelah register sesuai role (sama dengan halaman login)

Things to Explore:

- Multi-step form UX
- Auto-login after registration

3. Halaman Product Discovery / Home (Buyer)

Halaman product discovery berfungsi sebagai halaman utama bagi buyer untuk menjelajahi dan menemukan produk yang tersedia dari berbagai seller. Halaman ini menampilkan katalog produk dalam bentuk *grid* card yang memuat foto produk, nama, harga, dan nama toko penjual. Buyer dapat dengan mudah mencari produk spesifik menggunakan search bar,

menyaring produk berdasarkan kategori dan rentang harga, serta menavigasi melalui halaman-halaman produk dengan sistem pagination. **Halaman ini dirancang responsif** untuk memberikan pengalaman browsing yang optimal baik di desktop maupun mobile. Halaman ini dapat diakses oleh guest user (tidak perlu login) namun dengan fitur terbatas (tidak bisa add to cart atau checkout). Pastikan bahwa semua barang ditampilkan, namun untuk barang dengan **stok = 0** memiliki tampilan yang berbeda.

Acceptance Criteria:

- Search dengan debounce 500ms (substring, case-insensitive)
- Filter: kategori, range harga (min-max)
- Product grid: image, nama, harga, nama toko (clickable ke [detail](#))
- [Pagination](#)
- Loading skeleton saat fetch
- Empty state jika tidak ada hasil
- URL parameters untuk search, filter, page
- [Responsive](#)
- Guest user bisa lihat produk tapi tidak bisa add to cart
- Barang dengan stok yang sudah habis memiliki tampilan berbeda

Things to Explore:

- Debouncing (500ms)
- Lazy loading images
- Server-side pagination

4. Halaman Detail Produk (Buyer)

Halaman detail produk menampilkan informasi lengkap tentang sebuah produk termasuk foto, nama, harga, deskripsi detail, informasi stok, kategori, serta informasi toko penjual. Halaman ini memungkinkan buyer untuk melihat produk secara mendetail sebelum memutuskan untuk menambahkannya ke shopping cart. **Buyer** dapat **memilih quantity** yang diinginkan dan langsung **menambahkan** produk ke **cart** dari halaman ini. Halaman ini dapat diakses

oleh guest user. Namun, tombol "Tambah ke Keranjang" hanya muncul untuk buyer yang sudah login.

Acceptance Criteria:

- Product image
- Info produk: nama, harga, deskripsi (HTML), kategori (dari tabel [categories](#)), status stok
- Info toko: nama toko, deskripsi toko (dari tabel [stores](#))
- Form add to cart: quantity selector (+/- button), validasi max stok, tombol "Tambah ke Keranjang" (disabled jika stok habis)
- Guest user: pesan "Login untuk menambahkan ke keranjang" dengan link login
- Toast notification saat berhasil add to cart
- Cart badge update otomatis
- Loading state

Things to Explore:

- HTML rendering (XSS prevention)

5. Halaman Detail store (Buyer)

Halaman public yang menampilkan informasi toko dan katalog semua produk dari toko tertentu. Halaman ini dapat diakses oleh siapa saja (guest/buyer/seller) ketika mereka klik nama toko di product card, product detail, shopping cart, atau order history. Halaman ini berfungsi seperti "toko virtual" dimana visitor dapat melihat profil toko dan browse semua produk yang dijual toko tersebut. Untuk Acceptance Criteria halaman ini, dapat mengikuti halaman home dari buyer, dengan tambahan adanya informasi dari toko terkait.

6. Halaman Shopping Cart (Buyer)

Halaman shopping cart menampilkan daftar semua produk yang telah ditambahkan buyer ke keranjang belanja. Halaman ini berfungsi sebagai tempat untuk review, mengubah quantity, atau menghapus produk sebelum

melanjutkan ke checkout. Cart data disimpan secara persisten di database sehingga tidak hilang meskipun buyer logout atau menutup browser. Halaman ini juga menampilkan total harga dari semua item di cart. Jika cart berisi produk dari beberapa seller berbeda, produk akan dikelompokkan berdasarkan toko (karena checkout membuat order terpisah per toko).

Acceptance Criteria:

- Daftar item: thumbnail, nama, harga, nama toko, quantity selector (+/-), subtotal, tombol hapus
- Group items by toko (tampilkan header nama toko untuk setiap group)
- Auto-increment quantity jika produk sama ditambahkan lagi
- Confirmation dialog saat hapus item
- Summary panel: total item, total harga (per toko dan grand total), tombol checkout (disabled jika cart kosong)
- Empty state dengan tombol "Mulai Belanja"
- Loading state saat update/hapus
- Cart badge update otomatis
- Cart persisten di database
- Grouping items by store

Things to Explore:

- Optimistic UI updates
- Cart persistence
- Debouncing untuk quantity updates

7. Halaman Checkout (Buyer)

Halaman checkout memungkinkan buyer untuk menyelesaikan pembelian produk yang ada di shopping cart. Halaman ini menampilkan ringkasan order, validasi balance buyer, dan proses pembayaran menggunakan balance yang tersedia. Saat checkout, sistem akan membuat order terpisah untuk setiap toko (jika cart berisi produk dari berbagai toko), melakukan hold pada balance buyer, mengurangi stok produk, dan membuat order dengan status **waiting_approval** yang perlu dikonfirmasi oleh seller.

Acceptance Criteria:

- Ringkasan order: daftar produk (grouped by toko), quantity, harga, total per toko, grand total
- Info alamat pengiriman (dari profil, editable)
- Balance info: balance saat ini, balance setelah checkout, warning jika kurang dengan link "Top-up Balance" (buka modal)
- Tombol checkout disabled jika balance tidak cukup atau loading
- Validasi balance, stok sebelum proses, dan apakah produk dihapus (soft delete). Bila produk dihapus, maka transaksi tidak bisa dilakukan.
- Confirmation modal sebelum checkout
- Proses: hold balance, kurangi stok, buat order per toko (status: waiting_approval)
- Redirect ke order history dengan pesan sukses
- Error handling jika stok/balance tidak cukup
- Barang akan otomatis hilang dari *cart* ketika berhasil melakukan *checkout*

8. Halaman Order History (Buyer)

Halaman order history menampilkan daftar semua pembelian yang pernah dilakukan buyer beserta status order masing-masing. Buyer dapat melihat detail setiap order, melacak status pengiriman, dan menerima refund otomatis jika order ditolak oleh seller. Buyer perlu melakukan konfirmasi untuk barang-barang yang sedang berstatus **on_delivery**, tetapi sudah melewati delivery time (mengkonfirmasi bahwa barang terkait sudah sampai) dan status baru akan berubah menjadi received.

Acceptance Criteria:

- Daftar order: order ID, tanggal, nama toko, daftar produk (thumbnail, nama, quantity), total harga, status badge, tombol detail
- Status: Waiting Approval, Approved, On Delivery, Received, Rejected
- Detail order (expand/modal): info lengkap produk, alamat pengiriman, estimasi delivery (jika on_delivery), alasan reject (jika rejected)

- Filter per status dan juga all.
- Sort by tanggal (newest first)
- Adanya tampilan mengenai jumlah dana yang dikembalikan serta alasan penolakan untuk order-order yang di *reject* oleh penjual.
- Empty state jika belum ada order

9. Halaman Profile (Buyer)

Halaman profile **Buyer** memungkinkan pengguna untuk melihat dan mengubah informasi profil mereka termasuk **nama**, **password**, dan **alamat**. **Email tidak dapat diubah** karena digunakan sebagai **unique identifier** untuk [login](#). Halaman ini memberikan kontrol penuh kepada buyer untuk mengelola data personal mereka.

Acceptance Criteria:

- Terdapat *form* untuk **edit profile** yang berisi **nama** (editable), **email** (readonly), **alamat** (editable), pranala ke form ubah password
- Form ubah password (section terpisah)
 - Berisi **password lama**, **password baru**, **confirm password**. Jangan lupa untuk menampilkan password visibility toggle
 - Validasi password
 - Minimal 8 karakter, harus mengandung huruf besar, huruf kecil, angka, dan simbol
 - Validasi password match dan field tidak kosong
 - Verifikasi *server-side* password lama correct
- Tampilkan *confirmation modal* sebelum save
- Saat proses berjalan, disable tombol simpan dan munculkan loading indicator
- Toast notification sukses/error

Things to Explore:

- Password change validation
- Session invalidation after password change

10. Halaman Dashboard (Seller)

Halaman **dashboard seller** adalah landing page setelah login yang menampilkan *overview* dan *quick access* ke fitur-fitur utama. Halaman ini bukan untuk manage produk secara detail, melainkan memberikan gambaran umum kondisi toko dan shortcut ke halaman-halaman lain.

Acceptance Criteria:

- **Header Info:** Seluruh data yang dimiliki toko, serta bisa edit informasi di tempat.
- **Quick Stats Cards:**
 - Total Produk Unik (Bukan quantity)
 - Pending Orders
 - Produk Stok Menipis (< 10)
 - Total Pendapatan
- **Quick Actions Buttons:**
 - "Kelola Produk" → redirect ke [Product Management](#)
 - "Lihat Orders" → redirect ke [Order Management](#)
 - "Tambah Produk Baru" → redirect ke [Add Product](#)

11. Halaman Product Management (Seller)

Halaman ini menampilkan tabel/list semua produk dengan fitur search, filter, sort, dan action buttons (Edit/Delete). Ini adalah halaman utama untuk melihat dan mengelola *inventory* produk **seller**.

Acceptance Criteria:

- Tombol "Tambah Produk Baru", redirect ke [Add Product](#)
- Tabel produk: thumbnail, nama, kategori, harga, stok, status, aksi (Edit, Delete)
- Search produk (debounce, substring, case-insensitive)
- Filter: kategori (dropdown dari [tabel categories](#))
- Sort: nama, harga, stok
- Tombol Edit untuk redirect ke [Halaman Edit Product \(Seller\)](#)
- Tombol Delete dengan confirmation modal

- Loading state saat fetch/delete
- Empty state dengan tombol "Tambah Produk Pertama"
- [Responsive](#)

Things to Explore:

- Soft delete vs hard delete
- Cascade delete validation (cek apakah ada order pending dengan produk ini)
- Product indexing

12. Halaman Add Product (Seller)

Halaman ini adalah form kosong untuk menambahkan produk baru ke inventory seller.

Acceptance Criteria:

- Form: nama produk (max **200** char), deskripsi (max 1000 char) dengan [rich text](#), kategori (dropdown dari tabel categories), harga (min 1000), stok (min 0), foto (max **2MB**, **1** foto)
- Preview foto setelah upload dengan tombol "Ganti Foto"
- Validasi client-side: required fields, format *numeric*, *file size*
- Validasi server-side: *data integrity*, *file type* (**jpg, jpeg, png, webp**)
- Tombol "Simpan" disabled saat loading/validasi gagal
- Tombol "Batal" redirect ke Product Management
- Loading state dan progress bar upload (opsional)
- Toast notification sukses: "Produk berhasil ditambahkan"
- Redirect ke [Product Management](#) setelah sukses

Things to Explore:

- File upload handling
- Image optimization (resize, compress)
- File validation
- File storage path structure

13. Halaman Edit Product (Seller)

Halaman ini adalah form terisi dengan data produk yang sudah ada untuk diedit. Untuk memudahkan, pastikan bahwa sifat perubahan *quantity* yang ada pada suatu barang adalah **mengubah, bukan *increment***.

Acceptance Criteria

- Preview foto saat ini dengan tombol "Ganti Foto"
- Upload foto baru (opsional, max 2MB) untuk replace foto lama
- Preview foto baru jika di-upload
- Tombol "Simpan Perubahan" disabled saat loading
- Tombol "Batal" redirect ke [Product Management](#)
- Confirmation modal
- Loading state
- Toast notification sukses: "Produk berhasil diupdate"
- Redirect ke [Product Management](#) setelah sukses

Things to Explore:

- Image replacement handling
- Price change impact (perhatikan apa yang terjadi kepada transaksi-transaksi lama)

14. Halaman Order Management (Seller)

Halaman order management menampilkan daftar order yang masuk untuk toko seller dengan berbagai status. Seller dapat melihat order dengan status **waiting_approval** dan melakukan **approve** atau **reject**. Saat approve, sistem akan mengubah status ke **approved** (barang sedang disiapkan). Saat **reject**, *balance* buyer dikembalikan dan Seller perlu memasukkan alasan penolakan. Ketika seller selesai mengatur **delivery_time** untuk order yang sudah di-approve, sistem akan mengubah status ke **on_delivery**. Seller juga dapat mencari suatu order dengan menggunakan order ID atau nama buyer.

Acceptance Criteria:

- Filter/tab status: Waiting Approval, Approved, On Delivery, Received, Rejected

- Tabel order: Order ID, tanggal, nama buyer, daftar produk, total harga, status badge, aksi
- Tombol "Lihat Detail", expand/modal detail order
- Detail: info buyer, alamat, produk lengkap, total, status history
- Approve (waiting_approval): tombol "Approve Order", untuk mengubah status ke approved
- Reject (waiting_approval): tombol "Reject Order", modal input alasan (required), proses: refund balance buyer + ubah status + simpan alasan, toast sukses
- Set Delivery (approved): tombol "Kirim Barang", modal input delivery_time (date picker atau input hari), proses: set time + ubah status on_delivery, toast sukses
- Search: order ID atau nama buyer
- [Pagination](#)
- Empty state per status

Things to Explore:

- Refund transaction handling

Spesifikasi Bonus

Catatan: Kerjakan dahulu spesifikasi wajib sebelum mengerjakan bonus.

1. **All Responsive Web Design**

Semua tampilan dibuat **responsif** (minimal untuk ukuran 1280 x 768 dan 400 x 800). Artinya, tampilan mungkin berubah menyesuaikan ukuran layar. Hint: gunakan CSS @media rule, lebih lanjut: https://www.w3schools.com/css/css_rwd_mediaqueries.asp.

2. **UI/UX Seperti Tokopedia**

Membuat UI/UX yang serupa dengan tampilan website Tokopedia, dengan desain yang konsisten dan responsif untuk setiap halaman. Siapa tahu nanti kalian bisa dapat pekerjaan di Tokopedia. Kalian dapat menyesuaikan dengan fitur yang ada pada spesifikasi ini.

3. **Data Export**

Fitur ini memungkinkan *seller* untuk mengunduh daftar performa ke dalam format **Excel** atau **CSV**. Ini berguna untuk keperluan internal seperti analisis, pengarsipan, atau pelaporan. Fungsionalitas ini bisa ditambahkan dengan menggunakan tombol “Export” di halaman dashboard. Nilai yang didapat dari fitur ini akan semakin baik seiring dengan semakin lengkap dan *insightful* file CSV yang dihasilkan.

4. **Advanced Search**

Bayangkan situasi dimana kalian mencari “laptop” di marketplace kesayangan kalian. Namun, kurang majunya algoritma *searching* yang mereka gunakan membuat produk-produk seperti “Macbook”, “Asus”, “Lenovo”, “Acer” dan sebagainya tidak keluar di hasil pencarian. Hal ini akan sangat mengurangi *user experience* dan juga *usability* dari marketplace kalian. Oleh karena itu, implementasikan suatu algoritma *smart searching* yang dapat menyelesaikan persoalan terkait.

Algoritma maupun mekanisme pencarian yang kalian gunakan **DIBEBASKAN**. Manipulasi struktur basis data juga diperbolehkan.

5. Google Lighthouse

Google Lighthouse adalah alat otomatis *open-source* untuk meningkatkan kualitas halaman web. Lighthouse dipakai sebagai alat pengukuran dan audit untuk performance kualitas website, aksesibilitas, aplikasi web progresif, dan banyak lagi. Tugas kalian adalah melakukan pengecekan skor di Lighthouse untuk seluruh halaman dan pastikan bahwa skor untuk **best practices**, **aksesibilitas** yang didapatkan memiliki **nilai di atas 90** dan **performance** yang didapatkan memiliki **nilai di atas 80**. Jika pada saat pertama kali pengukuran skor masih kurang, tetap *capture* dan jelaskan perubahan apa yang kalian lakukan untuk mencapai nilai yang lebih baik. Lampirkan bukti tangkapan layar pada README.

Keyword & Tips

Untuk meringankan beban tugas ini, ada beberapa *keyword* yang bisa kalian cari untuk menyelesaikan tugas ini.

- **HTTP methods:** GET, POST, PUT, PATCH, DELETE, OPTIONS.
- **CSS:** margin, padding, font-size, text-align, flex, grid, border, color, div, span, anchor tag, box-shadow.
- **Javascript:** XMLHttpRequest, addEventListener.
- **PHP:** mysqli_connect, mysqli_query, pg_connect, pg_query, PDO, \$_GET, \$_POST, \$_COOKIE, var_dump, print_r, echo, require.
- **SQL:** SELECT, INSERT, UPDATE, DELETE, WHERE, LIKE, ORDER BY.
- **DOCKER:** Dockerfile, Docker Compose, Docker Container, Docker Image, Docker Volume.

Berikut adalah *tips* dari asisten tercinta.

- Usahakan untuk membuat code yang maintainable (*reusable, loosely coupled, dll*). Sebagai contoh, dalam membuat *navbar*, jangan meng-copy navbar pada setiap berkas php namun manfaatkan **PHP templating**, lebih lanjut: <https://css-tricks.com/php-templating-in-just-php/>.
- Gunakan [autoload](#) pada PHP untuk mempermudah modularitas kode.
- Gunakan design pattern untuk mempermudah arsitektur program, misalnya MVC (Model View Controller). Kalian bisa mengimplementasikan *routing* sendiri alih-alih *routing* berdasarkan direktori file php.
- Perhatikan *performance* dan juga *Access Control* yang baik di aplikasi kalian. Beberapa contohnya adalah membatasi akses ke file-file serta halaman tertentu untuk memastikan bahwa tidak ada orang yang mengakses sesuatu yang bukan hak mereka.
- Dalam menggunakan Docker, pahami dulu istilah-istilah yang dipakai pada Docker, jangan langsung mencoba, karena akan sulit apabila tidak berjalan atau menemukan *error*. Misalnya, pahami dulu perbedaan pembuatan

docker image dan *docker container*. Lalu, ketahui cara kerja *docker container* mulai dari *port forwarding*, *docker volume*, dan *docker network*.

- Jangan berantakan saat membuat kode PHP. Ini saran agar keberlangsungan milestone 2 kalian lancar dan tidak terhambat oleh milestone 1 ini. Apabila tugas besar ini dipakai untuk matkul lain, alur program dapat dibaca dan diingat dengan cepat.

Lain-Lain

Responsi

Diadakan responsi per kelompok bersama asisten menjelang pengumpulan tugas.

- Responsi bersifat **opsional**. Artinya **tidak wajib**.
- Kelompok yang ingin mendaftarkan diri dapat mengisi pada [pranala](#) berikut.
- Deadline pengisian jadwal responsi adalah **11 Oktober 2025 pukul 23.59 WIB**.
- Perlu diperhatikan bahwa ketersediaan asisten **terbatas**.
Siapkan terlebih dahulu hal-hal yang ingin ditanyakan / dikonsultasikan dan lakukan setup project di komputer kalian **sebelum** sesi responsi dimulai.
- Pastikan sudah riset dan **mencoba** mempelajari sebelum bertanya ke asisten. Agar bisa bertanya lebih banyak pertanyaan dan pertanyaan lebih dekat ke pengerjaan bukan *understanding*.
- Silahkan **membuat undangan** di Google Calendar sesuai jadwal yang telah diisi menggunakan email STD asisten.

Daftar Pertanyaan

Jika masih ada pertanyaan mengenai spesifikasi tugas, dapat dilakukan melalui [QnA](#) berikut.

Pastikan **sudah membaca spesifikasi sepenuhnya hingga akhir** dan tidak ada pertanyaan yang berulang.

Deliverables

Berikut adalah hal yang harus diperhatikan untuk pengumpulan tugas ini:

1. Buatlah tim pada github classroom [berikut](#) dengan format nama tim **<kelas>-<nomor kelompok>**, contoh: **K01-02** oleh salah satu anggota kelompok. Anggota lainnya bisa *join* ke tim yang sudah dibuat.
2. Hal-hal yang harus diperhatikan.
 - a. Silakan *commit* pada *repository* kalian yang dibuat otomatis..
 - b. Lakukan beberapa *commit* dengan pesan yang bermakna, contoh: "add register form", "fix logout bug", jangan seperti "final", "benerin dikit", "fix bug".
 - c. Disarankan untuk tidak melakukan *commit* dengan perubahan yang besar karena akan mempengaruhi penilaian (contoh: hanya melakukan satu *commit* kemudian dikumpulkan).
 - d. *Commit* dari setiap anggota tim akan mempengaruhi penilaian.
Jadi, setiap anggota tim harus melakukan commit yang berpengaruh terhadap proses pembuatan aplikasi.
 - e. Sebagai panduan, kalian bisa mengikuti [semantic commit](#).
3. Buatlah berkas **README** yang berisi:
 - a. Deskripsi aplikasi web
 - b. Daftar *requirement*
 - c. Cara instalasi
 - d. Cara menjalankan *server*
 - e. Tangkapan layar tampilan aplikasi (tidak perlu semua kasus, minimal 1 per halaman), dan
 - f. Penjelasan mengenai pembagian tugas masing-masing anggota (lihat formatnya pada bagian pembagian tugas).

Pengumpulan Tugas

- Tenggat waktu tugas adalah pada hari **31 Oktober 2025 pukul 23.59 WIB**.
- Tahap pengumpulan adalah sebagai berikut.

- a. Buat *release* dengan format **v1.x** pada Github dan judul bebas tetapi bermakna.
X adalah nomor yang dimulai dari 1.
- b. Apabila kalian ingin merevisi tugas, buat *release* dengan penambahan angka sebelumnya.
Contoh, kelompok kalian sudah merilis v1.1, tetapi ternyata ada revisi sedikit, silakan buat rilis v1.2. **Tetap gunakan major version 1 yaitu v1.x untuk milestone 1 ini.**
Tugas yang dinilai adalah *release* versi terakhir.
- c. Waktu pengumpulan tugas yang dilihat adalah **waktu *release version* terakhir ke server Github** terakhir.
Akan ada pengurangan terhadap *release version* yang melebihi waktu tenggat waktu tugas.

Pembagian Kelompok

Pembagian kelompok ada di [pranala](#) berikut.

Bagi yang belum mendapatkan kelompok dapat mengontak asisten melalui Teams.

Pembagian Tugas

Setiap anggota kelompok **diwajibkan** untuk mengerjakan bagian *server-side* (PHP) dan *client-side* (HTML, CSS, JS), dengan harapan kalian bisa mencoba mempelajari semua bagian dan cara kerjanya.

Server-side

Login : 13522xxx, 13522xxx

Register : 13522xxx

(Lanjutkan ...)

Client-side

Login : 13522xxx, 13522xxx

Register : 13522xxx

(Lanjutkan...)

Catatan Tambahan

Milestone ini akan berhubungan dengan *milestone* selanjutnya; artinya tugas yang kalian buat di sini **akan digunakan kembali pada *milestone* berikutnya**. Usahakan untuk menyelesaikan spesifikasi utama agar tidak terhambat pada *milestone* berikutnya.