

# **Spesifikasi Tugas Besar Milestone**

## **2 IF3110 Web Based Development:**

### **Nimonspedia**

Tim Asisten Lab Pemrograman 2022

Versi : 1

Tgl. Revisi Terakhir : **Kamis 11 Desember 2025 pukul 21.28**

Deadline : **Jumat 12 Desember 2025 pukul 21.00 WIB**

Revisi:

- Penghapusan End Time di User Story
- Penambahan keterangan terkait Load Test

# Daftar Isi

<b>Daftar Isi.....</b>	<b>2</b>
<b>Deskripsi Persoalan.....</b>	<b>5</b>
<b>Tujuan Pembelajaran.....</b>	<b>8</b>
<b>Spesifikasi Sistem.....</b>	<b>9</b>
Nginx (Reverse Proxy & Static File Server).....	9
PHP Backend.....	9
Node.js Backend.....	10
React SPA (Client-Side Application).....	10
Database (MySQL/PostgreSQL).....	11
Spesifikasi Umum.....	11
Tabel User (Ada tambahan Role Admin).....	12
Tabel Auctions.....	12
Tabel Auction_Bids.....	13
Tabel Chat_Room.....	13
Tabel Chat_Messages.....	14
Tabel Push_Subscriptions.....	14
Tabel Push_Preferences.....	15
Tabel User_Feature_Access.....	15
User Story Buyer.....	16
1. Auction.....	16
2. Chat.....	16
3. Push Notification.....	17
User Story Seller.....	17
1. Auction Management.....	17
2. Chat.....	17
3. Product Management.....	17
4. Push Notification.....	18
User Story Admin.....	18
1. Autentikasi dan Profil.....	18
2. User Management.....	18

3. User Feature Flags Management.....	18
<b>Spesifikasi Fitur.....</b>	<b>19</b>
1. Autentikasi pengguna.....	19
2. Navigation bar.....	19
3. Keamanan.....	20
4. Error handling dan State Indicator.....	20
5. Responsivitas.....	20
6. Feature Flags.....	20
7. Push Notifications.....	21
8. Websocket.....	22
9. Pagination.....	22
10. Load Test.....	23
11. Docker.....	23
<b>Spesifikasi Tiap Halaman.....</b>	<b>24</b>
<b>Halaman Baru (React).....</b>	<b>24</b>
1. Halaman Auction List (Buyer).....	24
Acceptance Criteria:.....	24
Things to Explore:.....	24
2. Halaman Auction Detail (Buyer & Seller).....	25
Acceptance Criteria:.....	25
3. Halaman Chat (Buyer & Seller).....	26
Acceptance Criteria:.....	26
Things to Explore:.....	27
4. Halaman Admin Dashboard (Admin).....	27
Acceptance Criteria:.....	27
5. Halaman Admin Login (Admin).....	28
Acceptance Criteria:.....	28
<b>UPDATE HALAMAN.....</b>	<b>28</b>
1. Halaman Detail Produk (Buyer).....	28
2. Halaman Product Management (Seller).....	28
3. Halaman Profile (Buyer & Seller).....	28
<b>Spesifikasi Bonus.....</b>	<b>30</b>
1. All Responsive Web Design.....	30

2. UI/UX Seperti Tokopedia.....	30
3. Review & Rating System.....	30
4. Payment Webhook Integration.....	31
5. Google Lighthouse.....	32
<b>Keyword &amp; Tips.....</b>	<b>33</b>
<b>Lain-Lain.....</b>	<b>36</b>
Daftar Pertanyaan.....	36
Deliverables.....	36
Pengumpulan Tugas.....	37
Pembagian Kelompok.....	38
Pembagian Tugas.....	38

## 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. Luiy, 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.

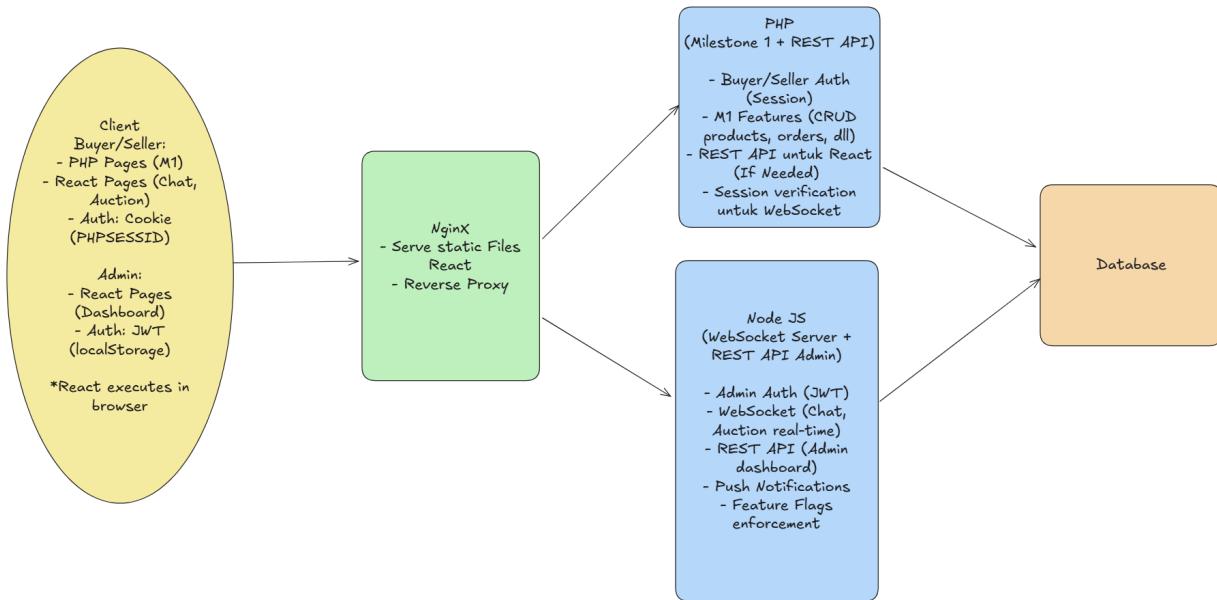
*"Nimonspedia, because buying bananas shouldn't be a Gru-eling task!"*

(Tugas ini merupakan simplifikasi dari problem di dunia nyata. Beberapa asumsi dibuat hanya untuk konteks perkuliahan ini.)

## **Tujuan Pembelajaran**

- Mahasiswa mampu mengintegrasikan React SPA dengan backend PHP existing
- Mahasiswa memahami dan mengimplementasikan JWT authentication
- Mahasiswa mampu mengimplementasikan WebSocket untuk fitur real-time
- Mahasiswa memahami konsep Service Worker
- Mahasiswa mampu mengimplementasikan Push Notification di browser
- Mahasiswa mampu melakukan load testing dan optimasi kinerja
- Mahasiswa memahami konsep reverse proxy dengan Nginx

# Spesifikasi Sistem



Gambaran Umum untuk Milestone 2

Milestone 2 merupakan kelanjutan dari Milestone 1 dengan penambahan fitur real-time menggunakan WebSocket, React SPA, dan sistem administrasi. Arsitektur aplikasi akan menggunakan kombinasi PHP (legacy features), Node.js (real-time features), dan React (modern UI).

## Komponen Sistem

### Nginx (Reverse Proxy & Static File Server)

- **Wajib digunakan** sebagai reverse proxy dan static file server
- **Routing request** ke PHP atau Node.js berdasarkan path
- **Serving static files** untuk React SPA (HTML, JS, CSS, assets)
- **WebSocket proxy** untuk real-time communication

### PHP Backend

- Menjalankan semua fitur dari Milestone 1
- Menangani autentikasi buyer/seller (session-based)

- Menyediakan **REST API** untuk React
- Generate session untuk WebSocket authentication
- Berkommunikasi dengan database MySQL/PostgreSQL

## Node.js Backend

- **WebSocket server** untuk real-time features
- Menangani auction logic (bidding, countdown, winner determination)
- Menangani chat real-time
- Menangani admin authentication (**JWT-based**)
- Trigger push notifications
- Enforce feature flags dan global flags
- Membuat order otomatis saat auction selesai
- Menyediakan **REST API** untuk admin
- Berkommunikasi dengan database MySQL/PostgreSQL
- **Recommended tools:**
  - Express, Hono, atau Nest.js
  - Socket.io

## React SPA (Client-Side Application)

- **Berjalan di browser client** setelah di-download dari Nginx
- Static files (HTML, JS, CSS) di-build menggunakan Vite/CRA dan di-host oleh Nginx
- **Wajib digunakan untuk:**
  - Chat Page
  - Auction Page (list & detail)
  - Admin Dashboard
- **Kommunikasi:**
  - Dengan PHP Backend via **REST API**
  - Dengan Node.js Backend via **WebSocket & REST API**
- **State management** dengan React hooks (useState, useContext, useReducer)
- **Recommended setup:**
  - Vite sebagai build tool (alternatif: Create React App)
  - React Router DOM
  - Socket.io-client untuk WebSocket client
  - Axios atau Fetch API untuk HTTP requests
  - Boleh menggunakan TailwindCSS tapi tidak dengan UI library.

## Database (MySQL/PostgreSQL)

- Shared database antara PHP dan Node.js
- Schema dari Milestone 1 + tambahan untuk Milestone 2
- Menyimpan data users, products, auctions, bids, chat messages, push subscriptions, dll.

## Spesifikasi Umum

Berikut ada beberapa ketentuan umum yang harus dipenuhi oleh sistem.

1. Untuk **client-side PHP**, mengikuti aturan Milestone 1: wajib menggunakan JavaScript, HTML, dan CSS secara murni tanpa framework. Untuk halaman baru (auction dan chat) **wajib menggunakan React**.
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 (Ada tambahan Role Admin)

Nama	Tipe	Keterangan
user_id	int	Primary key, auto increment
email	string	email address, unique
password	string	Hashed password
role	enum	"BUYER", "SELLER", "ADMIN"
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 Auctions

Nama	Tipe	Keterangan
auction_id	int	Primary key, auto increment
product_id	int	Foreign key to Product
starting_price	int	Harga awal lelang
current_price	int	Harga bid tertinggi saat ini (default = starting_price)
min_increment	int	Minimum kenaikan bid
quantity	int	Jumlah item yang dilelang (dari stock product)

start_time	datetime	Waktu mulai auction
end_time	datetime	Waktu selesai auction, nullable, diisi ketika auction sudah berakhir.
status	enum	"scheduled", "active", "ended", "cancelled"
winner_id	int	Foreign key to User (buyer), nullable
created_at	datetime	Auction creation timestamp

### Tabel Auction\_Bids

Nama	Tipe	Keterangan
bid_id	int	Primary key, auto increment
auction_id	int	Foreign key to Auctions
bidder_id	int	Foreign key to User (buyer)
bid_amount	int	Jumlah bid
bid_time	datetime	Waktu bid dibuat (default now)

### Tabel Chat\_Room

Nama	Tipe	Keterangan
store_id	int	Foreign key to Store, PK
buyer_id	int	Foreign key to User (buyer), PK
last_message_at	datetime	Timestamp pesan terakhir, nullable
created_at	datetime	Room creation timestamp (default now)

updated_at	datetime	Last update timestamp (default now)
------------	----------	-------------------------------------

### Tabel Chat\_Messages

Nama	Tipe	Keterangan
message_id	int	Primary key, auto increment
store_id	int	Foreign key ke Chat_Rooms (bagian dari composite key)
buyer_id	int	Foreign key ke Chat_Rooms (bagian dari composite key)
sender_id	int	Foreign key to User
message_type	enum	"text", "image", "item_preview"
content	text	Message content (text atau path untuk image)
product_id	int	Foreign key to Product (jika type = item_preview), nullable
is_read	boolean	Default false
created_at	datetime	Message timestamp (default now)

### Tabel Push\_Subscriptions

Nama	Tipe	Keterangan
subscription_id	int	Primary key, auto increment

user_id	int	Foreign key to User
endpoint	text	Push service endpoint URL
p256dh_key	string	Public key untuk encryption
auth_key	string	Auth secret untuk encryption
created_at	datetime	Subscription timestamp (default now)

**Tabel Push\_Preferences**

Nama	Tipe	Keterangan
user_id	int	Primary key, Foreign key to User
chat_enabled	boolean	Enable push untuk chat (default true)
auction_enabled	boolean	Enable push untuk auction (default true)
order_enabled	boolean	Enable push untuk order (default true)
updated_at	datetime	Last update timestamp (default now)

**Tabel User\_Feature\_Access**

Nama	Tipe	Keterangan
access_id	int	Primary key, auto increment
user_id	int	Foreign key to User, Null apabila menandakan global
feature_name	enum	"checkout_enabled", "chat_enabled",

		"auction_enabled"
<b>is_enabled</b>	boolean	Default true
<b>reason</b>	text	Alasan disable feature, nullable
<b>updated_at</b>	datetime	Last update timestamp (default now)

## **User Story Buyer**

### **1. Auction**

- a. Buyer dapat melihat daftar auction yang sedang aktif dan yang akan datang (scheduled)
- b. Buyer dapat melihat detail auction dengan informasi lengkap dan countdown timer real-time
- c. Buyer dapat join auction session dan place bid
- d. Buyer melihat bid update secara real-time tanpa refresh
- e. Buyer menerima push notification saat di-outbid, menang auction, atau auction akan berakhir

### **2. Chat**

- a. Buyer dapat melihat daftar chat rooms dan memulai chat baru dengan store
- b. Buyer dapat mengirim message text, image attachment, dan item preview
- c. Buyer melihat message baru, typing indicator, dan read receipts secara real-time
- d. Buyer menerima push notification saat ada message baru dari seller
- e. Buyer tidak dapat send message jika feature flag disabled

### **3. Push Notification**

- a. Buyer dapat mengatur preferensi push notification per-category (chat, auction, order) di Profile

## User Story Seller

### 1. Auction Management

- a. Seller dapat membuat auction dari produk yang sudah ada dengan input quantity, starting price, min increment, ~~dan end time~~
- b. Seller hanya dapat memiliki 1 auction aktif dalam satu waktu
- c. Seller dapat melihat auction yang sedang berjalan dengan bid history dan countdown timer real-time
- d. Seller dapat cancel auction sebelum ada bid
- e. Saat auction berakhir dengan winner, order otomatis dibuat dengan status "approved"

### 2. Chat

- a. Seller dapat melihat daftar chat rooms dengan buyers
- b. Seller dapat mengirim message text, image attachment, dan item preview
- c. Seller melihat message baru, typing indicator, dan read receipts secara real-time
- d. Seller menerima push notification saat ada message baru dari buyer
- e. Seller tidak dapat send message jika feature flag disabled

### 3. Product Management

- a. Seller dapat melihat badge "DALAM LELANG" pada produk yang sedang dilelang
- b. Seller tidak dapat edit/delete produk yang sedang dilelang
- c. Seller dapat membuat auction dari produk via tombol "Jadikan Lelang"

### 4. Push Notification

- a. Seller dapat mengatur preferensi push notification per-category (chat, auction, order) di Profile

## User Story Admin

### 1. Autentikasi dan Profil

- a. Admin dapat login menggunakan email dan password via [Admin Dashboard](#)
- b. Admin login menggunakan JWT authentication (berbeda dari buyer/seller)
- c. JWT disimpan di localStorage dan valid selama 1 jam (ketika demo)
- d. Admin dapat logout dari sistem

### 2. User Management

- a. Admin dapat melihat daftar semua users (buyer dan seller)
- b. Admin dapat search user berdasarkan nama atau email
- c. Admin dapat filter user berdasarkan role (buyer/seller)

### 3. User Feature Flags Management

- a. Admin dapat melihat feature flags untuk user tertentu
- b. Admin dapat enable/disable feature flags untuk user tertentu
- c. Admin harus memberikan reason saat disable feature
- d. User yang di-disable akan melihat reason saat mencoba menggunakan feature
- e. Admin dapat re-enable feature kapan saja

### 4. Global Feature Flags Management

- a. Admin dapat enable/disable global feature flags

## **Spesifikasi Fitur**

Berikut ada beberapa ketentuan terkait fitur-fitur apa saja yang ada di dalam sistem. Spesifikasi dari Milestone 1 tetap berlaku.

### **1. Autentikasi pengguna**

- Admin Authentication

Admin menggunakan sistem autentikasi yang berbeda:

1. Login melalui Admin Dashboard (React)
2. Menggunakan JWT (JSON Web Token) yang di-generate oleh Node.js
3. JWT disimpan di localStorage browser
4. Setiap request ke admin endpoints harus include JWT di header
5. Jika JWT expired atau invalid, redirect ke halaman login admin

- WebSocket Authentication

Untuk koneksi WebSocket (Chat & Auction):

1. Menggunakan PHP session cookie yang sudah ada
2. Browser otomatis mengirim session cookie saat WebSocket handshake
3. Node.js verify session dengan query ke database atau Redis
4. Jika session valid, connection accepted dengan metadata user (user\_id, role)
5. Jika session invalid, connection rejected dengan error 401

### **2. Navigation bar**

Berikut merupakan tambahan dari hal yang perlu ada di *Navigation Bar*

- *Buyer (sudah login):*
  - Link "Lelang" (link ke Auction Page Buyer)
  - Link "Chat" (link ke Chat Page Buyer)
- *Seller (sudah login):*
  - "Lelang" (link ke Auction Page Seller)
  - "Chat" (link ke Chat Page Seller)

### **3. Keamanan**

- WebSocket connections harus ter-autentikasi (verify session/JWT)

- Feature flags harus di-enforce di server-side (PHP & Node.js), bukan hanya UI
- Push notification payload harus di-encrypt (handled by Web Push API)
- Admin endpoints harus protected dengan JWT verification
- XSS prevention untuk chat messages (sanitize input, escape output)

#### 4. Error handling dan State Indicator

Implementasikan beberapa hal berikut untuk page-page baru kalian:

1. Loading states untuk async operations (skeleton screens, spinners)
2. Error boundaries untuk handle React errors
3. Toast notifications untuk success/error messages
4. WebSocket connection status indicator
5. Optimistic UI updates dengan rollback jika error

#### 5. Responsivitas

Implementasikan minimal [Halaman Chat](#) yang **responsif** (minimal untuk ukuran 1280 x 768 dan 400 x 800). Artinya, tampilan mungkin berubah menyesuaikan ukuran layar.

#### 6. Feature Flags

Feature flags memungkinkan admin untuk enable/disable fitur tertentu untuk user atau globally. Untuk fitur yang di *disable* diharapkan dihapus dari navbar.

Buatlah sebuah halaman *disabled state* yang menjadi tempat *default* ketika ada pengguna yang mencoba untuk mengakses halaman yang sedang tidak tersedia baginya. Tampilkan pesan alasan dimati其实nya fitur tertentu kepada pengguna (Mainenance jika fitur global, alasan spesifik jika untuknya).

- User-Specific Feature Flags

Disable feature untuk user tertentu dengan reason.

- Global Feature Flags

Disable feature untuk semua user (maintenance mode).

3 Flag yang dapat dinyalakan/matikan:

1. auction\_enabled

2. chat\_enabled
3. Checkout\_enabled (Disable dari navbar dan juga functionality untuk add to cart)

## 7. Push Notifications

Push notification adalah cara yang bagus untuk membuat pengguna tetap terlibat dengan aplikasi. Server dapat mengirimkan push notification ke client bahkan saat client tidak aktif.

Push notification di aplikasi web melibatkan kombinasi dari browser APIs, service worker, dan server yang dapat mengirim pesan. Jika browser sedang offline, pesan akan di-queue (diantrikan) hingga browser online. Pengguna dapat memilih untuk menerima notifikasi ini dari situs web.

### Use Case Notifikasi

#### 1. Chat Notification

Saat pengguna menerima pesan dari pengguna lain, maka notifikasi harus dimunculkan dengan isi notifikasi adalah pengguna lain yang mengirim dan pesan yang dikirim. Saat pengguna click notifikasi, maka akan langsung di-redirect ke chat tersebut.

**NOTE:** Pastikan notifikasi hanya dimunculkan pada user yang bersangkutan.

#### 2. Auction Notification

Notifikasi untuk event auction:

1. Outbid Notification
2. Win Auction Notification
3. Ending Soon Notification

#### 3. Order Notification

Saat order status berubah (approved, rejected, on\_delivery, received), notifikasi harus dimunculkan kepada buyer atau seller yang bersangkutan.

### Referensi

Berikut adalah beberapa referensi terkait Implementasi Push Notification:

1. [PushSubscription - Web APIs | MDN](#)
2. [Push API - Web APIs | MDN](#)
3. [Push Notification in React and Next.js app using Node.js | by Reetesh Kumar | Medium](#)

## 8. **Websocket**

WebSocket digunakan untuk komunikasi real-time di Chat dan Auction.

- WebSocket Server (Node.js)
  - Menggunakan Socket.io atau ws library
  - Handle connection authentication (verify PHP session)
  - Handle events: connect, disconnect, error
  - Broadcast updates ke clients yang relevant
  - Implement heartbeat/ping-pong untuk keep-alive
- WebSocket Client (React)
  - Connect ke WebSocket server dengan session cookie
  - Auto-reconnect pada disconnect
  - Emit events ke server (send message, place bid)
  - Listen events dari server (new message, bid update)

## 9. **Pagination**

Ketentuan Pagination sama seperti Milestone 1. Untuk memudahkan, tidak terdapat ketentuan mengenai berapa jumlah halaman yang menggunakan *infinite scroll* dan juga *server side pagination*. Silakan implementasikan yang lebih sesuai dengan preferensi kalian. Gunakan *Cursor Based Pagination* untuk memastikan performa yang lebih baik pada data yang relatif sering untuk berubah (minimal 1).

## 10. **Load Test**

Load testing adalah proses menguji performa aplikasi dengan memberikan beban yang tinggi untuk mengukur response time dan kemampuan sistem menangani concurrent requests. Pada tugas besar ini, load testing akan dilakukan untuk 4 endpoint GET products dengan skenario berbeda.

1. **Get All Products**

2. **Get Products with Range**
3. **Get Products with Filter**
4. **Get Products with Search**

Setelah mempertimbangkan beberapa hal, mekanisme load test akan diberikan setelah UAS, berdekatan dengan jadwal demo. Hal yang perlu dipastikan adalah kalian memiliki endpoint GET product yang mampu menangani keempat fungsionalitas tersebut. Load test akan dilakukan menggunakan k6 dan saat demo akan dijalankan di VPS milik asisten.

**Khusus untuk perubahan yang berkaitan dengan penyesuaian aplikasi terhadap mekanisme load test, perubahan setelah batas waktu (deadline) akan diperbolehkan.**

## 11. 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. Beberapa hal yang direkomendasikan untuk ada di dalam docker adalah:

1. nginx service (reverse proxy)
2. php service
3. node service
4. db service
5. redis service (opsional, untuk session sharing)

## Spesifikasi Tiap Halaman

Spesifikasi disini bersifat sebagai *guide* dari halaman yang perlu dibuat. Kalian diberikan 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).

### Halaman Baru (React)

#### 1. Halaman Auction List (Buyer)

Halaman yang menampilkan daftar auction yang tersedia, baik yang sedang aktif maupun yang akan datang (scheduled). Tampilan diberikan formatnya.

##### Acceptance Criteria:

- Terdapat tab/section untuk "Lelang Aktif" dan "Lelang Akan Datang"
- Countdown timer update setiap detik tanpa refresh (client-side) di setiap produk untuk menunjukkan berapa lama lagi lelang mereka akan dibuka / ditutup.
- Terdapat pagination
- Terdapat search berdasarkan toko dan nama produk (tidak perlu filter) dengan debounce.
- Setiap Card Auction setidaknya mengandung product image, nama produk, current bid/startling price, countdown timer (aktif) atau waktu mulai (scheduled), jumlah bidders
- Apabila lelang baru dimulai ketika pengguna sudah masuk di satu halaman, maka UI akan otomatis berubah tanpa perlu refresh.
- Card Auction akan redirect ke [halaman Auction Detail](#)

##### Things to Explore:

1. WebSocket connection management dan auto-reconnect
2. Real-time countdown timer dengan setInterval

3. Optimistic UI updates
- 



Contoh Countdown yang menempel ke lelang tertentu.

## 2. Halaman Auction Detail (Buyer & Seller)

Halaman detail untuk satu auction spesifik dengan fitur real-time bidding. Lelang dapat berhenti dalam 2 kondisi utama:

1. Tidak terdapat bid yang lebih tinggi dalam waktu 15 detik setelah bid terakhir masuk ke sistem (tampilkan counter mundur).
2. Penjual menghentikan (bukan cancel), lelang dengan sendirinya.

Initial countdown dihitung dari server timestamp sebagai source of truth. Countdown berjalan di client-side dengan update setiap detik. WebSocket broadcast dilakukan saat ada bid baru untuk recalculate countdown (kembali menjadi 15 detik). Auto-sync dengan server dilakukan setiap 30 detik untuk koreksi drift. Countdown stop di 0 akan trigger auction end logic

Ketika pembeli melakukan *bid*, *balance* mereka akan langsung berkurang, tetapi akan kembali apabila mereka bid mereka dikalahkan. Ketika sudah terdapat pemenang, akan langsung terbentuk order dengan status 'approved' dan penjual hanya perlu untuk menentukan tanggal pengirimannya. Pastikan integritas data tetap terjaga di kondisi ini.

Seller juga bisa mengakses halaman ini, tetapi tidak dapat melakukan *bidding*. Mereka hanya bisa melihat kondisi lelang sekarang serta dapat menghentikan lelang.

### Acceptance Criteria:

- Ada Section product information: large image, nama produk, deskripsi lengkap (rich text), quantity yang dilelang, store name (clickable ke store detail)

- Section auction information: starting price, current highest bid (real-time), minimum increment, countdown timer (update setiap detik), waktu mulai & berakhir
- Section bid input (buyer only): input bid amount dengan validation real-time, display minimum bid requirement, display user balance, button "Place Bid" (disabled jika insufficient balance atau invalid input)
- Section bid history: tabel dengan kolom (Bidder name (boleh anonim)/Anda, Bid amount, Time), real-time update, highlight bid dari user sendiri, show latest 10 bids dengan button "Load More", display total bidders count
- Seller view: tidak bisa bid, button "Cancel Lelang" (hanya jika belum ada bid), atau "Hentikan Lelang" (jika sudah ada bid). Confirmation modal untuk cancel.
- Status handling berbeda untuk SCHEDULED (show countdown to start, disable bid), AKTIF (enable bid), BERAKHIR (show winner info), CANCELLED (show reason)
- Validation: bid amount  $\geq$  current\_price + min\_increment, balance  $\geq$  bid amount, auction status = AKTIF

### **3. Halaman Chat (Buyer & Seller)**

Halaman chat yang menampilkan list chat rooms (sidebar) dan chat room interface (main panel) dalam satu halaman.

#### **Acceptance Criteria:**

##### **Sidebar (Chat List):**

- Tampilkan semua chat rooms user dengan search box di atas
- Setiap room card menampilkan: store logo & name (buyer) atau buyer name (seller), last message preview (truncate 50 chars), timestamp (format: "5 menit lalu", "Kemarin", "12 Jan"), unread badge count
- Sort by last\_message\_at (newest first)
- Search by store/buyer name (debounce 300ms)
- Empty state: "Belum ada percakapan" dengan button "Mulai Chat" (buyer only)

- Button "Chat Baru" (buyer) → modal dengan list stores untuk memulai chat
- Highlight active room
- Real-time update: new message → update preview & move to top, update unread count

### **Main Panel (Chat Room):**

- Header menampilkan logo & nama toko (pembeli) atau nama pembeli (penjual).
- Pesan ditampilkan urut dari yang paling lama ke yang terbaru.
- Tampilan otomatis scroll ke bawah ketika ada pesan baru.
- Bubble pesan berbeda: pesan sendiri di kanan (biru), pesan lawan bicara di kiri (abu-abu).
- Setiap pesan menampilkan timestamp.
- Status pesan memakai ikon centang untuk terkirim, diterima, dan dibaca (format dibebaskan)
- Indikator mengetik muncul sebagai “[Nama] sedang mengetik...” selama 3 detik setelah aktivitas mengetik terakhir.
- Saat membuka chat, sistem memuat 50 pesan terakhir.
- Saat scroll ke atas, sistem memuat 50 pesan tambahan yang lebih lama.
- Menampilkan loading skeleton saat pesan sedang dimuat.

### **Things to Explore:**

- Typing indicator

## **4. Halaman Admin Dashboard (Admin)**

Halaman dashboard admin dengan semua fitur dalam satu halaman. Halaman ini berisi daftar user beserta dengan sebuah search bar. Untuk masing-masing pengguna, admin dapat mengatur feature flags secara individual. Terdapat pula sebuah section untuk menyalakan/mematikan Global Feature Flags.

#### **Acceptance Criteria:**

- Redirect ke login jika JWT tidak valid
- Header dengan info admin, dan tombol logout
- Search bar untuk cari user by nama/email (debounce 300ms)
- Tabel user: ID, Nama, Email, Role, Balance, Tanggal Daftar, button "Kelola Flags"
- Pagination
- Click "Kelola Flags" → expand form atau modal dengan 3 checkboxes (Checkout, Chat, Auction)
- Jika uncheck flag → textarea "Alasan" (required, min 10 char)
- Tampilkan current status flags (highlight merah jika disabled)
- Section Global Flags dengan warning banner dan 3 toggle: Auction Enabled, Chat Enabled, Checkout Enabled
- Jika toggle off → textarea "Alasan" (required, min 20 char)
- Tampilkan status global flags dengan warna (hijau/merah)
- Button "Simpan" dengan confirmation modal untuk global flags
- Toast notification success/error
- Loading state dan empty state

#### **5. Halaman Admin Login (Admin)**

Halaman login khusus untuk admin dengan JWT authentication. Tampilan dibebaskan, selama memenuhi ketentuan dari M1.

#### **Acceptance Criteria:**

Sama dengan Halaman Login di M1. Untuk akun Admin, silahkan HARDCODE di database.

### **UPDATE HALAMAN**

#### **1. Halaman Detail Produk (Buyer)**

Tambahkan tombol "Start Chat" ke toko terkait. Serta "Pergi ke Lelang" apabila barang terkait memiliki sesi lelang yang *available*.

#### **2. Halaman Product Management (Seller)**

Tambahkan opsi untuk memulai lelang. Kemudian munculkan sebuah popup / halaman baru untuk memasukkan informasi lelang, yaitu:

1. Jam mulai lelang
2. Kuantitas barang yang ingin di lelang
3. Minimum increment ( jumlah perubahan minimal dari bid terakhir)
4. Starting price

Berikan pula opsi untuk melihat lelang / hasil lelang kepada barang-barang yang terdaftar di lelang tertentu.

### **3. Halaman Profile (Buyer & Seller)**

Tambahkan Section "Pengaturan Notifikasi" dengan 3 checkboxes:

- a. Notifikasi Chat
- b. Notifikasi Lelang
- c. Notifikasi Pesanan

# **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](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. Review & Rating System**

Fitur *review* dan *rating* untuk produk setelah order diterima. Fitur ini juga memberikan kewenangan bagi admin untuk melakukan modifikasi secara khusus seperti yang dijelaskan pada ketentuan nomor 5.

### **Ketentuan:**

1. Buyer dapat memberikan review setelah order status = "received"
2. Review mencakup: rating (1-5 stars), text review (max 500 char), photo review (optional, max 3 images)
3. Menampilkan **average rating** dan **review count** di [product detail](#)
4. Menampilkan **latest reviews** di [product detail \(pagination\)](#)
5. Admin dapat me-moderasi reviews: hide inappropriate reviews, respond to reviews.
6. Seller dapat respond to reviews (one reply per review)

## 4. Payment Webhook Integration

Integrasi dengan payment gateway untuk menerima notifikasi status pembayaran secara real-time via webhook.

Ketentuan:

1. Integrasi dengan payment provider (pilih salah satu):
  - a. [Midtrans \(Sandbox/Development mode\)](#)
  - b. Xendit (Test mode)
  - c. Mock payment gateway (self-implemented)
2. Buyer dapat top-up balance atau pay order via payment gateway
3. Webhook endpoint untuk menerima notification dari payment provider
4. Update balance/order status otomatis setelah payment confirmed
5. Handle berbagai payment status: pending, success, failed, expired

Flow:

1. User klik "Top-up Balance" atau "Bayar Order"
2. Redirect ke payment gateway page (Midtrans/Xendit)
3. User complete payment
4. Payment gateway kirim webhook ke /api/node/webhook/payment
5. Verify webhook signature
6. Update balance/order status di database
7. Send push notification ke user

PERLU DIPERHATIKAN:

- **IDEMPOTENCY.** Pastikan update database tidak double jika webhook dikirim lebih dari sekali. *Payment gateway mungkin saja retry.*

Notes: Eksplor pula secara mandiri, baik langkah-langkah yang perlu dilakukan untuk mencegah security concern yang mungkin terjadi, ataupun best practice dalam melakukan payment webhook integration ini.

## 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 & API:** GET, POST, PUT, DELETE, REST API, JSON, AJAX, Fetch API, WebSocket, Socket.io, CORS, JWT
- **CSS & Styling:** margin, padding, flex, grid, Tailwind CSS, media queries, transitions
- **JavaScript & React:** useState, useEffect, useContext, React Router, addEventListener, fetch, async/await, map
- **PHP:** PDO, \$\_SESSION, \$\_POST, \$\_GET, json\_encode, json\_decode, session\_start, move\_uploaded\_file
- **Node.js:** Express.js, middleware, Socket.io, rooms, emit, broadcast, jsonwebtoken, bcrypt, web-push
- **SQL:** SELECT, INSERT, UPDATE, DELETE, JOIN, WHERE, ORDER BY, LIMIT, INDEX, TRANSACTION
- **Docker:** Dockerfile, Docker Compose, services, volumes, networks, environment variables
- **Nginx:** proxy\_pass, location, WebSocket proxy, static files, reverse proxy
- **Web APIs:** Notification API, pushManager, Service Worker, localStorage, WebSocket AP

Berikut adalah *tips* dari asisten tercinta.

- Pisahkan React components ke folder terpisah (components/, pages/, layouts/) dan buat **reusable components** seperti Button, Modal, dan Card untuk kemudahan maintenance. Gunakan **custom hooks** untuk extract logic yang sering dipakai seperti useWebSocket atau useAuth.
- Untuk Node.js backend, pisahkan routes, controllers, dan services agar kode menjadi lebih modular. Gunakan **.env** file untuk menyimpan **credentials**

seperti *JWT secret*, *VAPID keys*, dan *database credentials*. **JANGAN PERNAH** commit file `.env` ke git.

- Selalu **cleanup side effects** di `useEffect` seperti *disconnect WebSocket* dan *clear intervals* untuk mencegah **memory leaks**. Berikan key yang unik saat map array di React, jangan gunakan index sebagai key (**reference**).
- Untuk WebSocket, *verify session* atau *JWT* di setiap connection dan gunakan rooms untuk group connections seperti chat rooms atau auction sessions. **Handle disconnect/reconnect dengan baik** dan **implement auto-reconnect** di client-side.
- Buat **indexes** di database untuk columns yang sering di-query seperti foreign keys dan search fields. Gunakan prepared statements untuk prevent SQL injection dan gunakan **transactions** untuk operations yang harus **atomic** seperti deduct balance + create order.
- Dalam menggunakan Docker, pahami dulu perbedaan **`docker image`** dan **`docker container`**, serta cara kerja **`port forwarding`, `volume`**, dan **`network`**. Gunakan `.dockerignore` untuk exclude `node_modules` dan gunakan named volumes untuk data persistence seperti database data.
- Untuk Nginx, pahami cara **konfigurasi reverse proxy** dengan `proxy_pass` dan cara setup WebSocket proxy dengan `proxy_http_version`, `Upgrade`, dan `Connection` headers. Pisahkan routing untuk PHP, Node.js API, dan React static files.
- **Selalu validate dan sanitize input di server-side**, jangan hanya di client-side. Gunakan bcrypt untuk hash password, set expiration time untuk JWT, dan enforce feature flags di backend bukan hanya di UI.
- Implement **debouncing** untuk search input dan typing indicator (300-500ms), gunakan **pagination** untuk list data, dan optimize images dengan **compression** dan **lazy loading**. Minimize re-renders di React dengan **`React.memo`, `useMemo`, dan `useCallback`**.
- Test WebSocket dan Push Notification di **BERBAGAI browsers** (Chrome, Firefox, Safari) karena implementasinya bisa berbeda. Gunakan **Browser DevTools Network tab** untuk debug WebSocket messages dan API calls.

- **JANGAN COMMIT `node_modules`, `vendor folders`, atau `.env` ke git.**

# Lain-Lain

## 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. Silahkan lakukan perubahan di *repository* yang telah kalian buat bersama tim sebelumnya. Kalian diperbolehkan untuk melakukan perubahan atas M1, tetapi hal tersebut tidak akan merubah penilaian dari M1. Namun hal tersebut dapat membantu kalian untuk mengerjakan M2.
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 **9 Desember 2025 pukul 22.10 WIB**.
- Tahap pengumpulan adalah sebagai berikut.
  - a. Buat *release* dengan format **v2.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 v2.1, tetapi ternyata ada revisi sedikit, silakan buat rilis v2.2. **Tetap gunakan major version 2 yaitu v2.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

Kelompok yang digunakan tetap sama seperti pada Milestone 1.

## Pembagian Tugas

Setiap anggota kelompok **diwajibkan** untuk mengerjakan bagian *server-side* (PHP / Node) dan *client-side* (React), 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...)