

PEMBINA Phase 9 — Codebase Inspection Report

Generated: 2025-12-16 16:43 WIB

Status: Read-Only Inspection Completed

SECTION 1 — GLOBAL HEADER IMPLEMENTATION

Q1: What component renders the GLOBAL HEADER / TOP NAVBAR for PEMBINA?

- Component Name: TopNavbar
- File Path: src/components/layout/TopNavbar.tsx
- Shared or PEMBINA-specific: SHARED across all roles (STUDENT, PEMBINA, ADMIN)

Evidence:

- Used in [pembina/layout.tsx](#)
 - Used in [admin/layout.tsx](#)
 - Used in [StudentLayoutClient.tsx](#)
-

Q2: Does this header already include:

Feature	Status	Evidence
Notification bell icon	YES	<Bell className="h-5 w-5" /> at line 105
User/avatar menu	YES	Custom <DropdownMenu> with <Avatar> at lines 113-177

SECTION 2 — CLERK USER UI INTEGRATION

Q3: Is Clerk's <UserButton /> currently used anywhere?

- NO

No occurrences of <UserButton /> found in the codebase.

Q4: Is any Clerk-provided account management UI already embedded?

- NO

No occurrences of:

- <UserProfile />
- Clerk's native sign-out menu
- Clerk's session management components

Note: Sign-out is handled via custom implementation using `useClerk().signOut()` in `TopNavbar.tsx` line 66.

SECTION 3 — AVATAR SOURCE OF TRUTH

Q5: Where does the user avatar currently come from?

- Source: Clerk user image (`user.imageUrl`)

Evidence across all layouts:

Role	File	Avatar Source
PEMBINA	pembina/layout.tsx	avatarUrl: user?.imageUrl
ADMIN	admin/layout.tsx	avatarUrl: user?.imageUrl
STUDENT	StudentLayoutClient.tsx	avatarUrl: user?.imageUrl

Q6: Is avatar data stored in Prisma (User / Profile models)?

- YES
- Model: User
- Field: avatar_url (optional String?)

Evidence from [schema.prisma](#):

```
model User {  
  ...  
  avatar_url  String?  
  ...  
}
```

[!IMPORTANT] The `avatar_url` field exists in Prisma but is **NOT currently consumed** by the layouts. All layouts fetch avatar directly from Clerk's `user.imageUrl`.

SECTION 4 — ROLE ISOLATION CHECK

Q7: Is the global header:

- Answer: Conditionally rendered by role

Component: Fully shared (`TopNavbar.tsx`)

Behavior: Conditionally renders based on `user.role` prop

Conditional Logic Evidence:

Element	Behavior	Code Reference
Notification Bell Link	SISWA → <code>/student/notifications</code> Others → # (no-op)	Line 102: <code>href={user?.role === "SISWA" ? "/student/notifications" : "#"}</code>
Profile Menu Link	SISWA → <code>/student/profile</code> PEMBINA → <code>/pembina/profile</code> ADMIN → <code>/admin/profile</code>	Lines 147-153

SUMMARY TABLE

#	Question	Answer

1	Header component	TopNavbar in <code>src/components/layout/TopNavbar.tsx</code> (shared)
2a	Bell icon present	YES
2b	User/avatar menu present	YES (custom dropdown)
3	Clerk <code><UserButton /></code> used	NO
4	Clerk account UI embedded	NO
5	Avatar source	Clerk user.imageUrl
6	Avatar in Prisma	YES — <code>User.avatar_url</code> (but unused by layouts)
7	Header isolation	Conditionally rendered by role

IMPLICATIONS FOR PHASE 9

[!NOTE] Based on this inspection:

1. **Notifications for PEMBINA:** Bell icon exists but links to `#`. Requires new route `/pembina/notifications` and conditional update in `TopNavbar.tsx`.
2. **Profile for PEMBINA:** Dropdown links to `/pembina/profile`. Route must be created.
3. **Avatar Management:** Current system relies on Clerk. To allow local avatar upload, would need to:
 - o Consume `User.avatar_url` from Prisma
 - o Add fallback logic: Prisma → Clerk → fallback initial
4. **No Clerk UI Components:** Custom implementations required for any account management features.