

SOAL TIPE 2

APLIKASI PEMANTAUAN BAHAN BAKU MBG BERBASIS WEB

A. Gambaran Umum

Untuk mencegah kasus keracunan pada program Makan Siang Gratis (MBG), pengelolaan bahan baku menjadi hal yang sangat penting. Bahan yang sudah mendekati atau melewati tanggal kadaluarsa harus terdeteksi, dan hanya bahan layak konsumsi yang boleh digunakan dapur. Oleh karena itu, dibutuhkan sistem berbasis web sederhana untuk memastikan **bahan baku aman** (tidak kadaluarsa) dan **penyerahan bahan ke dapur** terkelola rapi. Sistem fokus pada:

- **Pemantauan stok & status kedaluwarsa** (Tersedia, segera kadaluarsa H-3, kadaluarsa/habis).
- **Permintaan bahan** oleh Petugas Dapur
- **Persetujuan Permintaan bahan** oleh Petugas Gudang (Challenge/ Bonus).
- **Pengurangan stok otomatis** saat permintaan di-ACC (Challenge/ Bonus).

B. Stakeholder yang Terlibat

1. **Petugas Gudang (Admin)**
Menginput bahan masuk, memantau status kedaluwarsa, mengelola stok, memproses (ACC/Tolak) permintaan bahan dari dapur.
2. **Petugas Dapur (Client)**
Mengajukan permintaan bahan (hanya bahan stok > 0 & tidak kadaluarsa), memantau status permintaan (disetujui/ditolak).

C. Kebutuhan Sistem

1. Fitur Petugas Gudang (Admin)

a. Tambah Bahan Baku

- Admin dapat menginput bahan baku baru dengan atribut: **nama**, **kategori**, **jumlah**, **satuan**, **tanggal_masuk**, **tanggal_kadaluarsa**.
- Sistem menyimpan data bahan baku ke database dengan status awal **Tersedia**.

b. Lihat Data Bahan Baku

- Sistem harus menampilkan daftar bahan baku dengan atribut lengkap termasuk status bahan baku (otomatis ditentukan oleh sistem).
- Sistem harus menghitung dan menampilkan status bahan baku berdasarkan aturan:
 - **Segera Kadaluarsa**: jika $\text{tanggal_kadaluarsa} - \text{hari_ini} \leq 3$ dan stok > 0.
 - **Kadaluarsa**: jika $\text{hari_ini} \geq \text{tanggal_kadaluarsa}$.

- **Tersedia:** jika stok > 0 dan tidak masuk kondisi di atas.
- **Habis:** jika jumlah = 0.

c. Update Stok Bahan Baku

- Admin dapat mengubah jumlah stok bahan baku.
- Sistem harus menolak update jika nilai stok < 0.

d. Hapus Bahan Baku

- Admin dapat menghapus data bahan baku Jika status ≠ kadaluarsa, sistem menolak aksi hapus.
- Pada saat admin melakukan penghapusan data, sistem terlebih dahulu menampilkan data bahan baku yang akan dihapus.
- Sistem hanya mengizinkan penghapusan bahan baku yang berstatus **kadaluarsa**. Tampilkan informasi bahwa bahan baku tersebut tidak dapat dihapus.

e. Proses Permintaan Bahan dari Dapur

- Sistem harus menampilkan daftar permintaan bahan dari bagian dapur dengan status **menunggu**.
- Admin dapat melakukan persetujuan permintaan dengan pilihan **(Challenge)**:
 - **Disetujui:** mengurangi stok sesuai jumlah yang dipesan.
 - Jika stok akhir = 0, sistem otomatis mengubah status bahan menjadi **habis**.
 - **Tolak:** mengubah status permintaan menjadi **ditolak**, dengan opsi menambahkan inputan alasan.
- Sistem harus memperbarui status permintaan menjadi **disetujui/ ditolak** sesuai dengan aksi dari Admin **(Challenge)**.

2. Fitur Petugas Dapur (Client)

a. Buat Permintaan Bahan (H-1)

- Bagian Dapur dapat melakukan permintaan bahan baku dengan input seperti dibawah ini:

Tanggal Masak : [date]

Menu yang akan dibuat : [teks]

Jumlah Porsi yang dibuat: [numeric]

Daftar Bahan Baku yang diminta :

No	Nama Bahan Baku	Jumlah
1	[teks]	[numerik]
2	[teks]	[numerik]

3	[teks]	[numerik]
---	--------	-----------

- Bagian dapur dapat menambahkan nama bahan baku beserta jumlahnya (tidak dibatasi jumlah bahan baku nya).
- Sistem harus menampilkan daftar bahan yang tersedia untuk dipilih, dengan syarat:
 - `stok > 0`
 - `status ≠ kadaluarsa`.
- Sistem harus menyimpan data permintaan dengan status awal **menunggu**.

b. Lihat Status Permintaan (Challenge).

- Sistem harus menampilkan daftar permintaan beserta status: **menunggu / disetujui / ditolak**.
- Sistem harus menampilkan detail permintaan (tgl_masak, menu, bahan, jumlah diminta).

D. Struktur Tabel

1. Tabel User

Field	Tipe Data	Length	FK	Auto Increment	Keterangan
id	INT	11	-	Yes	Primary Key
name	VARCHAR	100	-	-	Nama user
email	VARCHAR	150	-	-	Email unik
password	VARCHAR	255	-	-	Password yang diencrypt
role	ENUM('gudang','dapur')	-	-	-	Peran user
created_at	DATETIME	-	-	-	Waktu dibuat

2. Tabel bahan_baku


Field	Tipe Data	Length	FK	Auto Increment	Keterangan
id	INT	11	-	Yes	Primary Key
nama	VARCHAR	120	-	-	Nama bahan
kategori	VARCHAR	60	-	-	Kategori bahan
jumlah	INT	4	-	-	Stok tersedia
satuan	VARCHAR	20	-	-	Satuan bahan
tanggal_masuk	DATE	-	-	-	Tanggal masuk
tanggal_kadaluarsa	DATE	-	-	-	Tanggal kadaluarsa
status	ENUM('tersedia','segera_kadaluarsa','kadaluarsa','habis')	-	-	-	Status bahan
created_at	DATETIME	-	-	-	Waktu dibuat

3. Tabel permintaan

Field	Tipe Data	Length	FK	Auto Increment	Keterangan
id	INT	11	-	Yes	Primary Key
pemohon_id	INT	11	users.id	-	Relasi ke tabel users (role=dapur)
tgl_masak	DATE	-	-	-	Tanggal rencana memasak
menu_makan	VARCHAR	255	-	-	Deskripsi menu
jumlah_porsi	INT	4	-	-	Jumlah porsi
status	ENUM('menunggu','disetujui','ditolak')	-	-	-	Status permintaan
created_at	DATETIME	-	-	-	Waktu dibuat

4. Tabel permintaan_detail

Field	Tipe Data	Length	FK	Auto Increment	Keterangan
id	INT	11	-	Yes	Primary Key
permintaan_id	INT	11	permintaan.id	-	Relasi ke tabel permintaan
bahan_id	INT	11	bahan_baku.id	-	Relasi ke tabel bahan_baku
jumlah_diminta	INT	4	-	-	Jumlah bahan diminta

Contoh data dapat dilihat disini  [Soal Tipe 2 - Dummy Data](#) . Data boleh ditambah/ diubah sesuai kebutuhan.

Physical Data Model:

<https://dbdiagram.io/d/Proyek-3-Pemantauan-Bahan-Baku-MBG-68dca2add2b621e422b3ba81>

Data Instance:

<https://gist.github.com/alifiharafi/0d842caf727bc2ba2183938b826e87bc>

E. Aturan dan Validasi Umum

1. Tersedia Autentikasi (login/logout) sesuai role access yaitu bagian gudang (admin) dan bagian dapur (client). Login menggunakan email dan password.

Sample Akun :

name	email	password
Gudang A	gudang.a@example.com	gudang.a
Gudang B	gudang.b@example.com	gudang.b
Dapur A	dapur.a@example.com	dapur.a
Dapur B	dapur.b@example.com	dapur.b
Dapur C	dapur.c@example.com	dapur.c

2. Tambahkan Konfirmasi dialog pada saat pengguna melakukan aksi tambah/ ubah/ hapus. Jika ya proses penyimpanan ke database dilakukan, jika tidak proses dibatalkan.
3. Anda dapat menambahkan asumsi lain (tidak mengurangi) terkait desain tabel namun tetap memperhatikan kebutuhan sistem
4. Diperbolehkan menggunakan Framework CSS atau Alternatif Framework PHP lain selain CodeIgniter
5. Basis data boleh menggunakan alternatif lain selain MySQL

F. Tata Cara Pengumpulan

1. Buat Repository GitHub: Format : `MBG_NIM_Kelas_Nama`
2. Commit per Fitur (gunakan format ini, ubah `Nama_Mahasiswa`):
 - Fitur Login
 - Commit Message : `Add Fitur Login dan Logout by Nama_Mahasiswa`
 - Fitur Tambah Bahan Baku
 - Commit Message : `Add Fitur Tambah Bahan Baku by Nama_Mahasiswa`
 - Fitur Lihat Data Bahan Baku
 - Commit Message : `Add Fitur Lihat Bahan Baku by Nama_Mahasiswa`
 - Fitur Update Stok Bahan :
 - Commit Message : `Add Fitur Update Stok Bahan by Nama_Mahasiswa`
 - Fitur Hapus Bahan Kadaluarsa :
 - Commit Message : `Add Fitur Hapus Bahan Kadaluarsa by Nama_Mahasiswa`
 - Fitur Lihat Status Permintaan
 - Commit Message : `Add Fitur Lihat Status Permintaan di akun gudang by Nama_Mahasiswa`
 - Fitur Permintaan Bahan (Challenge) :
 - Commit Message : `Add Fitur Permintaan Bahan oleh akun Dapur by Nama_Mahasiswa`
 - Proses Persetujuan Permintaan dari Dapur (Challenge) :
 - Commit Message : `Add Fitur Persetujuan Permintaan oleh akun gudang by Nama_Mahasiswa`

G. Rubrik Penilaian

1. Penilaian Kemampuan Teknis (80%)

Kriteria	Dasar (<75)	Mahir (≥75)
1. Manipulasi Data dengan Framework (CRUD)	<ul style="list-style-type: none"> - Create: input & simpan data tanpa validasi/rule. - Read: menampilkan data sederhana (tanpa join / grouping). - Update: ubah data tanpa validasi/rule. - Delete: hapus data tanpa validasi/rule. 	<ul style="list-style-type: none"> - Create: input & simpan data dengan validasi/rule. - Read: menampilkan data dengan query JOIN atau grouping agregasi. - Update: ubah data dengan validasi/rule. - Delete: hapus data dengan validasi/rule.
2. Desain HTML & CSS	Tampilan menggunakan HTML & CSS dasar (tanpa framework).	Tampilan menggunakan framework CSS (misalnya Bootstrap, Tailwind) dengan komponen responsif.
3. JavaScript	Menggunakan JavaScript dasar (misalnya alert/confirm) tanpa manipulasi DOM.	Menggunakan JavaScript dengan memanfaatkan DOM (validasi input form, dynamic table, manipulasi elemen halaman).
4. Authentication	Login/logout tanpa role; semua user memiliki hak akses yang sama.	Login/logout dengan role (misalnya Admin vs Client/Public) untuk membatasi hak akses.
5. Clean Code	Tata tulis kode rapi (indentasi, komentar seperlunya, penamaan variabel konsisten).	Penerapan clean code lanjutan: kode modular dengan reusable method, komponen, atau model; minim duplikasi; struktur project terorganisir.

2. Penilaian Kualitas Luaran Aplikasi (20%)

Aspek	Kurang (0-59)	Sebagian (60-89)	Lengkap (90-100)
1. Kesesuaian dengan Spesifikasi	Aplikasi jauh dari spesifikasi, kebutuhan sistem inti banyak yang hilang.	Aplikasi memenuhi sebagian (minimal Read dan Create), ada beberapa fitur/aturan bisnis yang belum ada.	Aplikasi memenuhi seluruh kebutuhan sistem sesuai soal (CRUD, pencarian, validasi, role, dan aturan bisnis yang diminta).
2. Kelengkapan Fitur	Fitur sangat terbatas, hanya sebagian kecil dari requirement yang dibuat.	Hanya fitur inti yang tersedia (minimal 50%), fitur tambahan atau detail tidak dikerjakan.	Semua fitur utama tersedia sesuai soal (misalnya pengelolaan data master, transaksi, pencarian, detail, validasi, laporan sederhana).

3. Role & Hak Akses	Tidak ada penerapan role, semua user memiliki akses yang sama.	Role ada tetapi implementasi tidak konsisten atau hanya sebagian berjalan (Minimal 1 role berfungsi).	Role diterapkan sesuai kebutuhan (misalnya Admin vs Client/Public) dan bekerja konsisten.
4. Antarmuka & Usability	Antarmuka seadanya, sulit digunakan, tidak ramah pengguna.	Antarmuka cukup rapi namun navigasi atau alur penggunaan membingungkan.	Antarmuka rapi, navigasi jelas, user mudah memahami alur, ada feedback interaksi (notifikasi/konfirmasi).
5. Konsistensi Data & Aturan Bisnis	Data tidak konsisten, aturan bisnis utama tidak jalan, banyak bug yang mengganggu.	Data sebagian konsisten (minimal 1 tabel datanya muncul lengkap), namun ada bug/aturan bisnis belum terimplementasi penuh.	Data konsisten mengikuti aturan bisnis (misalnya stok otomatis berkurang, status otomatis, perhitungan gaji/take home pay benar).