**Rekapitulasi Rancangan & Alur Kerja Aplikasi CBT**

Dokumen ini memberikan gambaran menyeluruh tentang arsitektur, struktur, dan alur kerja aplikasi CBT yang telah kita bangun.

**1. Arsitektur & Teknologi**

Aplikasi ini dibangun di atas tumpukan teknologi berbasis web yang umum dan andal:

* **Backend (Logika Server):** **PHP** digunakan untuk semua pemrosesan di sisi server, termasuk interaksi dengan database, manajemen sesi, dan logika bisnis.
* **Frontend (Tampilan Pengguna):** **HTML** sebagai struktur, **Tailwind CSS** untuk styling yang cepat dan responsif, serta **JavaScript (JS)** untuk interaktivitas dinamis seperti modal (popup), validasi form, dan komunikasi *real-time* dengan server (AJAX).
* **Database:** **MySQL/MariaDB** digunakan untuk menyimpan semua data, mulai dari akun pengguna, soal, ujian, hingga hasil pengerjaan.

**2. Struktur Database**

Database adalah fondasi dari aplikasi ini, terdiri dari beberapa tabel utama yang saling terhubung:

* **users**: Menyimpan data login. Kolom role membedakan antara 'admin' dan 'student'.
* **questions**: Ini adalah **Bank Soal** utama. Menyimpan semua detail soal, termasuk teks, kategori, media (gambar/audio), dan opsi jawaban (disimpan dalam format **JSON** yang fleksibel).
* **tests**: Menyimpan informasi umum tentang setiap ujian, seperti judul, kategori, dan durasi.
* **test\_questions**: Tabel **penghubung** krusial. Tabel ini mencatat soal mana saja dari questions yang "dirakit" ke dalam ujian mana di tests. Di sinilah **urutan soal** dan **poin per soal** disimpan.
* **test\_results**: Merekam setiap sesi pengerjaan ujian. Setiap kali siswa memulai ujian, sebuah baris baru dibuat di sini untuk mencatat ID siswa, ID ujian, waktu mulai, dan nantinya skor akhir.
* **student\_answers**: Menyimpan setiap jawaban yang dipilih oleh siswa untuk setiap soal dalam sebuah sesi ujian.

**3. Alur Kerja Aplikasi Secara Detail**

Berikut adalah penjelasan langkah demi langkah tentang bagaimana pengguna berinteraksi dengan sistem.

**Alur Kerja Admin**

1. **Login:** Admin masuk melalui login.php. Sistem memverifikasi data dengan tabel users dan membuat sesi ($\_SESSION) dengan role 'admin'.
2. **Dasbor:** Admin diarahkan ke admin/index.php. Halaman ini secara dinamis menghitung dan menampilkan data statistik (jumlah ujian, soal, siswa) dan ringkasan hasil ujian terbaru.
3. **Manajemen (CRUD Dinamis):**
   * Untuk **Manajemen Ujian, Bank Soal, dan Siswa**, alurnya sama. Halaman utama (misal: manage\_tests.php) awalnya kosong.
   * JavaScript kemudian secara otomatis melakukan permintaan **AJAX** ke file itu sendiri (dengan parameter ?fetch\_list=true).
   * PHP mendeteksi permintaan ini, mengambil data dari database sesuai filter/pencarian, dan mengirimkannya kembali dalam format **JSON**.
   * JavaScript menerima data JSON ini dan "menggambar" tabel di layar.
   * Saat admin menekan tombol **Tambah/Edit**, sebuah modal (popup) muncul. Saat form di-submit, data dikirim lagi via **AJAX** ke file yang sama (dengan parameter action=add\_test), PHP memprosesnya, dan JavaScript me-refresh tabel tanpa perlu memuat ulang seluruh halaman.
4. **Rakit Ujian (assemble\_test.php):**
   * Admin memilih ujian, lalu diarahkan ke halaman perakitan.
   * Di sisi kiri, ditampilkan soal yang sudah ada di ujian tersebut, lengkap dengan input untuk **poin** dan antarmuka **drag-and-drop**.
   * Di sisi kanan, ditampilkan daftar soal dari Bank Soal dengan **filter, pencarian, dan paginasi (navigasi halaman)**.
   * Saat admin menyimpan perubahan (urutan/poin), JavaScript mengirimkan data baru via **AJAX** ke update\_test\_questions.php untuk disimpan.
5. **Laporan Hasil (reports.php):**
   * Sama seperti halaman manajemen lainnya, halaman ini menggunakan **AJAX** untuk menampilkan daftar semua test\_results yang sudah selesai.
   * Saat admin menekan tombol "Lihat Detail", JavaScript memanggil get\_result\_details.php via **AJAX** untuk mengambil semua detail jawaban dan menampilkannya dalam sebuah modal.

**Alur Kerja Siswa**

1. **Login:** Siswa masuk melalui login.php. Sistem membuat sesi dengan role 'student'.
2. **Dasbor Siswa (student/index.php):** Halaman ini menampilkan daftar semua ujian yang tersedia. Sistem juga memeriksa tabel test\_results untuk menentukan status setiap ujian bagi siswa tersebut (Mulai Ujian, Lanjutkan, atau Lihat Hasil).
3. **Konfirmasi Ujian (confirm\_page.php):** Sebelum memulai, siswa melihat detail ujian (jumlah soal, durasi, total poin) dan harus menekan tombol konfirmasi.
4. **Memulai Ujian:** Saat konfirmasi, sistem membuat baris baru di test\_results dengan status 'in\_progress' dan mencatat start\_time.
5. **Pengerjaan Ujian (test\_page.php):**
   * Halaman ini memuat semua soal yang telah dirakit untuk ujian tersebut.
   * Sebuah **timer** berjalan mundur berdasarkan durasi ujian.
   * Siswa dapat bernavigasi antar soal menggunakan tombol **Next/Back** atau panel navigasi. Soal yang sudah dijawab akan ditandai.
   * Soal yang memiliki **gambar atau audio** akan menampilkannya secara otomatis.
6. **Selesai Ujian:**
   * Saat waktu habis atau siswa menekan "Selesaikan Ujian", semua jawaban dikirim ke submit\_test.php.
   * submit\_test.php melakukan kalkulasi: membandingkan jawaban siswa dengan kunci jawaban, menghitung skor berdasarkan **poin per soal** yang telah diatur admin, lalu menyimpan setiap jawaban ke student\_answers dan skor akhir ke test\_results.
7. **Halaman Hasil (result\_page.php):**
   * Siswa diarahkan ke halaman ini untuk melihat **skor akhir** dan ringkasan jawaban (total benar/salah).
   * Terdapat tombol **dropdown** untuk melihat review detail dari setiap soal, jawaban yang dipilih, dan kunci jawaban yang benar.