Kelompok 12
Topik 4 – Perpustakaan Online

a) SP pada stakeholder : Member

- **Register (CREATE)**
  member_register.sql
  1. Menerima data diri lengkap member.
  2. Cek apakah username yang diterima sudah pernah digunakan sebelumnya di tabel Pengguna.
  3. Jika sudah pernah, return false.
  4. Sebaliknya, masukkan username dan password ke tabel Pengguna.
  5. Lalu, masukkan data diri lainnya ke tabel Member dengan foreign key id_pengguna yang mengarah ke primary key id_pengguna di tabel Pengguna.

- **Log in (sama dengan admin) (READ)**
  member_admin_login.sql
  1. Menerima username dan password.
  2. Cek apakah username ada di tabel Pengguna.
  3. Jika username ada di tabel Pengguna, cek apakah passwordnya sesuai.
  4. Jika username tidak ada di tabel Pengguna atau password salah, return false.
  5. Sebaliknya jika username dan password tepat, return true.

- **Edit Profile (UPDATE)**
  member_update_data.sql
  1. Menerima data diri member (kecuali username dan password tidak bisa diganti).
  2. Mengupdate data diri member yang ada di tabel Member sesuai dengan parameter data diri yang diterima.

- **Melihat Status Keanggotaan Saat Ini (READ)**
  member_lihat_status_keanggotaan_pribadi.sql
  1. Menerima id_member yang ingin dilihat status keanggotaannya.
  2. Return status_keanggotaan member tersebut dari tabel Member.

- **Melihat Menu Upgrade Keanggotaan (sama dengan admin) (READ)**
  member_admin_lihat_menu_keanggotaan.sql
  1. Return seluruh entry yang ada di tabel Keanggotaan.

- **Melakukan Transaksi Keanggotaan (CREATE)**
  member_transaksi_keanggotaan.sql
  1. Menerima id_member yang akan melakukan transaksi dan id_keanggotaan yang akan dibeli.
  2. Ambil data harga keanggotaan yang dibeli, tanggal dilakukannya transaksi.
  3. Masukkan entry baru ke dalam tabel Transaksi_Keanggotaan dengan status_validasi di set 0 (belum tervalidasi) dan id_admin diset 1 (akan diupdate menjadi id_admin yang memvalidasi nantinya).
  4. Return true

- **Melihat Seluruh Artikel (sama dengan admin menggunakan filtering SP) (READ)**

- **Filtering Artikel berdasarkan kategori, judul, penulis(READ)**
  member_filter_artikel.sql
  1. Menerima keyword filter yang diinginkan, termasuk kategori, judul, penulis dan status premium artikel.
  2. Buat query dynamic yang berisi pengambilan data dari function yang berfungsi untuk memfilter berdasarkan sejumlah n kategori (format string kategori 'kategori1,kategori2'), dengan syarat WHERE filter judul dan penulis.
  3. Setelah query lengkap untuk setiap filternya, maka eksekusikan query.

- **Memilih Artikel untuk dibaca (sama dengan admin) (READ)**
  member_admin_pilih_artikel.sql
  1. Menerima id_member yang akan membaca dan id_artikel yang dipilih.
  2. Cek apakah status keanggotaan member >= dengan kode is_premium pada tabel Artikel.
  3. Jika >= maka return entry artikel yang dipilih.
  4. Sebaliknya, maka return false

- **Mencatat log artikel yang dibaca (CREATE)**
  member_catat_log_baca_artikel.sql
  1. Menerima id_member yang sedang membaca dan id_artikel yang dibacanya.
  2. Ambil current date lalu masukan dengan parameter lainnya sebagai entry pada tabel Membaca.

- **Menulis Artikel (CREATE)**
  member_tulis_artikel.sql
  1. Menerima id_member yang menulis, disertai data diri artikel (nama, path, current date).
  2. Mengambil status_keanggotaan penulis artikel
  3. Menambahkan entry baru pada tabel Artikel dengan data artikel yang diterima, set status_validasi 0 an id_admin 1 (nanti akan diupdate dengan id admin yang memvalidasinya).

- **Melihat artikel-artikel yang pernah ditulis beserta status validasinya (READ)**
  member_penulis_lihat_artikel_dan_status.sql
  1. Menerima id_member pengakses
  2. Return data lengkap artikel dari tabel Artikel dengan id_penulisnya adalah id_member yang diterima.

- **Melihat artikel-artikel yang pernah dibaca oleh suatu member (READ)**
  member_lihat_log_baca_artikel.sql
  1. Menerima id_member yang ingin lihat log baca.
  2. Return entry pada tabel Membaca yang memiliki id_member sesuai.

- **Review artikel (CREATE)**
  member_review_artikel.sql
  1. Menerima id_member pemberi review, id_artikel yang direview, current date, komentar serta ratingnya.
  2. Tambahkan entry baru pada tabel Review.

3. Return true

- **Member melihat review artikel (READ)**
  member_lihat_review_artikel.sql
  1. Menerima id_artikel yang akan diakses.
  2. Return seluruh entry review dengan FK id_artikel yang sesuai dengan parameter.

- **Memfavoritkan artikel (CREATE / DELETE)**
  member_favorite_artikel.sql
  1. Menerima id_member, id_artikel yang difavoritkan, serta current date
  2. Masukan entry favoritkan artikel.

- **Melihat jumlah favorit sebuah artikel (READ)**
  member_lihat_jumlah_favorit_artikel
  1. Menerima id_artikel yang akan dilihat jumlah favoritnya.
  2. Apabila id_artikel berisi NULL, maka akan di-return jumlah favorit seluruh artikel yang berurut dari jumlah favorit tertinggi.
  3. Apabila id_artikel tidak NULL, maka return jumlah favorit dari id_artikel

b) SP pada stakeholder : Admin

- **Login (sama percis dengan member punya) (READ)**
  member_admin_login.sql
  1. Menerima username dan password .
  2. Cek apakah usernamenya ada di tabel Pengguna.
  3. Jika ada, cek apakah passwordnya sesuai. Jika sesuai, maka return true
  4. Sebaliknya return false.

- **Melihat daftar member (READ)**
  admin_melihat_daftar_member.sql
  1. Return seluruh entry dari tabel Member.

- **Melihat laporan seluruh transaksi keanggotaan (READ)**
  admin_lihat_daftar_transaksi_keanggotaan.sql
  1. Return seluruh entry dari tabel Transaksi_Keanggotaan.

- **Melihat laporan transaksi keanggotaan salah satu member (READ)**
  admin_lihat_transaksi_keanggotaan_satu_member.sql
  1. Menerima id_member yang akan dilihat laporan transaksinya.
  2. Return entry pada tabel Transaksi_Keanggotaan yang memiliki id_member sesuai.

- **Validasi transaksi keanggotaan (READ)**
  admin_validasi_transaksi_keanggotaan.sql
  1. Menerima id_transaksi_keanggotaan dan id_admin validator.
  2. Update value status_validasinya menjadi -1 jika ditolak, atau 1 jika diterima.
  3. Update value id_admin validator menjadi id_admin yang memvalidasi.

- **Melihat daftar seluruh tipe keanggotaan (sama dengan yang member) (READ)**
  member_admin_lihat_menu_keanggotaan.sql

1. Return seluruh entry dari tabel Keanggotaan

- **Mengupdate keterangan tipe keanggotaan (CREATE / UPDATE)**
  admin_update_daftar_tipe_keanggotaan.sql
  1. Menerima id_keanggotaan yang akan diupdate, id_admin, serta keterangan lainnya yang dibutuhkan untuk mengupdate keanggotaan.
  2. Cari keanggotaan yang dituju, lalu update data-datanya sesuai parameter yang diterima
  3. Update juga id_admin yang terakhir mengeditnya.

- **Melihat daftar seluruh artikel (sama dengan member menggunakan filtering SP) (READ)**

- **Filtering artikel berdasarkan status_validasinya (READ)**
  admin_filter_artikel.sql
  1. Menerima filter status_validasi.
  2. Return entry dari tabel artikel sesuai dengan filter status_validasi yang diinginkan.

- **Melihat artikel terpilih (sama dengan member) (READ)**
  member_admin_pilih_artikel.sql
  1. Menerima id_artikel yang akan dilihat.
  2. Return entry dari tabel Artikel sesuai dengan id yang diinginkan.

- **Validasi artikel (UPDATE)**
  admin_validasi_artikel_baru.sql
  1. Menerima id_admin validator, id_artikel yang akan divalidasi, status_validasi.
  2. Cari entry di tabel Artikel yang memiliki id sesuai dengan id_artikel yang diterima.
  3. Update value status_validasinya menjadi -1 jika tidak disetujui, atau 1 jika disetujui, serta update juga id_admin validatornya.

- **Validasi dan cek status aktif dari keanggotaan suatu member (READ & UPDATE)**
  admin_validasi_status_aktif_keanggotaan.sql
  1. Menerima id_member yang akan dicek status aktif keanggotaannya (cek aktivasi keanggotaan akan dilakukan tiap kali login).
  2. Ambil entry transaksi member tersebut, urutkan sehingga dapat diambil top 1 transaksi yang paling terakhir terjadi.
  3. Ambil juga durasi dari status keanggotaan yang dibelinya.
  4. Hitung sisa durasi berlakunya status keanggotaan tersebut.
  5. Jika selisih <= 0 maka update status_keanggotaan member menjadi Free /reguler. Sebaliknya jika masih ada sisa durasi berlakunya maka biarkan status_keanggotaan apa adanya.