

Nama : Yoga Pratama
Nim : 312210042
Kelas : TI. 22.A.1
Mata Kuliah : Pemrograman Web
UTS

1. Jelaskan secara singkat konsep dasar yang akan Anda gunakan dalam mengimplementasikan sistem alert ini untuk penanganan percobaan login yang gagal dengan menggunakan One Time Password (OTP).

Jawaban :

Konsep dasar yang akan digunakan dalam mengimplementasikan sistem alert untuk penanganan percobaan login yang gagal dengan menggunakan One Time Password (OTP) adalah sebagai berikut:

- Pengecekan Gagal Login: Setiap kali pengguna mencoba login, sistem akan memeriksa apakah percobaan login berhasil atau gagal.
 - Pengecekan Batas Gagal: Jika percobaan login gagal, sistem akan mencatatnya dan memeriksa apakah jumlah percobaan gagal telah melewati batas yang ditentukan (misalnya, tiga kali).
 - Pengiriman OTP: Jika batas percobaan gagal telah tercapai, sistem akan mengunci akun
 - pengguna gagal dalam mencoba login lebih dari tiga kali, akses mereka akan dikunci selama 10 menit sebelum mereka dapat mencoba login kembali.
 - Validasi OTP: Sistem akan memvalidasi OTP yang dimasukkan pengguna.
2. Buatlah algoritma atau pseudocode sederhana yang menggambarkan langkah-langkah dalam penanganan percobaan login yang gagal dan penerapan sistem alert, dengan mempertimbangkan penggunaan OTP melalui email, pesan teks (SMS), atau aplikasi pesan instan (WhatsApp).

Jawaban :

Berikut adalah algoritma dan pseudocode yang lebih lengkap untuk menangani percobaan login yang gagal dan penerapan sistem alert dengan menggunakan OTP yang dapat dikirimkan melalui email, SMS, atau WhatsApp:

Algoritma

- Inisialisasi Sistem:

Siapkan variabel untuk menyimpan jumlah percobaan login yang gagal, status kunci akun, dan informasi pengguna.

- Percobaan Login:

Periksa apakah akun pengguna terkunci. Jika terkunci, tampilkan pesan bahwa akun terkunci dan waktu penguncian tersisa.

- Verifikasi Kredensial:

Jika akun tidak terkunci, verifikasi kredensial pengguna.

Penanganan Gagal Login:

Jika kredensial salah, tingkatan jumlah percobaan login yang gagal.

Jika jumlah percobaan gagal melebihi tiga kali, kunci akun pengguna selama 10 menit dan kirimkan OTP melalui metode yang dipilih (email, SMS, atau WhatsApp).

- Pengiriman OTP:

Pilih metode pengiriman OTP (email, SMS, atau WhatsApp) berdasarkan preferensi pengguna.

Kirim OTP menggunakan layanan pihak ketiga yang sesuai.

- Validasi OTP:

Saat pengguna mencoba login lagi, validasi OTP yang dimasukkan.

Jika OTP valid dan kredensial benar, reset jumlah percobaan gagal dan berikan akses.

Jika OTP tidak valid, tampilkan pesan kesalahan.

Pseudocode

```
function login(username, password, method):
    if is_account_locked(username):
        if current_time < get_lock_end_time(username):
            display "Account is locked. Try again later."
            return
        else:
            unlock_account(username)

    if authenticate(username, password):
        reset_failed_attempts(username)
        grant_access(username)
    else:
        increment_failed_attempts(username)
        if get_failed_attempts(username) > 3:
            lock_account(username)
            otp = generate_otp()
            send_otp(username, otp, method)
            display "Account is locked due to multiple failed login
attempts. An OTP has been sent to your chosen method. Please try again
in 10 minutes."
        else:
```

```

        display "Login failed. Please check your credentials."

function authenticate(username, password):
    user = get_user(username)
    return user.password == hash(password)

function increment_failed_attempts(username):
    current_attempts = get_failed_attempts(username)
    set_failed_attempts(username, current_attempts + 1)

function reset_failed_attempts(username):
    set_failed_attempts(username, 0)

function get_failed_attempts(username):
    return db_query("SELECT failed_attempts FROM users WHERE username = ?", username)

function set_failed_attempts(username, attempts):
    db_query("UPDATE users SET failed_attempts = ? WHERE username = ?", attempts, username)

function lock_account(username):
    lock_end_time = current_time + 10 minutes
    set_account_lock(username, lock_end_time)

function unlock_account(username):
    clear_account_lock(username)
    reset_failed_attempts(username)

function is_account_locked(username):
    return get_lock_end_time(username) > current_time

function get_lock_end_time(username):
    return db_query("SELECT lock_end_time FROM users WHERE username = ?", username)

function set_account_lock(username, lock_end_time):
    db_query("UPDATE users SET lock_end_time = ? WHERE username = ?", lock_end_time, username)

function clear_account_lock(username):
    db_query("UPDATE users SET lock_end_time = NULL WHERE username = ?", username)

function generate_otp():
    return random_number(100000, 999999)

function send_otp(username, otp, method):

```

```

    if method == "email":
        send_otp_via_email(username, otp)
    elif method == "sms":
        send_otp_via_sms(username, otp)
    elif method == "whatsapp":
        send_otp_via_whatsapp(username, otp)

function send_otp_via_email(username, otp):
    email = get_user_email(username)
    email_body = "Your OTP is: " + otp
    send_email(email, "Your OTP Code", email_body)

function send_otp_via_sms(username, otp):
    phone_number = get_user_phone_number(username)
    sms_body = "Your OTP is: " + otp
    send_sms(phone_number, sms_body)

function send_otp_via_whatsapp(username, otp):
    phone_number = get_user_phone_number(username)
    whatsapp_body = "Your OTP is: " + otp
    send_whatsapp(phone_number, whatsapp_body)

function get_user_email(username):
    return db_query("SELECT email FROM users WHERE username = ?",
username)

function get_user_phone_number(username):
    return db_query("SELECT phone_number FROM users WHERE username =
?", username)

function validate_otp(username, entered_otp):
    stored_otp = get_stored_otp(username)
    return entered_otp == stored_otp

function get_stored_otp(username):
    return db_query("SELECT otp FROM users WHERE username = ?",
username)

function login_with_otp(username, password, otp):
    if is_account_locked(username):
        display "Account is locked. Try again later."
        return

    if validate_otp(username, otp):
        if authenticate(username, password):
            reset_failed_attempts(username)
            grant_access(username)
        else:

```

```
increment_failed_attempts(username)
display "Login failed. Please check your credentials."
else:
display "Invalid OTP. Please try again."
```

3. Identifikasi teknologi atau layanan pihak ketiga yang akan Anda gunakan untuk mengirimkan OTP melalui email, pesan teks (SMS), atau aplikasi pesan instan (WhatsApp), serta teknologi atau framework yang akan Anda gunakan untuk mengimplementasikan sistem alert ini dalam lingkungan web.

Jawaban :

Untuk pengiriman code otp saya menggunakan Pengiriman OTP nya melalui library phpmailer smtp

Sedangkan Teknologi/Framework:

Saya menggunakan berbagai teknologi dan framework dalam mengimplementasikan sistem alert ini, misalnya menggunakan Php, Mysql, Java script dan Css.

4. Diskusikan potensi risiko keamanan yang mungkin terjadi terkait dengan implementasi sistem alert ini, terutama dalam pengiriman dan penggunaan OTP melalui email, pesan teks (SMS), atau aplikasi pesan instan (WhatsApp), serta strategi yang akan Anda terapkan untuk mengurangi risiko tersebut.

Jawaban :

Ada beberapa potensi risiko keamanan terkait dengan pengiriman dan penggunaan OTP, seperti pencurian identitas, peretasan akun, dan serangan phishing. Strategi untuk mengurangi risiko tersebut meliputi enkripsi data, penggunaan HTTPS, validasi input, dan penerapan praktik keamanan terbaik.

5. Jelaskan bagaimana Anda akan mengintegrasikan sistem alert ini ke dalam sistem Siakad Kampus yang sudah ada, termasuk langkah-langkah yang perlu Anda ambil untuk memastikan ketersediaan dan performa yang optimal.

Jawaban :

- **Penambahan Modul Autentikasi:** Integrasikan modul autentikasi OTP ke dalam sistem Siakad yang sudah ada.
- **Penyesuaian Database:** Tambahkan tabel atau kolom untuk menyimpan informasi percobaan login dan waktu penguncian.
- **Pengujian dan Validasi:** Lakukan pengujian menyeluruh untuk memastikan bahwa fitur penguncian dan notifikasi berfungsi dengan baik.
- **Monitoring dan Logging:** Implementasikan logging untuk memantau percobaan login dan kinerja sistem.

- **Penggunaan LIBRARY PHPMAILER:** Gunakan PHPMAILER untuk menghubungkan sistem Siakad dengan layanan pengiriman OTP yang dipilih.

.