

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Penelitian**

Metode penelitian yang dipakai dalam studi ini adalah kualitatif deskriptif, yang bertujuan untuk menggambarkan dan menjelaskan hasil penelitian. Untuk pengembangan, digunakan Model waterfall yang berfokus pada penerapan MVC (*Model View Controller*) dalam merancang Aplikasi Presensi berbasis Web. Proses pengerjaannya dilakukan secara berurutan, tetapi perancangan dibatasi hanya sampai tahap Desain.

#### **3.2 Subjek Penelitian**

Subjek penelitian terdiri dari:

1. Pegawai di Kantor Kampung Tanjung Ratu Ilir yang akan menggunakan sistem daftar hadir.
2. Pengelola sistem yang bertanggung jawab atas implementasi dan pemeliharaan sistem.

#### **3.3 Pengumpulan Data**

Pada proses ini, beberapa cara digunakan untuk mengumpulkan data yang berkaitan dengan tujuan penelitian:

##### **3.3.1 Observasi**

Observasi ini dilakukan untuk memahami secara menyeluruh bagaimana proses administrasi presensi pegawai dijalankan di Kantor Kampung Tanjung Ratu. Fokus utamanya mencakup seluruh tahapan, mulai dari pencatatan kehadiran harian, perhitungan dan rekap data presensi selama periode tertentu, hingga penyusunan laporan yang mencatat kehadiran dan ketidakhadiran pegawai. Laporan ini nantinya menjadi dasar untuk mengevaluasi kinerja pegawai maupun keperluan administrasi lainnya. Melalui observasi ini, diharapkan dapat diperoleh gambaran yang jelas tentang seberapa efektif dan efisien sistem administrasi presensi yang digunakan.

##### **3.3.2 Wawancara**

Wawancara dilakukan dengan pegawai Kantor Kampung Tanjung Ratu Ilir untuk mengetahui pengalaman dan respon mereka terhadap penggunaan sistem presensi berbasis web. Wawancara dalam penelitian ini dilakukan secara terstruktur, yaitu dengan menyusun terlebih dahulu daftar pertanyaan yang telah disesuaikan dengan tujuan penelitian. Daftar

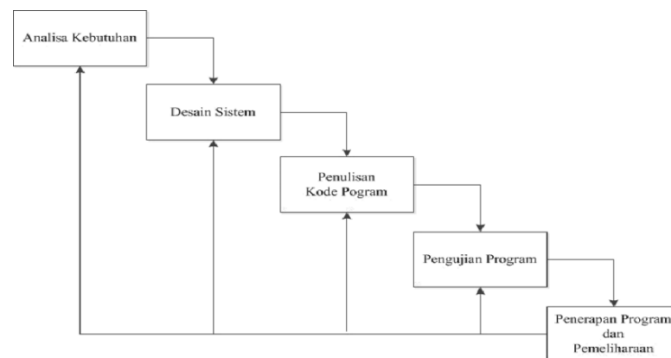
pertanyaan tersebut kemudian digunakan sebagai panduan saat mewawancarai informan di Kantor Kampung Tanjung Ratu Ilir. Dengan adanya pertanyaan yang telah dirancang sebelumnya, proses wawancara menjadi lebih terarah dan fokus, sehingga informasi yang diperoleh dapat menggambarkan kondisi nyata di lapangan serta mendukung pencapaian tujuan penelitian secara lebih efektif.

### 3.4 Rancangan Sistem dengan Metode Waterfall

Metode *Waterfall* atau yang sering disebut Model air terjun, merupakan salah satu pendekatan pengembangan perangkat lunak yang paling tua dan banyak digunakan. Model ini diperkenalkan pertama kali oleh **Winston W. Royce** pada tahun 1970. Sesuai dengan namanya, metode ini menggambarkan proses pengembangan sistem yang mengalir dari satu tahap ke tahap berikutnya, seperti air terjun yang mengalir ke bawah.

Menurut **Roger S. Pressman** dalam bukunya *Software Engineering: A Practitioner's Approach* (2010), Model Waterfall terdiri dari tahapan-tahapan yang saling terhubung secara linear. Artinya, setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya.

Model waterfall digunakan sebagai Gambaran dalam perancangan sistem presensi berbasis *web*, gambaran Model *waterfall* dapat dilihat pada Gambar 3.2 berikut;



Gambar 3. 1 Model Proses Waterfall

Rancangan sistem pada penelitian ini mengikuti tahapan model Waterfall yang disesuaikan secara langsung dengan kebutuhan operasional Kantor Kampung Tanjung Ratu Ilir. Model ini diterapkan bukan sekadar sebagai metodologi umum, tetapi sebagai alur kerja teknis yang digunakan mulai dari perumusan masalah absensi pegawai yang masih manual, sampai menghasilkan sistem daftar hadir berbasis web dengan arsitektur Model-View-Controller (MVC) yang benar-benar dapat digunakan oleh pegawai di lingkungan kantor. Pada tahap awal, analisis dilakukan untuk mengidentifikasi kendala nyata yang dihadapi pegawai seperti pencatatan absensi yang sering terlambat, tidak konsisten, serta sulit direkap oleh admin

kemudian tiap tahapan Waterfall diarahkan untuk menghasilkan solusi sistematis terhadap masalah tersebut. Setiap tahapan dirancang sedemikian rupa agar dapat menghasilkan sistem absensi yang mampu mencatat kehadiran secara real time, menyimpan data secara terpusat, menampilkan rekap otomatis, serta menyediakan antarmuka yang mudah digunakan oleh pegawai yang memiliki tingkat literasi digital beragam. Implementasi tahapan Waterfall ini mencakup analisis kebutuhan lapangan, perancangan alur proses MVC, implementasi modul absensi masuk–keluar, integrasi database berbasis waktu, pengujian fungsi kehadiran pada real environment kantor, hingga penyediaan mekanisme pemeliharaan yang berkelanjutan sesuai kebutuhan administrasi kepegawaian Kampung Tanjung Ratu Ilir. Dengan penerapan model Waterfall secara spesifik terhadap kondisi kantor, setiap tahap pengembangan tidak berhenti pada perancangan abstrak, tetapi menghasilkan komponen sistem yang secara langsung menjawab permasalahan absensi yang terjadi di lapangan.

### **3.5 Analisis Kebutuhan**

Analisis kebutuhan dilakukan melalui observasi langsung di Kantor Kampung Tanjung Ratu Ilir serta wawancara dengan pegawai dan admin yang bertanggung jawab terhadap pengelolaan presensi harian. Analisis ini menghasilkan gambaran mendalam mengenai alur kerja aktual dan permasalahan presensi manual yang selama ini digunakan.

#### **3.5.1 Kebutuhan Fungsional**

Kebutuhan fungsional merupakan fungsi inti yang harus tersedia dalam sistem agar proses presensi dapat berjalan efektif berdasarkan kondisi nyata lapangan.

#### **1. Fungsi Presensi Pegawai**

Hasil observasi menunjukkan pegawai sering mengalami antrean saat melakukan presensi manual pada pagi hari. Buku presensi yang diletakkan di meja depan kantor membuat pegawai harus datang bergantian untuk menandatangani, sehingga menimbulkan penumpukan.

Untuk mengatasi masalah tersebut, sistem dirancang memiliki:

- a. **Fitur Presensi Masuk dan Presensi Pulang** yang dapat dilakukan melalui ponsel masing-masing pegawai. Pada jam 08.00, pegawai dapat langsung menekan tombol presensi tanpa harus datang ke meja admin.
- b. **Notifikasi presensi berhasil**, karena beberapa pegawai sering ragu apakah presensi mereka tercatat saat masih menggunakan sistem manual.

- c. **Opsi keterangan (izin/sakit/hadir)** yang dapat dipilih pegawai jika mereka tidak bisa hadir ke kantor. Keterangan ini akan muncul otomatis di dashboard admin.

## **2. Fungsi Login dan Autentikasi**

- a. Setiap pegawai memiliki akun pribadi berisi username dan password yang diberikan oleh admin. Hal ini penting karena beberapa pegawai sebelumnya sering melakukan titip tanda tangan pada sistem manual.
- b. Sistem memvalidasi status akun: aktif/nonaktif, pegawai/admin, serta kecocokan password.
- c. Pegawai yang akunnya tidak aktif (misalnya pegawai honorer yang sudah tidak bertugas) tidak dapat melakukan presensi.

## **3. Fungsi Rekap Presensi**

- a. Admin dapat melihat rekap presensi harian, mingguan, dan bulanan. Sebelumnya admin harus menjumlahkan secara manual dari buku presensi, yang memakan waktu lama.
- b. Admin dapat melakukan koreksi waktu pulang untuk pegawai yang lupa presensi. Hal ini sesuai kejadian nyata yang cukup sering terjadi saat pegawai pulang terburu-buru.
- c. Sistem menyediakan opsi unduh laporan dalam format PDF untuk kebutuhan arsip dan pelaporan kepada kepala kampung.

## **4. Fungsi Pengelolaan Data Pegawai**

- a. Admin dapat menambah pegawai baru ketika ada penempatan personel tambahan.
- b. Admin dapat memperbarui data seperti jabatan, nomor induk pegawai lokal, dan username.
- c. Admin dapat menonaktifkan akun pegawai yang sudah berhenti bekerja. Ini penting karena beberapa pegawai honorer memiliki masa tugas tertentu.

### **3.5.2 Kebutuhan Non-Fungsional**

Di samping kebutuhan fungsional, sistem juga perlu memenuhi sejumlah kebutuhan non-fungsional yang berfokus pada kualitas, stabilitas, dan keandalan kinerja sistem secara keseluruhan. Kebutuhan ini memastikan bahwa sistem tidak hanya berfungsi sesuai tujuan,

tetapi juga aman, responsif, dan nyaman digunakan di berbagai situasi. Adapun beberapa kebutuhan non-fungsional yang diterapkan dalam pengembangan sistem ini antara lain:

1. Keamanan otentikasi pengguna.
2. Fitur *backup database* secara berkala.
3. Waktu respons sistem maksimal  $\leq 2$  detik.
4. Dukungan untuk berbagai browser dan perangkat mobile.

### 3.6 Perancangan Sistem

#### 3.6.1 Arsitektur MVC

Dalam pengembangan aplikasi ini, digunakan arsitektur Model-View-Controller (MVC) untuk memastikan sistem dibangun dengan struktur yang rapi, mudah dikembangkan, dan mudah dikelola. Pendekatan ini memisahkan antara bagian yang mengelola data (Model), tampilan antarmuka pengguna (View), serta logika atau alur proses aplikasi (*Controller*). Dengan memisahkan tanggung jawab setiap komponen, proses pengembangan menjadi lebih teratur dan sistem dapat dengan mudah diperbarui atau dikembangkan di kemudian hari.

##### 3.6.1.1 Model

Model adalah bagian yang berfungsi untuk mengelola data serta menjalankan logika yang berkaitan dengan sistem. Komponen ini terhubung langsung dengan *database* dan bertugas untuk mengambil, menyimpan, serta memperbarui data sesuai kebutuhan aplikasi.

Dalam sistem ini, terdapat dua tabel utama yang digunakan pada bagian Model, yaitu:

1. **Tabel Users:** digunakan untuk menyimpan informasi dasar pegawai seperti ID, username, password, dan *role*.
2. **Tabel presensi:** berisi data terkait kehadiran pegawai, seperti ID presensi, ID pegawai yang bersangkutan, tanggal kehadiran, waktu *check-in* dan *check-out*, status (hadir, izin, sakit, dll.), lokasi, serta keterangan tambahan bila ada.

##### 3.6.1.2 View

View merupakan bagian dari sistem yang langsung berinteraksi dengan pengguna. Komponen ini berfungsi untuk menampilkan data yang diambil dari Model serta menampung input atau perintah dari pengguna. Dalam sistem ini, View diwujudkan dalam bentuk berbagai halaman antarmuka, antara lain:

1. **Halaman Login:** digunakan oleh pengguna untuk masuk ke dalam sistem melalui proses otentikasi.

2. **Dashboard:** menyajikan ringkasan data seperti informasi presensi, grafik kehadiran, serta notifikasi penting.
3. **Halaman Laporan:** menampilkan data rekapitulasi kehadiran pegawai yang bisa difilter berdasarkan kriteria tertentu dan diekspor ke dalam format PDF.

### 3.6.1.3 Controller

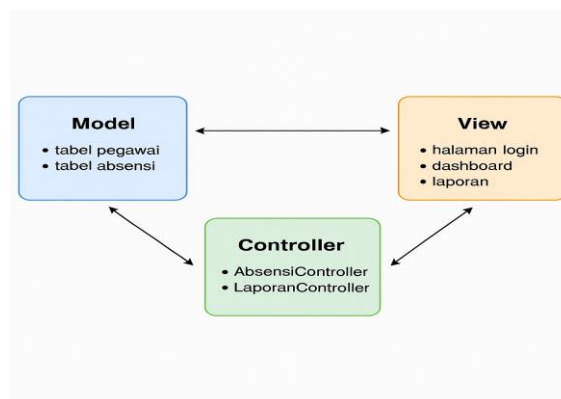
Controller bertugas sebagai penghubung antara View dan Model. Controller menerima input dari View, memprosesnya (termasuk berinteraksi dengan Model), lalu mengirimkan data kembali ke View. Beberapa Controller yang digunakan dalam sistem ini antara lain:

1. **AuthController:** mengelola proses *login* dan *logout* pengguna.
2. **PresensiController:** menangani proses *check-in* dan *check-out*, serta pengambilan data kehadiran.
3. **LaporanController:** mengelola proses pembuatan dan penyajian laporan presensi.
4. **PegawaiController:** mengatur data pegawai seperti tambah, edit, dan hapus.

Setiap komponen dalam arsitektur MVC saling berinteraksi dan membentuk alur kerja sistem yang terstruktur. Prosesnya dimulai ketika pengguna mengakses tampilan antarmuka, seperti halaman *login* atau *dashboard*, yang termasuk dalam komponen View.

View kemudian meneruskan permintaan tersebut ke *Controller*, misalnya *PresensiController*, yang bertugas mengelola logika aplikasi. Controller akan memproses permintaan tersebut dan, jika dibutuhkan, berkomunikasi dengan Model untuk mengambil atau memodifikasi data dalam *database* — seperti data pada tabel pegawai atau presensi.

Setelah mendapatkan data dari Model, Controller akan mengirimkan kembali hasilnya ke View agar informasi tersebut dapat ditampilkan kepada pengguna secara jelas dan interaktif, seperti yang dapat dilihat pada gambar 3.3.



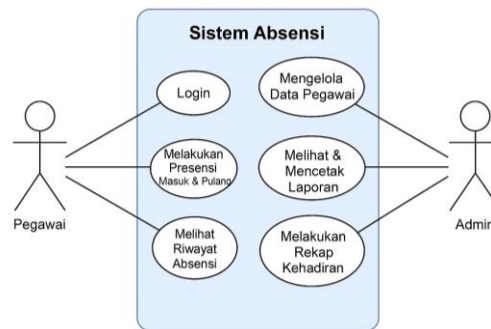
Gambar 3. 2 Relasi Pada MVC

Sumber : draw.io (Authors)

### 3.6.2 Diagram UML

#### 3.6.2.1 Use case

*Use case* diagram dirangkai yang merujuk pada kebutuhan input dan output yang telah dibuat sebelumnya. Bahwa pegawai dapat melakukan *login*, daftar akun, absen masuk dan pulang. Kemudian admin dapat melakukan *login*, melakukan rekap data presensi, melihat lokasi presensi pegawai, dan melihat data pegawai.



Gambar 3. 3 Use-case Sistem Presensi

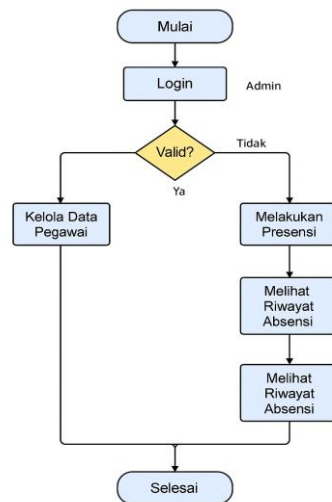
Sumber: draw.io (Authors)

Pada gambar 3.4 diagram ini menggambarkan interaksi antara pengguna (aktor) dengan sistem. Dalam sistem presensi ini, terdapat dua aktor utama, yaitu:

1. **Admin**, yang memiliki akses untuk mengelola data pegawai, melihat laporan, dan memantau presensi.
2. **Pegawai**, yang berinteraksi dengan sistem untuk melakukan presensi (absen masuk dan keluar).

#### 3.6.2.2 Flowchart

Langkah-langkah *flowchart* untuk Implementasi Sistem Daftar Hadir Pegawai Berbasis Web Menggunakan Algoritma Model-View-Controller (MVC) di Kantor Kampung Tanjung Ratu Ilir.



Gambar 3. 4 Flowchart Sistem Presensi

Sumber: draw.io (Authors)

1. **Mulai**

Proses dimulai ketika pegawai atau admin mengakses sistem presensi.

2. **Login**

Pengguna diminta untuk *login* dengan memasukkan username dan password. Sistem akan memverifikasi kredensial tersebut.

3. **Validasi**

**Login**

**(Valid?)**

Di tahap ini, sistem memeriksa apakah data *login* valid:

- Jika valid dan peran = Admin, maka diarahkan ke halaman kelola data pegawai.
- Jika valid dan peran = Pegawai, maka diarahkan ke fitur presensi.
- Jika tidak valid, sistem akan menampilkan pesan error (tidak digambarkan eksplisit di *flowchart*, bisa diasumsikan terjadi di balik layar).

4. **(Admin)**

**Kelola**

**Data**

**Pegawai**

Admin dapat mengakses dan mengelola data pegawai seperti menambah, mengedit, atau menghapus data.

5. **(Pegawai)**

**Melakukan**

**Presensi**

Pegawai melakukan presensi masuk atau pulang. Sistem mencatat waktu dan lokasi (jika berbasis lokasi).

6. **(Pegawai)**

**Melihat**

**Riwayat**

**Presensi**

Setelah melakukan presensi, pegawai dapat melihat histori kehadiran mereka.

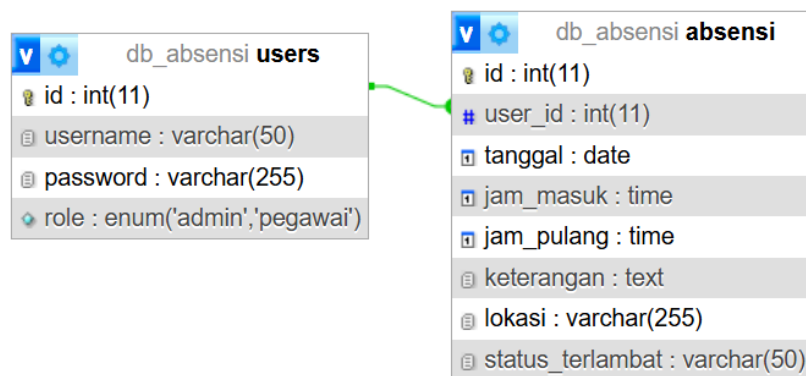
7. **Selesai**

Proses berakhir setelah pengguna menyelesaikan aktivitasnya.



### 3.6.2.3 Perancangan Database

Dalam sistem ini, rancangan databasenya terdiri dari dua tabel utama, yaitu *users* dan *presensi*. Tabel *users* menyimpan informasi pengguna seperti *id\_user*, nama, username, password, serta peran (admin atau pegawai). Sementara itu, tabel *presensi* digunakan untuk mencatat data kehadiran pegawai, yang mencakup *id\_presensi*, *id\_user* (sebagai kunci relasi ke tabel *users*), tanggal, jam masuk, jam keluar, status kehadiran, dan keterangan tambahan. Kedua tabel tersebut saling terhubung melalui *id\_user*, sehingga setiap catatan kehadiran bisa dikaitkan langsung dengan pengguna yang bersangkutan. Dengan relasi ini, sistem dapat mengelola serta menyajikan informasi presensi secara efisien berdasarkan data pengguna, seperti yang terlihat pada gambar 3.6.



Gambar 3. 5 ERD database sistem presensi

Sumber: localhost/phpmyadmin

### 3.6.3 Algoritma Penentuan Keterlambatan

Penentuan keterlambatan dalam sistem presensi dilakukan dengan membandingkan waktu kedatangan pegawai (*check-in time*) dengan batas waktu maksimal (*cut-off time*) yang telah ditentukan, ditambah dengan toleransi keterlambatan tertentu. Jika waktu *check-in* melebihi batas tersebut, maka pegawai dianggap terlambat dan status kehadirannya dicatat sebagai "Terlambat" (*Late*). Sistem kemudian secara otomatis menghitung jumlah menit keterlambatan dengan mengurangi waktu *check-in* dari waktu *cut-off*. Sebaliknya, jika pegawai melakukan *check-in* sebelum atau sama dengan waktu *cut-off* ditambah toleransi, maka statusnya akan tercatat sebagai "Tepat Waktu" (*On Time*) dan tidak ada menit keterlambatan yang dihitung. Dengan logika ini, sistem dapat mengidentifikasi dan membedakan secara adil antara kehadiran

yang tepat waktu dan yang terlambat, sambil tetap memberikan ruang toleransi yang wajar. Berikut adalah pseudocode dari logika tersebut:

```
if checkin_time <= cutoff_time + toleransi then
    status = "On Time"
    menit_terlambat = 0
else
    status = "Late"
    menit_terlambat = selisih(checkin_time, cutoff_time)
```

Berikut ini adalah penjelasan dari kode diatas:

1. cutoff\_time adalah batas jam masuk (misalnya pukul 08:00).
2. toleransi adalah waktu toleransi keterlambatan (misalnya 10 menit).
3. checkin\_time adalah waktu aktual pegawai melakukan presensi.
4. Jika pegawai datang lewat dari cutoff\_time + toleransi, maka status dicatat sebagai "Late", dan jumlah keterlambatan dihitung dalam menit dari cutoff\_time.
5. Jika masih dalam batas waktu dan toleransi, maka dianggap "On Time" tanpa keterlambatan.

### 3.7 Perancangan Keamanan dan Privasi

Dalam pengembangan sistem presensi ini, aspek keamanan dan privasi dirancang untuk melindungi data pengguna dan mencegah akses yang tidak sah. Sistem *login* telah menggunakan bcrypt untuk mengenkripsi password, sehingga walaupun terjadi kebocoran data, password pengguna tetap aman dan tidak bisa langsung dibaca.

Meskipun sistem saat ini **belum menerapkan CSRF token secara eksplisit**, keamanan tetap dijaga dengan membatasi akses hanya melalui metode **POST** dan validasi sesi pengguna yang aktif. Ke depan, penerapan *token* CSRF akan menjadi salah satu prioritas untuk meningkatkan perlindungan terhadap serangan pemalsuan permintaan.

Akses ke fitur-fitur tertentu diatur berdasarkan peran pengguna (*role*). Melalui penggunaan *middleware* dalam pola MVC, sistem memastikan bahwa hanya pengguna dengan hak tertentu, seperti admin, yang bisa mengakses halaman pengelolaan data. Pegawai hanya dapat menggunakan fitur sesuai dengan kewenangannya.

Untuk menjaga keandalan data, sistem ini juga dirancang dengan strategi *backup* dan pemulihan bencana (disaster recovery). Proses pencadangan data dijadwalkan secara berkala

dan disimpan di tempat yang aman, sehingga jika terjadi gangguan atau kehilangan data, sistem dapat dipulihkan dengan cepat dan data penting tidak hilang.

### 3.9 Pengujian (Testing)

Setelah sistem selesai dibuat, dilakukan serangkaian pengujian untuk memastikan bahwa semua fitur berjalan sesuai rencana. Pengujian dilakukan dalam dua tahap:

1. **Black-box testing**, untuk menguji apakah sistem merespons input dan menghasilkan output yang sesuai.
2. **Uji coba langsung oleh pengguna**, untuk mengetahui apakah sistem mudah digunakan dan sesuai dengan kebutuhan kerja mereka.

Dari pengujian ini, beberapa perbaikan kecil dilakukan, seperti penyesuaian tampilan dan perbaikan pada proses validasi input.

### 3.10 Penerapan (Deployment)

Setelah sistem dinyatakan siap, sistem diunggah ke server lokal atau hosting dan mulai digunakan secara terbatas oleh pegawai kantor kampung. Pada tahap ini, dilakukan juga pelatihan singkat kepada pengguna agar mereka memahami cara menggunakan sistem dengan benar.

### 3.11 Pemeliharaan (Maintenance)

Walaupun proyek ini bersifat penelitian, peneliti tetap membuka kemungkinan untuk melakukan perbaikan setelah sistem digunakan. Jika ditemukan *bug* atau pengguna mengajukan saran fitur tambahan, maka sistem dapat diperbarui sewaktu-waktu.