











**LEMBAR PENGESAHAN**

Nama Instansi : Badan Pusat Statistik

Bidang kajian : Sistem Layer Monitoring data dan informasi Survey pada website monitoring.bps.go id

Nama Mahasiswa : Hengky Rachmadhani G64120031

Andy Eka Saputra G64130036

Muhammad Alvian Supriadi G64130046

Muhammad Kamal Hidayatullah G64130072

Disetujui

Komisi Pembimbing

|  |  |
| --- | --- |
| Dr Ir Sri Wahjuni, MT  Pembimbing | M. Asyhar Agmalaro, S.Si, M.Kom  Pembimbing Lapangan |

Mengetahui,

Ketua Departemen Ilmu Komputer

Dr Ir Agus Buono, MSi, MKom

19660702 199302 1 001

**PRAKATA**

Puji dan syukur kami panjatkan kehadirat Allah *subhanahu wa ta’ala* yang telah melimpahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan Praktik Kerja Lapangan (PKL). Kegiatan PKL dilaksanakan selama kurun waktu 35 hari di Badan Pusat Statistik yang berlokasi di Jalan Dr. Sutomo no 6-8 Jakarta Pusat. Tujuan dari kegiatan ini adalah mendapatkan pengalaman di dunia kerja serta sebagai salah satu wadah dalam menerapkan pengetahuan yang dipelajari selama kuliah.

Dalam pelaksanaan PKL selama 40 hari ini kami ucapkan terima kasih atas bimbingan dan kerja sama dari beberapa pihak, yaitu:

1. Bapak Imam sebagai pembimbing di Badan Pusat Statistik
2. Pak Dudy selaku Manajer Departemen IT PT Goodyear Indonesia.
3. Pak Mizwar dan Pak Fahmi selaku *Senior* *IT* *Staff* di Badan Pusat Statistik,
4. Bapak Asyhar Agmalaro, S.Si, M.Kom, sebagai pembimbing akademik Departemen Ilmu Komputer IPB.
5. Semua pihak Badan Pusat Statistik yang turut serta membantu dalam kelancaran kegiatan PKL.

Kami menyadari bahwa laporan akhir PKL ini masih terdapat kekurangan sehingga kritik dan saran dari pembaca kami harapkan untuk perbaikan laporan akhir PKL ini. Untuk itu kami berharap semoga laporan akhir PKL ini dapat memberikan manfaat, khususnya bagi kami dan umumnya bagi para pembaca.

Bogor, November 2016

*Tim Penyusun*

**DAFTAR ISI**

[DAFTAR TABEL](#h.30j0zll)

[DAFTAR GAMBAR](#h.1fob9te)

[DAFTAR LAMPIRAN](#h.3znysh7)

[BADAN PUSAT STATISTIK](#h.2et92p0)

[Visi dan Misi](#h.tyjcwt)

[Struktur Organisasi](#h.3dy6vkm)

[Waktu dan Jadwal Praktik Kerja Lapangan](#h.1t3h5sf)

[Deskripsi Singkat Kajian](#h.2s8eyo1)

[SISTEM ROTATION SCHEDULE](#h.3rdcrjn)

[Latar Belakang](#h.26in1rg)

[Tujuan](#h.1ksv4uv)

[Ruang Lingkup](#h.44sinio)

[Metode Pengembangan](#h.2jxsxqh)

[Pengembangan Sistem Rotation Schedule](#h.3j2qqm3)

[1 *Communication*](#h.1y810tw)

[2 *Quick plan and modeling quick design*](#h.1ci93xb)

[3 *Construction of prototype*](#h.49x2ik5)

[4 *Deployment delivery and feedback*](#h.ihv636)

[SISTEM LAYER AUDIT DATABASE](#h.2grqrue)

[Latar Belakang](#h.vx1227)

[Tujuan](#h.3fwokq0)

[Ruang Lingkup](#h.1v1yuxt)

[Metode Pengembangan](#h.2u6wntf)

[Pengembangan Sistem Layer Audit Database](#h.3tbugp1)

[1 *Communication*](#h.28h4qwu)

[2 *Quick plan and modeling quick design*](#h.37m2jsg)

[3 *Construction of prototype*](#h.3l18frh)

[*4* *Deployment delivery and feedback*](#h.25b2l0r)

[SIMPULAN DAN SARAN](#h.43ky6rz)

[Simpulan](#h.2iq8gzs)

[Saran](#h.xvir7l)

[DAFTAR PUSTAKA](#h.1x0gk37)

[LAMPIRAN](#h.4h042r0)

DAFTAR TABEL

# 

[1 Kebutuhan fungsional Sistem Rotation Schedule](#h.4i7ojhp)

[2 Hasil pengujian halaman *user* *Rotation Schedule*](#h.32hioqz)

[3 Hasil pengujian halaman *admin* *Rotation Schedule*](#h.1hmsyys)

[4 Kebutuhan fungsional Sistem Layer Audit Database](#h.nmf14n)

[5 Hasil pengujian halaman *admin* *Layer Audit Database*](#h.kgcv8k)

[6 Hasil pengujian halaman *auditor* *Layer Audit Database*](#h.34g0dwd)

[7 Hasil pengujian halaman *owner* *Layer Audit Database*](#h.1jlao46)

# DAFTAR GAMBAR

[1 Struktur Organisasi PT Goodyear Indonesia](#h.4d34og8)

[2 *Flowchart Rotation Schedule* sebelum pembuatan sistem](#h.lnxbz9)

[3 *Flowchart Rotation Schedule* sesudah pembuatan sistem](#h.35nkun2)

[4 Paradigma *prototyping* (Pressman 2010)](#h.z337ya)

[5 *Use case diagram* Sistem Rotation Schedule](#h.3whwml4)

[6 *Context diagram* Sistem Rotation Schedule](#h.2bn6wsx)

[7 DFD Level 1](#h.qsh70q)

[8 DFD Level 2](#h.3as4poj)

[9 ERDSistem Rotation Schedule](#h.1pxezwc)

10 Halaman *Login*

11 Halaman *Rotation Schedule*

[12 Halaman *Template*](#h.2p2csry)

[13 Halaman *Example*](#h.147n2zr)

14 Halaman *Login* untuk *admin*

[15 Halaman *User Maintenance*](#h.3o7alnk)

[16 Halaman *Rotation Schedule*](#h.23ckvvd)

17 *Flowchart Layer Audit Database* sebelum pembuatan sistem

18 *Flowchart Layer Audit Database* sesudah pembuatan sistem

[19 Paradigma *prototyping* (Pressman 2010)](#h.19c6y18)

[20 *Use case diagram* Sistem Layer Audit Database](#h.1mrcu09)

[21 *Context* *diagram*](#h.46r0co2)

[22 DFD Level 1](#h.2lwamvv)

[23 ERDSistem Layer Audit Database](#h.111kx3o)

[24 Halaman *Login* untuk *user*](#h.206ipza)

[25 Halaman Audit](#h.4k668n3)

[26 Halaman *Calendar*](#h.2zbgiuw)

[27 Halaman *Login* untuk *admin*](#h.1egqt2p)

[28 Halaman *Schedule*](#h.3ygebqi)

[29 Halaman *Question*](#h.2dlolyb)

[30 Halaman *Result*](#h.sqyw64)

[31 Halaman *Database Auditor*](#h.3cqmetx)

[32 Halaman *Auditor Notification Logs*](#h.1rvwp1q)

[33 Halaman *Owner*](#h.4bvk7pj)

[34 Halaman *Notification Logs* (*Owner Area*)](#h.2r0uhxc)

[35 Halaman *Login* untuk *owner*](#h.1664s55)

[36 Halaman *Corrective Action*](#h.3q5sasy)

# DAFTAR LAMPIRAN

[1 Deskripsi *use case* Sistem Rotation Schedule](#h.2w5ecyt)

[2 Deskripsi *use case* Sistem Layer Audit Database](#h.1baon6m)

# BADAN PUSAT STATISTIK

Badan Pusat Statistik atau BPS merupakan sebuah lembaga pemerintahan non kementrian yang memiliki tanggung jawab kepada presiden. Pada awalnya BPS merupakan Biro pusat Statistik, yang dibentuk berdasarkan UU No 6 Tahun 1960 mengenai Sensus dan UU Nomor 7 tahun 1960 tentang Statistil. Sebagai Pengganti kedua UU tersebut ditetapkan UU Nomor 16 Tahun 1997 tentang Statistik. Peran dari Badan Pusat Statistik berdasarkan Undang 0 undang yaitu menyediakan kebutuhan data bagi pemerintah dan masyarakat, mengembangkan dan mempromosikan standar teknik dan metodologi statistik, menyediakan pelayanan pada bidang pendidikan dan pelatihan statistik, membangun kerjasama institusi internasional dan negara lain serta membantu kegiatan statistik didepartemen, lembaga pemerintah atau institusi

## Visi dan Misi

**Visi**:

Pelopor data statistik terpercaya untuk semua

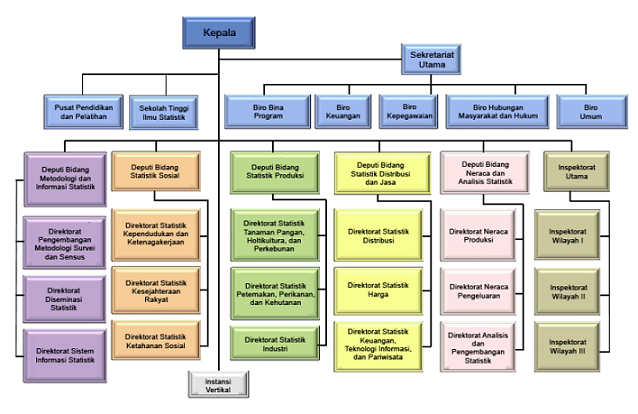
**Misi**:

1. Menyediakan data statistik berkualitas melalui kegiatan statistik yang terintegrasi dan berstandar nasional maupun internasional
2. Memperkuat Sistem Statistik Nasional yang berkesinambungan melalui pembinaan dan koordinasi di bidang statistik
3. membangun insan statistik nasional, berintegritas dan amanah untuk kemajuan perstatistikan.

sumber dapus https://www.bps.go.id/index.php/masterMenu/view/id/1

## Struktur Organisasi

Badan Pusat Statistik dipimpin oleh seorang kepala yang mempunyai tugas memimim BPS sesuai dengan ketentuan perundang-undangan yang berlaku, meyiapkan kebijakan nasional dan kebijakan umum ssuai dengan tugas BPS, menetapkan kebijakan teknis pelaksanaan tugas BPS yang menjadi tanggung jawabnya, serta membina dan melaksanakan kerjasama dengan instansi dan organisasi lain. Kepala dibantu oleh seorang Sekretaris Utama, Lima Deputi dan Insperktorat Utama, Struktur organisasi tersebut dapat dilihat pada Gambar 1.



## Waktu dan Jadwal Praktik Kerja Lapangan

Waktu pelaksanaan praktik kerja lapangan di Badan Pusat Statistik Pusat yaitu 35 hari kerja dengan 5 hari kerja per minggu dari hari Senin sampai Jum’at yang dimulai dari tanggal 11 Juni hingga 26 Agustus 2016. Waktu kerja pukul 07.30-16.00.

Gambar 1 Struktur Organisasi Badan Pusat Statistik Tahun 2015)

## Deskripsi Singkat Kajian

Dalam pelaksanaan praktik kerja lapangan di Badan Pusat Statistik, bidang kajian yang dikerjakan adalah Sistem Layer Monitoring Survey. Bidang kajian tersebut diberikan oleh divisi Sistem Informasi

Penanggung jawab dari sistem tersebut yaitu:

Hengky Rachmadhani G64120031

Andy Eka Saputra G64130036

Muhammad Alvian Supriadi G64130046

Muhammad Kamal Hidayatullah G64130072

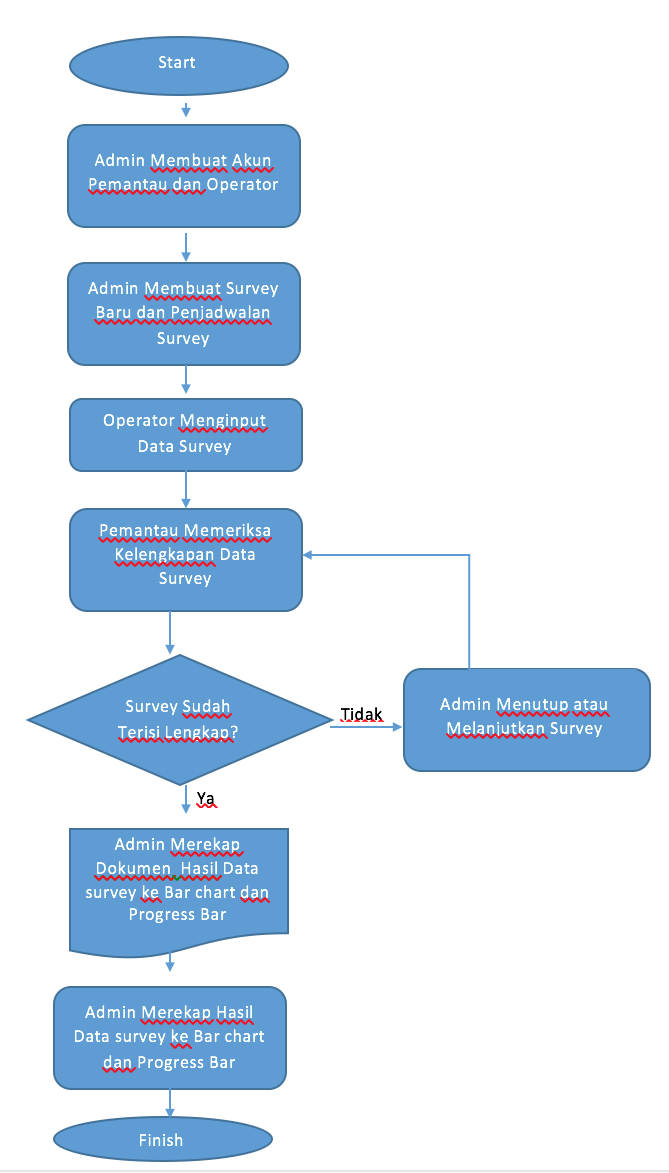
* **Sistem Layer Monitoring Survei**

Sistem ini dirancang dan dibangun untuk dapat melakukan pengisian data survey oleh pihak operator yang tersebar diberbagai daerah. Operator dapat mengisi data survei dengan 2 cara, meliputi *input entry data* melalui template yang telah ditentukan oleh SuperAdmin dan mengupload data survei melalui file excel. Proses pengisian data survei memiliki waktu *expire* sehingga pihak operator hanya bisa mengisi pada waktu yang ditentukan. Pihak pengawas akan melihat proses berjalannya suatu survei. Admin memiliki hak untuk CRUD pada tiap/sebuah provinsi, membatasi hak akses Operator dan Pemantau. SuperAdmin memiliki hak untuk membuat survei, penjadwalan, Menyusun tahapan dalam pembuatan survei, pemberian hak akses untuk (**Admin, Pemantau, Operator**), Menentukan target waktu pengisian survei. Membatasi hak akses dari Admin untuk CRUD pada suatu provinsi

## Latar Belakang

*Job rotation* atatu perputaran jabatan merupakan suatu mutasi personal yang dilakukan secara horizontal tanpa menimbulkan perubahan dalam hal gaji ataupun pangkat/golongan dengan tujuan untuk menambah pengetahuan seseorang tenaga kerja dan menghindarkan terjadinya kejenuhan. Dalam metode *job rotation* para karyawan yang mengalami perpindahan secara rutin dalam suatu pekerjaan kepekerjaan lain untuk memperoleh pengetahuan organisasi atau perusahaan secara menyeluruh (Wahyudi 2002).

Di Badan Pusat Statistik, departemen yang bertanggung jawab atas *Monitoring Sensus Survey* adalah departemen *Data Center*. Sebelumnya, sistem yang dibuat untuk *Monitoring Sensus* menggunakan Program atau Coding (Statis) sehingga proses ini masih bersifat manual. Dengan adanya Sistem Administrasi yang bersifat Dinamik, proses monitoring sensus menjadi terotomatisasi input dan output dataserta dapat meringankan pekerjaan dari departemen *Data Center*. Bagan alir *rotation schedule* sebelum menggunakan sistem dapat dilihat pada Gambar 2 dan bagan alir *rotation schedule* setelah menggunakan sistem dapat dilihat pada Gambar 3.

Gambar 2 *Flowchart Sistem Layer Monitoring*

## Tujuan

Tujuan dibuatnya Sistem Monitoring Data adalah:

1. Memudahkan setiap perwakilan departemen dalam mengatur sistem.
2. Memudahkan proses input dan output data sensus

## Ruang Lingkup

Ruang lingkup dari pembuatan Sistem Monitoring Administrasi yaitu:

1. Sistem dikembangkan pada sistem operasi Windows 10 dengan perangkat lunak Sublime, Notepad++. *Database* yang digunakan adalah Microsoft SQL Server. Sementara itu, Framework yang digunakan Laravel dengan bahasa pemrograman yang diimplementasikan pada sistem ini yaitu *PHP Hypertext Preprocessor*.
2. Sistem hanya diakses oleh Super Admin serta perwakilan setiap departemen pada Badan Pusat Statistik.

## Metode Pengembangan

Teknik dan metode yang digunakan untuk mengembangkan Sistem Monitoring Administrasi yaitu metode Extreme Programming (XP). Menurut keny Extreme Programming merupakan metode ringan, efisien, beresiko rendah, fleksibel, dapat diprediksi, ilmiah dan merupakan cara menyenangkan untuk mengembangkan software. Metode ini digunakan karena ingin membuat sistem yang fokus untuk menekankan kepuasan pelanggan. menekankan kerja tim, menekankan diskusi bersama client tentang proses bisnis dan apabila terjadi perubahan fungsinya.

Tahapan pada Extreme Programming untuk membangun sistem informasi Monitoring Administrasi yaitu:

1. Planning/Perencanaan

Pada tahap perencanaan ini dimulai dari pengumpulan kebutuhan yang membantu tim teknikal untuk memahami konteks bisnis dari sebuah aplikasi. Selain itu pada tahap ini juga mendefinisikanoutput yang akan dihasilkan, fitur yang dimiliki oleh aplikasi dan fungsi dari aplikasi yang dikembangkan.

1. Design/Perancangan

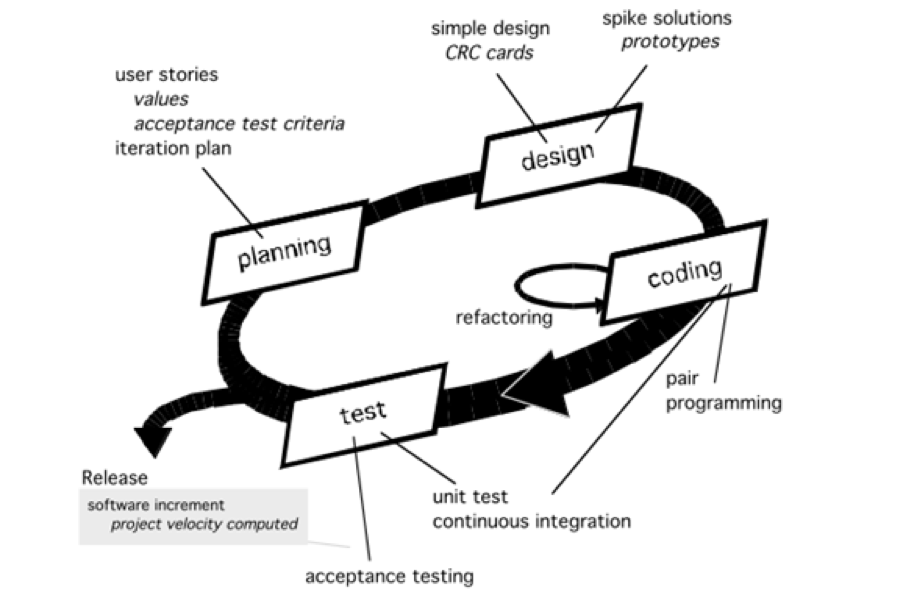
Metode ini menekankan desain aplikasi yang sederhana, untuk mendesain aplikasi dapat menggunakan Class-Responsibility-Collaborator(CRC) cards yang mengidentifikasi dan mengatur class pada object-oriented.

1. Coding/Implementasi

Konsep utama dari tahapan pengkodean pada extreme programming adalah pair programming, melibatkan lebih dari satu orang untuk menyusun kode.

1. Testing/Pengujian

Pada tahapan ini lebih fokus pada pengujian fitur dan fungsionalitas dari aplikasi

r

Gambar 4 Paradigma *Extreme Programming* (Pressman 2010)

Metode *Extreme Programming* memiliki tahapan sebagai berikut:

1. *Planning/Perencanaan*

Tahap ini yang dilakukan adalah wawancara untuk menghasilkan tabel - tabel dari fitur yang tersedia pada sistem Monitoring Administrasi.

1. *Design/Perancangan*

Tahap ini kami membuat use case diagram yang didasarkan pada tabel kebutuhan.

1. *Coding/Implementasi*

Bahas pemograman yang digunakan pada sistem monitoring admininstrasi adalah PHP dengan framework Lavarel.

## Pengembangan Sistem Monitoring Administrasi

Terdapat tiga tahapan dalam proses *extreme programming* yang digunakan dalam pengembangan Sistem Layer Monitoring Administrasi.

### *Planning/Perencanaan*

Dalam tahapan ini, kami melakukan pendefinisian masalah yang dihadapi oleh pengguna terkait sistem yang akan dirancang. Berikut merupakan identifikasi format Sistem Layer Monitoring.

a Analisis Pengguna Sistem

Dalam sistem ini penguna dibagi menjadi empay, yaitu:

1 *Super Admin*, yaitu perwakilan dari departemen Data Center memiliki hak membuat survei, penjadwalan, menyusun tahapan survei, pemberian dan membatasi hak akses (Admin, Pemantau, Operator).

2 *Admin*, yaitu perwakilan dari setiap departemen Data Center yang memiliki hak akses untuk CRUD pada setiap provinsi, memberikan dan membatasi hak akses Operator dan Pemantau.

3. Supervisor, yaitu perwakilan BPS dalam masing - masing wilayah memiliki hak akses memonitoring aktifitas operator sesuai pembagian wilayah yang diberikan admin.

4. Operator, yakni perwakilan yang bertugas pada wilayah yang ditentukan yang memilii hak akses menginput data

b Analisis Kebutuhan Fungsional

Berdasarkan hasil diskusi dengan *stakeholder*, kebutuhan fungsional Sistem Rotation Schedule dapat dilihat pada Tabel 1.

Tabel 1 Kebutuhan fungsional Monitoring Layer Administrasi

|  |  |  |
| --- | --- | --- |
| Kategori Pengguna | Kebutuhan Fungsional | Hak Akses Terhadap Aplikasi |
| *Super Admin* | Membuat *user* (Admin, operator & pemantau) | MLA-0001 |
| Mengubah data *user* | MLA-0002 |
| Menghapus *user* | MLA-0003 |
| Memberikan hak akses  Membuat Survei  Mengubah Survei  Menghapus Survei  Membuat Template Tabel  Menghapus Template Tabel | MLA-0004  MLA-0005  MLA-0006  MLA-0007  MLA-0008  MLA-0009 |
| *Admin* | Membuat user baru (operator, & pemantau) | MLA-0010 |
| Mengubah data user | MLA-0011 |
| Menghapus user | MLA-0012 |
| Memberikan hak akses | MLA-0008 |
| Operator | Memasukan data wilayah  Memonitoring survei | MLA-0015  MLA-0016 |

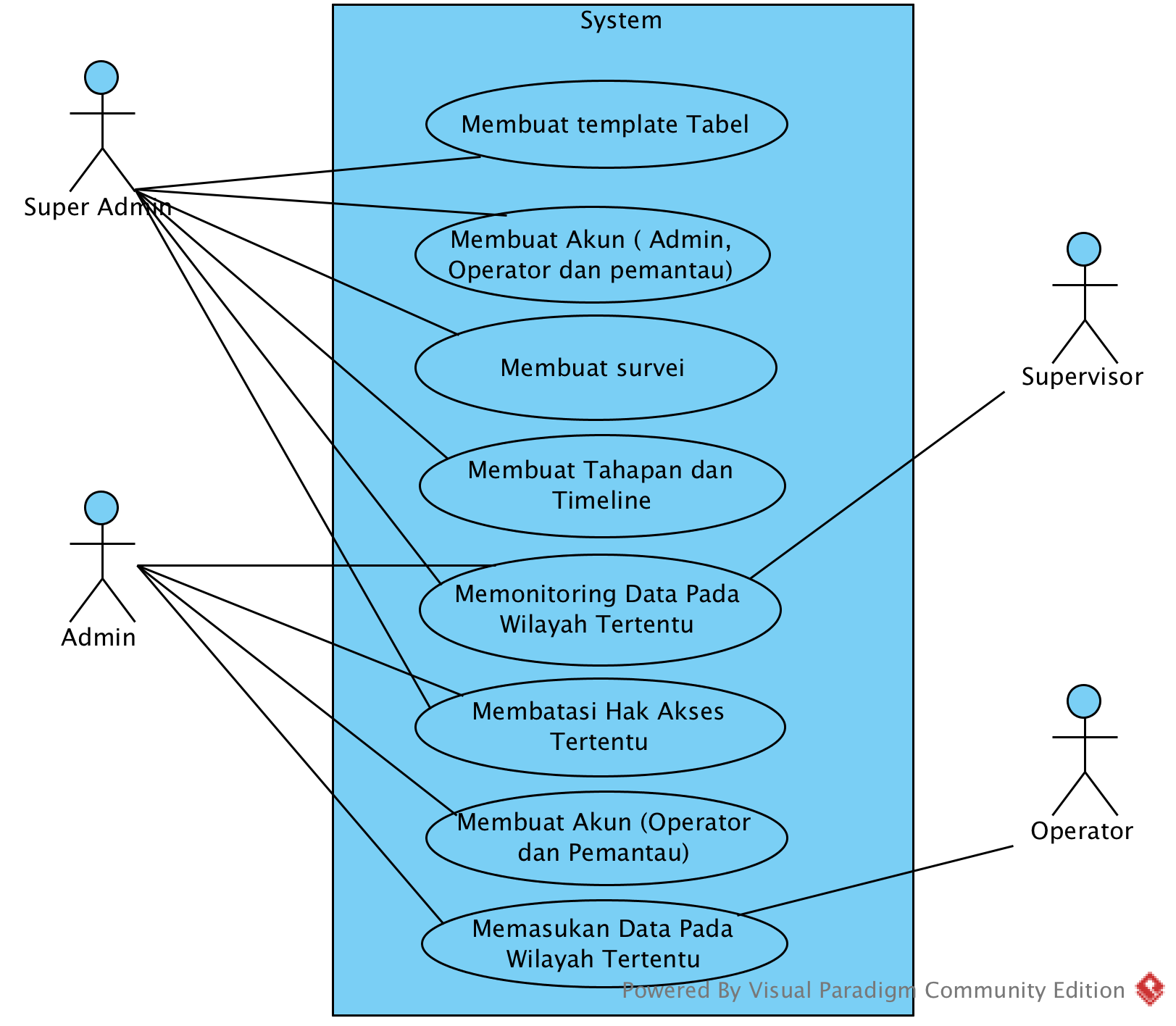
MLA-xxxx adalah kode yang digunakan untuk merepresentasikan kebutuhan (*requirement*) pada Sistem Monitoring Layer Administrasi dengan MLA merupakan kode sistem dan xxxx adalah digit/nomor kebutuhan (*requirement*).

### *Design / perancangan*

Pada tahap ini dibangun perancangan sementara yang merepresentasikan seluruh proses yang terjadi di dalam sistem, *input*, dan *output.* Representasi proses untuk Sistem Rotation Schedule dibangun dalam bentuk *use case diagram, context diagram,* dan *data flow diagram* level1 dan level 2*.*

2.1 *Use Case Diagram*

*Use case diagram* dirancang untuk menentukan fungsi-fungsi dari setiap aktor yang akan dibuat. Gambar 5 menunjukkan aktor dan fungsi yang terlibat dalam pembuatan Sistem Layer Monitoring.



Gambar 5 *Use case diagram* Sistem Rotation Schedule

Dalam Sistem Layer Monitoringterdapat empat aktor yaitu Super*Admin, Admin, Pemantau* dan *Operator*. *SuperAdmin* yang diwakili oleh departemen Data Center dapat melakukan enam fungsi utama yaitu membuat akun (Admin, Operator dan Pemantau), membuat survei, Membuat tahapan dan timeline, Memonitoring Data dari wilayah tertentu, membatasi hak akses tertentu. Admin yang diwakili oleh setiap departemen dapat melakukan lima fungsi utama yaitu membuat survei, membuat tahapan dan timeline, memonitoring data pada wilayah tertentu, membatasi hak akses tertentu, membuat akun (operator dan pemantau), memasukan data pada wilayah tertentu. Supervisor yang diwakili oleh perwakilan BPS dalam masing - masing wilayah memiliki fungsi utama yakni memonitoring data pada wilayah tertentu sedangkan operator memiliki fungsi utama yaitu memasukan data pada wilayah tertentu. Deskripsi masing - masing *use case* Sistem Layer Monitoring dapat dilihat pada Lampiran 1.

2.2 *Context Diagram*

*Context diagram* menjelaskan aliran data yang dimiliki sistem antara *SuperAdmin, Admin, Supervisor dan operator*, *admin,* dan Sistem Layer Monitoring. Gambar 6 menjelaskan aliran data yang direpresentasikan melalui *context diagram*.

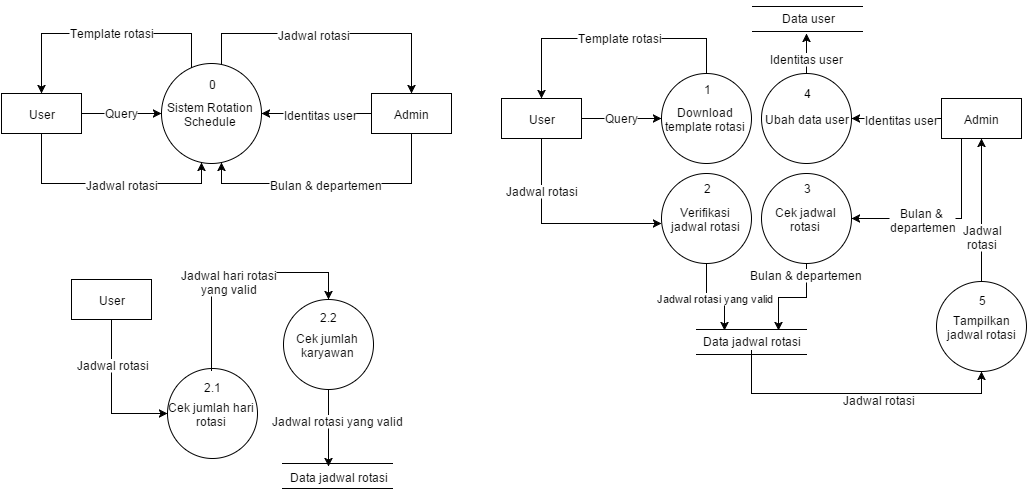
Survey making context diagram.png

Gambar 6 *Context diagram* Sistem Layer Monitoring

Pada *context diagram* digambarkan aliran data pada Sistem Layer Monitoringyaitu *SuperAdmin* melakukan proses pembuatan survei dengan di monitoring oleh Admin dan Supervisor lalu operator memasukan data survey dan diproses oleh admin dan super admin.

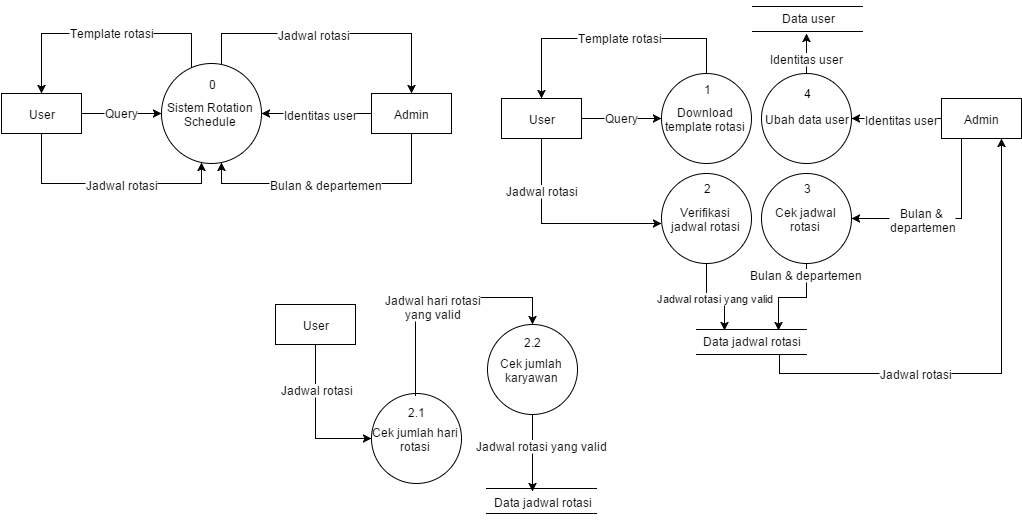
2.3 *Data Flow Diagram* (DFD)

DFD dirancang untuk menyusun tahapan yang harus dilakukan oleh *user* dan *admin* dalam mencapai fungsi utama dari Sistem Rotation Schedule. Gambar 7 menjelaskan tentang DFDlevel1yaitu perluasan aliran data yang dimiliki Sistem Rotation Schedule. Sementara itu, Gambar 8 menjelaskan tentang DFDlevel 2yaitu perincian dari proses padaDFDlevel 1.



Gambar 7 DFD Level 1

DFD level 1 menggambarkan aliran data yang dilakukan oleh *admin* dan *user* terhadap sistem. *User* memberi *input* berupa *query* untuk mengunduh *template* rotasi. Kemudian jadwal rotasi yang diunggah oleh user diverifikasi oleh sistem. Jadwal rotasi yang *valid* akan tersimpan di *database* dan ditampilkan oleh sistem. Sementara *admin* dapat mengecek jadwal rotasi sesuai *input* bulan dan departemen. *Admin* juga dapat mengubah data *user* dengan *input* identitas *user*.



Gambar 8 DFD Level 2

Pada DFDlevel 2 digambarkan secara rinci aliran data ketika *user* memberi *input* berupa jadwal rotasi yang diunggah ke dalam sistem. Sistem akan mengecek jumlah hari pada jadwal rotasi. Apabila *valid*, sistem akan melanjutkan pengecekan terhadap jumlah karyawan. Jika jumlah karyawan sesuai dengan yang ada di dalam *database*, maka jadwal rotasi tersebut *valid* dan dapat ditampilkan oleh sistem.

2.4 *Entity Relationship Diagram* (ERD)

ERD menggambarkan relasi antar objek yang digunakan pada Sistem Layer Monitoring. Sistem ini menggunakan empat entitas yaitu otoritas, provinsi,kabkot, users, survey, dan transaksi data survey. Setiap entitas tersebut memiliki hubungan tersendiri dengan entitas lainnya. Hubungan tersebut dapat dilihat pada Gambar 9.

Entitas departemen memiliki hubungan *one-to-many* dengan entitas karyawan. Satu departemen dapat memiliki banyak karyawan dan satu karyawan memiliki satu departemen. Selain itu entitas departemen memilki hubungan *one-to-one* dengan entitas *user*. Satu departemen memiliki satu *user* yang dapat mengakses Sistem Rotation Schedule. Sementara itu, entitas karyawan memiliki hubungan *one-to-many* dengan entitas *rotation* *schedule*. Satu karyawan dapat memiliki banyak *rotation schedule* maupun tidak dan satu *rotation schedule* dapat dimiliki oleh satu karyawan.

ERD BPS.png

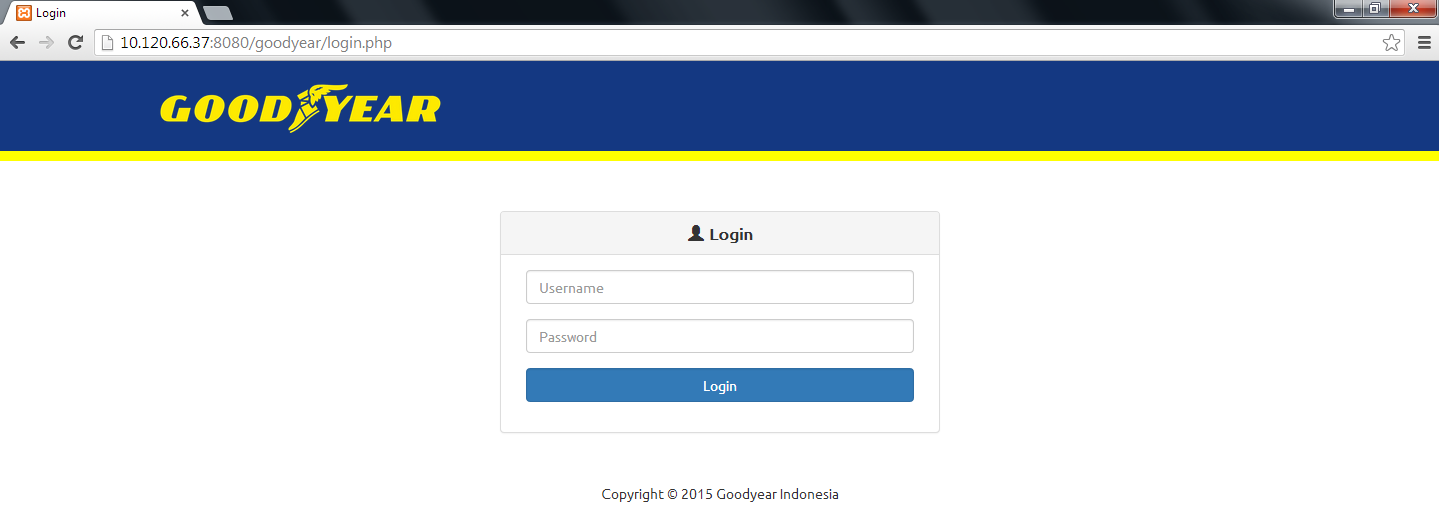
Gambar 9 ERD Sistem Layer Monitoring

### *Construction of prototype*

Pada tahap ini dilakukan implementasi hasil perancangan pada tahap *design* menjadi baris kode program sesuai bahasa pemrograman yang digunakan. Dalam sistem untuk *user* terdapat beberapa halaman yang diimplementasikan, diantaranya sebagai berikut:

* 1. Halaman *Login*

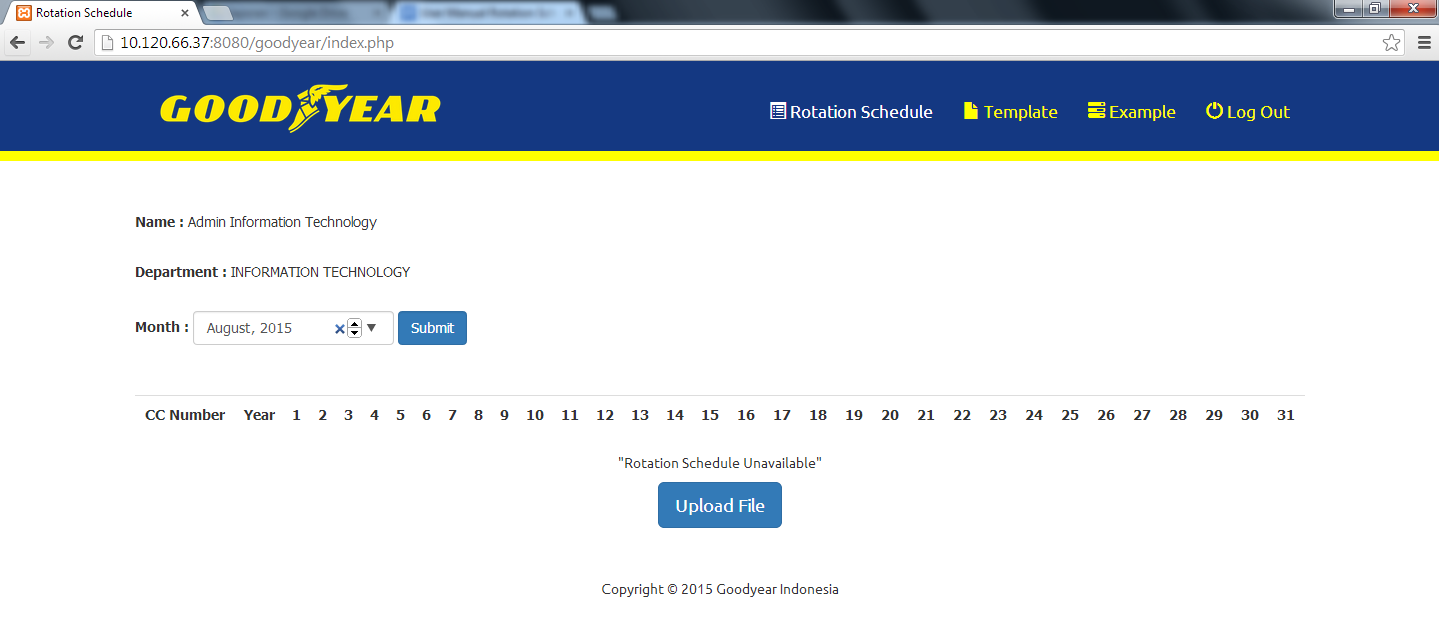
Halaman *Login* diimplementasikan untuk masuk dalam Sistem RotationSchedule dengan menggunakan *input username* dan *password*. Tampilan Halaman *Login* dapat dilihat pada Gambar 10.



* 1. Halaman *Rotation Schedule*

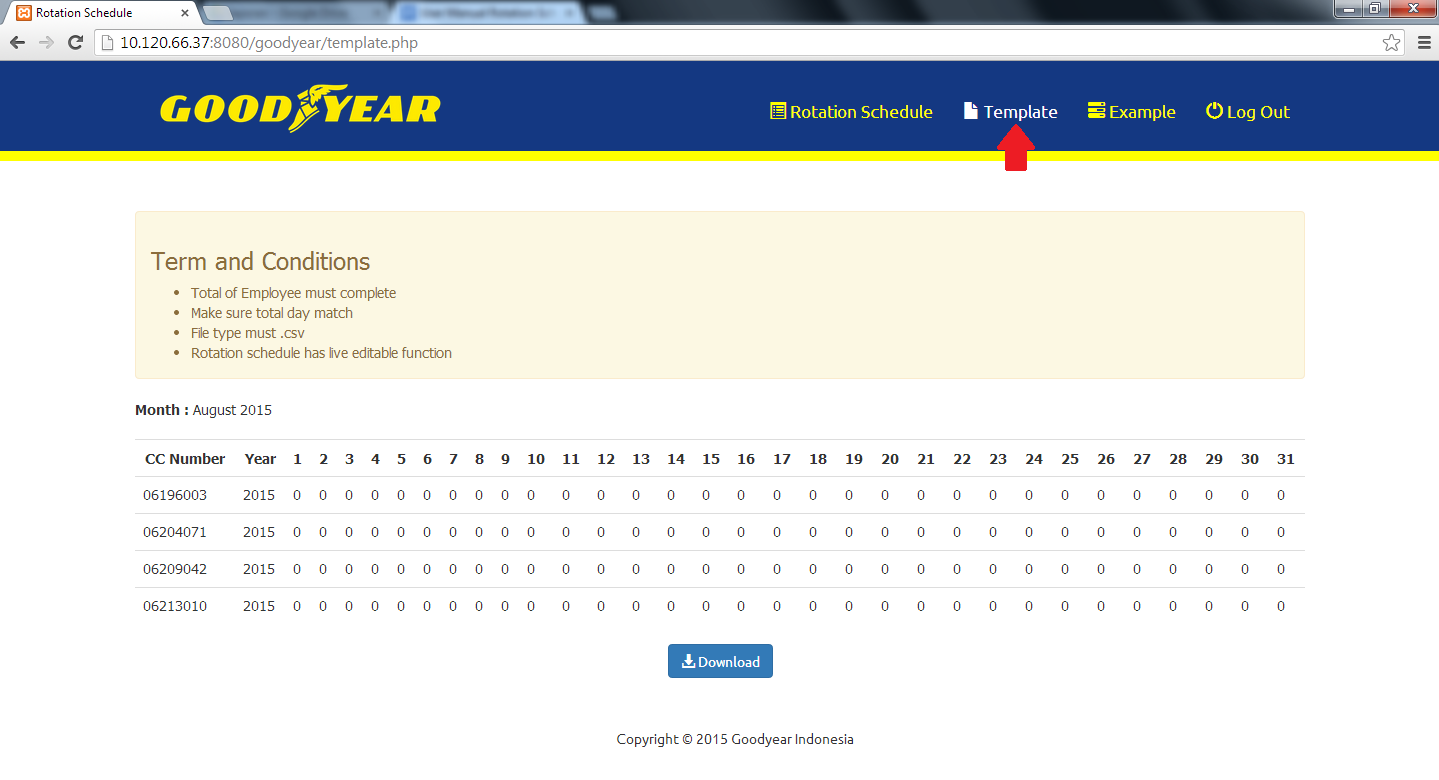
Halaman *Rotation Schedule* diimplementasikan untuk melakukan akses terhadap jadwal rotasi. Tabel jadwal rotasi berisi *cc\_number* yang merupakan nomor identitas karyawan, tahun, dan kode *job rotation*. Pada halaman ini user dapat mengunggah, mengunduh serta menghapus tabel jadwal rotasi sesuai dengan bulan yang dipilih. Tampilan Halaman *Rotation Schedule* dapat dilihat pada Gambar 11.





* 1. Halaman *Template*

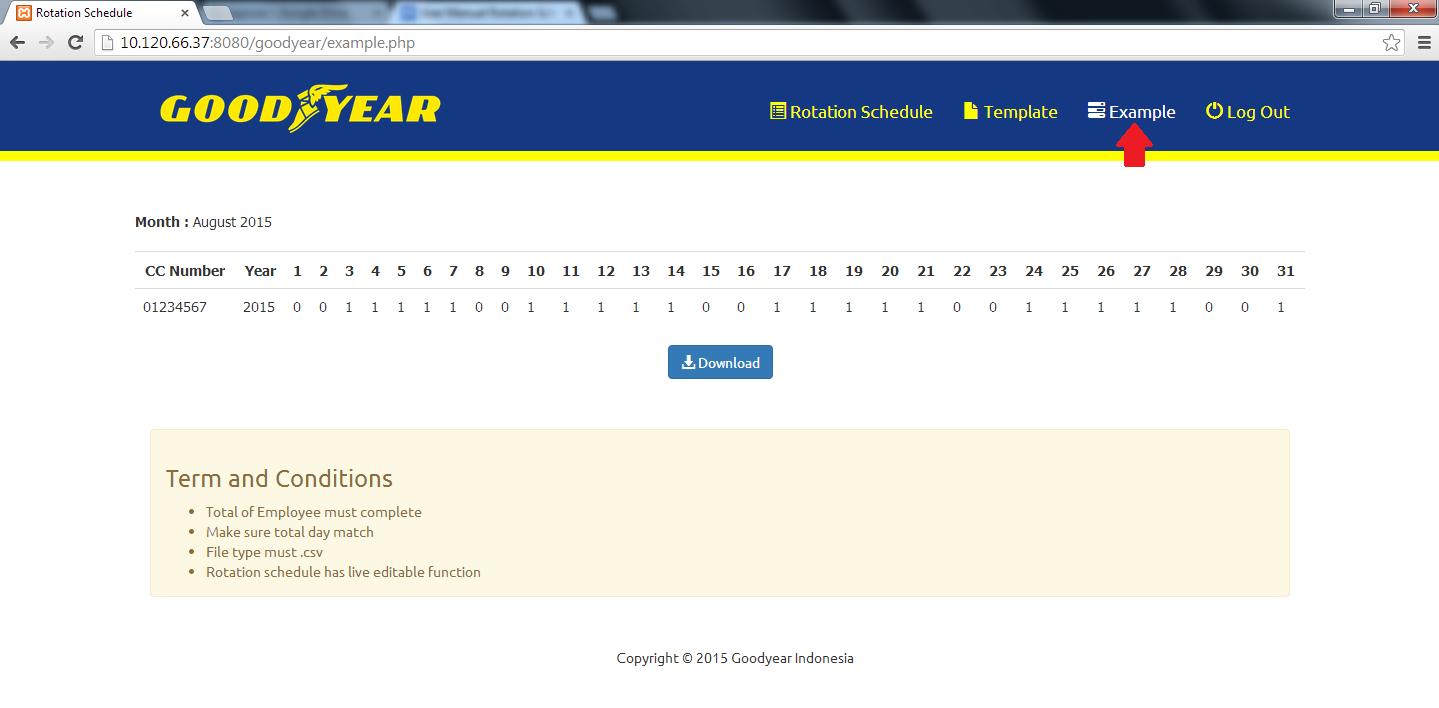
Halaman *Template* ditujukan untuk memudahkan *user* mengetahui tipe dokumen dan format *rotation schedule* yang benar. Dalam halaman ini terdapat fungsi *download*. Hasil dokumen unduhan tersebut berisi format *rotation schedule* dengan jumlah hari yang sesuai pada bulan saat itu dan jumlah karyawan yang lengkap sesuai departemen terkait. Pada halaman tersebut terdapat *Term and Condition* yang berguna sebagai petunjuk untuk *user* agar dapat meminimalisir kesalahan. Tampilan Halaman *Template* dapat dilihat pada Gambar 12.



Gambar 12 Halaman *Template*

* 1. Halaman *Example*

Halaman *Example* diimplementasikan untuk mengunduh *example* dari jadwal rotasi. Halaman *Example* ditujukan untuk memudahkan *user* mengetahui pola *rotation schedule* yang benar dan sesuai dengan bulan saat itu. Pola tersebut didasari jumlah hari kerja departemen. Jika departemen yang memiliki hari kerja dari hari Senin hingga Sabtu maka pada hari Minggu pola yang muncul yaitu 0. Kemudian jika departemen yang memiliki hari kerja dari hari Senin hingga Jumat maka pada hari Sabtu dan Minggu pola yang muncul yaitu 0. Pada halaman tersebut terdapat *Term and Condition* yang berguna sebagai petunjuk untuk *user* agar dapat meminimalisir kesalahan. Tampilan Halaman *Example* dapat dilihat pada Gambar 13.

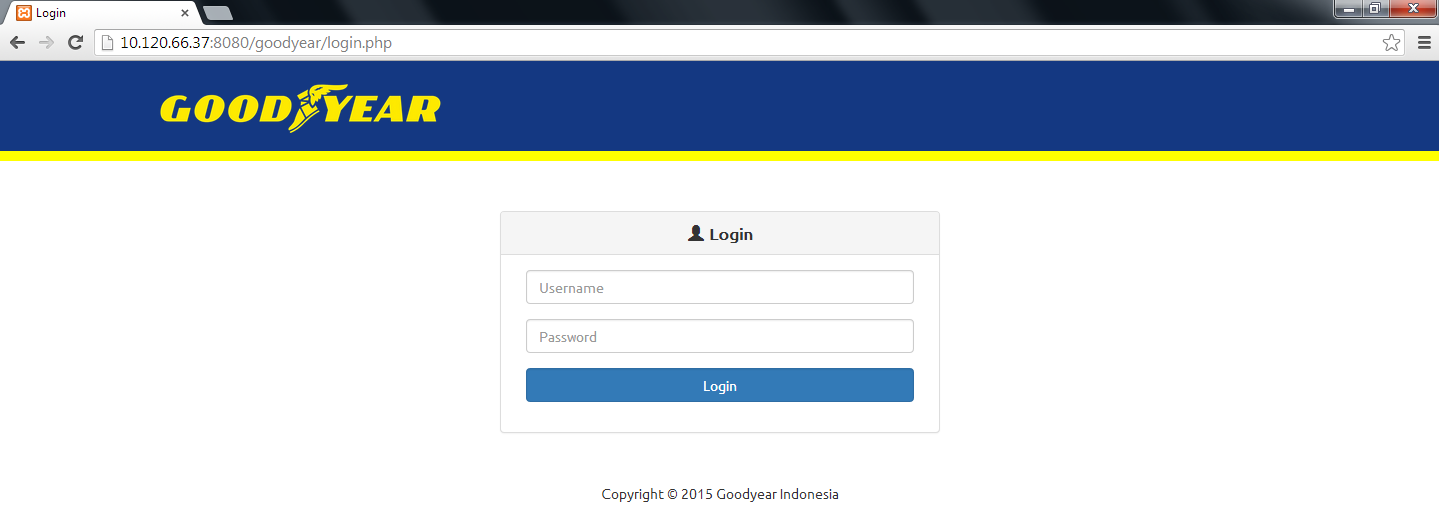


Gambar 13 Halaman *Example*

Dalam sistem untuk *admin* terdapat beberapa halaman, diantaranya sebagai berikut:

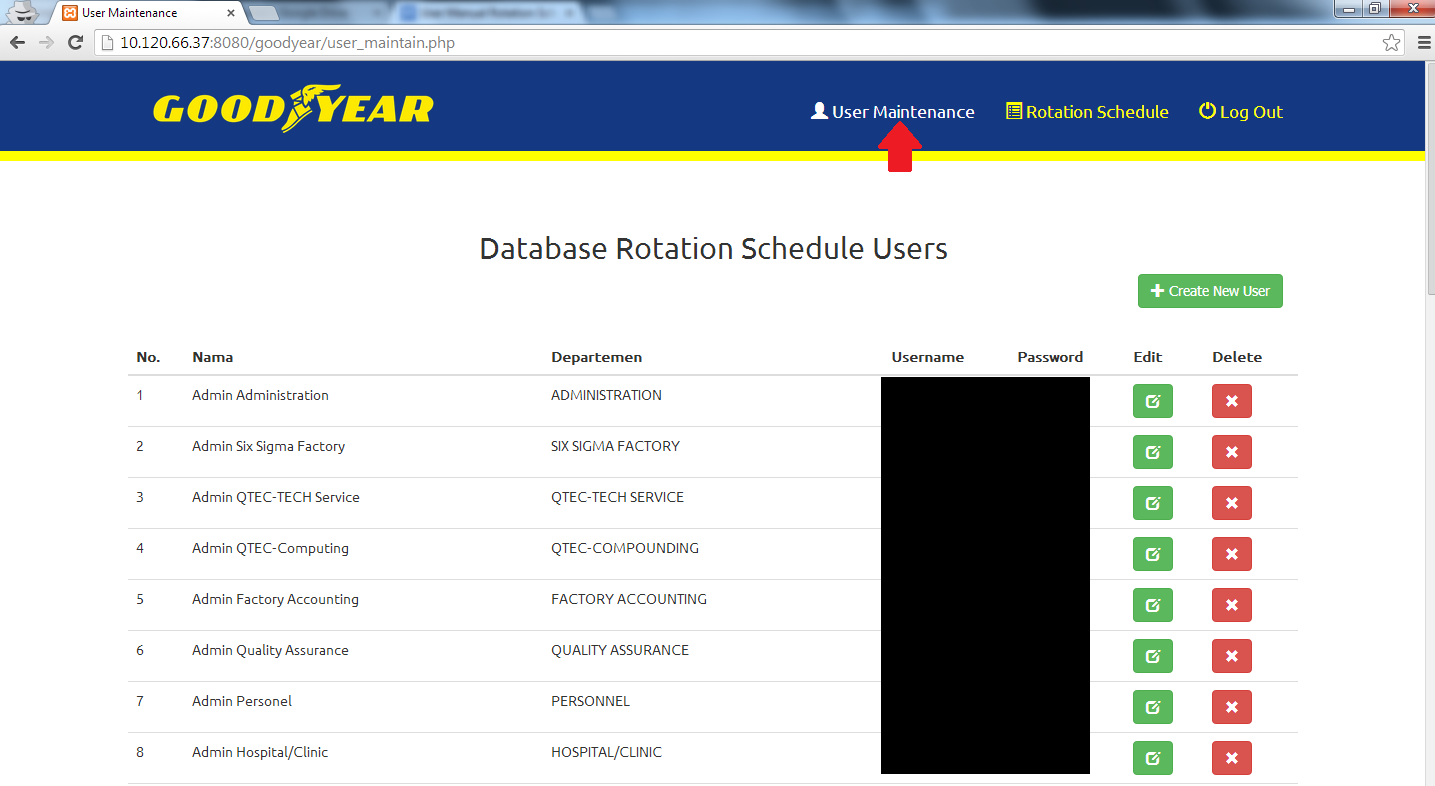
* 1. Halaman *Login*

Halaman *Login* diimplementasikan untuk masuk dalam Sistem Rotation Schedule dengan menggunakan *input username* dan *password*. Tampilan Halaman *Login* dapat dilihat pada Gambar 14.



* 1. Halaman *User Maintenance*

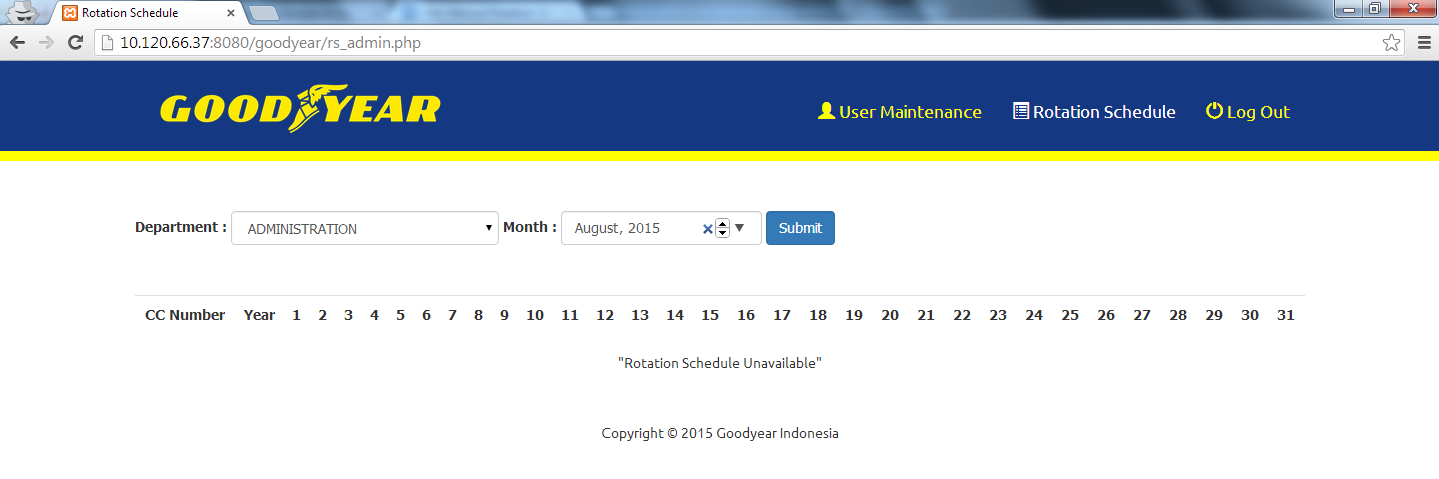
Halaman *User Maintenance* diimplementasikan untuk dapat melakukan akses terhadap data *user*. Data *user* terdiri dari *nama*, *departemen*, *username*, dan *password*. Dalam halaman ini *admin* dapat menambah *user* baru, mengubah data *user* dan menghapus data *user*. Tampilan Halaman *User Maintenance* dapat dilihat pada Gambar 15.



Gambar 15 Halaman *User Maintenance*

* 1. Halaman *Rotation Schedule*

Halaman *Rotation Schedule* diimplementasikan untuk dapat mengunduh jadwal rotasi tiap departemen. Dalam halaman ini *admin* tidak dapat mengubah data jadwal rotasi. Tampilan Halaman *Rotation Schedule* dapat dilihat pada Gambar 16.



Gambar 16 Halaman *Rotation Schedule*

### 4 *Deployment delivery and feedback*

Pada tahap ini, dilakukan pengujian sistem dengan *stakeholder*. Hasil pengujian halaman *Rotation Schedule* yang diuji oleh *user* dapat dilihat pada Tabel 2. Sementara itu, hasil pengujian halaman *Rotation Schedule* yang diuji oleh *admin* dapat dilihat pada Tabel 3.

Tabel 2 Hasil pengujian halaman *user* *Rotation Schedule*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Skenario pengujian | *Test case* | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Mengisi *username* yang tidak terdaftar, lalu mengeklik tombol ‘Login’ | Username: test  Password: goodyear | Sistem akan menampilkan pesan “Username doesn’t exist” | Sesuai harapan | *Valid* |
| 2 | Mengisi *username* yang telah terdaftar namun *password* salah, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: notvalid | Sistem akan menampilkan pesan “Wrong Password” | Sesuai harapan | *Valid* |
| 3 | Mengisi *username* dan *password* yang telah terdaftar, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: goodyear | Sistem akan menampilkan halaman utama yaitu halaman *Rotation Schedule* | Sesuai harapan | *Valid* |
| 4 | Memilih bulan yang diinginkan pada halaman *Rotation Schedule*, lalu mengeklik ‘Submit’ |  | Sistem akan menampilkan tabel *Rotation Schedule* pada bulan yang telah dipilih, tetapi jika pada bulan tersebut belum terdapat *Rotation Schedule* maka system akan menampilkan tabel kosong dengan tombol ‘Upload File’ | Sesuai harapan | *Valid* |
| 5 | Mengeklik tombol ‘Upload File’ |  | Sistem akan menampilkan *pop up* untuk mengunggah dokumen | Sesuai harapan | *Valid* |
| 6 | Mengeklik ‘Submit’ pada *pop up* ‘Upload File’ tanpa memilih dokumen yang akan diunggah |  | Sistem akan menampilkan pesan “No *file* selected” | Sesuai harapan | *Valid* |
| 7 | Mengeklik ‘Choose File’ pada *pop up* ‘Upload File’ dan memilih dokumen yang bukan bertipe .csv, lalu mengeklik ‘Submit’ |  | Sistem akan menampilkan pesan “*File* type must .csv” | Sesuai harapan | *Valid* |
| 8 | Mengeklik ‘Choose File’ pada *pop up* ‘Upload File’ dan memilih dokumen bertipe .csv tetapi jumlah karyawan pada dokumen yang akan di unggah tidak sesuai dengan jumlah karyawan yang berada dalam *database*, lalu mengeklik ‘Submit’ |  | Sistem akan menampilkan pesan “Employee not complete” | Sesuai harapan | *Valid* |
| 9 | Mengeklik ‘Choose File’ pada *pop up* ‘Upload File’ dan memilih dokumen bertipe .csv dengan jumlah karyawan sesuai dengan jumlah karyawan yang berada dalam *database* tetapi jumlah kolom hari pada bulan dalam dokumen yang akan diunggah tidak sesuai dengan jumlah hari pada bulan yang dipilih, lalu mengeklik ‘Submit’ |  | Sistem akan menampilkan pesan “Coloumn doesn’t match” dan “Make sure total day match” | Sesuai harapan | *Valid* |
| 10 | Mengeklik ‘Choose File’ pada *pop up* ‘Upload File’ dan memilih dokumen bertipe .csv dengan jumlah karyawan sesuai dengan jumlah karyawan yang berada dalam *database* dan jumlah kolom hari sesuai dengan dengan jumlah hari pada bulan yang dipilih, lalu mengeklik ‘Submit’ |  | Sistem akan menampilkan pesan “Uploaded successfully” dan menampilkan tabel *Rotation Schedule* yang telah diunggah | Sesuai harapan | *Valid* |
| 11 | Mengeklik tombol ‘Download’ pada halaman *Rotation Schedule* |  | Sistem akan memberikan tabel *Rotation Schedule* kedalam komputer *user* dengan format csv | Sesuai harapan | *Valid* |
| 12 | Mengedit nilai pada tabel *Rotation Schedule* langsung dalam sistem |  | Sistem akan langusng menampilkan tabel yang telah diubah | Sesuai harapan | *Valid* |
| 13 | Mengeklik tombol ‘Delete’ pada halaman *Rotation Schedule* |  | Sistem akan menghapus tabel *Rotation Schedule* dalam *database* dan menampilkan halaman *Rotation Schedule* dengan tabel kosong | Sesuai harapan | *Valid* |
| 14 | Memilih halaman ‘*Template*’ pada *menu bar* |  | Sistem akan menampilkan halaman *Template* yang berisi format tabel *Rotation Schedule* | Sesuai harapan | *Valid* |
| 15 | Mengeklik tombol ‘Download’ pada halaman *Template* |  | Sistem akan memberikan tabel *template* *Rotation Schedule* ke dalam komputer *user* dengan format csv | Sesuai harapan | *Valid* |
| 16 | Memilih halaman ‘Example’ pada *menu bar* |  | Sistem akan menampilkan halaman *Example* yang berisi tabel pola *Rotation Schedule* | Sesuai harapan | *Valid* |
| 17 | Mengeklik tombol ‘Download’ pada halaman *Example* |  | Sistem akan memberikan tabel contoh *Rotation Schedule* ke dalam komputer *user* dengan format csv | Sesuai harapan | *Valid* |

Tabel 3 Hasil pengujian halaman *admin* *Rotation Schedule*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Skenario pengujian | *Test case* | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Mengisi *username* yang tidak terdaftar, lalu mengeklik tombol ‘Login’ | Username: test  Password: goodyear | Sistem akan menampilkan pesan “Username doesn’t exist” | Sesuai harapan | *Valid* |
| 2 | Mengisi *username* yang telah terdaftar namun *password* salah, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: notvalid | Sistem akan menampilkan pesan “Wrong Password” | Sesuai harapan | *Valid* |
| 3 | Mengisi *username* dan *password* yang telah terdaftar, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: goodyear | Sistem akan menampilkan halaman utama yaitu halaman *User Maintenance* yang berisi tabel *Database* *Rotation Schedule* pengguna | Sesuai harapan | *Valid* |
| 4 | Mengeklik tombol ‘Edit’ pada tabel *user* dalam halaman *User Maintenance* |  | Sistem akan menampilkan *pop up* ‘Edit User’ | Sesuai harapan | *Valid* |
| 5 | Mengubah data *user* pada *pop up* ‘Edit’, lalu mengeklik tombol ‘Update’ |  | Sistem akan menampilkan halaman *User Maintenance* dengan data *user* yang baru dalam tabel *Database* *Rotation Schedule* pengguna dan menampilkan pesan “Update Successfully” | Sesuai harapan | *Valid* |
| 6 | Mengeklik tombol ‘Delete’ pada tabel *user* dalam halaman *User Maintenance* |  | Sistem akan menampilkan *pop up* ‘Delete User’ | Sesuai harapan | *Valid* |
| 7 | Mengeklik tombol ‘Delete’ pada *pop up* ‘Delete User’ |  | Sistem akan menampilkan halaman *User Maintenance* dengan menghilangkan data *user* yang telah dihapus dalam tabel *Database* *Rotation Schedule* pengguna dan menampilkan pesan “User has been deleted” | Sesuai harapan | *Valid* |
| 8 | Mengeklik tombol ‘Create New User’ pada halaman *User Maintenance* |  | Sistem akan menampilkan *pop up* ‘Create New User’ | Sesuai harapan | *Valid* |
| 9 | Mengisi data *user* baru pada *pop up* ‘Create New User’, lalu mengeklik tombol ‘Add’ |  | Sistem akan menampilkan halaman *User Maintenance* dengan menambahkan data *user* baru dalam tabel *Database* *Rotation Schedule* pengguna dan menampilkan pesan “New User Added Successfully” | Sesuai harapan | *Valid* |
| 10 | Memilih halaman *Rotation Schedule* pada *menu bar* |  | Sistem akan menampilkan halaman *Rotation Schedule* | Sesuai harapan | *Valid* |
| 11 | Memilih departemen dan bulan yang diinginkan, lalu mengeklik tombol ‘Submit’ |  | Sistem akan menampilkan halaman *Rotation Schedule* dengan tabel *Rotation Schedule* sesuai departemen dan bulan yang telah dipilih | Sesuai harapan | *Valid* |
| 12 | Mengeklik tombol ‘Download’ pada halaman *Rotation Schedule* |  | Sistem akan memberikan tabel *Rotation Schedule* kedalam komputer *admin* dengan format csv | Sesuai harapan | *Valid* |

Berdasarkan hasil *testing* dengan pengguna, secara fungsional Sistem Rotation Schedule sudah bisa digunakan oleh *admin* dan *user* (perwakilan setiap departemen). Sistem Rotation Schedule baru diterapkan di departemen HRD dan IT. Untuk departemen yang belum menerapkan sistem ini akan diberikan pengarahan sebelumnya oleh departemen HRD sebagai penanggung jawab sistem.

# SISTEM LAYER AUDIT DATABASE

Sistem Layer Audit Database merupakan sistem yang dikembangkan berbasis web dengan *database* terpusat. Sistem ini dikembangkan untuk dapat melakukan proses *auditing* di departemen *Manufacturing* khususnya divisi *Quality and Technology* di PT Goodyear Indonesia pada waktu yang berkala. Dalam sistem ini, setiap *auditor* ditugaskan untuk melakukan audit pada jadwal yang sudah ditentukan. Hasil audit akan diteruskan kepada *owner* dari area yang dilakukan proses audit tersebut untuk ditindaklanjuti dengan melakukan *corrective action*. *Corrective action* merupakan solusi yang diberikan *owner* atas permasalahan di area terkait. Dengan adanya proses audit dalam sistem ini, sistem pelaporan atas permasalahan yang terjadi pada suatu area akan lebih efektif dan efisien.

## Latar Belakang

Menurut Mulyadi (2002), *auditing* merupakan suatu proses sistematik untuk memperoleh dan mengevaluasi bukti secara objektif mengenai pernyataan-pernyataan tentang kegiatan dan kejadian ekonomi dengan tujuan untuk menetapkan tingkat kesesuaian antara pernyataan-pernyataan tersebut dengan kriteria yang telah ditetapkan, serta penyampaian hasil-hasilnya kepada pemakai yang berkepentingan.

Selama ini prosesaudit yang dilakukan oleh divisi *Quality and Technology* masih bersifat manual. Pertanyaan-pertanyaan audit yang diberikan kepada *auditor* dicetak dan kemudian hasil evaluasi dari audit tersebut ditulis dengan tangan. Data-data tersebut tidak masuk ke dalam *database* yang terpusat dan juga rentan hilang. Dengan adanya Sistem Layer Audit Database, proses audit dikerjakan di dalam satu kesatuan sistem dan hasilnya tersimpan di dalam *database* yang terpusat. Dalam sistem ini juga terdapat fungsi notifikasi yang akan masuk ke *email* masing-masing *auditor*. Notifikasi ini berfungsi mengingatkan *auditor* untuk melakukan audit pada bulan itu. Bagan alir *layer audit database* sebelum menggunakan sistem dapat dilihat pada Gambar 17 dan bagan alir *layer audit database* setelah menggunakan sistem dapat dilihat pada Gambar 18.

## Tujuan

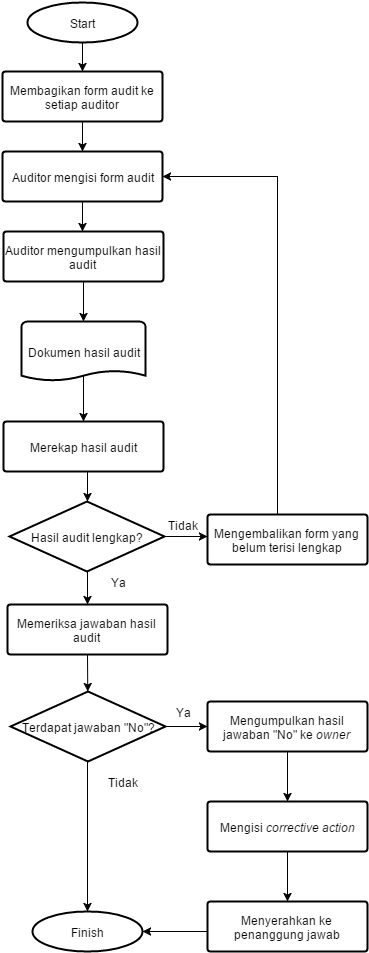
Tujuan dibuatnya Sistem Layer Audit Database adalah:

1. Membuat proses audit yang semula manual menjadi lebih otomatis.
2. Membuat notifikasi *email* berupa *reminder* bagi *auditor* untuk melakukan audit.

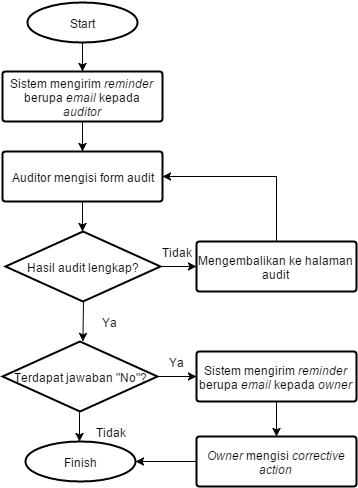
## Ruang Lingkup

Ruang lingkup dari pembuatan Sistem Rotation Schedule yaitu:

* + - 1. Sistem dikembangkan pada sistem operasi Windows 7 dengan perangkat lunak Adobe Dreamweaver, Notepad++. *Database* yang digunakan adalah MySQL. Sementara itu, bahasa pemrograman yang diimplementasikan pada sistem ini yaitu *PHP Hypertext Preprocessor*.
      2. Sistem hanya diakses oleh divisi *Quality and Technology* PT Goodyear Indonesia.





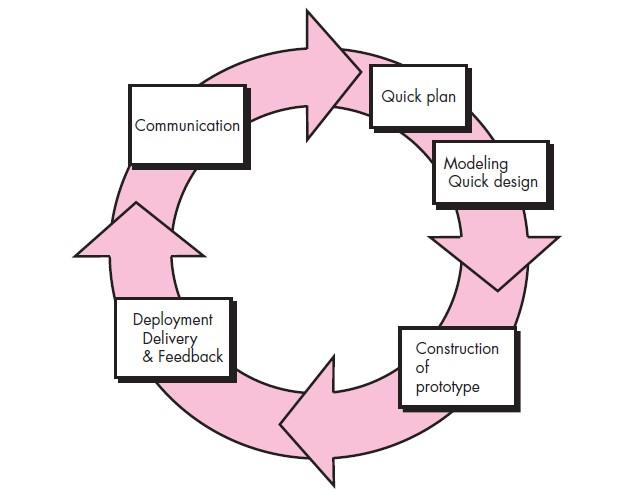




## Metode Pengembangan

Teknik dan metode yang digunakan untuk mengembangkan Sistem Rotation Schedule yaitu metode *prototyping*. *Prototyping* adalah metode pengembangan yang cepat dan pengujian terhadap model kerja dari aplikasi baru melalui proses interaksi berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis (O’Brien 2005).

Menurut Pressman (2010), seringkali pelanggan mendefinisikan satu set tujuan umum untuk perangkat lunak, tetapi tidak mengidentifikasi persyaratan rinci untuk fungsi dan fitur. Dalam lain kasus, pengembang mungkin tidak yakin efisiensi algoritma, adaptasi dari sistem operasi, atau bentuk interaksi manusia-mesin yang harus diambil. Dalam hal ini, paradigma *prototyping* dapat menjadi pendekatan yang terbaik. Proses-proses model pengembangan perangkat lunak *prototyping* dapat dilihat pada Gambar 19.



Gambar 19 Paradigma *prototyping* (Pressman 2010)

## Pengembangan Sistem Layer Audit Database

Terdapat empat tahapan dalam proses *prototyping* yang digunakan dalam pengembangan Sistem Layer Audit Database.

### *Communication*

Dalam tahapan ini, kami melakukan pendefinisian masalah yang dihadapi oleh pengguna terkait sistem yang akan dirancang. Berikut merupakan identifikasi format Sistem Layer Audit Database.

a Analisis Pengguna Sistem

Dalam sistem ini penguna dibagi menjadi tiga, yaitu:

1 *Admin*, yaitu perwakilan dari divisi *Quality and Technology* memiliki hak akses untuk mengatur jadwal audit, melakukan *maintenance* terhadap data *auditor* dan *owner* serta mengunduh jawaban hasil audit.

2 *Auditor*, yaitu karyawan dari departemen *Manufacturing* yang melakukan audit dengan menjawab pertanyaan-pertanyaan audit yang disediakan pada setiap area.

3 *Owner*, yaitu penanggung jawab dari setiap area yang memiliki akses untuk memberikan *corrective* *action* yang merupakan solusi dari permasalahan yang didapatkan dari hasil audit.

b Analisis Kebutuhan Fungsional

Berdasarkan hasil diskusi dengan *stakeholder*, kebutuhan fungsional Sistem Layer Audit Database dapat dilihat pada Tabel 4.

Tabel 4 Kebutuhan fungsional Sistem Layer Audit Database

|  |  |  |
| --- | --- | --- |
| Kategori Pengguna | Kebutuhan Fungsional | Hak Akses Terhadap Aplikasi |
| *Admin* | Mengatur jadwal audit | SLADB-0001 |
| Mengubah pertanyaan audit | SLADB-0002 |
| Menghapus pertanyaan audit | SLADB-0003 |
| Mengunduh jawaban hasil audit | SLADB-0004 |
| Menambah *auditor* | SLADB-0005 |
| Mengubah data *auditor* | SLADB-0006 |
| Menghapus data *auditor* | SLADB-0007 |
| Menambah *owner* | SLADB-0008 |
| Mengubah data *owner* | SLADB-0009 |
| Menghapus data *owner* | SLADB-0010 |
| *Auditor* | Menjawab pertanyaan audit | SLADB-0011 |
| *Owner* | Menjawab *corrective action* | SLADB-0012 |

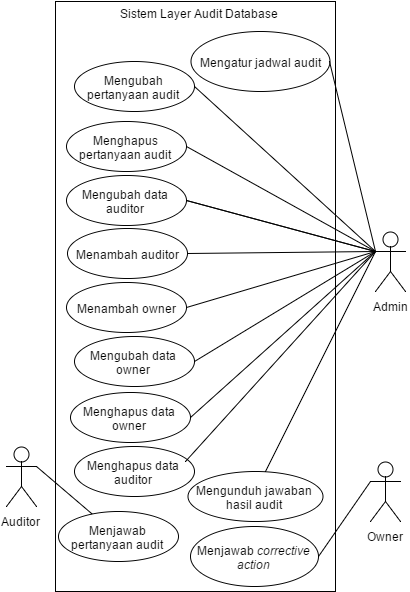
SLADB-xxxx adalah kode yang digunakan untuk merepresentasikan kebutuhan (*requirement*) pada Sistem Layer Audit dengan SLADB merupakan kode sistem dan xxxx adalah digit/nomor kebutuhan (*requirement*).

### *Quick plan and modeling quick design*

Pada tahap ini dibangun perancangan sementara yang merepresentasikan seluruh proses yang terjadi di dalam sistem, *input*, dan *output.* Representasi proses untuk Sistem Layer Audit Database dibangun dalam bentuk *use case diagram, context diagram,* dan *data flow diagram* level1*.*

2.1 *Use Case Diagram*

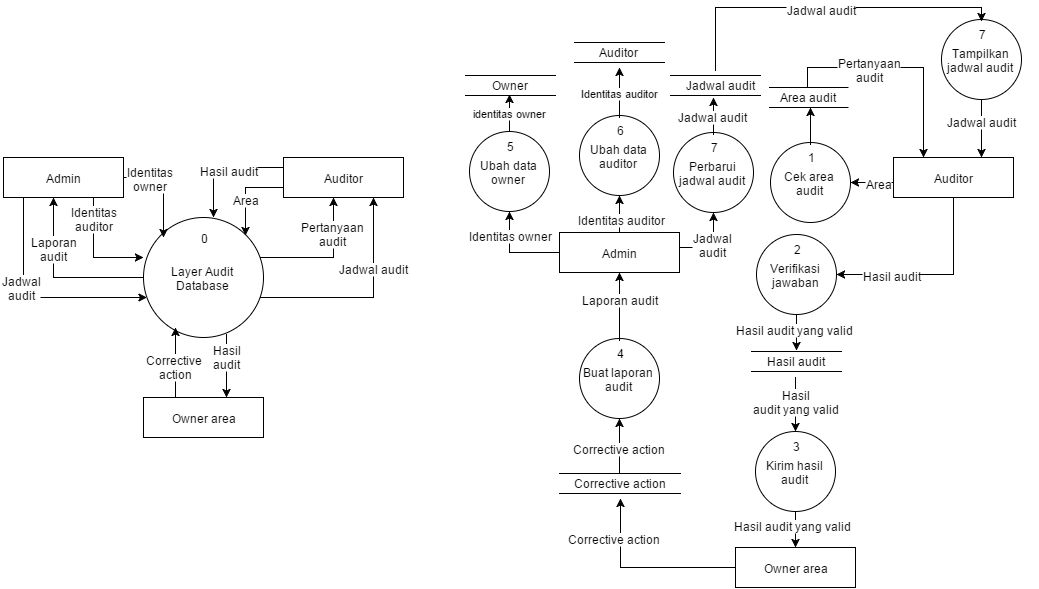
*Use case diagram* Sistem Layer Audit Database setelah melalui tahap evaluasi *prototype* dapat dilihat pada Gambar 20. Dalam Sistem Layer Audit Database terdapat tiga aktor yaitu *admin*, *auditor*,dan *owner*. *Admin* yang diwakili oleh departemen *Quality and Technology* dapat melakukan sembilan fungsi utama yaitu mengatur jadwal audit, mengubah pertanyaan audit, mengubah data *auditor*, menghapus data *auditor*, menambah *auditor*, menambah *owner*, mengubah data *owner*, menghapus data *owner*, dan mengunduh jawaban hasil audit. Sementara itu, *auditor* yang merupakan karyawan dari departemen *Manufacturing* dapat melakukan satu fungsi utama yaitu menjawab pertanyaan audit dari masing-masing area dan *owner* yang merupakan penanggung jawab dari setiap area dapat melakukan satu fungsi utama yaitu menjawab menjawab *corrective action*. Deskripsi masing-masing *use case* Sistem Layer Audit Database dapat dilihat pada Lampiran 2.



Gambar 20 *Use case diagram* Sistem Layer Audit Database

2.2 *Context Diagram*

*Context diagram* menjelaskan aliran data yang dimiliki sistem antara *auditor*, *owner,* *admin,* dan Sistem Layer Audit Database. Gambar 21 menjelaskan aliran data yang direpresentasikan melalui *context diagram*.



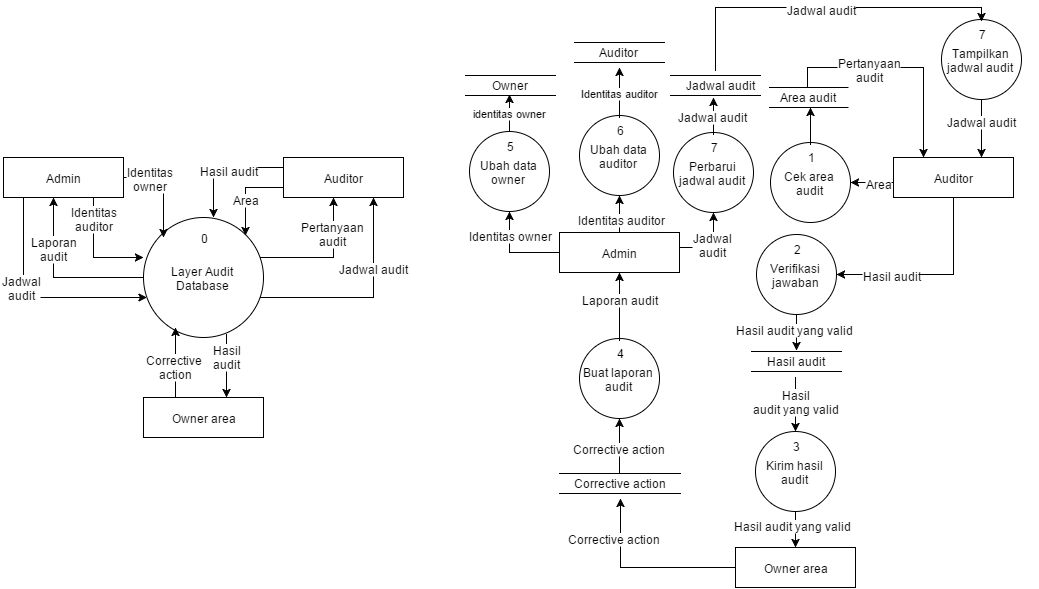
Gambar 21 *Context* *diagram*

*Context diagram* menggambarkan aliran data pada sistem dengan pengguna. *Admin* dapat memberikan *input* berupa identitas *auditor* dan jadwal audit. Kemudian sistem akan memberikan *output* kepada *admin* berupa laporan audit. *Owner area* diberikan *output* berupa hasil audit dan dapat melakukan *corrective action* dari hasil audit. Sementara *auditor* dapat menjawab pertanyaan audit sesuai jadwal yang ditentukan dan hasil audit berdasarkan area tersebut akan masuk ke dalam sistem.

2.3 *Data Flow Diagram* (DFD)

DFD dirancang untuk menyusun tahapan yang harus dilakukan oleh *User* dan *Admin* dalam mencapai fungsi utama dari Sistem Rotation Schedule. Gambar 22 menjelaskan tentang DFD level1yaitu perluasan aliran data yang dimiliki Sistem Rotation Schedule.

Perluasan aliran data yang digambarkan pada DFD level 1 mencakup *input* dan *output* antara sistem dengan pengguna. Sistem akan terlebih dahulu mengecek area yang terjadwal untuk diaudit. Kemudian *auditor,* yang bertugas untuk mengaudit area tersebut, dapat memberi *input* dengan menjawab pertanyaan audit. Sistem akan melakukan verifikasi terhadap hasil audit yang apabila *valid*, hasil audit tersebut akan dikirimkan kepada *owner area*. *Owner area* memberikan *corrective action* terhadap hasil audit. Laporan hasil audit tersebut akan diberikan kepada *admin*. Dalam Sistem Layer Audit Database, *admin* dapat mengubah data *auditor* dan *owner*.

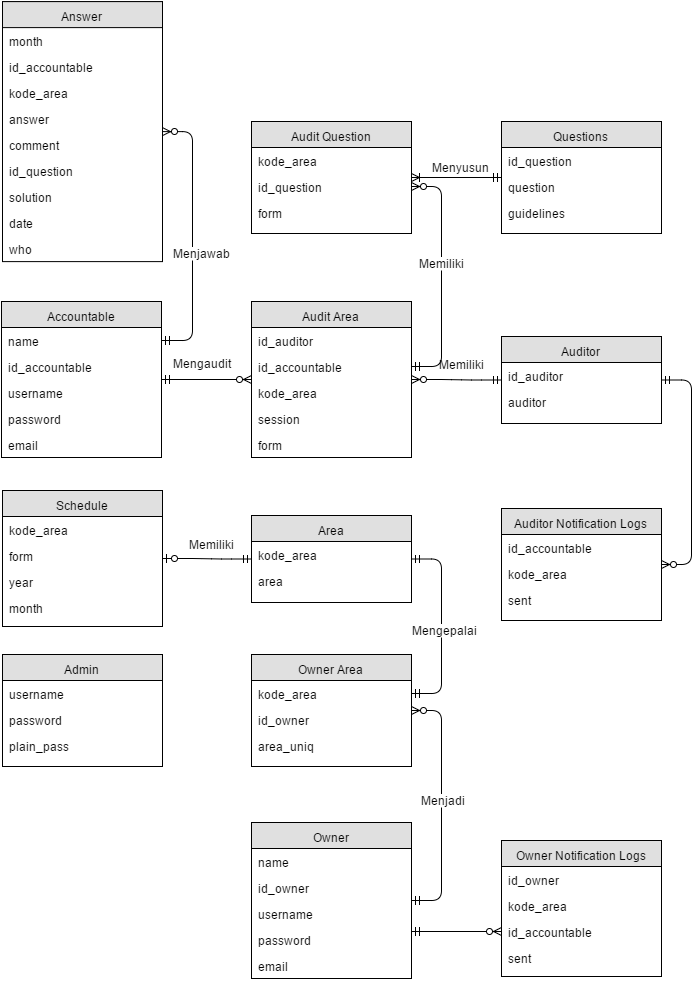


Gambar 22 DFD Level 1

2.4 *Entity Relationship Diagram* (ERD)

ERD menggambarkan relasi antar objek yang digunakan pada Sistem Layer Audit Database. Sistem ini menggunakan 13 entitas. Setiap entitas tersebut memiliki hubungan tersendiri dengan entitas lainnya. Hubungan tersebut dapat dilihat pada Gambar 23.

Entitas *accountable* memiliki hubungan *one-to-many* dengan entitas *answer*. Satu *accountable* dapat menjawab banyak *answer*. Satu *accountable* juga dapat mengaudit banyak area. Area yang diaudit dimiliki satu *auditor*. Sementara itu, *auditor* dapat memiliki banyak area. Setiap area yang diaudit sesuai *schedule* memiliki banyak pertanyaan audit yang disusun oleh entitas *questions*. Satu *owner* dapat menjadi banyak *owner area* yang setiap *owner area* tersebut dikepalai oleh satu area.



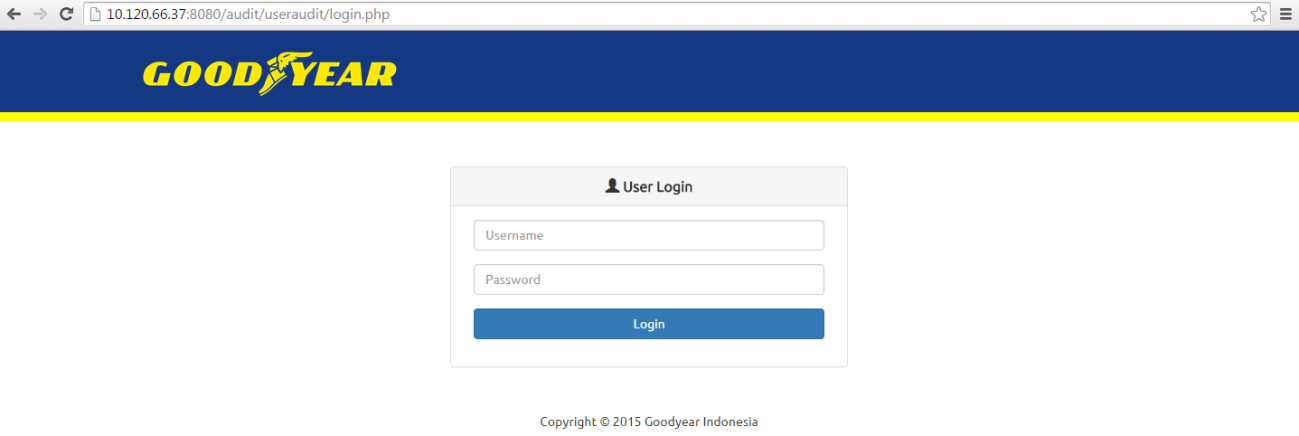
Gambar 23 ERD Sistem Layer Audit Database

### *Construction of prototype*

Pada tahap ini dilakukan implementasi hasil perancangan pada tahap *design* menjadi baris kode program sesuai bahasa pemrograman yang digunakan. Dalam Sistem Layer Audit Database untuk *auditor* terdapat beberapa halaman yang diimplementasikan, diantaranya sebagai berikut:

* + - * 1. Halaman *Login*

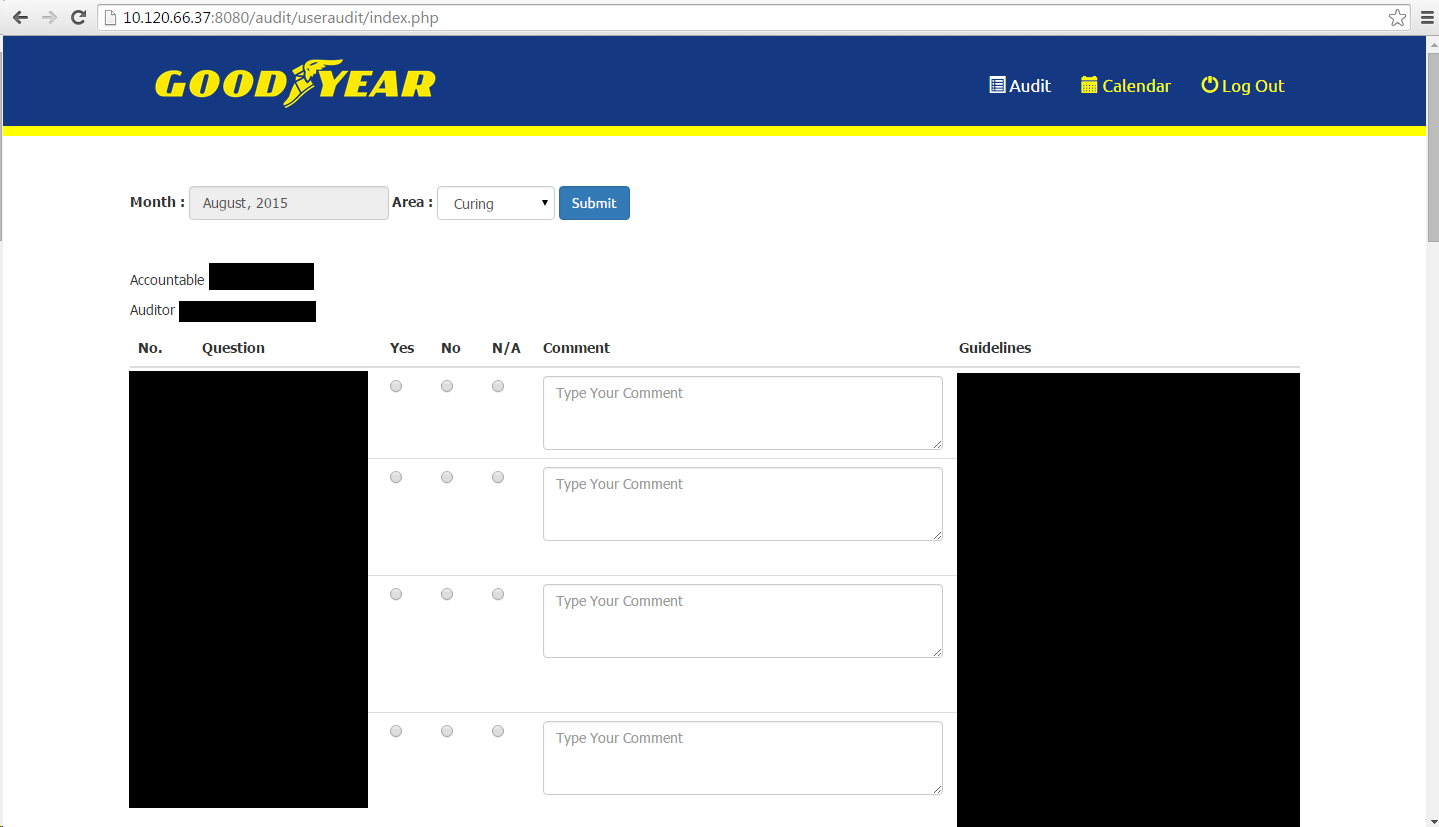
Halaman *Login* diimplementasikan untuk masuk dalam Sistem Layer Audit Database dengan menggunakan *input username* dan *password*. Tampilan Halaman *Login* dapat dilihat pada Gambar 24.



Gambar 24 Halaman *Login* untuk *user*

* 1. Halaman Audit

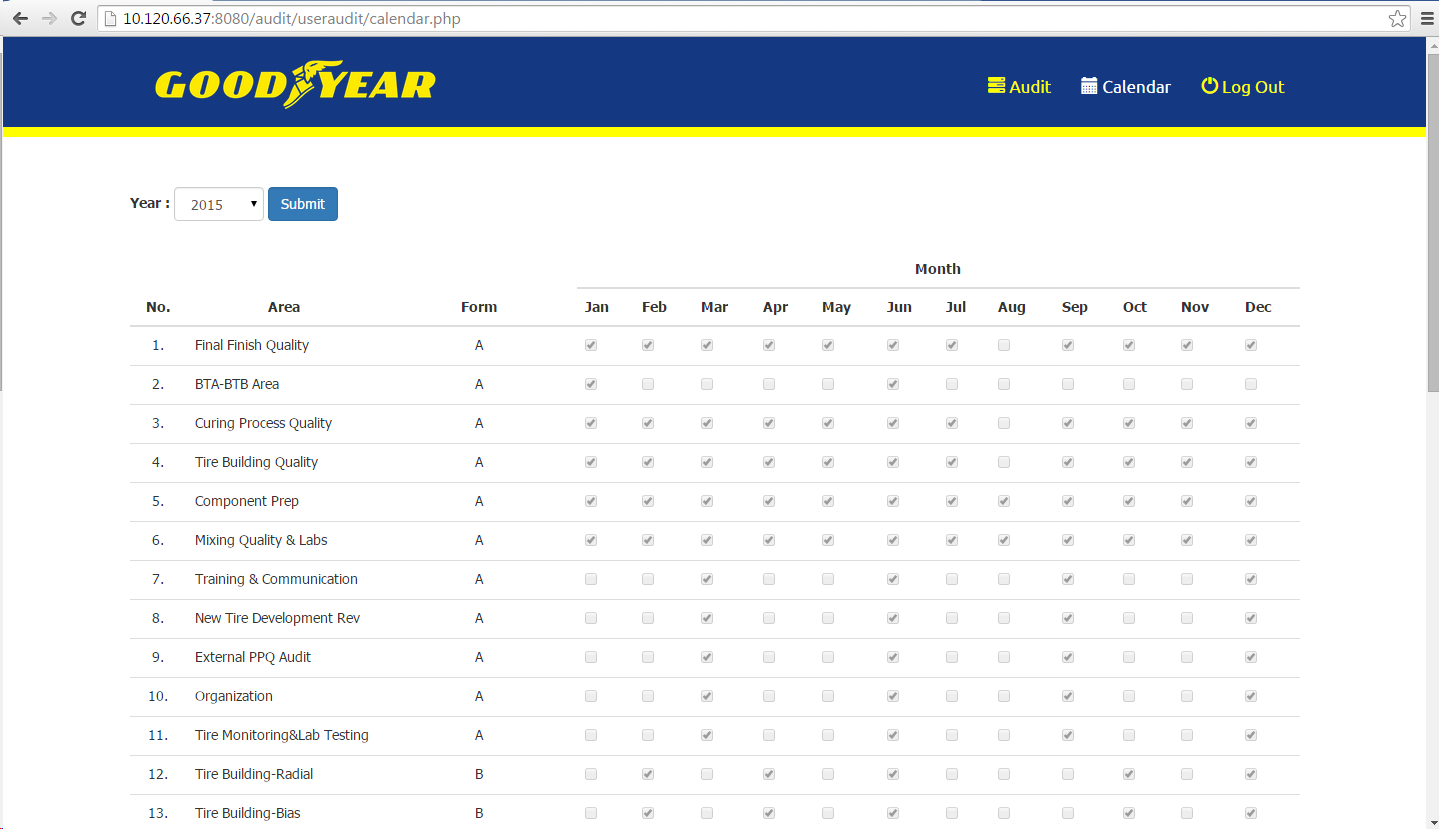
Halaman Audit diimplementasikan untuk *auditor* agar dapat melakukan proses audit dengan menjawab pertanyaan audit yang disediakan. Pada halaman ini terdapat 4 kolom yaitu *question*, jawaban (*yes*, *no*, n/a), *comment*, dan *guideline*. *Question* adalah pertanyaan-pertanyaan audit yang diberikan sesuai area. *User* dapat memberikan jawaban berupa *yes* atau *no* atau n/a, jika *user* menjawab *no* maka *user* harus mengisi permasalahan yang dihadapi pada kolom *comment*. *Guideline* merupakan penjelasan dari pertanyaan audit. Tampilan Halaman Audit dapat dilihat pada Gambar 25.



Gambar 25 Halaman Audit

* 1. Halaman *Calendar*

Halaman *Calendar* diimplementasikan untuk *auditor* agar dapat melihat jadwal audit sesuai area masing-masing. Tampilan Halaman *Calendar* dapat dilihat pada Gambar 26.

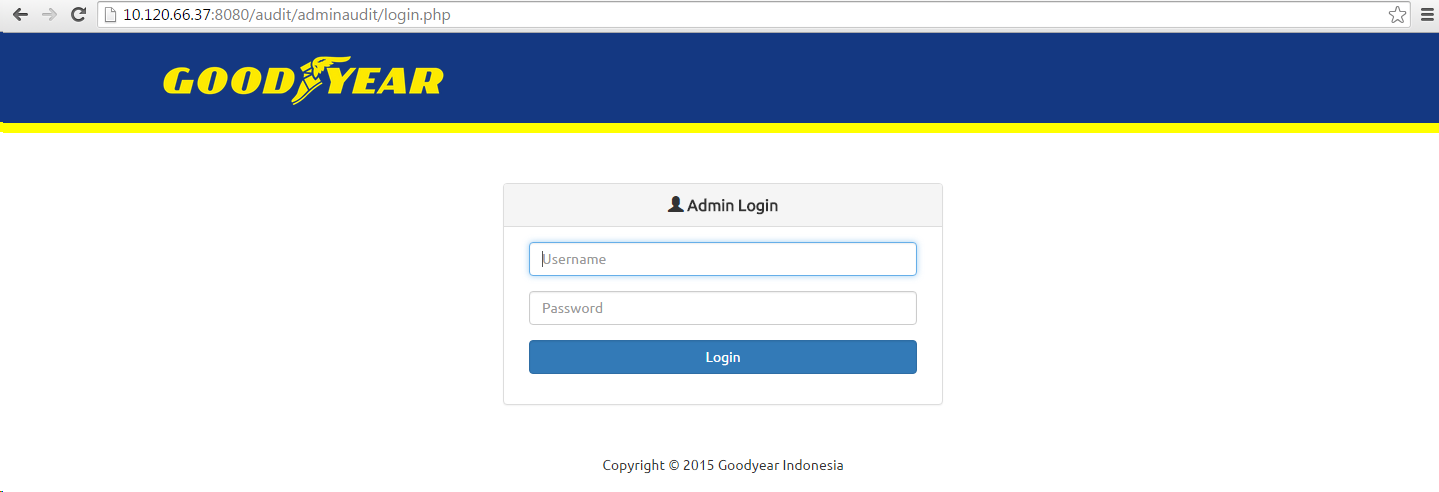


Gambar 26 Halaman *Calendar*

Dalam sistem untuk *admin* terdapat beberapa halaman, diantaranya sebagai berikut:

* + - * 1. Halaman *Login*

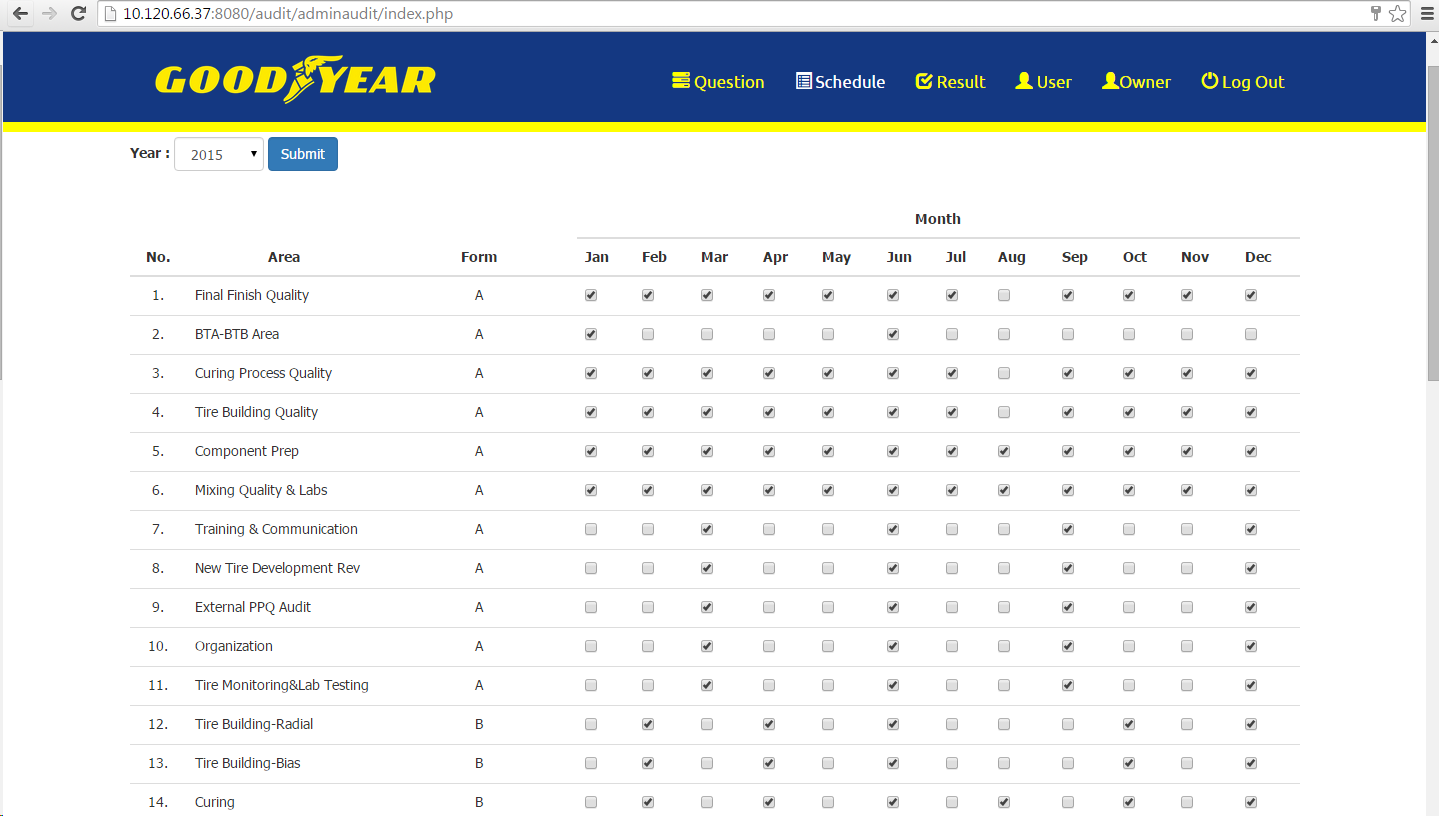
Halaman *Login* diimplementasikan untuk masuk dalam Sistem Layer Audit Database dengan menggunakan *input username* dan *password*. Tampilan Halaman *Login* dapat dilihat pada Gambar 27.



Gambar 27 Halaman *Login* untuk *admin*

* + - * 1. Halaman *Schedule*

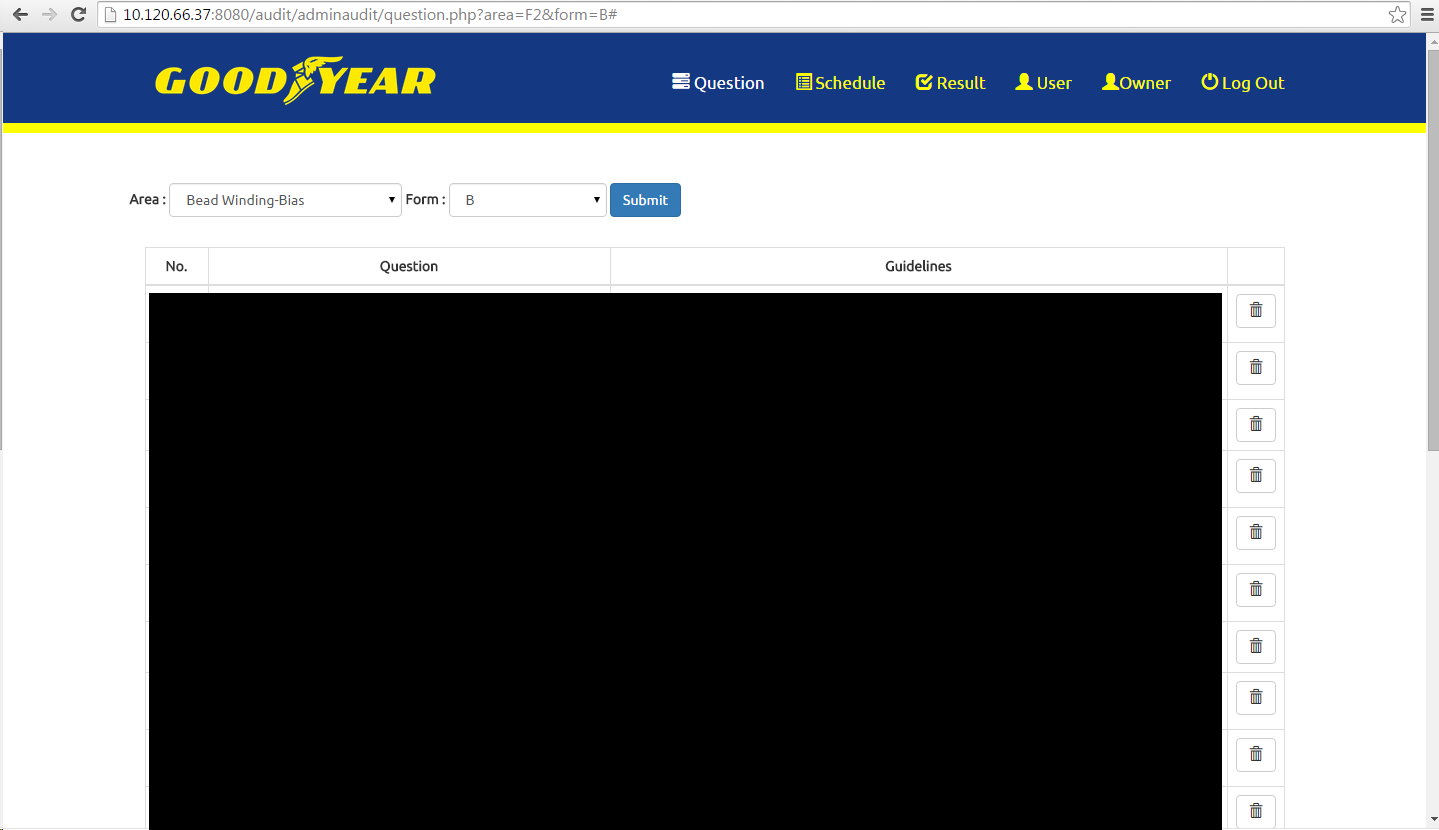
Halaman *Schedule* diimplementasikan untuk *admin* agar dapat mengatur jadwal audit. Setiap bulan, sistem akan mengirim *email* notifikasi kepada *auditor* yang terjadwal untuk melakukan audit. Tampilan Halaman *Schedule* dapat dilihat pada Gambar 28.



Gambar 28 Halaman *Schedule*

* + - * 1. Halaman *Question*

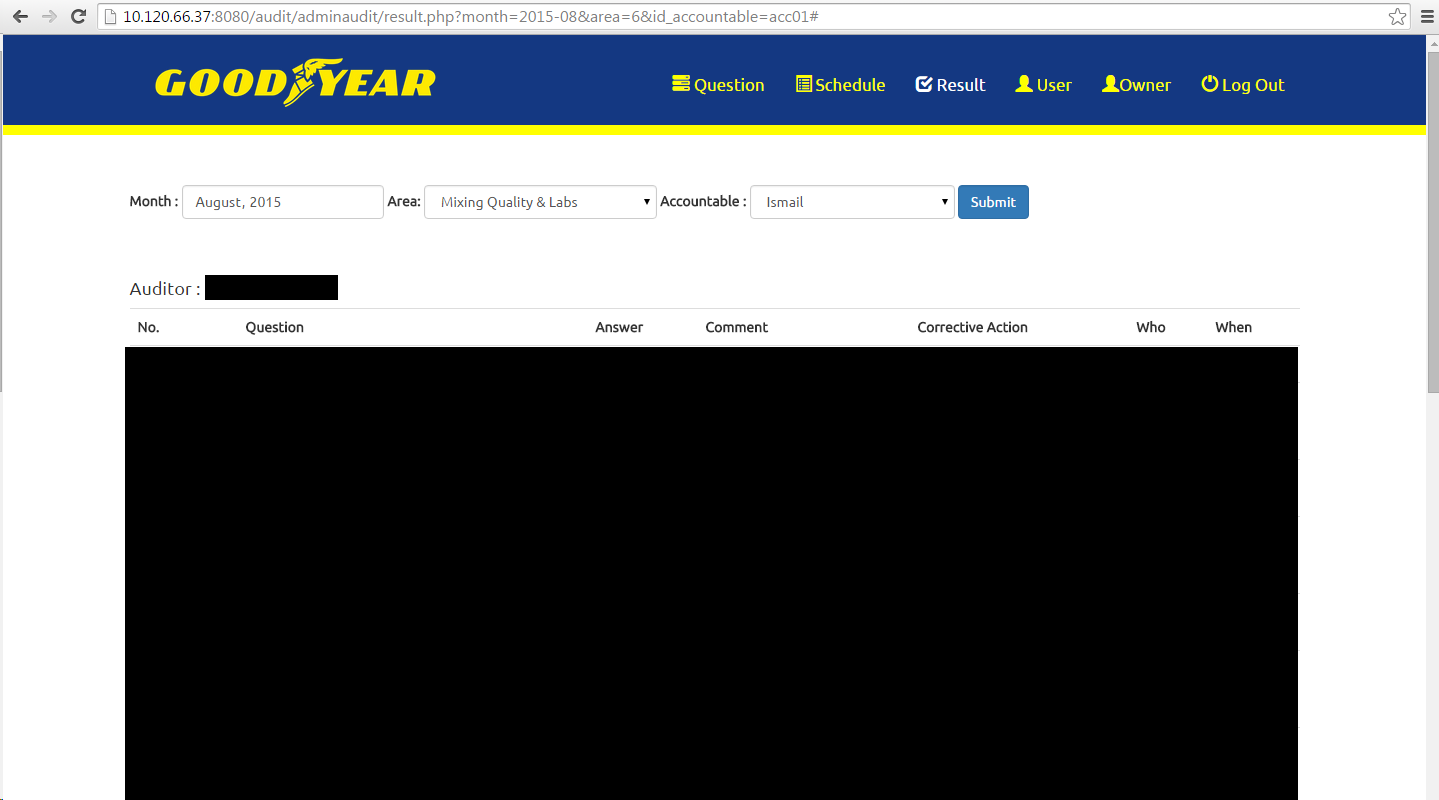
Halaman *Question* diimplementasikan untuk melihat serta melakukan akses terhadap pertanyaan audit. Pada halaman ini terdapat *question* dan *guideline* sesuai dengan area dan *form*. Tampilan Halaman *Question* dapat dilihat pada Gambar 29.



Gambar 29 Halaman *Question*

* 1. Halaman *Result*

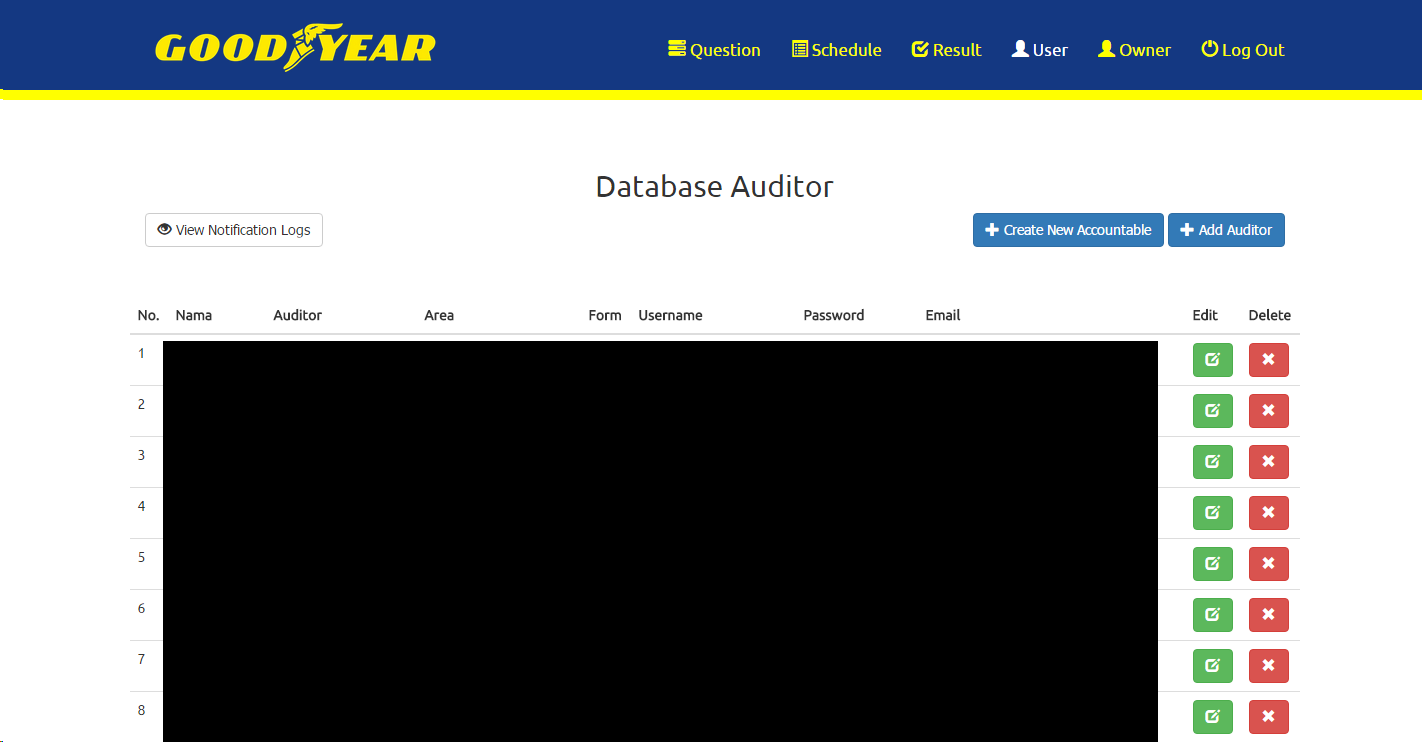
Halaman *Result* diimplementasikan untuk melihat serta mengunduh hasil audit sesuai dengan bulan, area, dan *accountable* yang dipilih. Tampilan Halaman *Result* dapat dilihat pada Gambar 30.



Gambar 30 Halaman *Result*

* 1. Halaman *Database Auditor*

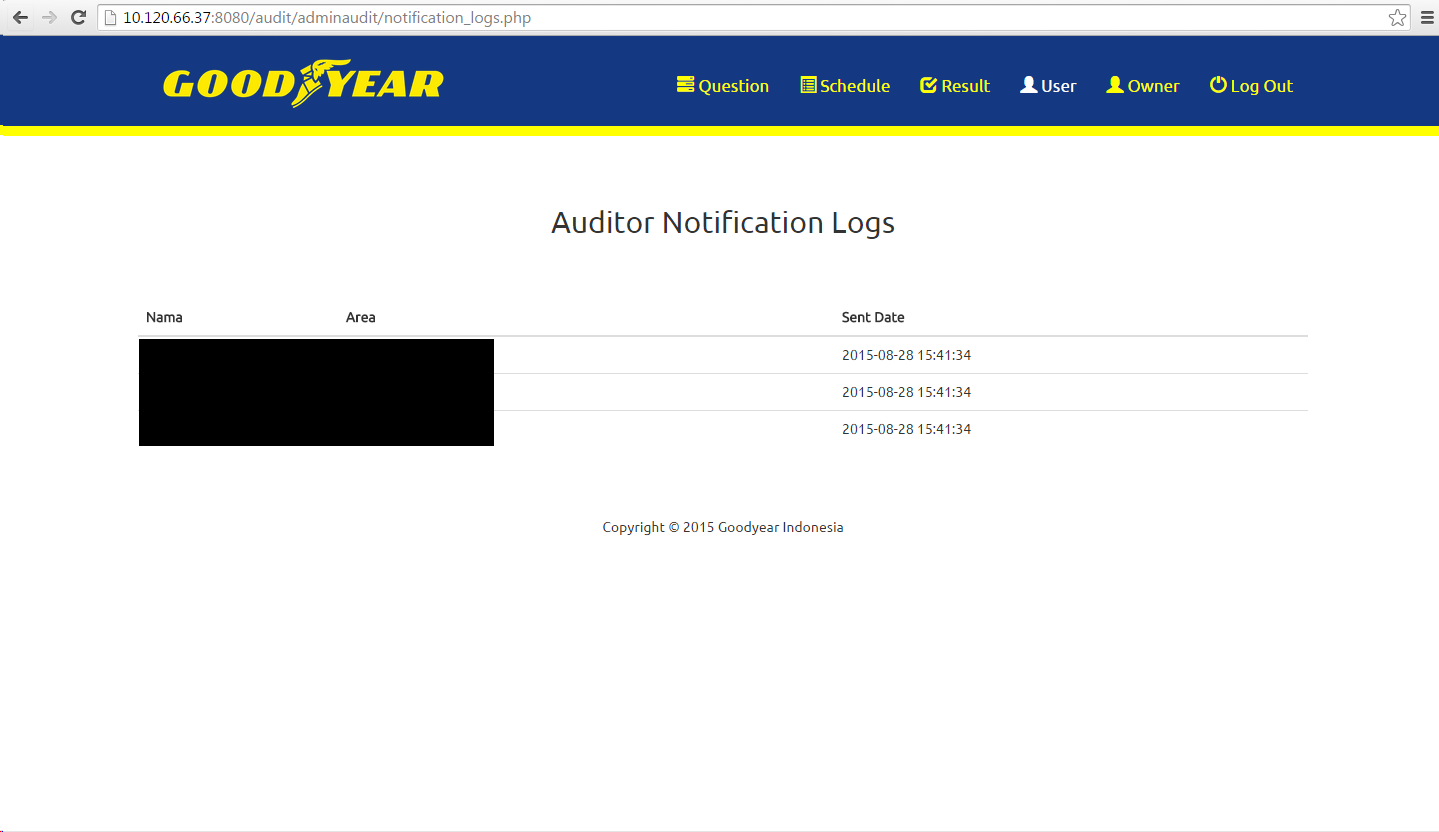
Halaman *Database Auditor* diimplementasikan untuk mengelola *database auditor*. Data *auditor*  terdiri dari nama, *auditor* (jabatan), area, *form*, *username*, *password*, dan *email*. Dalam halaman ini *admin* dapat menambah *auditor* baru, mengubah data *auditor*, dan menghapus data *auditor*. Tampilan Halaman *Database Auditor* dapat dilihat pada Gambar 31.



Gambar 31 Halaman *Database Auditor*

* 1. Halaman *Auditor Notification Logs*

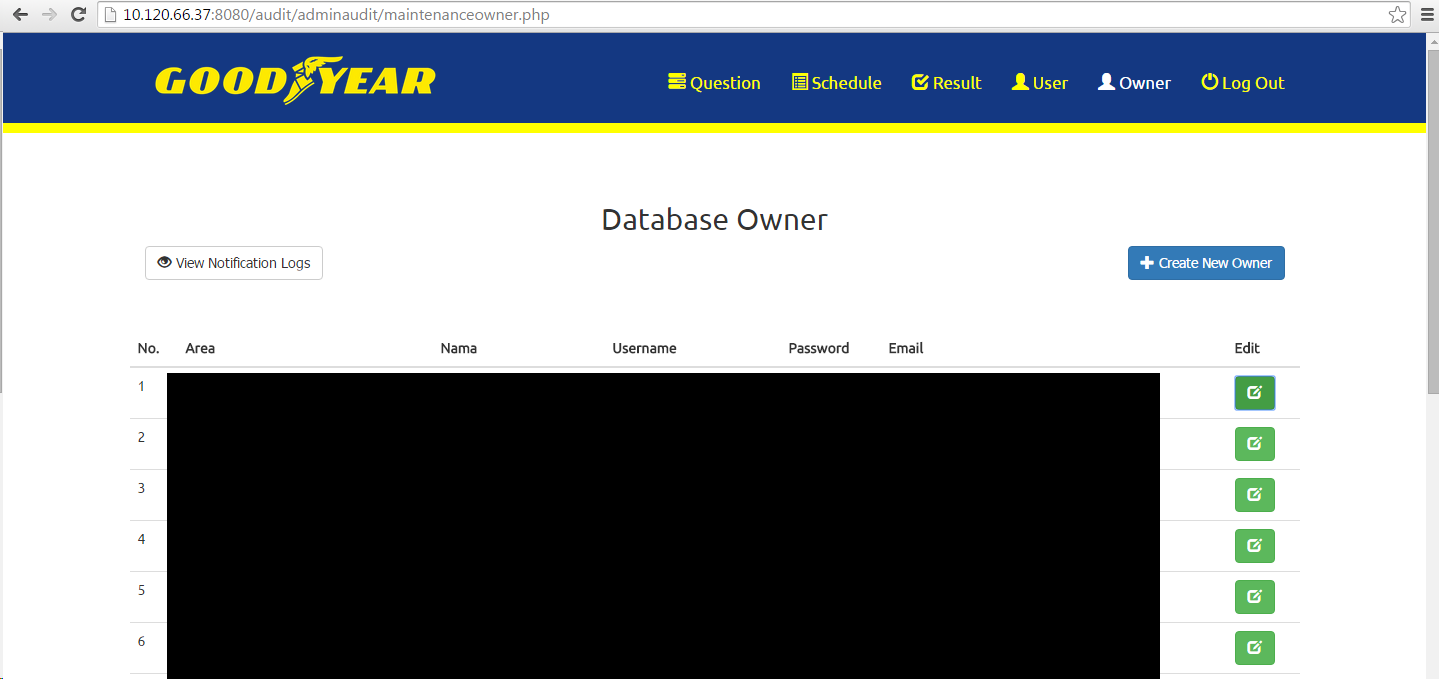
Halaman *Auditor Notification Logs* diimplementasikan untuk melihat *logs* pengiriman *email* untuk *auditor*. Pada halaman ini terdapat 3 kolom yaitu nama *auditor*, area, dan *send* *date* yang merupakan waktu pengiriman *email* untuk *auditor* yang terjadwal untuk melakukan audit. Tampilan Halaman *Auditor Notification Logs* dapat dilihat pada Gambar 32.



Gambar 32 Halaman *Auditor Notification Logs*

* 1. Halaman *Database Owner*

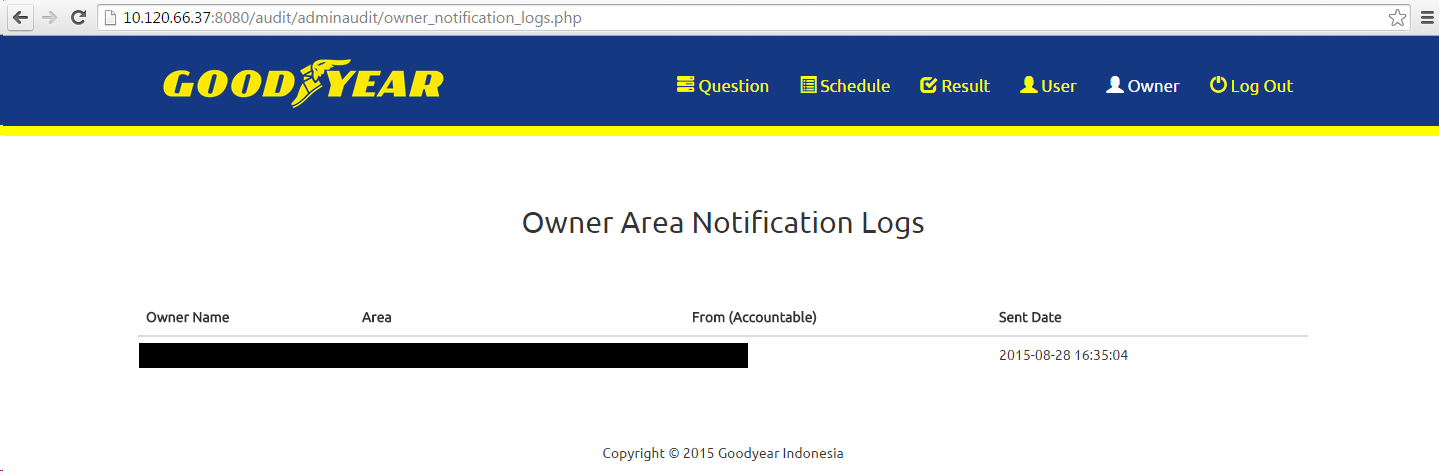
Halaman *Database Owner* diimplementasikan untuk mengelola *database owner*. Data *owner*  terdiri dari area, nama, *username*, *password*, dan *email*. Dalam halaman ini *admin* dapat menambah *owner* baru, mengubah data *owner*, dan menghapus data *owner*. Tampilan Halaman *Database Owner* dapat dilihat pada Gambar 33.



Gambar 33 Halaman *Owner*

* 1. Halaman *Owner Notification Logs*

Halaman *Owner Notification Logs* diimplementasikan untuk melihat *logs* pengiriman *email* untuk *owner*. Pada halaman ini terdapat 4 kolom yaitu nama *owner*, area, *from* (*accountable*), dan *send* *date*. Kolom *from* berisi nama *accountable* yang melakukan audit dan kolom *send* *date* merupakan waktu pengiriman *email* untuk *owner*. *Email* tersebut akan terkirim otomatis apabila *accountable* yang terkait telah menyeleseikan proses audit. Tampilan Halaman *Owner Notification Logs* dapat dilihat pada Gambar 34.

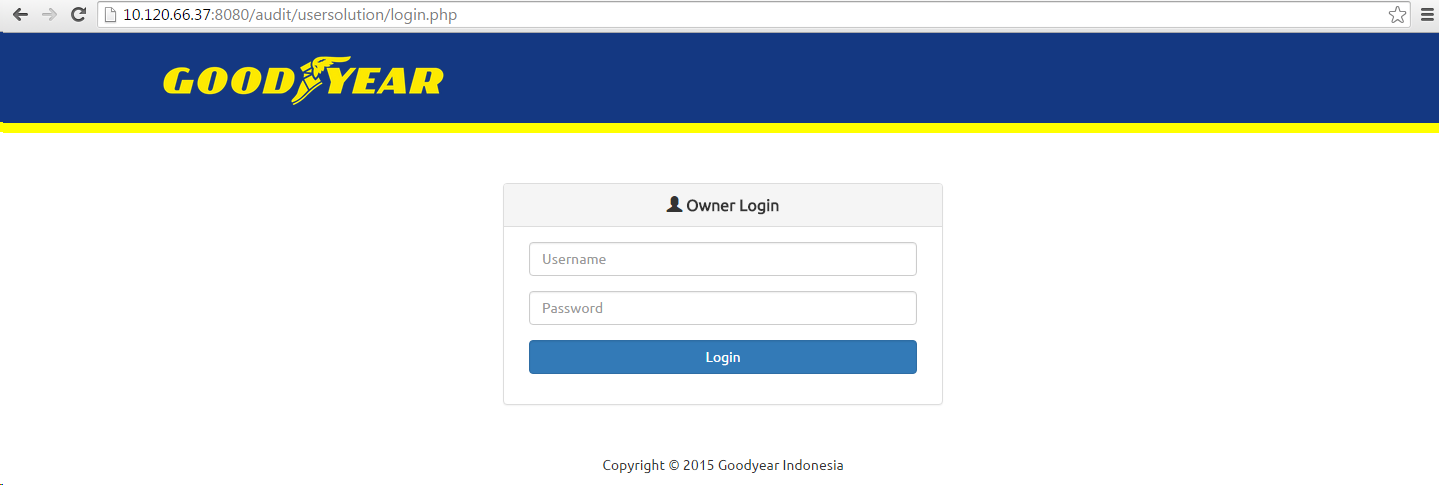


Gambar 34 Halaman *Notification Logs* (*Owner Area*)

Dalam sistem untuk *owner* terdapat beberapa halaman, diantaranya sebagai berikut:

* + - * 1. Halaman *Login*

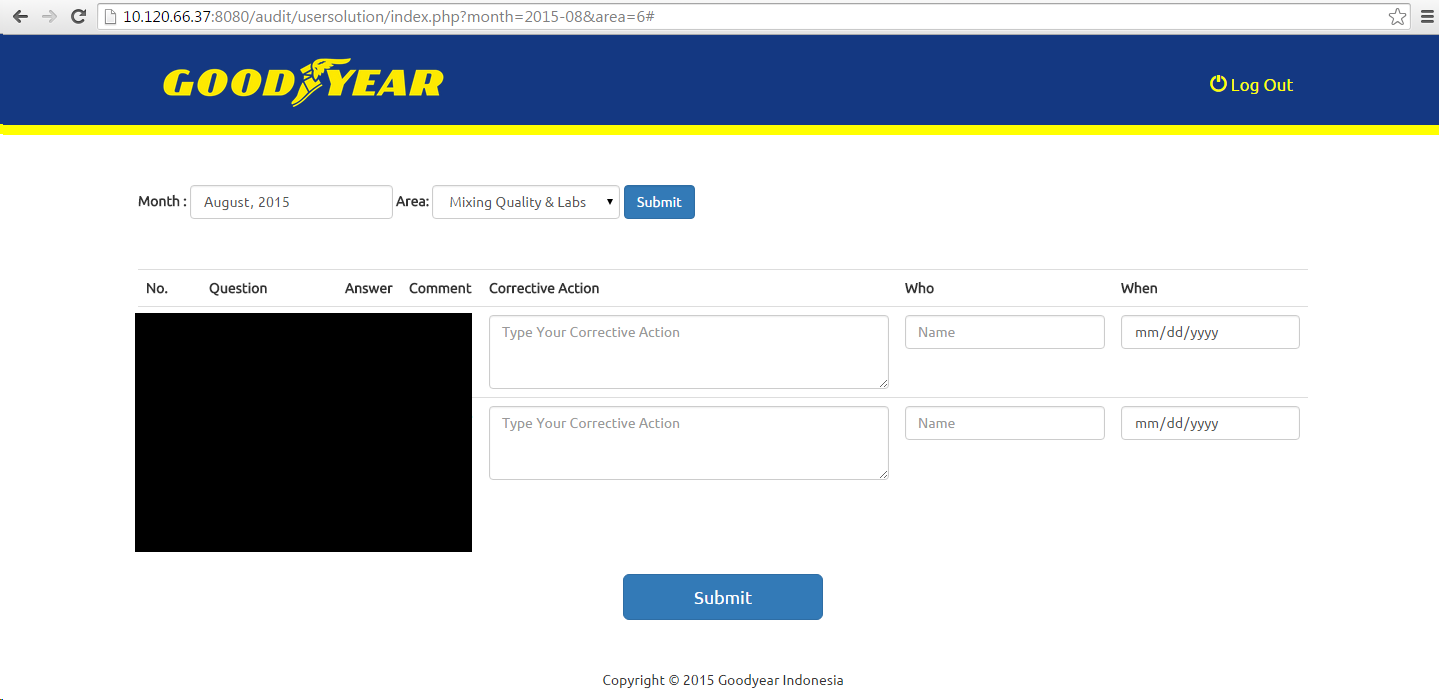
Halaman *Login* diimplementasikan untuk masuk dalam Sistem Layer AuditDatabase dengan menggunakan *input username* dan *password*. Tampilan Halaman *Login* dapat dilihat pada Gambar 35.



Gambar 35 Halaman *Login* untuk *owner*

* + - * 1. Halaman *Corrective Action*

Halaman *Corrective Action* diimplementasikan untuk *owner* memberikan *corrective action* terhadap jawaban hasil audit sesuai dengan bulan dan area yang dipilih. Pada halaman ini terdapat 6 kolom yaitu *question*, *answer*, *comment*, *corrective* *action*, *who*, dan *when*. *Question* merupakan pertanyaan yang diberikan pada saat audit kepada *auditor*, *answer* merupakan jawaban *auditor* untuk pertanyaan tersebut, *comment* merupakan permasalahan dari *auditor* terkait dengan pertanyaan tersebut, *corrective* *action* merupakan solusi dari *owner* untuk permasalahan yang didapatkan dari hasil audit, *who* merupakan orang yang bertanggung jawab dalam melaksanakan *corrective* *action* dan *when* merupakan waktu untuk melakukan *corrective* *action*. Tampilan Halaman *Corrective Action* dapat dilihat pada Gambar 36.



Gambar 36 Halaman *Corrective Action*

### *Deployment delivery and feedback*

Pada tahap ini, dilakukan pengujian sistem dengan *stakeholder*. Hasil pengujian halaman *admin* dapat dilihat pada Tabel 5. Sementara itu, hasil pengujian halaman audit yang diuji oleh *auditor* dapat dilihat pada Tabel 6. Kemudian hasil pengujian halaman *owner* dapat dilihat pada Tabel 7.

Tabel 5 Hasil pengujian halaman *admin* *Layer Audit Database*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Skenario pengujian | *Test case* | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Mengisi *username* yang tidak terdaftar, lalu mengeklik tombol ‘Login’ | Username: test  Password: goodyear | Sistem akan menampilkan pesan “Username doesn’t exist” | Sesuai harapan | *Valid* |
| 2 | Mengisi *username* yang telah terdaftar namun *password* salah, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: notvalid | Sistem akan menampilkan pesan “Wrong Password” | Sesuai harapan | *Valid* |
| 3 | Mengisi *username* dan *password* yang telah terdaftar, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: goodyear | Sistem akan menampilkan halaman utama yaitu halaman *Schedule* yang berisi jadwal audit | Sesuai harapan | *Valid* |
| 4 | Mengubah dan mengatur jadwal audit dengan mengeklik *checkbox* pada tabel jadwal audit di halaman *Schedule* |  | Sistem akan langsung menampilkan tabel jadwal audit yang telah diubah | Sesuai harapan | *Valid* |
| 5 | Memilih halaman ‘Question’ pada *menu bar* |  | Sistem akan menampilkan halaman *Question* | Sesuai harapan | *Valid* |
| 6 | Memilih area dan *form* yang diinginkan pada halaman *Question*, lalu mengeklik tombol ‘Submit’ |  | Sistem akan menampilkan daftar pertanyaan audit sesuai dengan area dan *form* yang telah dipilih | Sesuai harapan | *Valid* |
| 7 | Mengeklik tombol ‘Delete’ pada salah satu pertanyaan pada halaman *Question* |  | Sistem akan menampilkan halaman *Question* dengan tabel pertanyaan yang telah diubah dan menampilkan pesan “Delete Successfully” | Sesuai harapan | *Valid* |
| 8 | Mengubah pertanyaan atau *guideline* pada tabel pertanyaan di halaman *Question* |  | Sistem akan menampilkan halaman *Question* dengan tabel pertanyaan yang telah diubah | Sesuai harapan | *Valid* |
| 9 | Memilih halaman ‘Result’ pada *menu bar* |  | Sistem akan menampilkan halaman *Result* | Sesuai harapan | *Valid* |
| 10 | Memilih bulan, area, dan *accountable* yang diinginkan pada halaman *Result*, lalu mengeklik tombol ‘Submit’ |  | Sistem akan menampilkan daftar jawaban hasil audit sesuai dengan bulan, area, dan *accountable* yang telah dipilih | Sesuai harapan | *Valid* |
| 11 | Mengeklik tombol ‘Download’ pada halaman *Result* |  | Sistem akan memberikan tabel jawaban hasil audit kedalam komputer *admin* dengan format csv | Sesuai harapan | *Valid* |
| 12 | Memilih halaman ‘User’ pada *menu bar* |  | Sistem akan menampilkan halaman *User* yang berisi tabel *Database* *Auditor* | Sesuai harapan | *Valid* |
| 13 | Mengeklik tombol ‘Create New Accountable’ pada halaman *User* |  | Sistem akan menampilkan *pop up* ‘Create New User’ | Sesuai harapan | *Valid* |
| 14 | Mengisi data *user* baru pada *pop up* ‘Create New User’, lalu mengeklik tombol ‘Add’ |  | Sistem akan menampilkan halaman *User* dengan pesan “New User Added Successfully” | Sesuai harapan | *Valid* |
| 15 | Mengeklik tombol ‘Add Auditor’ pada halaman *User* |  | Sistem akan menampilkan *pop up* ‘Add Auditor’ | Sesuai harapan | *Valid* |
| 16 | Mengisi data *auditor* baru pada *pop up* ‘Add Auditor’, lalu mengeklik tombol ‘Add’ |  | Sistem akan menampilkan halaman *User* dengan menambahkan data *auditor* baru dalam tabel *Database Auditor* dan menampilkan pesan “New Auditor Added Successfully” | Sesuai harapan | *Valid* |
| 17 | Mengeklik tombol ‘Edit’ pada salah satu data tabel *Database Auditor* dalam halaman *User* |  | Sistem akan menampilkan *pop up* ‘Edit User’ | Sesuai harapan | *Valid* |
| 18 | Mengubah data *auditor* pada *pop up* ‘Edit’, lalu mengeklik tombol ‘Update’ |  | Sistem akan menampilkan halaman *User* dengan data *auditor* yang baru dalam tabel *Database Auditor* dan menampilkan pesan “Update Successfully” |  |  |
| 19 | Mengeklik tombol ‘Delete’ pada salah satu data tabel *Database* *Auditor* dalam halaman *User* |  | Sistem akan menampilkan *pop up* ‘Delete User’ |  |  |
| 20 | Mengeklik tombol ‘Delete’ pada *pop up* ‘Delete User’ |  | Sistem akan menampilkan halaman *User* dengan menghilangkan data *auditor* yang telah dihapus dalam tabel *Database* Auditor dan menampilkan pesan “User has been deleted” |  |  |
| 21 | Mengeklik tombol ‘View Notification Logs’ pada halaman *User* |  | Sistem akan menampilkan halaman yang berisi *Notification Logs* dari *email* yang sudah terkirim ke *Auditor* | Sesuai harapan | *Valid* |
| 22 | Memilih halaman ‘Owner’ pada *menu bar* |  | Sistem akan menampilkan halaman *Owner* yang berisi tabel *Database Owner* | Sesuai harapan | *Valid* |
| 23 | Mengeklik tombol ‘Create New Owner’ pada halaman *Owner* |  | Sistem akan menampilkan *pop up* ‘Create New Owner’ | Sesuai harapan | *Valid* |
| 24 | Mengisi data owner baru pada *pop up* ‘Create New Owner’, lalu mengeklik tombol ‘Add’ |  | Sistem akan menampilkan halaman *Owner* dengan menambahkan data *owner* baru dalam tabel *Database Owner* dan menampilkan pesan “New Owner Added Successfully” | Sesuai harapan | *Valid* |
| 25 | Mengeklik tombol ‘Edit’ pada salah satu data tabel *Database Owner* dalam halaman *Owner* |  | Sistem akan menampilkan *pop up* ‘Edit User’ | Sesuai harapan | *Valid* |
| 26 | Mengubah data *owner* pada *pop up* ‘Edit’, lalu Mengeklik tombol ‘Update’ |  | Sistem akan menampilkan halaman *Owner* dengan data *owner* yang baru dalam tabel *Database* *Owner* dan menampilkan pesan “Update Successfully” | Sesuai harapan | *Valid* |
| 27 | Mengeklik tombol ‘Delete’ pada salah satu data tabel *Database Owner* dalam halaman *Owner* |  | Sistem akan menampilkan *pop up* ‘Delete User’ | Sesuai harapan | *Valid* |
| 28 | Mengeklik tombol ‘Delete’ pada *pop up* ‘Delete User’ |  | Sistem akan menampilkan halaman *Owner* dengan menghilangkan data *owner* yang telah dihapus dalam tabel *Database Owner* dan menampilkan pesan “User has been deleted” | Sesuai harapan | *Valid* |
| 29 | Mengeklik tombol ‘View Notification Logs’ pada halaman *Owner* |  | Sistem akan menampilkan halaman yang berisi *Notification Logs* dari *email* yang sudah terkirim ke *owner* | Sesuai harapan | *Valid* |

Tabel 6 Hasil pengujian halaman *auditor* *Layer Audit Database*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Skenario pengujian | *Test case* | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Mengisi *username* yang tidak terdaftar, lalu mengeklik tombol ‘Login’ | Username: test  Password: goodyear | Sistem akan menampilkan pesan “Username doesn’t exist” | Sesuai harapan | *Valid* |
| 2 | Mengisi *username* yang telah terdaftar namun password salah, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: notvalid | Sistem akan menampilkan pesan “Wrong Password” | Sesuai harapan | *Valid* |
| 3 | Mengisi *username* dan *password* yang telah terdaftar, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: goodyear | Sistem akan menampilkan halaman utama yaitu halaman Audit | Sesuai harapan | *Valid* |
| 4 | Memilih area lalu mengeklik tombol ‘Submit’ pada halaman Audit |  | Sistem akan menampilkan halaman Audit yang berisi pertanyaan-pertanyaan audit | Sesuai harapan | *Valid* |
| 5 | Mengisi beberapa pertanyaan audit, lalu mengeklik tombol ‘Submit’ pada halaman Audit |  | Sistem akan menampilkan kembali halaman Audit dengan menampilkan pesan “Your answer isn’t complete” | Sesuai harapan | *Valid* |
| 6 | Mengisi semua pertanyaan audit, lalu mengeklik tombol ‘Submit’ pada halaman Audit |  | Sistem akan menampilkan kembali halaman Audit yang sudah tidak berisi pertanyaan-pertanyaan audit | Sesuai harapan | *Valid* |
| 7 | Memilih halaman ‘Calendar’ pada *menu bar* |  | Sistem akan menampilkan halaman *Calendar* yang berisi jadwal audit semua area | Sesuai harapan | *Valid* |

Tabel 7 Hasil pengujian halaman *owner* *Layer Audit Database*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Skenario pengujian | *Test case* | Hasil yang diharapkan | Hasil pengujian | Kesimpulan |
| 1 | Mengisi *username* yang tidak terdaftar, lalu mengeklik tombol ‘Login’ | Username: test  Password: goodyear | Sistem akan menampilkan pesan “Username doesn’t exist” | Sesuai harapan | *Valid* |
| 2 | Mengisi *username* yang telah terdaftar namun *password* salah, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: notvalid | Sistem akan menampilkan pesan “Wrong Password” | Sesuai harapan | *Valid* |
| 3 | Mengisi *username* dan *password* yang telah terdaftar, lalu mengeklik tombol ‘Login’ | Username: goodyear  Password: goodyear | Sistem akan menampilkan halaman utama yaitu halaman *Corrective Action* | Sesuai harapan | *Valid* |
| 4 | Memilih bulan dan area lalu mengeklik tombol ‘Submit’ pada halaman *Corrective Action* |  | Sistem akan menampilkan halaman *Corrective Action* yang berisi jawaban-jawaban *auditor* | Sesuai harapan | *Valid* |
| 5 | Mengisi semua isian pada halaman *Corrective* *Action*, lalu mengeklik tombol ‘Submit’ |  | Sistem akan menampilkan halaman *Corrective* *Action* yang kosong | Sesuai harapan | *Valid* |

Berdasarkan hasil *testing* dengan *user*, secara fungsional Sistem Layer AuditDatabase sudah bisa digunakan oleh *admin, auditor*, *dan owner*. Sistem ini sudah diterapkan oleh *admin*, dua orang auditor yang mendapat jadwal mengaudit pada bulan September, serta satu orang *owner* yang memiliki area *Mixing Quality and Labs*. Sistem belum diterapkan oleh semua pengguna dikarenakan jadwal audit yang berbeda-beda.

# SIMPULAN DAN SARAN

## Simpulan

Sistem Rotation Schedule memudahkan tugas dari departemen *Human Resources* dalam mengatur jadwal rotasi tiap departemen. Dengan adanya sistem ini, setiap departemen dapat mengatur jadwal rotasi mereka masing-masing yang kemudian disatukan dalam *database* yang terpusat.

Sementara itu, pada Sistem Layer Audit Database dapat memudahkan *auditor* dalam melakukan proses audit yang semula dilakukan secara manual. Sistem ini dapat meminimalisir terlewatnya proses audit dikarenakan setiap jadwal audit, *auditor* mendapat notifikasi berupa *email* yang berisi *reminder* untuk mengaudit. Selain itu, *owner* juga akan langsung mendapat notifikasi apabila terdapat hasil audit yang membutuhkan *corrective action*.

## Saran

Dalam pengembangan Sistem Rotation Schedulediperlukan adanya *reminder* berupa notifikasi melalui *email* supaya *user* dapat menyusun jadwal rotasi secara rutin. Sementara dalam pengembangan Sistem Layer Audit Database diperlukan adanya fungsi untuk mencetak daftar pertanyaan audit agar *auditor* dapat mengaudit secara langsung di area terkait. Tampilan antarmuka dari Sistem Layer Audit Database disarankan dapat lebih interaktif agar lebih mudah dimengerti dan sesuai dengan keinginan pengguna.

# DAFTAR PUSTAKA

Mulyadi. 2002. *Auditing*. Jakarta: Salemba Empat.

O’Brien, James. A. 2005. *Pengantar Sistem Informasi Perspektif Bisnis dan Manajerial.* Jakarta: Salemba Empat.

Pressman, Roger S. 2010. *Software Engineering*: A Practitioner’s Approach,7th Edition. New York: The McGraw Hill Companies.

Wahyudi, Bambang. 2002. *Manajemen Sumber Daya Manusia*. Bandung: Sulita.

[PT Goodyear Indonesia]. 2012. Laporan Tahunan. Goodyear Indonesia [terhubung berkala] www.goodyear-indonesia.com (20 September 2015).

# LAMPIRAN

Lampiran 1 Deskripsi *use case* Sistem Rotation Schedule

|  |  |
| --- | --- |
| *Use case Name:* | Mengubah data jadwal rotasi |
| *Scenario:* | *User* mengubah data jadwal rotasi departemennya |
| *Triggering Event:* | Mengeklik *cell* pada tabel jadwal rotasi |
| *Brief Description:* | Jadwal rotasi harus sudah tersedia |
| *Actors:* | *User* |
| *Related Use Case:* | - |
| *Stakeholders:* | *User*: Mengganti data jadwal rotasi |
| *Preconditions:* | *User* belum masuk ke halaman *Rotation Schedule* |
| *Postconditions:* | Jadwal rotasi sudah terubah |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *User* mengeklik *cell* pada tabel jadwal rotasi |  | | 2 *User* mengetik data baru pada tabel jadwal rotasi | 2.1 Data baru ter-*update* di *database* | |
| *Exception Conditions:* | *User* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengunggah jadwal rotasi |
| *Scenario:* | *User* mengunggah jadwal rotasi |
| *Triggering Event:* | Mengeklik “Upload” pada halaman *Rotation Schedule* |
| *Brief Description:* | *User* mengunggah *file* jadwal rotasi departemen masing-masing pada bulan yang belum terdapat *Rotation Schedule* |

Lanjutan

|  |  |
| --- | --- |
| *Actors:* | *User* |
| *Related Use Case:* | - |
| *Stakeholders:* | *User*:  Mendapatkan informasi tentang *interface* yang baik |
| *Preconditions:* | Pengguna belum masuk ke halaman *Rotation Schedule* |
| *Postconditions:* | Jadwal rotasi ditampilkan pada halaman *Rotation Schedule* |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *User* memilih bulan sesuai dengan jadwal rotasi yang akan diunggah |  | | 2 *User* mengeklik “Submit” | 2.1 Sistem menampilkan tabel *Rotation Schedule* yang masih kosong | | 3 *User* mengeklik “Upload File” | 3.1 Sistem menampilkan *pop up* untuk mengunggah *file* | | 4 *User* memilih *file* yang akan diunggah dari computer |  | | 5 *User* melakukan *submit file* tersebut | 5.1 Memvalidasikan tipe *file*, jumlah hari dan jumlah karyawan pada *file* yang diunggah  5.2 Mengembalikan ke halaman *Rotation Schedule* dan menampilkan jadwal rotasi yang telah diunduh | |
| *Exception Conditions:* | *User* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengunduh jadwal rotasi |

Lanjutan

|  |  |
| --- | --- |
| *Scenario:* | *User* mengunduh *Rotation Schedule* |
| *Triggering Event:* | *User* mengeklik “Download” pada halaman *Rotation Schedule* |
| *Brief Description:* | *User* mengunduh *Rotation Schedule* pada bulan yang dipilih dan sudah terdapat *Rotation Schedule* sesuai dengan departemen *user* |
| *Actors:* | *User* |
| *Related Use Case:* | Mengunggah *Rotation Schedule* |
| *Stakeholders:* | *User*: Mendapatkan informasi tentang *interface* yang baik |
| *Preconditions:* | Pengguna belum masuk ke halaman *Rotation Schedule* |
| *Postconditions:* | Jadwal rotasi tersimpan dalam dokumen csv |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *User* memilih bulan yang diingkan untuk diunduh *file* *Rotation Schedule* nya |  | | 2 *User* mengeklik *submit* |  | | 3 *User* mengeklik “Download” pada halaman *Rotation Schedule* | 3.1 Sistem memberikan *file* *Rotation Schedule* | |
| *Exception Conditions:* | *User* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengunduh jadwal rotasi semua departemen |
| *Scenario:* | *Admin* mengunduh jadwal rotasi departemen yang dipilih sesuai bulannya |
| *Triggering Event:* | *Admin* mengeklik “Download” pada halaman *Rotation Schedule* |

Lanjutan

|  |  |
| --- | --- |
| *Brief Description:* | *Admin* mengunduh jadwal rotasi yang sudah tersedia sesuai departemen dan bulan yang dipilih |
| *Actors:* | *Admin* |
| *Related Use Case:* | Mengunggah jadwal rotasi |
| *Stakeholders:* | *Admin*: Mendapatkan informasi tentang *interface* yang baik |
| *Preconditions:* | Pengguna belum masuk ke halaman *Rotation Schedule* |
| *Postconditions:* | Jadwal rotasi tersimpan dalam dokumen csv |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* memilih departemen dan bulan yang diinginkan untuk mengunduh jadwal rotasi |  | | 2 *Admin* mengeklik submit |  | | 3 *Admin* mengeklik “Download” pada halaman *Rotation Schedule* | 3.1 Sistem memberikan *file* *Rotation Schedule* sesuai | |
| *Exception Conditions:* | *Admin* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Membuat *user* baru |
| *Scenario:* | *Admin* membuat *user* baru sesuai departemen |
| *Triggering Event:* | *Admin* mengeklik“Create New User” pada halaman *User Maintenance* |
| *Brief Description:* | *-* |
| *Actors:* | *Admin* |

Lanjutan

|  |  |
| --- | --- |
| *Related Use Case:* | *-* |
| *Stakeholders:* | *Admin:* Membuat *user* baru  *User:* Dapat melakukan *login* |
| *Preconditions:* | *Admin* belum masuk ke halaman *User Maintenance* |
| *Postconditions:* | *User* baru sudah terdaftar di *database* |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik “Create New User” pada halaman *User* *Maintenance* | 1.1 Sistem menmpilkan *pop up* yang berisi *form* pengisian *user* baru | | 2 *Admin* mengisi data *user* baru dalam *form* |  | | 3 *Admin* mengeklik “Submit” | 3.1 Sistem menyimpan data *user* baru dalam *database* | |
| *Exception Conditions:* | *Admin* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengubah data *user* |
| *Scenario:* | *Admin* mengubah data *user* yang sudah tersedia di *database* |
| *Triggering Event:* | *Admin* mengeklik “Edit” pada tabel *database user* di halaman *User Maintenance* |
| *Brief Description:* | *Admin* mengubah data *user* yang sebelumnya sudah ada di *database* |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |

Lanjutan

|  |  |
| --- | --- |
| *Stakeholders:* | *Admin*: Mengubah data *user*  *User*: Data *user* berubah |
| *Preconditions:* | *Admin* belum masuk ke halaman *User Maintenance* |
| *Postconditions:* | Data *user* sudah berubah di *database* |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik tombol “Edit” pada tabel *database user* | 1.1 Sistem menampilkan *pop up* | | 2 *Admin* mengisi *form* *database* *user* |  | | 3 *Admin* mengeklik “Submit” | 3.1 Sistem mengubah data *user* di *database* | |
| *Exception Conditions:* | *Admin* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Menghapus *user* |
| *Scenario:* | *Admin* menghapus data *user* dari kolom *database* |
| *Triggering Event:* | *Admin* mengeklik “Delete” pada halaman *User Maintenance* |
| *Brief Description:* | *Admin* menghapus *user* yang sebelumnya sudah tersimpan di *database* |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Menghapus data *user* |
| *Preconditions:* | *Admin* belum masuk ke halaman *User Maintenance* |
| *Postconditions:* | Data *user* sudah terhapus dari *database* |

Lanjutan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik tombol “Delete” di kolom *database* *user* | 1.1 Sistem menampilkan *pop up* berupa konfirmasi akan dihapusnya *user* | | 2 *Admin* mengeklik tombol “Delete” | 2.1 Sistem menghapus data *user* dari *database*  2.2 Sistem mengembalikan ke halaman *User Maintenance* | |
| *Exception Conditions:* | *Admin* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengunduh *template* |
| *Scenario:* | *User* mengunduh *template* jadwal rotasi departemen |
| *Triggering Event:* | *User* mengeklik “Download” pada halaman *Template* |
| *Brief Description:* | *User* mengunduh jadwal rotasi yang sudah tersedia di halaman *Template* |
| *Actors:* | *User* |
| *Related Use Case:* | - |
| *Stakeholders:* | *User* |
| *Preconditions:* | *User* belum masuk ke halaman *Template* |
| *Postconditions:* | *Template* jadwal rotasi tersimpan dalam dokumen csv |

Lanjutan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *User* memilih halaman *Template* | 1.1 Sistem menampilkan halaman *Template* | | 2 *User* mengeklik “Download” pada halaman *Template* | 2.1 Sistem memberikan dokumen *template* jadwal rotasi | |
| *Exception Conditions:* | *User* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengunduh *example* |
| *Scenario:* | *User* mengunduh *example* jadwal rotasi departemen |
| *Triggering Event:* | *User* mengeklik “Download” pada halaman *Example* |
| *Brief Description:* | *User* mengunduh jadwal rotasi yang sudah tersedia di halaman *Example* |
| *Actors:* | *User* |
| *Related Use Case:* | - |
| *Stakeholders:* | *User* |
| *Preconditions:* | *User* belum masuk ke halaman *Example* |
| *Postconditions:* | *Example* jadwal rotasi tersimpan dalam dokumen csv |

Lanjutan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *User* memilih halaman *Example* | 1.1 Sistem menampilkan halaman *Example* | | 2 *User* mengeklik “Download” pada halaman *Example* | 2.1 Sistem memberikan dokumen *example* jadwal rotasi | |
| *Exception Conditions:* | *User* belum melakukan *login* |

Lampiran 2 Deskripsi *use case* Sistem Layer Audit Database

|  |  |
| --- | --- |
| *Use case Name:* | Mengatur jadwal audit |
| *Scenario:* | *Admin* mengatur jadwal audit setiap bulan dalam setahun |
| *Triggering Event:* | Mengeklik *checkbox* pada tabel jadwal audit |
| *Brief Description:* | Memilih tahun yang akan diatur jadwal auditnya |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Mengatur jadwal audit  *Auditor*: Mendapat jadwal audit sesuai aturan *admin* |
| *Preconditions:* | *Admin* belum masuk ke halaman *Schedule* |
| *Postconditions:* | Jadwal audit sudah terubah |

Lanjutan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik *checkbox* pada tabel jadwal audit | 1.1 Sistem menyimpan jadwal terbaru di *database* | |
| *Exception Conditions:* | *Admin* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengubah pertanyaan audit |
| *Scenario:* | *Admin* mengubah pertanyaan audit sesuai area dan *form* yang dipilih |
| *Triggering Event:* | Mengeklik *cell* pada tabel pertanyaan |
| *Brief Description:* | Area dan *form* harus dipilih terlebih dahulu |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Mengubah pertanyaan audit |
| *Preconditions:* | *Admin* belum masuk ke halaman *Question* |
| *Postconditions:* | Pertanyaan audit sudah terubah |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik *cell* pada tabel pertanyaan |  | | 2 *Admin* mengetik isi pertanyaan pada tabel pertanyaan | 2.1 Data pertanyaan baru ter-*update* di *database* | |
| *Exception*  *Conditions:* | *Admin* belum melakukan *login* |

Lanjutan

|  |  |
| --- | --- |
| *Use case Name:* | Menghapus pertanyaan audit |
| *Scenario:* | *Admin* menghapus pertanyaan audit sesuai area dan *form* yang dipilih |
| *Triggering Event:* | Mengeklik tombol “Delete” pada tabel pertanyaan audit |
| *Brief Description:* | - |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Menghapus pertanyaan audit |
| *Preconditions:* | *Admin* belum masuk ke halaman *Question* |
| *Postconditions:* | Pertanyaan audit yang dipilih *admin* telah terhapus |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* memilih area dan *form* pertanyaan audit | 1.1 Sistem menampilkan halaman *Question* berdasarkan area dan *form* yang dipilih | | 2 *Admin* mengeklik tombol “Delete” | 2.1 Sistem menampilkan *pop up* berupa konfirmasi pertanyaan yang akan dihapus | | 3 *Admin* mengeklik tombol “Delete” | 3.1 Sistem menghapus pertanyaan yang terpilih di *database* | |
| *Exception Conditions:* | *Admin* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengubah data *auditor* |
| *Scenario:* | *Admin* mengubah data *auditor* |
| *Triggering Event:* | Mengeklik tombol “Edit” pada halaman *Database Auditor* |

Lanjutan

|  |  |
| --- | --- |
| *Brief Description:* | Data *auditor* harus lengkap |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Mengubah data *auditor*  *Auditor*: Data *auditor* mengalami perubahan |
| *Preconditions:* | *Admin* belum masuk ke halaman *Database Auditor* |
| *Postconditions:* | Data *auditor* sudah terubah |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik tombol “Edit” pada tabel *Database Auditor* | 1.1 Sistem menampilkan *pop up* berisi *form* data *auditor* | | 2 *Admin* mengetik data baru pada *form auditor* | 2.1 Data *auditor* baru ter-*update* di *database* | |
| *Exception Conditions:* | *Admin* tidak memasukkan data *auditor* secara lengkap |

|  |  |
| --- | --- |
| *Use case Name:* | Menambah *auditor* |
| *Scenario:* | *Admin* menambah *auditor* baru |
| *Triggering Event:* | Mengeklik tombol “Create Auditor” pada halaman *Database Auditor* |
| *Brief Description:* | *Auditor* yang ingin ditambah belum ada sebelumnya |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |

Lanjutan

|  |  |
| --- | --- |
| *Stakeholders:* | *Admin*: Menambah *auditor* baru  *Auditor*: Dapat menjawab pertanyaan audit |
| *Preconditions:* | *Admin* belum masuk ke halaman *Database Auditor* |
| *Postconditions:* | *Auditor* sudah terdaftar di *database* |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik tombol “Create Auditor” pada halaman *Database Auditor* | 1.1 Sistem menampilkan *pop up form auditor* yang belum terisi | | 2 *Admin* mengetik data *auditor* pada *form* |  | | 3 *Admin* mengeklik tombol “Submit” | 3.1 Data *auditor* berhasil ditambahkandi *database* | |
| *Exception Conditions:* | Data *auditor* sudah ada sebelumnya |

|  |  |
| --- | --- |
| *Use case Name:* | Mengubah data *owner* |
| *Scenario:* | *Admin* mengubah data *owner* |
| *Triggering Event:* | Mengeklik tombol “Edit” pada halaman *Database* *Owner* |
| *Brief Description:* | Data *owner* harus lengkap |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Mengubah data *owner*  *Owner*: Data *owner* mengalami perubahan |
| *Preconditions:* | *Admin* belum masuk ke halaman *Database* *Owner* |

Lanjutan

|  |  |
| --- | --- |
| *Postconditions:* | Data *owner* sudah terubah |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik tombol “Edit” pada tabel *Database Owner* | 1.1 Sistem menampilkan *pop up* berisi *form* data *owner* | | 2 *Admin* mengetik data baru pada *form owner* | 2.1 Data *owner* baru ter-*update* di *database* | |
| *Exception Conditions:* | *Admin* tidak memasukkan data *owner* secara lengkap |

|  |  |
| --- | --- |
| *Use case Name:* | Menambah *owner* |
| *Scenario:* | *Admin* menambah *owner* baru |
| *Triggering Event:* | Mengeklik tombol “Create Owner” pada halaman *Database Owner* |
| *Brief Description:* | *Owner* yang ingin ditambah belum ada sebelumnya |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Menambah *owner* baru  *Owner*: Dapat mengisi *corrective action* |
| *Preconditions:* | *Admin* belum masuk ke halaman *Database* *Owner* |
| *Postconditions:* | *Owner* sudah terdaftar di *database* |

Lanjutan

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik tombol “Create Owner” pada halaman *Database* *Owner* | 1.1 Sistem menampilkan *pop up* *form owner* yang belum terisi | | 2 *Admin* mengetik data *owner* pada *form* |  | | 3 *Admin* mengeklik tombol “Submit” | 3.1 Data *owner* berhasil ditambahkandi *database* | |
| *Exception Conditions:* | Data *owner* sudah ada sebelumnya |

|  |  |
| --- | --- |
| *Use case Name:* | Menghapus data *owner* |
| *Scenario:* | *Admin* menghapus data *owner* di *database* |
| *Triggering Event:* | mengeklik tombol “Delete”pada tabel *Database Owner* |
| *Brief Description:* | *Owner* yang dihapus sebelumnya sudah terdaftar di *database* |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Menghapus data *owner*  *Owner*: Data *owner* terhapus sehingga tidak dapat melakukan *corrective action* |
| *Preconditions:* | *Admin* belum masuk ke halaman *Database* *Owner* |
| *Postconditions:* | Data *owner* sudah terhapus |

Lanjutan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik tombol “Delete” pada tabel *Database Owner* | 1.1 Sistem menampilkan *pop up* yang berisi konfirmasi data *owner* akan dihapus | | 2 *Admin* mengeklik tombol “Delete” pada *pop up* | 2.1 Data *owner* terhapus di dalam *database*  2.2 Sistem menampilkan halaman *Database* *Owner* yang terbaru | |
| *Exception Conditions:* | *Admin* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Menghapus data *auditor* |
| *Scenario:* | *Admin* menghapus data *auditor* di *database* |
| *Triggering Event:* | mengeklik tombol “Delete”pada tabel *Database Auditor* |
| *Brief Description:* | *Auditor* yang dihapus sebelumnya sudah terdaftar di *database* |
| *Actors:* | *Admin* |
| *Related Use Case:* | - |
| *Stakeholders:* | *Admin*: Menghapus data *auditor*  *Auditor*: Data *auditor* terhapus sehingga tidak dapat menjawab pertanyaan audit |
| *Preconditions:* | *Admin* belum masuk ke halaman *Database* *Auditor* |
| *Postconditions:* | Data *auditor* sudah terhapus |

Lanjutan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* mengeklik tombol “Delete” pada tabel Database Auditor | 1.1 Sistem menampilkan *pop up* yang berisi konfirmasi data *auditor* akan dihapus | | 2 *Admin* mengeklik tombol “Delete” pada *pop up* | 2.1 Data *auditor* terhapus di dalam *database*  2.2 Sistem menampilkan halaman *Database Auditor* yang terbaru | |
| *Exception Conditions:* | *Admin* belum melakukan *login* |

|  |  |
| --- | --- |
| *Use case Name:* | Mengunduh jawaban hasil audit |
| *Scenario:* | *Admin* mengunduh jawaban hasil audit pada halaman *Result* |
| *Triggering Event:* | mengeklik tombol “Download” pada halaman *Result* |
| *Brief Description:* | Jawaban hasil audit harus sudah tersedia |
| *Actors:* | *Admin* |
| *Related Use Case:* | Menjawab pertanyaan audit |
| *Stakeholders:* | *Auditor*: Menjawab pertanyaan audit |
| *Preconditions:* | *Admin* belum masuk ke halaman *Result* |
| *Postconditions:* | Jawaban hasil audit sudah terunduh |

Lanjutan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Admin* memilih area dan *form* jawaban hasil audit pada halaman *Result* lalu klik “Submit” | 1.1 Sistem menampilkan jawaban hasil audit | | 2 *Admin* mengeklik tombol “Download” | 2.1 Jawaban hasil audit terunduh dalam dokumen csv | |
| *Exception Conditions:* | Jawaban hasil audit belum tersedia |

|  |  |
| --- | --- |
| *Use case Name:* | Menjawab pertanyaan audit |
| *Scenario:* | *Auditor* menjawab pertanyaan audit sesuai jadwal yang ditentukan *admin* |
| *Triggering Event:* | mengeklik tombol “Submit” pada halaman Audit |
| *Brief Description:* | Jadwal audit sudah sesuai bulannya |
| *Actors:* | *Auditor* |
| *Related Use Case:* | Mengunduh jawaban hasil audit, Menjawab *corrective action* |
| *Stakeholders:* | *Auditor*: Menjawab pertanyaan audit  *Admin*: Dapat mengunduh jawaban hasil audit  *Owner*: Dapat menjawab *corrective action* |
| *Preconditions:* | *Auditor* belum masuk ke halaman Audit |
| *Postconditions:* | Jawaban audit sudah masuk ke *database* |

Lanjutan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Auditor* menjawab pertanyaan audit dengan lengkap | 1.1 Sistem menampilkan pertanyaan audit | | 2 *Auditor* mengeklik tombol “Submit” | 2.1 Jawaban hasil audit tersimpan di *database* | |
| *Exception Conditions:* | *Auditor* tidak menjawab pertanyaan audit dengan lengkap |

|  |  |
| --- | --- |
| *Use case Name:* | Menjawab *corrective action* |
| *Scenario:* | *Owner* menjawab *corrective action* apabila terdapat jawaban “No” pada hasil audit |
| *Triggering Event:* | mengeklik tombol “Submit” pada halaman *Corrective Action* |
| *Brief Description:* | Jawaban “No” pada hasil audit tersedia |
| *Actors:* | *Owner* |
| *Related Use Case:* | Menjawab pertanyaan audit |
| *Stakeholders:* | *Owner*: Menjawab *corrective action*  *Auditor*: Menjawab pertanyaan audit dengan pilihan jawaban “No” |
| *Preconditions:* | *Owner* belum masuk ke halaman *Corrective Action* |
| *Postconditions:* | Hasil *corrective action* berhasil di *submit* ke *database* |

Lanjutan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Flow of Activities:* | |  |  | | --- | --- | | *Actor* | *System* | | 1 *Owner* mengisi *corrective action* | 1.1 Sistem menampilkan jawaban hasil audit dan kolom *corrective action* | | 2 *Owner* mengeklik “Submit” pada halaman *Corrective Action* | 2.1 Data *corrective action* tersimpan di dalam *database* | |
| *Exception Conditions:* | Tidak ada jawaban “No” dari hasil audit |