




# UMN Libtii SI 6

## Maria Claudia Samur\_00000056684\_Turnitin.pdf

 MARIA CLAUDIA SAMUR  
 2024 GANJIL - SKRIPSI SISTEM INFORMASI  
 Universitas Multimedia Nusantara

### Document Details

**Submission ID**

trn:oid::1:3125187365

**Submission Date**

Jan 7, 2025, 8:47 AM GMT+7

**Download Date**

Jan 7, 2025, 8:52 AM GMT+7

**File Name**

Maria\_Claudia\_Samur\_00000056684\_Turnitin.pdf

**File Size**

417.8 KB

45 Pages

9,784 Words

59,844 Characters




# 13% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- Bibliography
  - Quoted Text
- 

## Top Sources

- 12%  Internet sources
  - 4%  Publications
  - 4%  Submitted works (Student Papers)
-

## Top Sources

- 12% Internet sources
- 4% Publications
- 4% Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	kc.umn.ac.id	2%
2	Internet	123dok.com	2%
3	Internet	docplayer.info	1%
4	Internet	widuri.raharja.info	1%
5	Student papers	Universitas Pancasila	0%
6	Student papers	pnl	0%
7	Internet	jurnal.pancabudi.ac.id	0%
8	Internet	repository.nurulfikri.ac.id	0%
9	Student papers	Universitas Pamulang	0%
10	Internet	text-id.123dok.com	0%
11	Internet	repository.uniks.ac.id	0%

12	Internet	core.ac.uk	0%
13	Internet	id.wikipedia.org	0%
14	Internet	repository.unama.ac.id	0%
15	Publication	Guna yanti kemala Sari siregar, Dani Anggoro. "PERANCANGAN APLIKASI PENGOLAH...	0%
16	Internet	pt.scribd.com	0%
17	Internet	repository.upi.edu	0%
18	Student papers	Sriwijaya University	0%
19	Internet	repository.unisbablitar.ac.id	0%
20	Student papers	Universitas Bina Sarana Informatika	0%
21	Internet	repository.unika.ac.id	0%
22	Student papers	Universitas Putera Batam	0%
23	Student papers	Iain Palopo	0%
24	Student papers	Keimyung University	0%
25	Student papers	Politeknik Negeri Bandung	0%

26	Student papers	STT PLN	0%
27	Student papers	Universitas Islam Negeri Sumatera Utara	0%
28	Student papers	Universitas Lancang Kuning	0%
29	Student papers	Universitas Raharja	0%
30	Student papers	Universitas Teknologi Sumbawa	0%
31	Internet	computeraddict13.blogspot.com	0%
32	Publication	Muhammad Dedi Irawan, Herviana Herviana. "IMPLEMENTASI LOGIKA FUZZY DAL...	0%
33	Publication	Norma Ningsih, Karimatun Nisa, Endryco Farel Rianrachmatullah, Bintang Dest...	0%
34	Internet	jurnalunibi.unibi.ac.id	0%
35	Internet	moam.info	0%
36	Publication	Diana Diana, Alandika Dwi Rama, Rosa Fitriasari Asa. "Pemanfaatan Augmented ...	0%
37	Internet	adoc.tips	0%
38	Internet	eprints.walisongo.ac.id	0%
39	Internet	library.binus.ac.id	0%

40	Internet	ojs.palcomtech.ac.id	0%
41	Internet	pdfcoffee.com	0%
42	Internet	repository.dinamika.ac.id	0%
43	Internet	repository.unej.ac.id	0%
44	Internet	repository.upbatam.ac.id	0%
45	Internet	www.scribd.com	0%
46	Publication	Ragil Tegar Imanda, Anna Mukhayaroh. "Metode Design Thinking Perancangan U...	0%
47	Publication	Zaini Syarifudin, Yahfizham Yahfizham. "PERANCANGAN MANAJEMEN PROYEK SIS...	0%
48	Internet	eprints.pancabudi.ac.id	0%
49	Internet	eprints.upj.ac.id	0%
50	Internet	filopendi.blogspot.com	0%
51	Internet	ml.scribd.com	0%
52	Internet	repo.bunghatta.ac.id	0%
53	Publication	Agi Maehesa Putri, Neng Husnul Hamdiah. "POLA IMPLEMENTASI MANAJEMEN S...	0%

54	Publication	Imam Adlin Sinaga, Muhammad Zaki Zikrillah Baharuddin Nasution, Abdul Hakim...	0%
55	Internet	digilib.unila.ac.id	0%
56	Internet	epub.imandiri.id	0%
57	Internet	journal.uad.ac.id	0%
58	Internet	mpl-alfbs2004347.blogspot.com	0%
59	Internet	repository.teknokrat.ac.id	0%
60	Internet	repository.ub.ac.id	0%

3

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Perkembangan teknologi semakin cepat dengan munculnya perangkat-perangkat yang lebih canggih. Dampak yang diberikan oleh perkembangan teknologi adalah munculnya inovasi – inovasi berbasis internet yang dapat dijangkau oleh siapapun dan dimanapun[1]. Salah satu bidang bisnis yang sangat mencolok perubahannya adalah bisnis di bidang ritel.

Dahulu bisnis ritel dilakukan secara tradisional yaitu yang sebelumnya bergantung pada toko fisik sebagai tempat transaksi dan interaksi dengan pelanggan, kini menghadapi tantangan besar[2]. Dalam beberapa dekade terakhir, perusahaan ritel telah berkembang pesat dan menjadi salah satu sektor bisnis yang paling prospektif. Persaingan di bidang bisnis ritel yang semakin kompetitif mendorong seluruh pihak yang terlibat untuk meningkatkan efektifitas dan efisiensi kerja[3]. Pelaku usaha ritel dituntut untuk dapat melakukan pekerjaannya dengan usaha yang seminimal mungkin dan waktu yang sesingkat mungkin.

Dalam mengelola *user* salah satu bagian yang sangat penting, terutama bagi perusahaan dengan memiliki banyak karyawan seperti PT. Matahari Department Store Tbk. Perusahaan dapat mengelola *user* secara lebih efisien dan efektif, serta memastikan bahwa setiap *user* memiliki akses aplikasi sesuai dengan perannya.

Gambar 1. 1 Alur Proses Permintaan Akses User Aplikasi

Sumber: Alur proses permintaan akses *user* aplikasi pada PT. MDS

Pada gambar 1.1 merupakan proses permintaan dalam melakukan *request* aplikasi yang terjadi di Matahari tidak efektif, dikarenakan proses yang memakan

1

RANCANG BANGUN APLIKASI WEB UNTUK PENGELOLAAN HAK AKSES USER  
BERDASARKAN ROLE PADA PT.MDS MENGGUNAKAN MODEL WATERFALL, Maria  
Claudia Samur, Universitas Multimedia Nusantara

2



waktu. Karyawan harus meminta *approval* dari kepala bagian agar dapat melakukan permintaan akses yang nantinya akan dibuatkan oleh tim IT.

Selain proses permintaan akses *user* aplikasi, perusahaan Matahari mengalami kesulitan untuk memisahkan *user active* dan *inactive* dalam mengakses aplikasi – aplikasi yang ada di Matahari. Dalam memisahkan *user* tersebut, perusahaan Matahari harus melakukan konsolidasi data *user* secara manual untuk memisahkan antara *user active* dan *inactive*. Setelah didapatkan hasilnya, barulah ditemukan *user inactive* yang masih mendapatkan akses ke aplikasi yang digunakan oleh Matahari. Seharusnya, *user inactive* sudah tidak bisa mengakses aplikasi sebagai proses bisnis yang sedang berlangsung di Matahari dengan alasan *user inactive* adalah *user* yang sudah tidak lagi bekerja di Matahari. Contohnya seperti *user resign*, *user pensiun*, dan contoh lain sejenisnya. Data *user inactive* inilah yang nantinya akan di kelola untuk dihapuskan aksesnya ke aplikasi yang digunakan.

### Gambar 1. 2 Alur Proses Penghapusan Akses *User* Aplikasi

Sumber: Alur proses penghapusan akses *user* aplikasi pada PT MDS

Pada gambar 1.2 merupakan proses penghapusan akses *user*, pihak IT yang menghapus akan melakukan konfirmasi terlebih dahulu kepada pihak kepala bagian *user* yang akan dihapus akses ke aplikasi tersebut. IT akan mendapatkan list *user* yang sudah resign dari pihak HRD, kemudian pihak IT akan melakukan pengecekan akses aplikasi terhadap *user* tersebut. Pada saat pihak IT melakukan pengecekan memakan waktu, dikarenakan proses konfirmasi ini dilakukan *via email*. Hal tersebut membuat efisiensi waktu yang dimiliki tidak bisa optimal. Jika *user* yang mengalami perpanjangan kontrak kerja atau mengalami pembaharuan *user*, mendapatkan *user ID* yang berbeda dengan yang lama. Namun, saat ini pada saat *user* yang mengalami update *user ID* ini masih menggunakan *user ID* yang lama untuk mengakses aplikasi tersebut. Selain dalam *termination user*, dalam penambahan *user* untuk mengakses aplikasi yang digunakan, *user* harus mengisi

sebuah *form* yang akan di tandatangani oleh pihak terkait seperti kepala bagian dari *user* tersebut.

Berdasarkan latar belakang yang telah dipaparkan penelitian mengembangkan sebuah web untuk mempermudah proses pemisahan *user active* dan *inactive* serta penghapusan akses aplikasi *user inactive*. Dengan proses persetujuan yang memudahkan pihak terkait, hal ini diharapkan dapat mengoptimalkan waktu dengan waktu. Melalui aplikasi web ini, diharapkan dalam pengelolaan *user* dalam akses aplikasi dapat terpantau dengan mudah. Saat melakukan *request user* akan ada lampiran *approval* sehingga akan terjaga keamanannya.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan, ditemukan rumusan masalah sebagai berikut pada penelitian ini:

1. Bagaimana merancang aplikasi web untuk pengelolaan hak akses pengguna berdasarkan role di PT. MDS?
2. Bagaimana sistem memberikan kemudahan dalam pengajuan dan penghapusan pembuatan akses *user* aplikasi?
3. Bagaimana mengimplementasikan model *waterfall* dalam proses pengembangan aplikasi web?

## 1.3 Batasan Masalah

Batasan masalah yang ditemukan pada penelitian mengenai pengelolaan hak akses *user* adalah sebagai berikut

1. Fokus pada permasalahan pengelolaan *user* di PT Matahari Department Store.
2. Tantangan dalam pengelolaan akses aplikasi bagi *user* yang masih memiliki akses untuk aplikasi yang digunakan oleh Matahari sebagai proses bisnis, namun seharusnya *user* tersebut sudah tidak dapat mengakses aplikasi tersebut dikarenakan alasan yang telah disepakati.

3. Fokus pada masalah waktu yang diperlukan dalam proses konfirmasi penghapusan akses user yang dilakukan oleh pihak IT kepada kepala bagian user melalui *email*.

## 1.4 Tujuan dan Manfaat Penelitian

### 1.4.1 Tujuan Penelitian

Berdasarkan rumusan permasalahan yang telah ditemukan, maka tujuan penelitian ini adalah sebagai berikut:

1. Merancang aplikasi web untuk pengelolaan hak akses pengguna berdasarkan role di PT. MDS membutuhkan pemahaman tentang kebutuhan pengguna dan hak akses yang diperlukan setiap role. Proses desain mencakup analisis kebutuhan, desain arsitektur, pengembangan fitur utama, dan pengujian untuk memastikan aplikasi berjalan dengan baik dan aman. Dengan aplikasi web ini dapat mengelola hak akses pengguna secara efektif, mengurangi risiko kesalahan dalam melakukan konsolidasi data, dan meningkatkan keamanan sistem informasi perusahaan.
2. Aplikasi yang dikembangkan tidak hanya memfasilitasi pengelolaan akses, tetapi juga secara otomatis menghapus akses aplikasi untuk *user* yang sudah tidak bekerja di perusahaan. Langkah ini bertujuan untuk mengurangi risiko akses serta mempercepat proses pengelolaan *user*, sekaligus meminimalkan kesalahan dalam data yang bisa saja terjadi jika dilakukan secara manual.
3. Model *waterfall* diterapkan dalam pengembangan aplikasi web untuk pengelolaan hak akses pengguna berdasarkan role di PT. MDS dengan mengikuti tahapan yang terstruktur dan berurutan: analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Dengan model *waterfall*, setiap langkah akan dilakukan secara tuntas dan hasilnya dievaluasi sebelum melanjutkan ke langkah berikutnya.

Pendekatan ini memungkinkan pengembangan aplikasi yang terorganisir dengan baik, memastikan bahwa aplikasi berfungsi dengan baik, aman, dan mudah dipelihara.

#### 1.4.2 Manfaat Penelitian

Berikut adalah manfaat dari penelitian mengenai sistem pengelolaan hak akses *user* sebagai berikut:

1. Mengotomatisasi pengelolaan hak akses pengguna berdasarkan *role*, mengurangi waktu dan kesalahan dalam proses manual.
2. Web ini memberikan kontrol akses yang mudah sehingga struktur dalam pengelolaan hak akses *user* dapat terorganisir.
3. Dengan pendekatan model *waterfall*, aplikasi web ini mudah dipelihara dan dikembangkan sesuai kebutuhan yang berkembang.

### 1.5 Sistematika Penulisan

#### BAB I PENDAHULUAN

Bab ini menjelaskan alasan dari permasalahan pada penelitian ini termasuk dengan solusi yang diberikan pada penelitian ini. Berdasarkan dari pemaparan dari bab ini, disimpulkan dengan Rumusan Masalah, Batasan Masalah, Tujuan dan Manfaat dari penelitian tersebut.

#### BAB II LANDASAN TEORI

Pada bab ini, Landasan Teori menjelaskan mengenai teori - teori yang ada pada penelitian sehingga pembaca dapat memahami dengan baik isi dari penelitian ini.

### BAB III METODE PENELITIAN

Pada bab ini, peneliti memaparkan objek yang dijadikan sebagai penelitian. Kemudian, pada bab ini terdapat tahapan – tahapan dalam mengumpulkan data yang dibutuhkan.

### BAB IV ANALISA DAN HASIL

Pada bab ini menguraikan mengenai spesifikasi program dan pengelolaan data dari hasil pengujian sistem mulai dari tampilan program sampai pengujian terhadap sistem aplikasi.

### BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran untuk mendukung perbaikan dalam sistem pengelolaan hak akses *user* secara *realtime*.



56

## BAB II

### LANDASAN TEORI

#### 2.1 Peneliti Terkait

Pada tabel 2.1 merupakan penelitian yang berkaitan dengan *monitoring* dengan menggunakan sebuah metode berbasis web:

Pada tabel 2.1 adalah penelitian yang terkait untuk menjadi landasan dalam pengembangan penelitian yang sedang berlangsung. Penelitian terkait ini secara keseluruhan mengembangkan berbagai sistem yang berhasil memberikan solusi dalam keefektifan dalam setiap masalah yang sedang diteliti.

Hasil dari penelitian terkait dengan penelitian yang sedang dikembangkan menunjukkan bahwa seluruh sistem berfungsi sesuai dengan tujuan yang telah ditetapkan. Melalui model pengembangan yang tepat, seperti *waterfall* dan pengujian menggunakan *Black Box*, serta desain sistem yang baik dengan UML, sistem yang dikembangkan dapat meningkatkan kinerja dan mempermudah pengelolaan data.

#### 2.2 Tinjauan Teori

##### 2.2.1 Pengelolaan

Pengelolaan akses adalah proses untuk mengatur, memantau, dan mengendalikan akses *user* terhadap sumber daya, baik yang berada di sistem maupun di lokasi fisik[4]. Tujuan utamanya adalah memastikan bahwa hanya *user* yang memiliki otorisasi yang dapat mengakses sumber daya yang diperlukan, serta membatasi akses bagi *user* yang tidak berwenang.

##### 2.2.2 Termination

*Termination* atau pemutusan hubungan kerja dapat terjadi karena berbagai alasan, termasuk kinerja yang buruk, pelanggaran kebijakan perusahaan, atau pengurangan karyawan[5]. Dalam beberapa kasus, seorang

3

53

karyawan dapat dipecat tanpa alasan yang jelas, yang berarti perusahaan tidak memiliki alasan spesifik untuk mengakhiri hubungan kerja tersebut.

### 2.2.3 Database

*Database* adalah tempat penyimpanan berupa kumpulan tabel yang tersusun secara sistematis, sehingga bisa diakses melalui program komputer untuk mengelola data tersebut[6]. *Database* merupakan kumpulan informasi yang disimpan secara terorganisir di satu tempat, sehingga dapat diakses, dikelola, dianalisis, dan diperbarui dengan mudah[7].

#### a. MySQL

Dalam dunia pemrograman, SQL adalah bahasa yang digunakan untuk mengambil data dari relational database atau database terstruktur. Sederhananya, MySQL adalah sistem pengelolaan database yang memanfaatkan SQL sebagai bahasa penghubung antara aplikasi perangkat lunak dan server database[7][8].

Fungsi dari MySQL adalah untuk melakukan pengolahan terhadap informasi pada database dari sisi server dengan Bahasa pemrograman SQL[9]. Adapun kelebihan dari MySQL adalah sebagai berikut:

1. Dengan performa yang sangat baik, perangkat lunak ini merupakan sistem manajemen database yang sifatnya gratis dan dapat digunakan oleh siapapun.
2. MySQL memiliki keamanan sistem yang cukup baik, dengan memiliki basis host maupun enkripsi kata sandi membuat data yang ada di MySQL dapat dijamin keamanannya.
3. MySQL tidak membutuhkan RAM dengan ukuran yang besar atau dapat digunakan dengan spesifikasi hardware yang rendah. Walaupun tidak membutuhkan RAM yang besar, MySQL mampu menjalankan sistem dengan baik.



48

### Tabel 2. 1 Tipe Data MySQL

Pada tabel 2.1 adalah tabel tipe data pada MySQL dengan deskripsi berdasarkan tipe data masing – masing beserta dengan contohnya.

#### 2.2.4 Web

Tim Berners-Lee, menjelaskan bahwa sebuah web adalah "sekumpulan halaman yang saling terhubung secara logis dan dapat diakses melalui satu URL (*Uniform Resource Locator*)[10]." Menurutnya, web merupakan bagian dari struktur aplikasi yang lebih besar, di mana setiap halaman dapat diakses baik secara individu maupun sebagai bagian dari keseluruhan[10].

59

Fungsi dari web adalah sebagai berikut:

1. Memudahkan *user* dalam melakukan komunikasi.
2. *User* dapat menjangkau informasi secara *real time* atau informasi terbaru dapat langsung diakses oleh *user*.
3. Adapun hiburan yang diberikan seperti situs *game online*, *film*, music dan lainnya. *User* dapat menikmati berbagai hiburan melalui sebuah web.
4. Web dapat dijadikan sumber mata pencarian karena web dapat menjadi saran bisnis jual beli barang dan jasa. Kemudahan yang diberikan oleh web menjadi salah satu alasan utama saat ini banyak orang menggunakan web sebagai transaksi jual beli barang maupun jasa.

24

#### 2.2.5 Bahasa Pemrograman

##### a. *Hypertext Processor* (PHP)

*Hypertext Preprocessor* (PHP) merupakan bahasa pemrograman yang sering digunakan dalam pembangunan aplikasi web interaktif dan

2



dinamis[11]. PHP juga mendukung berbagai macam database seperti MySQL, PostgreSQL, dan Oracle, sehingga memungkinkan pembuatan aplikasi web yang lebih kompleks dan kaya fitur[9].

Kelebihan dari Bahasa pemrograman PHP adalah sebagai berikut[12]:

1. Dapat digunakan secara gratis dan sumber terbuka dan dapat diakses secara bebas. Hal ini memudahkan *user* dalam pengembangan aplikasi web tanpa bayaran yang mahal.
2. Mudah dipelajari khususnya bagi pemula.
3. Kemudahan dalam integrasi dengan teknologi web lainnya seperti HTML, CSS, dan JavaScript. PHP juga bisa diintegrasikan dengan berbagai *database* seperti MySQL dan Oracle.
4. Beberapa fitur pada PHP dapat membantu melindungi keamanan web dari kebocoran data. Salah satu contoh fiturnya adalah *magic quotes*.

## 2.2.6 Bahasa Markup

### a. *Hyper Text Markup Language (HTML)*

HTML bukan bahasa pemrograman, melainkan bahasa markup. HTML hanya memberikan struktur pada konten, sedangkan bahasa pemrograman digunakan untuk membuat logika dan interaksi dalam sebuah halaman web[13].

Kelebihan dari Bahasa pemrograman HTML adalah sebagai berikut:

1. Memiliki banyak sumber daya dan komunitas yang sangat besar.
2. HTML memiliki sifat *opensource* yang berarti HTML dapat diakses secara gratis.
3. Memiliki bahasa markup yang terstruktur dan konsisten.
4. Dapat dijalankan di semua browser.
5. Mudah dipelajari oleh pengembang web pemula

### 2.2.7 Skenario

Skenario tes dapat membantu dalam melakukan pengujian terhadap aplikasi yang sedang dikembangkan. Dengan adanya skenario, proses pengujian menjadi lebih efisien dan meminimalkan kemungkinan langkah-langkah yang terlewat. Skenario dibuat sebagai acuan dalam melakukan uji coba sebuah sistem yang telah dikembangkan[14]. Dalam hal ini, kegiatan testing dapat berjalan dengan efektif untuk menghindari adanya langkah – langkah yang terlewatkan dalam proses pengujian sistem, sehingga sistem dapat langsung diketahui bahwa sistem siap digunakan

### 2.2.8 *Retek Merchandising System (RMS)*

Perangkat lunak ini digunakan untuk mengelola ritel, khususnya oleh perusahaan ritel besar. Awalnya dikembangkan oleh Retek, yang kemudian diakuisisi oleh Oracle, RMS mendukung retailer dalam menangani berbagai aspek operasional ritel, seperti manajemen inventori, pembelian barang, penetapan harga, serta pelaporan performa penjualan[15].

Gambar 2. 1 Tampilan Aplikasi RMS

Sumber: [16]

Pada gambar 2.1 adalah contoh halaman RMS yang digunakan untuk mengelola stok, penetapan harga, promosi, serta tampilan produk baik di toko fisik maupun *platform e-commerce*.

### 2.2.9 *Oracle Financial Application (OFA)*

*Oracle Financial Application* merupakan kumpulan perangkat lunak yang menjadi bagian dari *Oracle E-Business Suite (EBS)*, dirancang untuk membantu perusahaan dalam mengelola berbagai aspek keuangan. Perangkat lunak ini menyediakan modul-modul yang berfungsi untuk mengotomatisasi, mengelola, serta memantau transaksi keuangan,

menghasilkan laporan keuangan, dan memastikan kepatuhan terhadap aturan akuntansi serta perpajakan.

#### Gambar 2. 2 Tampilan Aplikasi OFA

Sumber:[17]

Pada gambar 2.2 adalah contoh tampilan halaman OFA yang menyediakan gambaran umum dari status proses operasional, termasuk efisiensi dan kendala yang ada.

#### **2.2.10 Permintaan Penambahan Karyawan (PPK)**

Aplikasi Permintaan Penambahan Karyawan (PPK) adalah sebuah sistem yang dipakai oleh perusahaan untuk mengatur proses pengajuan penambahan tenaga kerja. Aplikasi ini umumnya digunakan oleh manajer atau departemen yang memerlukan tambahan staf untuk mengirimkan permintaan kepada divisi HRD.

#### Gambar 2. 3 Tampilan Aplikasi PPK

Sumber: Aplikasi PPK MDS

Pada gambar 2.3 adalah contoh tampilan PPK yang membantu untuk mengelola dan memantau anggaran, pengadaan barang dan jasa, serta proses keuangan terkait.

#### **2.2.11 Retail Store Inventory Management (RSIM)**

*Retail Store Inventory Management (RSIM)* adalah sistem atau aplikasi yang dikembangkan untuk membantu perusahaan ritel dalam mengelola inventori atau stok barang yang ada di toko. Sistem ini bertujuan untuk memastikan bahwa produk yang dijual selalu tersedia dalam jumlah yang memadai, mengurangi kemungkinan kekurangan atau kelebihan stok, serta meningkatkan efisiensi operasional[18].

## Gambar 2. 4 Tampilan Aplikasi RSIM

Sumber: Aplikasi RSIM MDS

Pada gambar 2.4 adalah contoh halaman utama menampilkan menu untuk mencari informasi penting tentang status toko, stok barang, penjualan, dan performa secara keseluruhan.

### 2.2.12 Use Case Diagram

*Use case diagram* adalah salah satu jenis diagram dalam UML (*Unified Modeling Language*) yang menunjukkan hubungan antara sistem dan aktor. *Use case* diagram tujuannya adalah memberikan fungsi yang disediakan oleh sistem, serta bagaimana aktor berinteraksi dengan fungsi tersebut untuk mencapai tujuan tertentu[19]. *Use case* dapat membantu penelitian untuk memahami pemodelan yang diperlukan dalam menggambarkan aksi aktor dan sistem secara jelas, seperti yang dilakukan oleh *use case* diagram.

### Tabel 2. 2 Simbol Use Case Diagram

Pada tabel 2.3 adalah komponen-komponen dalam diagram *use case* meliputi: batas sistem yang menunjukkan lingkup sistem dan mencakup bagian serta fungsi-fungsi sistem; aktor yang merupakan *user* yang berinteraksi dengan sistem dan dapat berupa individu, organisasi, atau sistem lain yang mempengaruhi fungsionalitas sistem; serta *use case* yang menggambarkan berbagai fungsionalitas dalam sistem secara visual.

*Use case* diagram dalam langkah awal melakukan pemodelan, diperlukan sebuah diagram yang dapat menggambarkan tindakan aktor dan tindakan sistem, seperti yang diilustrasikan dalam *use case* diagram. Selain itu, *use case* diagram relatif mudah dipahami.

### 2.2.13 Activity Diagram

*Activity diagram* digunakan untuk mewakili langkah – langkah yang berlangsung dalam sistem. Urutan proses dalam sistem tersebut

dilambangkan secara vertikal. Diagram aktivitas ini merupakan pengembangan dari *Use Case* yang menampilkan alur aktivitas.

Tabel 2. 3 Simbol Use Case Diagram

Pada tabel 2.4 adalah symbol *activity* diagram pada teknik yang digunakan untuk memvisualisasikan aktivitas-aktivitas dalam suatu proses bisnis yang dirancang dalam sistem[19]. Diagram ini dibuat berdasarkan satu atau beberapa use case yang terdapat pada use case diagram.

#### 2.2.14 Class Diagram

Class diagram dapat diimplementasikan dalam proyek yang menggunakan konsep object-oriented karena representasi class diagram cukup mudah dimengerti dan digunakan. Selain itu, class diagram dapat menggambarkan kelas, metode, atribut, dan hubungan antar objek[19]. Class diagram bersifat statis, artinya hanya dapat menjelaskan hubungan yang ada, tanpa menjelaskan dinamika yang terjadi ketika kelas-kelas tersebut berinteraksi.

Tabel 2. 4 Simbol Class Diagram

Tabel 2.5 simbol *class* diagram yang menggambarkan struktur kelas dalam sebuah sistem manajemen perpustakaan. Dalam diagram ini, terdapat beberapa kelas yang saling berhubungan untuk menggambarkan interaksi antar objek dalam sistem.

### 2.3 Teori tentang Framework/Algoritma yang digunakan

Kerangka kerja yang digunakan pada pengembangan sistem pengelolaan hak akses *user* dengan menggunakan model *waterfall*. *Framework* pada penelitian ini memberikan dukungan terhadap pengembangan sistem sehingga penelitian ini dapat terstruktur. Dalam penggunaan *framework* ini, dapat membantu penelitian dalam mempermudah secara sistematis.

2 Fungsi *framework* dalam penelitian ini adalah sebagai berikut :

1. Mempercepat proses pengembangan sistem pengelolaan hak akses *user*.
2. Menjaga ketetapan pada program yang telah dirancang.
3. Meningkatkan kualitas sistem pengelolaan hak akses *user*.

## 1 2.4 Teori tentang tools/software yang digunakan

### 2.4.1 Software Pengembangan Sistem

Dalam membangun pengembangan sistem pengelolaan hak akses *user* dibutuhkan *software* untuk mendukung pengembangan sistem web. Sebelum menentukan *software* yang akan digunakan, penelitian melakukan perbandingan antara *tools* yang akan digunakan.

Berikut adalah tabel perbandingan *tools* dalam pengembangan sistem monitoring pengelolaan akses user aplikasi[20]:

Table 2. 5 Tabel Perbandingan Tools

Pada tabel 2.4 adalah perbandingan *tools* yang digunakan dalam membuat aplikasi web. Penelitian ini menggunakan *tools* sublime text 3 dalam melakukan pengembangan sistem. Pemilihan *tools* pada penelitian ini karena *tools* yang digunakan akan menentukan produktivitas dalam pengembangan sistem pengelolaan hak akses *user*.

### 2.4.2 Server

Pengembangan aplikasi web diperlukan server untuk mengelola *database* yang akan digunakan pada pengembangan sistem sesuai dengan kebutuhan tanpa memerlukan jaringan internet pada sistem.

Pada pengembangan sistem ini server yang digunakan untuk menghubungkan antara *tools* dan *database* adalah XAMPP. XAMPP dapat mendukung berbagai macam sistem yang berfungsi sebagai server lokal seperti MySQL, PHP, dan Apache.



### BAB III

## METODOLOGI PENELITIAN

### 3.1 Gambaran Umum Objek Penelitian

#### 3.1.1 Sejarah Singkat Objek Penelitian

PT Matahari berdiri pada tahun 1958 yang dimulai sebagai sebuah toko biasa yang didirikan oleh Bapak Hari Dermawan. Awalnya toko Matahari berdiri dengan luas tanah 159 meter persegi di Pasar Baru, Jakarta Pusat dengan menjual pakaian impor. Nama perusahaan Matahari awalnya adalah PT Matahari Putra Prima. Kemudian, PT Matahari Putra Prima mulai mengembangkan jaringan dengan membuka toko serba ada, yang tersebar di Jakarta, Bandung, dan Bogor dengan total 11 toko. Delapan tahun setelah itu, jumlah toko Matahari bertambah menjadi 33 toko yang tersebar di 16 kota di seluruh Indonesia[21]. Kini Matahari tidak hanya menyediakan barang impor, tetapi juga menjual kebutuhan lainnya seperti pakaian, aksesoris, dan perlengkapan kecantikan.

Perubahan dalam upaya mengembangkan bisnisnya PT Matahari Putra Prima akhirnya berganti nama menjadi PT Matahari Department Store. Perubahan ini dilakukan untuk mentransformasikan bisnisnya menjadi penyedia barang-barang berkualitas bagi kalangan kelas menengah Indonesia. Matahari menjalin kemitraan dengan peritel asing dimulai pada tahun 1994. Kerja sama ini mendapatkan hasil yang baik, sehingga Matahari berhasil mempekerjakan sekitar 27.000 karyawan dengan total penjualan mencapai 2 miliar. Pada Januari 1997, Matahari tercatat memiliki 84 gerai yang tersebar di 30 kota di seluruh Indonesia[21].

Tujuan pendirian PT Matahari Department Store Tbk. adalah untuk mengembangkan usaha department store dan memanfaatkan peluang pasar yang lebih baik. Dengan menjadi perusahaan publik, PT Matahari dapat mengakses pasar modal, yang memungkinkan perusahaan untuk memberikan informasi yang lebih transparan dan jelas mengenai kinerjanya kepada

pemegang saham. Hal ini juga mempermudah para investor untuk berpartisipasi dalam perusahaan. Saat ini, PT Matahari telah memiliki 156 toko yang tersebar di seluruh Indonesia dan menjadi salah satu jaringan department store terkemuka di tanah air.. PT Matahari Department Store berhasil mencatat laba bersih tertinggi sepanjang sejarah yaitu sejumlah Rp652 Miliar pada tahun 2010[22].

Gambar 3. 1 Logo Perusahaan Tempat Penelitian

Sumber: [23]

Pada gambar 3.1 adalah logo dari MDS yang dijadikan tempat objek penelitian.

### 3.1.2 Struktur Organisasi

Gambar 3. 2 Struktur Organisasi

Sumber: PT. MDS

Pada gambar 3.1 adalah struktur organisasi di MDS dengan focus organisasi pada tempat objek melaksanakan penelitian. Penelitian ini berfokus pada divisi IT, sehingga organisasi yang dipaparkan adalah struktur organisasi IT.

Tabel 3. 1 Uraian Tugas Berdasarkan Jabatan

Pada tabel 3.1 adalah deskripsi dari setiap jabatan yang berada di organisasi MDS khususnya pada divisi IT.

### 3.1.3 Visi Misi Objek Penelitian

Visi dari objek penelitian adalah menjadi retailer dengan gaya hidup omni-channel terkemuka sebagai "*House of Specialists*" yang berpusat pada pelanggan. Misi dari objek penelitian adalah untuk memuaskan customer dengan



*fashion* yang aspiratif dengan kualitas yang bagus dan nyaman namun dengan harga yang terjangkau.

### 3.1.4 Filosofi Objek Penelitian

Filosofi Matahari tujuan utamanya adalah menciptakan tempat kerja yang nyaman, aman, dan lebih baik bagi karyawannya. Matahari memiliki organisasi yang jelas dengan dasar efektivitas dan efesiensi kerja yang maksimal. Matahari adalah salah satu perusahaan yang tidak memandang suku, agama, ras, dan golongan terhadap seluruh karyawannya. Matahari juga menjunjung tinggi *teamwork* yang baik dengan menciptakan kekeluargaan dalam *team* kerja untuk mencapai tujuan.

## 3.2 Kerangka Berpikir

### 3.2.1 Model Pengembangan

Berdasarkan pemaparan latar belakang, penelitian ini menggunakan model *waterfal* dalam menyelesaikan tugas akhir. Dalam pembangunan sistem perangkat lunak secara menyeluruh, terdapat berbagai tahapan yang perlu dilalui. Model *waterfall* merupakan model pengembangan perangkat lunak yang mendasar, berkat sifatnya yang bersifat alami dan terstruktur. Sebagai pendekatan pertama dalam SDLC, model *waterfall* mengikuti urutan yang sangat sistematis, dimulai dari tahap perencanaan, analisis, desain, hingga implementasi.

Gambar 3. 3 Model Pengembangan Waterfall

Sumber: [24]

Pada gambar 3.3 adalah proses dalam model ini dilakukan secara bertahap dan berurutan, mulai dari identifikasi kebutuhan sistem, dilanjutkan dengan analisis, desain, pengkodean, pengujian, hingga pemeliharaan. Setiap tahapan harus diselesaikan secara tuntas sebelum melanjutkan ke tahap berikutnya, tanpa adanya kemungkinan untuk melompat ke tahap yang lebih

lanjut[25]. Karena alur kerjanya yang berurutan seperti air yang mengalir, model ini dinamakan *waterfall*.

Pada tahap ini, pengembang mengidentifikasi semua kebutuhan perangkat lunak dan sistem yang akan dikembangkan.

1. *Requirements Analysis:*

Analisis kebutuhan merupakan tahapan pertama dalam model *waterfall*.

Pada tahap ini, dilakukan proses pengumpulan data yang sangat penting untuk merancang sistem yang akan dibangun. Proses pengumpulan data dilakukan melalui dua pendekatan utama, yaitu wawancara dan observasi lapangan.

2. *Design:*

Tujuan utama dari desain sistem ini adalah untuk memberikan gambaran yang jelas dan terstruktur tentang bagaimana sistem akan bekerja, sehingga mempermudah proses implementasi dan penulisan kode pada tahap selanjutnya. Dengan adanya desain yang rinci, pengembang dapat lebih mudah memahami alur dan fungsi dari sistem yang akan dibangun, yang pada gilirannya akan meningkatkan efisiensi dan mengurangi kemungkinan kesalahan dalam tahap implementasi.

3. *Development:*

Sistem ini dirancang untuk memastikan bahwa manajemen hak akses *user* dapat dilakukan dengan efektif dan efisien, memberikan kontrol yang ketat terhadap siapa saja yang dapat mengakses berbagai sumber daya dalam jaringan perusahaan.

4. *Testing:*

Setelah membuat program sistem mengenai web *monitoring* pengelolaan penggunaan multi aplikasi, dilakukan pengujian terhadap sistem agar sesuai

dengan fungsionalitas. Pada tahap ini pengujian sistem dilakukan secara keseluruhan untuk mengidentifikasi adanya kegagalan terhadap sistem yang sudah dibuat.

#### 5. *Maintenance*:

Tahapan terakhir pada model *waterfall* adalah evaluasi dan pemeliharaan. Tahap pemeliharaan melibatkan *user* yang terkait dalam pengelolaan web sesuai dengan kebutuhan. Apabila ditemukan ketidaksesuaian maka akan dilakukan perbaikan terhadap sistem agar dapat sesuai.

### 3.2.2 Perbandingan Model Pengembangan Sistem

Model yang digunakan dalam penelitian adalah model *waterfall*. Setelah melakukan pertimbangan terhadap model penelitian yang akan digunakan, terdapat perbandingan model penelitian dalam pembuatan aplikasi web untuk pengolahan akses *user* aplikasi.

Tabel 3. 2 Perbandingan Model Pengembangan Sistem

Pada tabel 3.2 setelah melakukan perbandingan terhadap dua model pengembangan sistem yang digunakan dalam perancangan dan pengembangan pengolahan akses *user* aplikasi untuk PT. Matahari Department Store Tbk. dengan model *waterfall*. Hal ini dikarenakan sistem yang dibangun pada model *waterfall* sesuai dengan kebutuhan dari penggunaannya. *User* dapat memantau perkembangan setiap tahap dalam perancangan sistem tersebut.

Pada sistem yang dirancang ini, model *waterfall* dilakukan dengan tingkat keahlian yang tidak terlalu tinggi, namun dapat menghasilkan sistem yang fleksibilitas. Landasan dalam memilih model *waterfall* untuk melakukan perancangan sistem pengolahan akses *user* aplikasi web dengan hasil yang sesuai dengan kebutuhan.

### 3.2.3 Design Prototype

Dalam perancangan pembuatan *design prototype*, penelitian ini menggunakan *tools* Figma. Figma adalah aplikasi yang digunakan untuk merancang sebuah desain UI/UX untuk mempermudah proses pembuatan sebuah sistem[26]. Dengan adanya proses desain, hal ini dapat menjadi lebih efisien dan efektif. Selain itu, Figma dapat diakses dimana saja dengan menggunakan *desktop* atau laptop yang terhubung dengan internet.

#### 1. Login

##### Gambar 3. 4 Tampilan Halaman Login

Gambar 3.4 merupakan tampilan terawal dari aplikasi yang dirancang. *User* akan melakukan *login* terlebih dahulu sebelum mengakses aplikasi. Dengan memasukkan *User ID* dan *password* yang sesuai, maka *user* dapat mengakses aplikasi tersebut.

#### 2. Tampilan Dashboard – Role Superadmin

##### Gambar 3. 5 Halaman Dashboard sebagai Role Superadmin

Gambar 3.5 adalah halaman setelah *user* Superadmin berhasil melakukan *login*. *User* akan melihat halaman *dashboard* dengan tampilan *pie chart* seperti pada gambar. Hal ini berfungsi untuk memudahkan *user* dalam melihat *summary* dari karyawan yang mengakses aplikasi dan karyawan perusahaan secara keseluruhan. Pengambilan *summary* ini berlangsung selama 1 periode atau 1 bulan.

### 3. Menu *User MDS – Role Superadmin*

#### Gambar 3. 6 Halaman Menu User MDS sebagai *Role Superadmin*

Pada gambar 3.6 terdapat 7 tabel informasi yang dapat digunakan oleh *user* untuk melihat setiap karyawan Matahari dengan status setiap karyawan. Tabel pertama menunjukkan nomor atau jumlah row yang berisikan informasi mengenai karyawan tersebut. Tabel *User ID* adalah *barcode* yang digunakan oleh Matahari untuk setiap karyawan terdapat 2 kriteria *user id* yaitu X dan M. X menandakan bahwa *user* adalah pihak eksternal Matahari seperti *intern*, *outsourcing*, dan lainnya, sementara *barcode* M adalah *user* karyawan Matahari. Tabel *user name* adalah nama dari *user id* tersebut. Tabel status adalah tabel untuk mengetahui status dari *user* tersebut. Terdapat 2 kriteria pada status yaitu *active* dan *inactive*. Status *active* adalah karyawan yang masih merupakan karyawan perusahaan, sementara *inactive* adalah karyawan yang sudah tidak lagi bekerja di perusahaan. Tabel *start date* adalah tabel yang memberitahukan informasi waktu karyawan mulai bekerja di Matahari, sementara tabel *end date* adalah waktu karyawan berhenti atau tidak lagi bekerja di perusahaan.

### 4. Menu Aplikasi – *Role Superadmin*

#### Gambar 3. 7 Halaman Menu Aplikasi sebagai *Role Super admin*

Gambar 3.7 adalah gambar menu pada setiap aplikasi. Terdapat 4 aplikasi yaitu PPK, OFA, RSIM, dan RMS. Pada tampilan setiap aplikasi terdapat 7 tabel yang memperlihatkan informasi terkait dengan karyawan yang dapat mengakses aplikasi – aplikasi tersebut.

Pada menu aplikasi, superadmin dapat melakukan *download report* dengan format pdf dan excel. Hasil *report* dapat diunduh kapan saja, namun periode

pada sistem berlangsung dalam 1 bulan, sehingga jika sudah berganti bulan maka daftar karyawan akan berganti pada tampilan.

#### 5. Menu *Role* – *Role* Superadmin

Gambar 3. 8 Halaman Menu *Role*

Gambar 3.8 adalah menu *Role* yang hanya dapat diakses oleh *role* super admin. Pada tabel *role* terdapat 3 *role* yang digunakan yaitu superadmin, admin HR, dan *Division*. Super admin didedikasikan dengan tim IT sebagai pembuatan *user* akses aplikasi yang akan di *request* dan super admin adalah *role* yang dapat mengakses menu *role* atau pembuatan *role* sesuai dengan kebutuhan.

Tabel 3. 3 Deskripsi *Role*

#### 6. *Create Role*

Gambar 3. 9 Halaman *Create Role*

Pada gambar 3.9 adalah formulir ketika *role* superadmin melakukan pembuatan *role*. Pembagian *role* akan disesuaikan dengan divisi dan jabatan yang ada di Matahari.

Tabel 3. 4 Divisi dan Jabatan *Role*

#### 7. Menu MDS – *Role Admin HR*

Gambar 3. 10 Halaman menu MDS sebagai *Role Admin HR*

Pada gambar 3.10 menampilkan bahwa menu MDS dengan menggunakan *role* admin HR dapat menambahkan karyawan pada aplikasi. Pada menu MDS, jika menggunakan *role* admin HR *user* dapat melakukan *download report* dengan format pdf dan excel.

#### 8. *Button Create Menu MDS – Role Admin HR*

##### Gambar 3. 11 Halaman *Create Kayawan*

Pada gambar 3.11 menampilkan formulir yang harus diisi oleh *user* dengan *role* Admin HR untuk menambahkan karyawan perusahaan pada aplikasi. Pada *field attachment* merupakan surat kontrak atau surat perjanjian prakerja yang sudah disepakati oleh kedua belah pihak antara perusahaan dan karyawan.

#### 9. *Menu MDS – Role Division*

##### Gambar 3. 12 Halaman Menu MDS sebagai Role Division

Pada gambar 3.12 menampilkan halaman menu MDS dengan *role* Division hanya dapat melakukan *download report*.

#### 10. *Menu Aplikasi – Role Division*

##### Gambar 3. 13 Halaman Menu Aplikasi sebagai *Role Division*

Pada gambar 3.13 menampilkan halaman menu aplikasi yang dapat diakses oleh *role Division*. *Role Division* dapat melakukan *request user* untuk mendapatkan akses ke setiap aplikasi. Dalam pengambilan *report*, *role* admin HR dapat *download report* dengan format pdf dan excel.

#### 11. *Halaman Create User Aplikasi*

##### Gambar 3. 14 Halaman Request User Aplikasi

Pada gambar 3.14 menampilkan formulir yang harus diisi oleh *role* Division jika melakukan *request user* untuk mendapatkan akses.



## 12. Halaman Notifikasi Berhasil *Request User*

### Gambar 3. 15 Halaman Notifikasi Request User Aplikasi

Pada gambar 3.15 adalah tampilan notifikasi jika *role Division* telah berhasil di *request*.

### 1.3 Variabel Penelitian

Variabel penelitian menggambarkan kejadian, kategori, jenis, atribut, atau karakteristik tertentu yang mencerminkan penggunaan setiap kebutuhan *user* dalam penelitian. Variabel memiliki nilai yang berbeda-beda, yang mana nilai-nilai tersebut ditetapkan berdasarkan penggambaran atau deskripsi objek yang diteliti. Variabel berfungsi untuk mengukur atau menggambarkan fenomena yang menjadi fokus dalam penelitian.

Variabel bebas atau independen adalah variabel yang mempengaruhi atau menjadi sebab munculnya perubahan pada variabel terikat atau dependen. Dalam konteks penelitian ini, variabel bebas (variabel X) adalah faktor yang diperkirakan dapat memberikan pengaruh atau menyebabkan perubahan pada variabel dependen (variabel Y):

X1 = Data Karyawan

X2 = Data Aplikasi

X3 = Data *User Login*

Y1 = Sistem Monitoring Pengelolaan Pengguna Multi Aplikasi

### 1.4 Teknik Analisis Data

#### 1.4.1 Pengumpulan Data

##### 1. Wawancara

Teknik wawancara adalah teknik tahapan awal pada penelitian yang digunakan dalam pengumpulan data. Teknik ini dilakukan dalam penelitian dengan melakukan wawancara terhadap pihak terkait.

Pada penelitian ini wawancara yang dilakukan sebanyak tiga kali dengan pihak yang terkait PT. Matahari Department Store Tbk. yaitu Ibu Rekha Bella sebagai *Business Analyst*. Hasil yang ditemukan dari



wawancara ini adalah kesulitan dalam memisahkan *user* yang sudah *termination*. Hal ini dikarenakan *user* yang tidak tercatat setiap periodenya.

Dalam penelitian ini contoh kasus yang diambil adalah saat karyawan yang ditugaskan melakukan *track user* akses telah pensiun. Namun, dikarenakan satu dan lain hal pengganti untuk karyawan pensiun ini belum ada hingga awal Januari, sehingga terjadi beberapa miss komunikasi pada *track user* ini.

## 2. Observasi

Dalam sebuah penelitian, metode observasi dijelaskan sebagai teknik yang digunakan untuk mengamati dan menggambarkan perilaku subjek. Sesuai dengan namanya, observasi merupakan cara untuk mengumpulkan informasi dan data yang relevan melalui pengamatan. Oleh karena itu, observasi disebut juga sebagai studi partisipatif, karena peneliti harus membangun hubungan dengan responden dan terlibat dalam lingkungan yang sama.

## 3. Studi Pustaka

Teknik dalam pengumpulan data juga melalui buku, jurnal, laporan atau referensi dari hasil penelitian sebelumnya yang masih berhubungan dengan masalah yang diteliti.

Penelitian ini dilakukan dengan mempelajari informasi terhadap perancangan sebuah aplikais web melalui berbagai sumber jurnal sebagai refensi dalam penelitian ini. Sumber – sumber yang dicari dalam penelitian ini berfokus pada kegiatan *monitoring* dan perancangan aplikais web melalui internet. Setelah mendapatkan berbagai referensi, penelitian ini mengumpulkan informasi yang digunakan sebagai landasan teori, metodologi penelitian, dan perancangan sistem.

10

## 2. Periode Pengumpulan Data

Dalam penelitian ini data yang dikumpulkan untuk menyelesaikan tugas akhir ini selama 3 bulan yaitu dari bulan Juni 2024 hingga Agustus 2024.

17

### 3.5 Teknik Analisis Data

Setelah seluruh data terkumpul, langkah berikutnya adalah menganalisis data tersebut untuk menarik kesimpulan. Dalam mendapatkan hasil yang akurat, penelitian ini menggunakan teknik analisis kritis. Analisis kritis merupakan metode yang digunakan untuk menganalisis dari sebuah masalah melalui sebuah diskusi secara lisan ataupun tertulis. Analisis kritis digunakan sebagai landasan penelitian berdasarkan informasi yang telah diberikan oleh sumber melalui wawancara. Analisis kritis biasanya didasarkan pada pandangan atau nilai-nilai tertentu yang diyakini pada.

Oleh karena itu, sudut pandang terhadap suatu masalah sangat memengaruhi cara penafsiran terhadap data. Kaitan dengan pembahasan yaitu sebagai salah satu upaya untuk memudahkan penelitian dalam menganalisis masalah yang terjadi pada *user* akses aplikasi.



## BAB IV

### ANALISIS DAN HASIL PENELITIAN

#### 4.1 Analisa Masalah

##### 4.1.1 Analisis Proses Bisnis

Sebelum melakukan penelitian lebih lanjut, tahap awal yang dilakukan adalah wawancara terhadap salah satu karyawan PT. Matahari Department Store Tbk. untuk mencari informasi yang akurat terkait masalah yang terjadi. Wawancara ini dilakukan dengan Bu Rekha Bella Novia yang salah satu tugasnya adalah mempersiapkan dokumen untuk *audit*.

Salah satu dokumen yang harus terpenuhi adalah dokumen terkait dengan *user* Matahari yang mengakses aplikasi. Kesulitan dalam pemisahan data *user* tersebut menjadikan gambaran untuk alur dari pembuatan sistem *monitoring* pengelolaan akses *user* aplikasi pada PT. Matahari Department Store Tbk. Informasi yang didapatkan dari sumber kemudian dikelola dan ditindaklanjuti untuk mendapatkan solusi yang lebih baik melalui penelitian ini.

##### 4.1.2 User Requirement

Pada hasil wawancara seperti terlampir, dibutuhkan sistem yang dapat melakukan *monitoring* user akses aplikasi. Dengan adanya web ini diharapkan dapat membantu *monitoring user* akses aplikasi dengan melakukan *request* akses dan untuk memantau *user termination* tersebut.

Berdasarkan kebutuhan sistem tersebut, didapatkan kebutuhan fungsional sebagai berikut:

1. Superadmin
  - Membuat dan mengelola *role user* yang akan digunakan oleh *user* sesuai dengan kebutuhan.
  - Membuat akses *user* aplikasi yang telah diajukan oleh Division dan membuat *approval* terhadap aplikasi web untuk menandakan bahwa *user* sudah dapat mengakses aplikasi yang telah diajukan
  - Mengunduh *report* data *user employee* dan *user* aplikasi pada setiap periode.
2. Admin HR
  - Melakukan *create* data *user employee*
  - Melakukan penyimpanan dokumen kontrak kerja terhadap *user employee* dengan memasukkan dokumen pada aplikasi web saat melakukan pembuatan *user employee*
  - Mengunduh *report* data *user employee* pada setiap periode
3. Division
  - Melakukan pengejauan akses *user* aplikasi kepada superadmin
  - Mengunduh *report* data *user* aplikasi pada setiap periode

## 4.2 Perancangan Sistem

Dalam pengembangan aplikasi web, adapun alur kerja yang telah dibuat untuk merancang aplikais web. Alur kerja ini menggunakan *Unified Modelling Language* (UML) dengan *use case diagram* dan *activity diagram* sehingga sistem dapat berinteraksi dengan manusia.

### 4.2.1 Use Case Diagram

*Use Case* Diagram pada pengembangan aplikasi web ini menunjukkan hierarki dan pembagian tugas dalam pengelolaan akses pengguna, di mana Superadmin memiliki kendali penuh, Admin HR mengelola data karyawan, dan Division mengajukan permintaan akses

#### Gambar 4. 1 Use Case Diagram Aplikasi Web Untuk Pengelolaan Hak Akses User

Gambar 4.1 menggambarkan struktur pengelolaan hak akses pengguna dalam suatu sistem, yang dibedakan berdasarkan peran masing-masing. Terdapat tiga aktor utama, yaitu Superadmin, Admin HR, dan Division. Masing-masing peran memiliki hak akses dan tanggung jawab yang berbeda.

1. Superadmin memiliki hak akses paling luas, yaitu dapat membuat dan mengelola *role user*, serta membuat akses pengguna terhadap aplikasi. Superadmin juga dapat mengunduh laporan data pengguna secara menyeluruh.
2. Admin HR memiliki tanggung jawab untuk membuat data *user* (employee), menginput dokumen kontrak kerja, dan mengunduh laporan data pengguna secara periode.
3. Division memiliki peran untuk membuat permintaan akses pengguna terhadap aplikasi dan mengunduh laporan data pengguna secara periode.

#### 4.2.2 Activity Diagram

*Activity diagram* yang digunakan untuk memodelkan langkah - langkah yang terjadi dalam sebuah sistem. Langkah – langkah tersebut termasuk dalam sistem yang digambarkan pada pengembangan sistem pengelolaan hak akses *user*. *Activity diagram* merupakan pengembangan dari *use case* setiap aktivitas yang dilakukan oleh setiap *role* pada sistem.

Alur atau aktivitas ini bisa berupa rangkaian menu atau proses bisnis yang ada dalam sistem tersebut.

Gambar 4. 2 Activity Diagram Aplikasi Web Untuk Pengelolaan Hak Akses User  
Gambar 4.2 menggambarkan proses pengelolaan data *user* dan hak akses *user* aplikasi dalam aplikasi web. Prosesnya dimulai dari Superadmin yang membuat user baru dengan role Admin HR dan Division. Selanjutnya, Admin HR akan membuat NIK (Nomor Induk Karyawan) untuk karyawan baru, mengisi data

karyawan, dan menentukan tanggal akhir masa kerja. Division kemudian mengajukan *request* akses untuk *user* ke aplikasi yang dibutuhkan. Kemudian, sistem akan melakukan pengecekan terhadap NIK yang dimasukkan. Jika NIK belum terdaftar maka proses pengajuan akses *user* aplikasi tidak dapat dilanjutkan. Jika NIK sudah terdaftar, sistem akan melanjutkan ke proses pengajuan akses *user* di aplikasi yang diajukan. Apabila karyawan tersebut telah melewati tanggal *termination*, sistem akan menghapus data *user* dan karyawan secara otomatis pada aplikais web.

#### 1. Activity Diagram Membuat *Role*

Gambar 4. 3 Activity Diagram Membuat *Role*

Pada gambar 4.3 merupakan rangkaian aktivitas dari super admin untuk membuat role dari sistem *monitoring* pengelolaan penggunaan multi aplikasi. Dalam aktivitas ini, super admin melakukan penambahan dan *edit role*. *Edit role* tersebut dengan kondisi *active* dan *inactive*. Saat *role* dibuat, maka status *role* tersebut adalah *inactive*. Kemudian, *super admin* akan *edit* menjadi *active*, sehingga *role* dapat melakukan kegiatan sesuai dengan akses yang telah diberikan. Hal ini untuk menghindari penyalahgunaan dalam wewenang yang telah diberikan aksesnya.

#### 2. Activity Diagram Super Admin Approval Request User Aplikasi

Gambar 4. 4 Activity Diagram Super Admin Approval Request User Aplikasi

Pada gambar 4.4 adalah kegiatan yang dilakukan oleh *super admin* dalam melakukan *approval* terhadap *request user* aplikasi yang diajukan oleh *Division*. *Super admin* dapat melakukan *approval* terhadap *user* yang diajukan oleh pihak *Division* untuk mengakses aplikasi yang akan diakses. *Super admin* akan mengecek pada menu aplikasi dengan status “*In Progress*”. Jika kondisi status *user* aplikasi “*In Progress*” maka *super admin*

akan melakukan proses akses di aplikasi yang diajukan terhadap *user* tersebut. Setelah akses *user* aplikasi telah dibuat, super admin akan melakukan *approve*. Kemudian, sistem akan menggantikan status *in progress* menjadi *done*. Status *done* ini menandakan bahwa *user* yang telah diajukan telah dibuatkan akses sehingga *user* dapat langsung mengakses aplikasi tersebut.

### 3. *Activiy Diagram Super Admin dan Division Mengambil Report Data User Aplikasi*

Gambar 4. 5 *Activiy Diagram Super Admin dan Division Mengunduh Report Data User Aplikasi*

Pada gambar 4.5 merupakan rangkaian dari kegiatan super admin dan division yang melakukan report data terhadap *user* aplikasi. *Super admin* dan division dapat melakukan report data tersebut dalam setiap periode. Sistem telah dirancang untuk dapat melakukan rangkuman data aplikasi pada setiap bulan, sehingga *report* dapat mengikutinya.

### 4. *Activity Diagram Division Request User Aplikasi*

Gambar 4. 6 *Activity Diagram Division Request User Aplikasi*

Pada gambar 4.6 adalah kegiatan Division dalam mengajukan *user* yang akan mengakses aplikasi. Division dapat melakukan *request user* tersebut pada menu aplikasi yang akan diakses oleh *user*. Setelah melakukan *request user* status pada web adalah “*In Progress*” tunggu hingga status berubah menjadi “*Done*” artinya akses *user* telah dibuatkan terhadap *user* yang telah diajukan.



## 5. Activity Diagram Admin HR *Create User* MDS

Gambar 4. 7 Activity Diagram Admin HR *Create User* MDS

Pada gambar 4.7 merupakan kegiatan dalam melakukan *create user* yang dimasukkan oleh admin HR ke dalam sistem. Admin HR akan memilih menu MDS untuk melakukan *create user* MDS, kemudian sistem akan menyimpan data dari *user* yang telah dibuat oleh admin HR.

## 6. Activity Diagram Admin HR Unduh Data Periode *User* MDS

Gambar 4. 8 Activity Diagram Admin HR Mengunduh Report Data Karyawan Setiap Periode

Pada gambar 4.8 merupakan rangkaian dari kegiatan admin HR yang melakukan report data terhadap *user* aplikasi. Admin HR dapat melakukan report data tersebut dalam setiap periode. Admin HR dapat melihat hasil *report* yang telah diunduh dengan format excel dan pdf. Sistem telah dirancang untuk dapat melakukan rangkuman data aplikasi pada setiap bulan, sehingga *report* dapat mengikutinya. Admin HR juga dapat melihat hasil *report* dalam dalam 1 bulan periode dari *chart* yang ada pada menu *Home*.

### 4.2.3 Class Diagram

*Class* diagram menggambarkan struktur data yang kompleks untuk mengelola informasi *user*, aplikasi yang digunakan *user*, dan *role* dalam aplikasi web. Diagram ini dapat digunakan sebagai dasar untuk membangun database yang mendukung berbagai macam fitur, seperti manajemen akses, pelaporan, dan analisis data *user*.



Gambar 4. 9 Class Diagram Aplikasi Web Untuk Pengelolaan Hak Akses User Aplikasi

Pada gambar 4.9 menggambarkan hubungan antar kelas dalam aplikasi web pengelolaan hak akses *user*. Kelas utama yang ada adalah *user*, *application*, *role*, *division\_role*, *hr\_role*, dan *superadmin\_role*. Hubungan antar kelas membentuk sebuah basis data yang digunakan untuk mengelola informasi pengguna, aplikasi yang digunakan, dan *role*.

1. Kelas *User*: menyimpan data *user* seperti ID *user*, nama, email, tanggal mulai dan akhir masa kerja, jabatan, dan ID kontrak. Kelas ini juga memiliki hubungan dengan kelas *role* yang menunjukkan peran *user* dalam sistem.
2. Kelas *Application*: menyimpan informasi tentang aplikasi yang ada pada aplikasi web, seperti ID aplikasi, nama aplikasi, tanggal terminasi aplikasi, dan memiliki hubungan dengan kelas *user* yang menggunakan aplikasi tersebut.
3. Kelas *Role*: mendefinisikan *role* yang ada dalam aplikasi web, seperti divisi, HR, dan superadmin. Setiap *role* memiliki tanggung jawab yang berbeda.
4. Kelas *division\_role*, *hr\_role*, dan *superadmin\_role*: merupakan turunan dari kelas *role* dan mendefinisikan secara spesifik tindakan yang dapat dilakukan oleh masing-masing *role*. Misalnya, *division\_role* memiliki tindakan untuk meminta akses aplikasi, sementara *superadmin\_role* memiliki tindakan untuk membuat akses *user* aplikasi yang telah diajukan.
5. Kelas *contract\_detail*: menyimpan *detail* kontrak karyawan, seperti ukuran file kontrak, ekstensi file, dan memiliki hubungan dengan kelas *user*.

#### 4.2.4 Kamus Data

Dalam sistem web yang dikembangkan, semua data dicatat dan disimpan dalam database. Sistem informasi ini memanfaatkan database yang dibuat dengan MySQL melalui phpMyAdmin, sehingga dapat terintegrasi dengan bahasa pemrograman PHP.

Berikut adalah tabel-tabel yang menyimpan data yang digunakan untuk aplikasi web pengelolaan hak akses *user* skripsi ini:

1. Tabel user

Tabel 4. 1 Tabel User MDS

2. Tabel application

Tabel 4. 2 Tabel application

3. Tabel app\_detail

Tabel 4. 3 Tabel app\_detail

4. Tabel user\_detail

Tabel 4. 4 Tabel user\_detail

5. Tabel contrac\_detail

Tabel 4. 5 Tabel contrac\_detail

6. Tabel role

Tabel 4. 6 Tabel role

7. Tabel division\_role

Tabel 4. 7 Tabel division\_role

8. Tabel superadmin\_role

Tabel 4. 8 Tabel superadmin\_role

9. Table hr\_role

Tabel 4. 9 Tabel hr\_role

### 4.3 Testing dan Implementasi Sistem

1. Halaman Login

Gambar 4. 10 Halaman Login

Pada gambar 4.10 adalah halaman *login* adalah halaman awal dari aplikasi web pengelolaan hak akses user aplikasi. Seluruh *user* yang akan mengakses aplikasi web ini harus melakukan *login* terlebih dahulu.

2. Halaman Dashboard

Gambar 4. 11 Halaman Dashboard

Pada gambar 4.11 merupakan halaman utama pada aplikasi web. Saat *user* sudah berhasil melakukan *login*, maka sistem akan menampilkan halaman *dashboard* tersebut. Halaman *dashboard* ini merupakan *report* setiap bulan pada *user* MDS dan *user* aplikasi.

3. Halamn Menu User MDS

#### Gambar 4. 12 Halaman Menu MDS sebagai Role Admin HR

Pada gambar 4.12 adalah menu *User MDS* merupakan menu yang menampilkan seluruh daftar karyawan MDS dengan memiliki 3 status yaitu: *new*, *rehire*, dan *inactive*. *New* adalah karyawan baru dalam satu periode. *Rehire* adalah karyawan yang sudah lebih dalam satu periode. *Inactive* adalah karyawan yang *resign* atau sudah tidak bekerja lagi di Matahari.

*User MDS* yang sudah *resign* akan diberikan *flag* sebagai penanda dalam satu periode. Setelah satu periode maka *user* tersebut tidak akan muncul pada daftar *user MDS*.

#### 4. Halaman *Create User MDS*

##### Gambar 4. 13 Halaman Menu Create User MDS

Pada gambar 4.13 adalah halaman *create user MDS* hanya *user* dengan *role* admin HR yang dapat mengakses halaman tersebut. *User* akan memasukkan nama, *user id*, *job title*, *email*, *employee status*, *termination date*, dan *upload file*. Pada saat *role* admin HR memilih *employee status* terdapat 2 pilihan *employee* yaitu *new* dan *rehire*.

##### Gambar 4. 14 Pilih Job Title

Pada gambar 4.14 merupakan posisi atau jabatan *user* yang akan disimpan datanya pada web pengelolaan penggunaan multi aplikasi pada PT. Matahari Department Store Tbk.

##### Gambar 4. 15 Pilih Status *Employee*

Pada gambar 4.15 adalah pemilihan status *employee* dari karyawan tersebut. Status *New* adalah karyawan yang baru satu bulan bekerja di Matahari, sementara status *Rehire* adalah status karyawan Matahari yang sudah habis masa kontraknya namun kontrak diperpanjang. Hal tersebut berfungsi untuk pencatatan *report* yang akan dikelola datanya pada setiap bulan.

#### Gambar 4. 16 *Upload Attachment*

Gambar 4.16 adalah *filed upload file* yang merupakan lampiran berisikan kontrak kerja yang telah disepakati antara karyawan dengan pihak Matahari.

#### 5. Halaman *View Detail*

##### Gambar 4. 1 Tampilan *View Detail* sebagai *Role* Admin HR

Pada gambar 4.17 adalah tampilan dari *view detail* dengan *role* admin HR. *Role* admin HR dapat mengakses dokumen yang diunggah dengan tujuan untuk memastikan *file* yang diunggah sudah sesuai.

#### 6. Hasil *Report User* MDS – Format pdf dan excel

##### Gambar 4. 17 Hasil *Report* Format PDF dan Excel Menu MDS

Pada gambar 4.17 adalah hasil report dengan format pdf dan excel yang ada pada tampilan menu aplikasi. Pada kolom status terdapat status *active* dan *inactive* yang diartikan bahwa status *active* adalah status karyawan yang masih bekerja di Matahari dan untuk status *inactive* adalah karyawan yang sudah tidak bekerja. Pengambilan *report* akan terunduh dengan format yang secara otomatis akan memperlihatkan waktu *user* melakukan pengunduhan.

#### 7. Hasil *Report User* Aplikasi – Format excel

##### Gambar 4. 18 Hasil *Report* Format excel Menu Aplikasi sebagai *role* Division

Pada gambar 4.18 adalah hasil *report* dengan format excel pada menu App yang dilakukan oleh Division.

#### 8. Tampilan *User App* Pada *Role* Admin HR

##### Gambar 4. 19 Tampilan Menu App sebagai *Role* Admin HR

Pada gambar 4.19 adalah tampilan menu aplikasi sebagai *role* admin HR. *Role* admin HR hanya dapat melakukan view terhadap menu aplikasi tanpa melakukan *action* apapun pada menu tersebut.

#### 9. Menu *User* MDS Pada *Role* Division

##### Gambar 4. 20 Tampilan Menu *User* MDS sebagai *Role* Division

Pada gambar 4.20 adalah tampilan halaman menu *user* MDS pada *role* Division. Jika *user* melakukan *login* sebagai Division maka tampilan pada halaman menu *user* MDS hanya bisa *view* atau *role* Division tidak dapat melakukan *action* apapun pada menu tersebut.

#### 10. Menu *User* App Pada *Role* Division

##### Gambar 4. 21 Tampilan Menu *User* Aplikasi sebagai *Role* Division

Pada gambar 4.21 adalah menu App merupakan menu yang menampilkan daftar *user* yang menggunakan aplikasi yang ada di Matahari. MDS memiliki banyak aplikasi untuk membantu proses bisnis agar dapat berjalan dengan baik sesuai dengan tujuan.

Pada menu *User* App terdapat name sebagai nama dari user yang menggunakan aplikasi tersebut, *user id* sebagai *barcode* yang digunakan oleh Matahari untuk setiap karyawannya, *job title* merupakan jabatan dari *user* tersebut, *create date* adalah waktu *user* dengan *role* Division melakukan *request* atas *user* tersebut, *termination date* adalah waktu berakhirnya *user* dapat mengakses aplikasi tersebut, status *user* adalah status yang telah tersistematis dengan waktu *termination*, status *request* adalah status yang telah diberikan *action* atau *approve* dari pihak *super admin*. Status *request* berfungsi sebagai tanda bahwa *user* dapat mengakses aplikasi tersebut. Jika status *request* adalah *in progress* maka *user* yang telah diajukan belum dapat mengakses aplikasi tersebut. Namun, jika status *request* adalah *done*, maka *user* sudah dapat mengakses aplikasi tersebut.

## 11. Create User App

### Gambar 4. 22 Tampilan Create User Aplikasi

Pada gambar 4.22 adalah halaman *create user* App hanya *user* dengan *role* Division yang dapat mengakses halaman tersebut. *User* akan memasukkan *user id* kemudian data akan tampil sesuai dengan *user id*. Seperti *name*, *job title*, *email* dan *create date*. *Termination date* adalah waktunya yang akan diajukan oleh *user* dengan *role* Division.

### Gambar 4. 23 Tampilan Pada Formulir Create User Aplikasi

Gambar 4.23 adalah contoh seperti pada gambar, *user* memasukkan *user id* dari karyawan MDS maka sistem akan menampilkan info terkait dengan *user id* tersebut.

### Gambar 4. 24 Tampilan Pemilihan Tanggal Termination

Selanjutnya pada gambar 4.24 *user* akan memasukkan waktu *termination* dari *user* yang akan mengakses aplikasi tersebut. Kemudian *user* akan melakukan *request* atas *user* tersebut. *User* akan menunggu konfirmasi penggunaan akses aplikasi melalui *email*. *Email* akan diterima oleh *user* ketika *role* superadmin sudah melakukan *approve* terhadap *user* yang diajukan.

*User* yang berhasil ditambahkan akan ditampilkan oleh sistem pada halaman menu app. Jika *user* memiliki *termination* sebelum tanggal *create*, maka status *user* akan *inactive*. Namun, jika *user* memiliki *termination* sesudah tanggal *create* maka status *user* yang ditampilkan oleh sistem adalah *active*.



## 12. Halaman Menu MDS Pada *Role* Super Admin

### Gambar 4. 25 Tampilan Menu MDS sebagai *Role* Super Admin

Pada gambar 4.25 merupakan tampilan dari halaman menu MDS dengan *login* sebagai super admin. Super admin hanya dapat *view* pada menu *User* MDS. *Role* super admin dapat mengunduh daftar *user* MDS dengan mengunduh dengan format excel maupun pdf.

## 13. Halaman Menu App Pada *Role* Super Admin

### Gambar 4. 26 Tampilan Menu Aplikasi sebagai *Role* Super Admin

Pada gambar 4.26 adalah super admin dapat melihat halaman menu app dan melakukan *approval* terhadap *user* yang diajukan dengan status *request* yaitu '*in progress*'. Jika status *request* adalah *done*, maka super admin tidak lagi melakukan *action* apapun.

## 14. Halaman Menu *Role*

### Gambar 4. 27 Tampilan Menu *Role*

Gambar 4.27 adalah tampilan pada menu *role*. Hanya super admin yang mengakses menu *role* untuk menghindari terjadinya kesalahan yang tidak diinginkan jika seluruh *role* dapat mengakses menu tersebut. Jika status pada *user* adalah *inactive*, maka *user* tersebut tidak dapat *login* web *monitoring* pengelolaan penggunaan multi aplikasi ini. Jika status adalah *active*, maka *user* dengan *role* yang telah diberikan dapat melakukan *login* sesuai dengan *role* masing – masing.



## 15. Create Role

### Gambar 4. 28 Tampilan *Create Role*

Pada gambar 4.28 adalah halaman super admin yang dapat menambah *role* terhadap *user* sesuai dengan kebutuhan. Super admin akan memasukan *user id*, kemudia sistem akan menampilkan keterangan sesuai dengan *field* yang ada.

### Gambar 4. 29 Tampilan Formulir *Create Role*

Pada gambar 4.29 adalah tampilan formulir untuk melakukan *create role* yang akan dilakukan oleh *role* super admin. Hal ini untuk memastikan bahwa *user* yang akan mendapatkan akses sebagai *role Division* ataupun admin HR adalah karyawan Matahari. Setelah *user id* dimasukkan dan *user* sesuai, maka *role* akan tampil.

### Gambar 4. 30 Tampilan Pilihan Role

Pada gambar 4.30 merupakan *role* yang dapat disesuaikan dengan kebutuhan *user*. Terdapat 3 *role* yang menjadi landasan untuk *user* menggunakannya sesuai dengan kebutuhan. *Role* super admin adalah *user* yang akan melakukan *action* terhadap *request user* aplikasi. Pemilihan terhadap *role* super admin adalah divisi IT yang bertugas untuk membuat akses *user* aplikasi. *Role Division role* dengan *user* yang melakukan *request user* aplikasi, *user* dengan *role* ini adalah semua divisi dengan jabatan kepala bagian setiap *department*. *Role* admin HR adalah *user* yang melakukan *create* seluruh karyawan Matahari. *User* dengan *role* tersebut adalah divisi HR, yang bertugas untuk merkrut karyawan.

#### 4.4 Hasil dan Diskusi

Setelah aplikasi web untuk pengelolaan hak akses user berdasarkan role selesai dikembangkan, perlu dilakukan pengujian. Dalam penelitian ini, pengujian dilakukan menggunakan metode *black box testing*, yang melibatkan dua orang: tim IT melakukan *action* dari *request* yang dilakukan oleh *all division* untuk mengajukan karyawan – karyawan yang akan mengakses aplikasi sesuai dengan kebutuhan. Data karyawan yang diajukan harus merupakan karyawan Matahari yang sudah di *create* oleh tim HR untuk mendaftarkan sebagai karyawan Matahari.

Berikut adalah tabel yang berisi hasil *User Acceptance Testing* (UAT) yang dilakukan dengan pendekatan *black box testing*:

Tabel 4. 10 Tabel Testing UAT Aplikasi Web Untuk Pengelolaan Hak Akses User

Setelah *user* melakukan UAT terhadap sistem yang telah dikembangkan terkait dengan pengelolaan hak akses *user* aplikasi, *user* akan mengisi kuisioner yang berisikan kepuasan *user* terhadap sistem. Terdapat 21 responden terhadap kepuasan *user* terhadap aplikasi yang telah dikembangkan.

Berdasarkan hasil evaluasi pengujian *user* UAT pada aplikasi pengelolaan hak akses, rata – rata *user* merasa puas terhadap aplikasi pengelolaan data *user*. Mulai dari kelayakan aplikasi, responsitiv, menu, dan manfaat dari aplikasi itu sendiri. Hasil yang didapatkan melalui survei tersebut adalah *user* merasa puas, sehingga aplikasi ini dapat digolongkan sebagai aplikasi layak pakai.

3

## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Simpulan

Simpulan terhadap permasalahan penelitian ini adalah memberikan solusi dengan mengembangkan aplikasi web untuk mengelola hak akses pengguna sesuai dengan *role* setiap *user* di PT. MDS. Setiap *role*, seperti super admin, Admin HR, Division, diberikan hak akses yang sesuai dengan fungsinya. Hal ini memungkinkan pengelolaan akses menjadi lebih terstruktur, dan aman sehingga setiap *user* hanya dapat mengakses data yang relevan dengan *role* mereka.

Aplikasi web yang dibangun mempermudah proses pengajuan dan penghapusan hak akses bagi *user*. *User* dapat mengajukan permintaan untuk mendapatkan akses sesuai kebutuhannya, yang kemudian dapat disetujui oleh super admin. Selain itu, hak akses untuk *user* yang sudah tidak aktif atau tidak lagi bekerja dapat dihapus secara otomatis, mengurangi risiko dalam penyalahgunaan akses aplikasi.

Model *waterfall* yang diterapkan dalam pengembangan aplikasi web melalui beberapa tahapan yang terstruktur, yakni analisis kebutuhan, desain sistem, pengembangan, pengujian, dan pemeliharaan. Setiap tahapan dilakukan secara berurutan untuk memastikan aplikasi yang dikembangkan memenuhi semua persyaratan yang telah ditentukan dan dapat diimplementasikan secara efektif.

Berdasarkan hasil evaluasi pengujian *user* UAT pada aplikasi pengelolaan hak akses, rata – rata *user* merasa puas terhadap aplikasi pengelolaan data *user*. Mulai dari kelayakan aplikasi, responsitiv, menu, dan manfaat dari aplikasi itu sendiri. Hasil yang didapatkan melalui survei tersebut adalah *user* merasa puas, sehingga aplikasi ini dapat digolongkan sebagai aplikasi layak pakai.

6

## 5.2 Saran

Berdasarkan kesimpulan diatas, peneliti memiliki beberapa saran yang dapat dijadikan dasar dalam mengembangkan lebih lanjut aplikasi ini, diantaranya adalah:

1. Desain arsitektur sistem agar mudah untuk diperbarui atau diperluas di masa yang akan datang. Misalnya, menggunakan desain berbasis *microservices* atau modularitas dalam pengelolaan hak akses, sehingga apabila ada perubahan di masa yang akan datang, sistem tetap fleksibel tanpa mengganggu keseluruhan aplikasi.
2. Implementasikan teknologi yang dapat mendukung pengelolaan hak akses secara real-time, seperti penggunaan WebSocket atau teknologi lain yang memungkinkan update status akses *user*.

