

## **SKRIPSI**

### **RANCANG BANGUN *E-LEARNING* DAN REPOSITORY DIGITAL DENGAN PLATFORM GITHUB**

**(Studi Kasus: Centratama Telekomunikasi Indonesia)**



**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA**

**2023 M / 1444 H**

## HALAMAN JUDUL

# RANCANG BANGUN *E-LEARNING* DAN REPOSITORY DIGITAL DENGAN PLATFORM GITHUB

(Studi Kasus: Centratama Telekomunikasi Indonesia)

Sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta



Disusun Oleh:

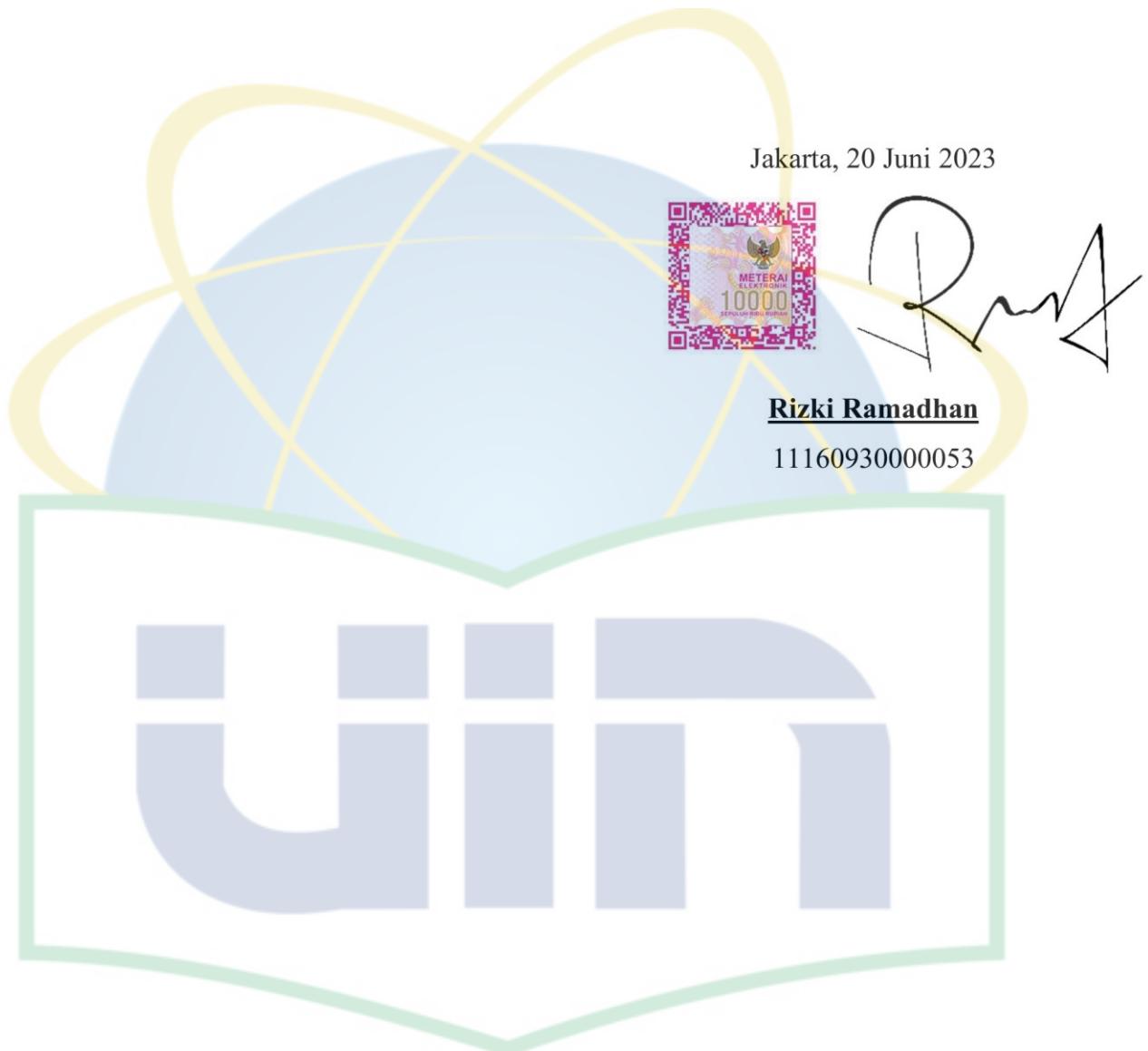
**RIZKI RAMADHAN**

11160930000053

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
2023 M / 1444 H**

## LEMBAR PERNYATAAN

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SENDIRI YANG BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI MANAPUN.



## ABSTRAK

**Rizki Ramadhan – 11160930000053**, Rancang Bangun *E-Learning* dan Repository Digital dengan Platform Github (Studi Kasus: Centratama Telekomunikasi Indonesia) di bawah bimbingan **Yuni Sugiarti** dan **Elvi Fetrina**.

Transformasi Digital di era endemi sudah tidak lagi menjadi sebuah opsi, melainkan sudah menjadi suatu keharusan agar perusahaan bisa terus berkembang. Centratama Teknologi Indonesia (CTI) merupakan salah satu perusahaan yang ingin terus mengembangkan perusahaannya. Mereka telah mentransformasikan beberapa proses kerja di perusahaannya ke berbasis digital. Di divisi HC, proses *training* karyawan menjadi salah satu proses kerja yang belum memiliki solusi digital. Kendala dengan proses *training* yang ada saat ini yaitu sulitnya membuat laporan hasil *training* karena data dan informasi terkait proses *training* tersebar di beberapa tempat penyimpanan, sulitnya menyiapkan *training* secara luring, dan tidak semua karyawan dapat mengikuti *training* secara daring. Selain itu, satu solusi digital yang dibuat oleh CTI hanya mencakup satu proses kerja. Akibatnya, pengelolaan solusi digital yang telah dibuat menjadi sulit untuk dilakukan. Tujuan penelitian ini adalah mengembangkan modul aplikasi *e-learning* pada HC Portal dan membuat repositori digital untuk membantu mewujudkan pengembangan HC Portal yang berkelanjutan dengan menggunakan platform Github. Metode pengembangan *software* yang digunakan pada penelitian ini adalah *iterative model*. Hasil penelitian ini tidak hanya memberikan solusi atas permasalahan proses *training* karyawan saja, melainkan juga memberikan sebuah lingkungan pengembangan yang dapat membantu proses pengembangan *software* yang berkelanjutan sehingga CTI bisa mengembangkan solusi digital yang dapat mencakup beberapa proses kerja.

**Kata Kunci:** Aplikasi Web (*Web Application*), *E-learning*, Github, *Iteration Model*, Repository Digital.

Bab I – V + 332 halaman + xxiv halaman + 171 Gambar + 93 tabel + Daftar Pustaka + Lampiran

Pustaka Acuan (100, 2012 – 2022)

## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

*Alhamdulillahirabbil'alamin*, puji syukur kepada Allah SWT yang telah memberikan Rahmat, Taufik, dan Hidayah-Nya, penulis pada akhirnya dapat menyelesaikan Skripsi yang berjudul “Rancang Bangun E-Learning dan Repositori Digital dengan Platform Github (Studi Kasus: Centratama Telekomunikasi Indonesia)”. Tak lupa Selawat dan Salam selalu tercurah kepada junjungan kita, Nabi Muhammad SAW beserta keluarga, sahabat, kerabat, serta muslimin dan muslimat. Semoga kita semua tetap istikamah dan selalu mendapatkan syafaat dalam menjalankan ajarannya hingga akhir zaman. Amin.

Dalam penyusunan Skripsi ini, penulis banyak mendapatkan bantuan, saran, dorongan, bimbingan, dan kritik yang membangun dari berbagai pihak. Ini semua adalah hal-hal yang tidak dapat diukur dengan materi. Oleh karena itu, penulis dengan segala hormat dan kerendahan hati ingin mengucapkan terima kasih kepada:

1. Bapak Husni Teja Sukmana S.T., M.Sc, Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.
2. Ibu Dr. Qurrotul Aini, M.T. selaku ketua Program Studi Sistem Informasi Fakultas Sains dan Teknologi periode saat ini, dan Bapak A'ang Subiyakto, M.Kom.,Ph.D selaku ketua Program Studi Sistem Informasi Fakultas Sains dan Teknologi periode sebelumnya.
3. Ibu Yuni Sugiarti, M.Kom. selaku Dosen Pembimbing I yang telah memberikan saran, bimbingan, ilmu, dan motivasi dalam menyelesaikan Skripsi ini.
4. Ibu Elvi Fetrina, B.Sc., M.IT selaku Dosen Pembimbing II yang juga telah memberikan saran, bimbingan, ilmu, dan motivasi dalam menyelesaikan Skripsi ini.
5. Dosen-dosen program studi Sistem Informasi Universitas Islam Negeri Syarif Hidayatullah Jakarta yang telah memberikan ilmu selama perkuliahan.
6. Kedua orang tua peneliti, Bapak Tinton dan Ibu Jumini yang tetap memberikan dukungan, kasih sayang, doa, dan semangat di masa-masa sulit penulis ini. Serta adik-adik dan kakak penulis yang selalu memberikan dua

- sisi koin kehidupan kepada penulis sehingga penulis bisa belajar untuk mengembangkan diri penulis menjadi lebih baik lagi.
7. Mas Yudhi selaku Pembimbing Lapangan, Mba Meutia, Mas Gandhi, dan seluruh karyawan lainnya di Divisi HC Centratama Telekomunikasi Indonesia yang telah memberikan ilmu dan pengalaman bekerja di dunia bisnis, serta masukan dan arahan saat magang.
  8. Teman-teman yang telah mau mendengarkan dan meladeni pikiran ruwet penulis, serta memberikan masukan kepada penulis untuk terus semangat dalam menyelesaikan perkuliahan ini, yaitu Akbar Riski, Taufan Yogi Adam, Bayu Wicaksono, Hanif Aulia Fikri, Muhammad Aulia Rahman, Nasirudin Sabiq, Rivaldi Zidan, Ridho Dhafi Fauzan, serta seorang teman spesial bagi penulis yaitu Elvy Farhati.
  9. Seluruh teman-teman JOGI yang selalu memberikan semangat dan hiburan bagi penulis.
  10. Keluarga besar Sistem Informasi angkatan 2016 dan kelas “B aja” yang menjadi motivasi penulis dalam menyelesaikan Skripsi ini.
  11. Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis dalam menyelesaikan Skripsi ini.

Semoga Allah SWT senantiasa melimpahkan rahmat dan hidayah-Nya, serta ganjaran yang setimpal atas kebaikan yang telah diberikan kepada penulis. Amin.

Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak sekali kekurangan yang disebabkan segala keterbatasan penulis. Untuk itu, penulis mengharapkan kritik dan saran yang membangun agar penulis bisa membuat karya tulis yang lebih baik lagi. Kritik dan saran ini bisa dikirimkan ke alamat surel pribadi penulis melalui [rizki\\_rama@hotmail.com](mailto:rizki_rama@hotmail.com).

Akhir kata, penulis berharap penelitian ini dapat bermanfaat bagi para pembaca, dan terima kasih telah membaca secercah kata pengantar ini.

Jakarta, 12 Juni 2023

**Rizki Ramadhan**

11160930000053

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>I</b>
<b>LEMBAR PENGESAHAN SKRIPSI .....</b>	<b>II</b>
<b>LEMBAR PENGESAHAN UJIAN .....</b>	<b>III</b>
<b>LEMBAR PERNYATAAN .....</b>	<b>IV</b>
<b>ABSTRAK .....</b>	<b>V</b>
<b>KATA PENGANTAR.....</b>	<b>VI</b>
<b>DAFTAR ISI.....</b>	<b>VIII</b>
<b>DAFTAR GAMBAR.....</b>	<b>XIII</b>
<b>DAFTAR TABEL .....</b>	<b>XXI</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Identifikasi Masalah.....	9
1.3    Rumusan Masalah.....	9
1.4    Batasan Masalah .....	9
1.5    Tujuan Penelitian .....	11
1.6    Manfaat Penelitian .....	11
1.7    Metode Penelitian .....	12
1.7.1    Metode Pengumpulan Data .....	12
1.7.2    Metode Pengembangan <i>Software</i> .....	12
1.8    Sistematika Penulisan .....	13
<b>BAB II LANDASAN TEORI .....</b>	<b>15</b>
2.1    Sistem Informasi .....	15
2.1.1    Pengertian Sistem.....	15
2.1.2    Pengertian Informasi .....	16
2.1.3    Pengertian Sistem Informasi .....	17

2.1.4	Komponen Sistem Informasi .....	17
2.2	Rancang Bangun .....	19
2.3	Perangkat Lunak ( <i>Software</i> ) .....	20
2.3.1	Perangkat Lunak Aplikasi ( <i>Application Software</i> ) .....	21
2.3.2	Modul Aplikasi ( <i>Application Module</i> ).....	21
2.3.3	Aplikasi Web ( <i>Web Application</i> ) .....	22
2.4	<i>E-Learning</i> .....	23
2.5	Pemrograman berbasis Objek ( <i>Object Oriented Programming/ OOP</i> ) ..	26
2.5.1	Abstraksi ( <i>abstraction</i> ) .....	27
2.5.2	Pewarisan ( <i>inheritance</i> ) .....	28
2.5.3	Polimorfisme ( <i>polymorphism</i> ).....	28
2.5.4	Penyembunyian Informasi ( <i>encapsulation</i> ) .....	29
2.5.5	Pengiriman Informasi ( <i>message sending</i> ) .....	30
2.5.6	Asosiasi ( <i>Association</i> ).....	31
2.5.7	Agregasi ( <i>aggregation</i> ) .....	31
2.5.8	Manfaat dan Keterbatasan OOP.....	32
2.6	Metode Pengumpulan Data.....	33
2.6.1	Observasi.....	33
2.6.2	Wawancara.....	33
2.6.3	Studi Literatur .....	34
2.7	Unified Modelling Language (UML) .....	35
2.7.1	Use Case Diagram.....	36
2.7.2	Activity Diagram.....	40
2.7.3	Class Diagram .....	43
2.7.4	Sequence Diagram .....	55
2.8	Iterative Model.....	63

2.8.1	Tahapan-tahapan Iterative Model .....	64
2.8.2	Kelebihan dan Kekurangan <i>Iterative Model</i> .....	65
2.8.3	Alasan Penggunaan <i>Iterative Model</i> .....	66
2.9	Basis Data ( <i>Database</i> ) .....	67
2.9.1	Sistem Manajemen Basis Data (DBMS).....	68
2.9.2	Skema Basis Data dan Pemodelannya dari <i>Class Diagram</i> .....	70
2.10	Git dan Github.....	76
2.10.1	Pengenalan Git .....	76
2.10.2	Kenapa Harus Menggunakan Git? .....	78
2.10.3	Pengenalan Github .....	80
2.10.4	Fitur-fitur Github.....	80
2.10.5	Kenapa Harus Menggunakan Github? .....	86
2.10.6	Berkas README.md untuk Repositori Github.....	88
2.11	Repositori Digital.....	89
2.12	Pengujian Perangkat Lunak ( <i>Software Testing</i> ).....	90
2.12.1	<i>Black Box Testing</i> dan <i>White Box Testing</i> .....	92
2.12.2	Level Testing.....	93
2.13	<i>Tools</i> Pengembangan Aplikasi.....	95
2.13.1	Draw.io ( <a href="#">diagrams.net</a> ).....	95
2.13.2	Figma .....	96
2.13.3	PHP .....	97
2.13.4	PHP <i>Framework</i> .....	99
2.13.5	Codeigniter.....	100
2.13.6	MySQL dan PHPMyAdmin.....	100
2.13.7	Bootstrap dan AdminLTE .....	103
2.14	Penelitian Sejenis .....	105

<b>BAB III METODE PENELITIAN .....</b>	<b>110</b>
3.1    Metode Pengumpulan Data.....	110
3.1.1    Observasi.....	110
3.1.2    Wawancara.....	112
3.1.3    Studi Literatur .....	112
3.2    Metode Pengembangan <i>Software</i> .....	113
3.3    Kerangka Penelitian .....	117
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>118</b>
4.1 <i>Requirement</i> (Kebutuhan).....	118
4.1.1    Gambaran umum Centratama Telekomunikasi Indonesia .....	118
4.1.2    Analisis Permasalahan .....	121
4.1.3    Identifikasi Aplikasi web HC Portal .....	123
4.1.4    Analisis Aplikasi Usulan.....	128
4.1.5    Analisis Struktur Aplikasi.....	130
4.1.6    Identifikasi Server Produksi.....	132
4.1.7    Persiapan Awal Pengembangan <i>Software</i> .....	133
4.1.8    Pembuatan Lingkungan Pengembangan <i>Software</i> dengan membuat repositori digital di platform Github .....	135
4.2 <i>Iteration</i> (Iterasi).....	144
4.2.1 <i>Analysis &amp; Design</i> (Analisis & Desain).....	144
4.2.2 <i>Implementation/Development</i> (Implementasi/Pengembangan) ....	307
4.2.3    Testing & Revise (Revisi) .....	314
4.2.4    Ulasan untuk tahapan <i>Production</i> (Produksi) dan Implikasi Penelitian	
	329
<b>BAB V PENUTUP .....</b>	<b>330</b>
5.1    Kesimpulan .....	330

5.2 Saran .....	331
<b>DAFTAR PUSTAKA .....</b>	<b>333</b>
<b>LAMPIRAN.....</b>	<b>349</b>



## DAFTAR GAMBAR

Gambar 1.1 Persentase Fokus Utama Bisnis pada Transformasi Digital.....	1
Gambar 2.1 Sebuah kelas memiliki <i>property</i> dan <i>behaviour</i> .....	27
Gambar 2.2 Wajan dan spatula merupakan bagian dari <i>Class</i> alat memasak .....	28
Gambar 2.3 Notasi-notasi diagram <i>Use Case</i> .....	38
Gambar 2.4 Contoh <i>Activity Diagram</i> .....	43
Gambar 2.5 <i>Class</i> dengan <i>attribute</i> , <i>visibility attribute</i> , dan <i>operation</i> -nya.....	45
Gambar 2.6 <i>Class</i> tanpa atribut dan fungsi .....	46
Gambar 2.7 <i>Class</i> dengan tanda elipsis.....	46
Gambar 2.8 <i>Class</i> dengan tanda <i>stereotype</i> .....	47
Gambar 2.9 Asosiasi dua <i>Class</i> .....	47
Gambar 2.10 Peran pada asosiasi dua <i>Class</i> .....	47
Gambar 2.11 Asosiasi beberapa <i>Class</i> dengan sebuah <i>Class</i> .....	48
Gambar 2.12 Contoh <i>Multiplicity</i> .....	49
Gambar 2.13 Mekanisme <i>Inheritance</i> .....	50
Gambar 2.14 <i>Abstract Class</i> karyawan karena tidak memiliki <i>instance</i> .....	50
Gambar 2.15 Dependensi <i>class Form</i> terhadap <i>class System</i> .....	51
Gambar 2.16 Contoh <i>Interface</i> .....	51
Gambar 2.17 Anak panah pada <i>dependency</i> , <i>interface</i> , dan <i>inheritance/generalisasi</i> .....	52
Gambar 2.18 Contoh Agregasi.....	52
Gambar 2.19 Penggunaan <i>Composite</i> dan Agregasi .....	53
Gambar 2.20 <i>Constraints</i> pada atribut <i>Class</i> ; atribut kapasitas memiliki tiga kemungkinan nilai, yaitu 7 kg, 12 kg, ataupun 15 kg .....	54
Gambar 2.21 <i>Constraints</i> pada asosiasi <i>Class</i> .....	54
Gambar 2.22 Relasi OR pada dua asosiasi dalam sebuah <i>Constraints</i> .....	54
Gambar 2.23 <i>Constraints</i> pada agregasi untuk menunjukkan komponen mana yang akan dipakai dengan relasi “atau” .....	54
Gambar 2.24 Sebuah <i>Participant</i> pada <i>Sequence Diagram</i> ; Nama <i>participant</i> merupakan komposisi dari nama <i>instance</i> dan nama <i>class</i> .....	56
Gambar 2.25 <i>Participant</i> dengan hanya nama <i>class</i> (kiri) dan <i>participant</i> dengan nama objek dari <i>class</i> yang tidak diketahui (kanan) .....	56

Gambar 2.26 Notasi-notasi <i>Message</i> .....	57
Gambar 2.27 <i>Recursive</i> pada <i>Sequence Diagram</i> .....	58
Gambar 2.28 Menghapus dan membuat <i>participant</i> .....	59
Gambar 2.29 Penggunaan <i>interaction frame</i> pada <i>Sequence Diagram</i> .....	60
Gambar 2.30 Sequence Diagram dengan notasi <i>stereotype</i> .....	63
Gambar 2.31 Lingkungan Sistem Basis Data .....	69
Gambar 2.32 <i>Class Book</i> memiliki struktur data sederhana dan bisa langsung menjadi table .....	70
Gambar 2.33 <i>Collection</i> pada <i>Class</i> dibuatkan tabel terpisah dan diberikan id sendiri.....	71
Gambar 2.34 Mengubah relasi <i>many-to-many</i> menjadi tabel relasional.....	72
Gambar 2.35 Mengubah relasi <i>one-to-one</i> menjadi tabel relasional.....	72
Gambar 2.36 Implementasi <i>superclass</i> menjadi tabel dan menambahkan atribut <i>role</i> pada tabel relasional untuk membedakan kedua jenis <i>user</i> .....	73
Gambar 2.37 Implementasi <i>subclass</i> menjadi tabel berbeda .....	74
Gambar 2.38 Implementasi <i>superclass</i> dan <i>subclass</i> untuk menjadi tabel yang terpisah .....	75
Gambar 2.39 Alur Perubahan Berkas pada Git.....	77
Gambar 2.40 Sebuah repositori publik yang dapat diakses dengan masuk ke akun (kiri) dan secara bebas tanpa masuk ke akun (kanan).....	81
Gambar 2.41 Sebuah repositori <i>private</i> yang dapat diakses oleh pemilik repositori ataupun kontributor (kiri).....	81
Gambar 2.42 Contoh perubahan visual pada baris kode yang dihapus ditandai dengan warna merah dan ditambahkan ditandai dengan warna hijau; disebut tampilan <i>diff</i> yang berarti perbedaan.....	82
Gambar 2.43 Tampilan aplikasi draw.io berjalan pada platform desktop dengan sistem operasi Windows 10.....	96
Gambar 2.44 Tampilan aplikasi Figma berjalan pada peramban web Firefox .....	96
Gambar 2.45 Contoh penggunaan bahasa <i>scripting</i> PHP .....	98
Gambar 2.46 Hasil dari eksekusi kode PHP sebelumnya .....	98
Gambar 2.47 Contoh <i>query</i> SQL untuk memberikan perintah pada sistem basis data .....	101

Gambar 2.48 Tipe-tipe <i>query</i> SQL .....	102
Gambar 2.49 Tampilan PHPMyAdmin menampilkan tabel-tabel dari satu basis data .....	103
Gambar 2.50 Kode HTML yang menerapkan Bootstrap .....	104
Gambar 2.51 Halaman web resmi untuk mengunduh Bootstrap .....	104
Gambar 3.1 Kerangka Penelitian .....	117
Gambar 4.1 Logo perusahaan .....	118
Gambar 4.2 Struktur Organisasi PT Centratama Telekomunikasi Indonesia (CTI) .....	120
Gambar 4.3 Tampilan halaman awal HC Portal; tombol <i>Login</i> di pojok kanan atas akan memunculkan jendela <i>popup Login</i> ketika diklik. ....	124
Gambar 4.4 Tampilan jendela <i>popup Login</i> aplikasi HC Portal. ....	124
Gambar 4.5 Tampilan modul aplikasi <i>Job Profile</i> aplikasi HC Portal; sebelah kiri gambar merupakan <i>sidebar</i> untuk navigasi ke modul aplikasi lainnya. ....	125
Gambar 4.6 Diagram Konteks <i>Class</i> sebelum pengembangan <i>e-learning</i> .....	126
Gambar 4.7 Pemodelan Basis Data sebelum pengembangan <i>e-learning</i> .....	127
Gambar 4.8 Alur proses <i>e-learning</i> secara keseluruhan .....	128
Gambar 4.9 <i>E-Learning</i> sebagai modul aplikasi HC Portal.....	129
Gambar 4.10 Struktur folder bawaan CodeIgniter .....	130
Gambar 4.11 Struktur folder <i>application</i> bawaan CodeIgniter.....	131
Gambar 4.12 Pengaturan repositori baru yang penulis buat. ....	138
Gambar 4.13 Tampilan awal setelah membuat repositori baru tanpa memilih opsi-opsi pada <i>initialize this repository</i> dan opsi <i>repository template</i> .....	139
Gambar 4.14 Folder baru kosong untuk membuat repositori baru .....	140
Gambar 4.15 <i>Command line</i> dengan perintah <i>git init</i> yang telah dimasukkan ...	140
Gambar 4.16 Github menyarankan untuk membuat berkas teks README.md, LICENSE, dan .gitignore .....	140
Gambar 4.17 Tampilan folder repositori lokal sebelum ditambahkan berkas <i>framework</i> CodeIgniter .....	142
Gambar 4.18 Tampilan folder repositori lokal setelah ditambahkan berkas <i>framework</i> CodeIgniter .....	142

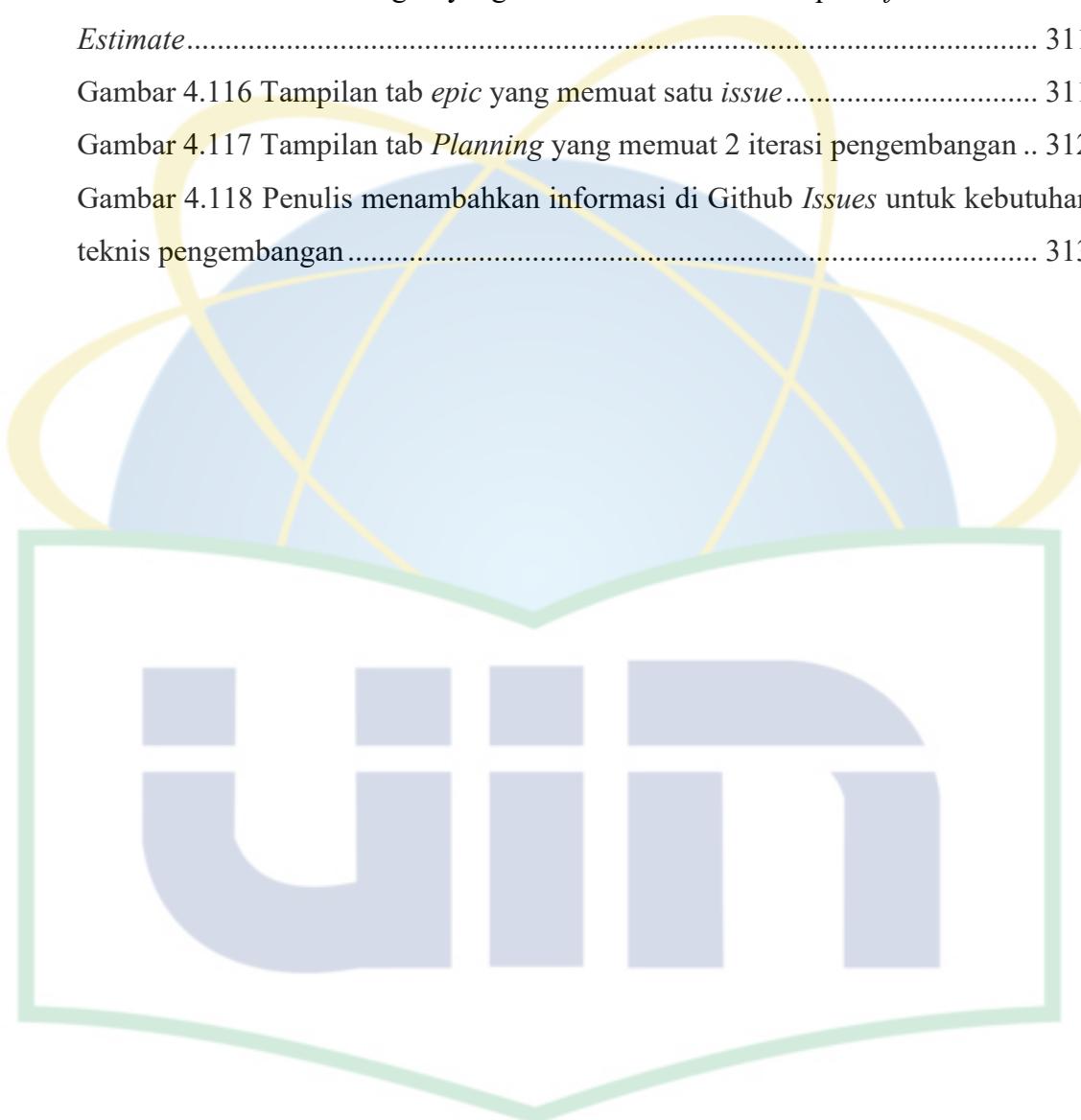
Gambar 4.19 Rangkaian perintah Git untuk menambahkan repositori lokal ke dalam repositori <i>remote</i> .....	143
Gambar 4.20 <i>Rich Picture</i> proses berjalan <i>training</i> karyawan baru .....	145
Gambar 4.21 <i>Rich Picture</i> proses berjalan pengajuan <i>training</i> karyawan lama. .	146
Gambar 4.22 <i>Rich Picture</i> proses berjalan <i>training</i> karyawan lama secara luring .....	147
Gambar 4.23 <i>Rich Picture</i> proses berjalan <i>training</i> secara daring.....	148
Gambar 4.24 <i>Rich Picture</i> proses berjalan diskusi <i>training</i> yang terjadi secara luring (atas) dan daring (tengah), serta melalui platform <i>chat</i> grup (bawah).....	150
Gambar 4.25 <i>Rich Picture</i> analisis sistem usulan .....	152
Gambar 4.26 <i>Use Case Diagram</i> <i>e-Learning</i> dan <i>Forum</i> pada HC Portal.....	159
Gambar 4.27 <i>Activity Diagram</i> Buka Halaman Utama.....	207
Gambar 4.28 <i>Activity Diagram</i> <i>Login</i> .....	208
Gambar 4.29 <i>Activity Diagram</i> <i>Logout</i> .....	209
Gambar 4.30 <i>Activity Diagram</i> Kelola Profil Akun .....	210
Gambar 4.31 <i>Activity Diagram</i> Kelola Data Karyawan.....	211
Gambar 4.32 <i>Activity Diagram</i> Akses Modul Belajar <i>Induction</i> .....	212
Gambar 4.33 <i>Activity Diagram</i> Isi Formulir Pengajuan <i>Training</i> Baru (PTB) ..	213
Gambar 4.34 <i>Activity Diagram</i> Persetujuan <i>Training</i> Baru.....	215
Gambar 4.35 <i>Activity Diagram</i> Kelola Modul <i>Training</i> /Penambahan Modul <i>Training</i> Baru.....	216
Gambar 4.36 <i>Activity Diagram</i> Kelola Modul <i>Training</i> /Pengarsipan Modul <i>Training</i> .....	217
Gambar 4.37 <i>Activity Diagram</i> Kelola Modul <i>Training</i> /Pengeditan Konten Modul <i>Training</i> Bertipe Tekstual dan Gambar.....	219
Gambar 4.38 <i>Activity Diagram</i> Kelola Modul <i>Training</i> /Pengeditan Konten Bertipe Video .....	220
Gambar 4.39 <i>Activity Diagram</i> Kelola Modul <i>Training</i> /Pengeditan konten bertipe PDF .....	221
Gambar 4.40 <i>Activity Diagram</i> Kelola Modul <i>Training</i> /Pembuatan Konten Kuis .....	223
Gambar 4.41 <i>Activity Diagram</i> Penjadwalan <i>Training</i> Luring .....	224

Gambar 4.42 <i>Activity Diagram</i> Penjadwalan VLT ( <i>Virtual Live Training</i> ) .....	225
Gambar 4.43 <i>Activity Diagram</i> Kelola Akses PIC Modul <i>Training</i> .....	226
Gambar 4.44 <i>Activity Diagram</i> Kelola Akses Peserta Modul <i>Training</i> .....	227
Gambar 4.45 <i>Activity Diagram</i> Akses Laporan <i>Training</i> .....	228
Gambar 4.46 <i>Activity Diagram</i> Akses Modul <i>Training</i> .....	229
Gambar 4.47 <i>Activity Diagram</i> Akses Forum <i>Training/Menampilkan Posting Forum</i> .....	230
Gambar 4.48 <i>Activity Diagram</i> Akses Forum <i>Training/Balasan Posting Forum</i>	231
Gambar 4.49 <i>Activity Diagram</i> Akses Forum <i>Training/Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul Training</i> .....	233
Gambar 4.50 <i>Activity Diagram</i> Akses Forum <i>Training/Pemberian Voting dan Penandaan Balasan Forum Training sebagai Solusi</i> .....	234
Gambar 4.51 <i>Activity Diagram</i> Moderasi Forum <i>Training</i> .....	236
Gambar 4.52 <i>Class Diagram</i> Modul Aplikasi <i>e-learning</i> dan Forum <i>Training</i> .	237
Gambar 4.53 Pemodelan Basis Data untuk modul <i>e-learning</i> .....	238
Gambar 4.54 Pemodelan Master Data Aplikasi HC Portal untuk Mengelola Data Pengguna .....	239
Gambar 4.55 Pemodelan Main App Data untuk Keperluan Dasar Aplikasi.....	240
Gambar 4.56 Pemodelan Basis Data Keseluruhan untuk Studi Kasus Penelitian Ini .....	241
Gambar 4.57 <i>Sequence Diagram</i> Buka Halaman Awal .....	267
Gambar 4.58 <i>Sequence Diagram</i> Login .....	268
Gambar 4.59 <i>Sequence Diagram</i> Logout .....	269
Gambar 4.60 <i>Sequence Diagram</i> Kelola Profil Akun.....	270
Gambar 4.61 <i>Sequence Diagram</i> Kelola Data Karyawan.....	271
Gambar 4.62 <i>Sequence Diagram</i> Akses Modul Belajar <i>Induction</i> .....	272
Gambar 4.63 <i>Sequence Diagram</i> Isi Formulir Pengajuan <i>Training Baru (PTB)</i>	273
Gambar 4.64 <i>Sequence Diagram</i> Persetujuan <i>Training Baru</i> .....	274
Gambar 4.65 <i>Sequence Diagram</i> Penambahan Modul <i>Training Baru</i> .....	275
Gambar 4.66 <i>Sequence Diagram</i> Pengarsipan Modul <i>Training</i> .....	276
Gambar 4.67 <i>Sequence Diagram</i> Pengeditan Konten Modul <i>Training</i> Bertipe Tekstual dan Gambar .....	277

Gambar 4.68 <i>Sequence Diagram</i> Pengeditan Konten Modul <i>Training</i> Bertipe Video .....	278
Gambar 4.69 <i>Sequence Diagram</i> Pengeditan Konten Modul <i>Training</i> Bertipe PDF .....	279
Gambar 4.70 <i>Sequence Diagram</i> Pembuatan Konten Kuis .....	280
Gambar 4.71 <i>Sequence Diagram</i> Penjadwalan <i>Training Luring</i> .....	281
Gambar 4.72 <i>Sequence Diagram</i> Penjadwalan VLT ( <i>Virtual Live Training</i> ) ....	282
Gambar 4.73 <i>Sequence Diagram</i> Kelola Akses PIC Modul <i>Training</i> .....	283
Gambar 4.74 <i>Sequence Diagram</i> Kelola Akses Peserta Modul <i>Training</i> .....	284
Gambar 4.75 <i>Sequence Diagram</i> Akses Laporan <i>Training</i> .....	285
Gambar 4.76 <i>Sequence Diagram</i> Akses Modul <i>Training</i> .....	286
Gambar 4.77 <i>Sequence Diagram</i> Menampilkan Posting Forum .....	287
Gambar 4.78 <i>Sequence Diagram</i> Balasan Posting Forum .....	288
Gambar 4.79 <i>Sequence Diagram</i> Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul <i>Training</i> .....	289
Gambar 4.80 <i>Sequence Diagram</i> Pemberian <i>Voting</i> dan Penandaan Balasan Forum <i>Training</i> sebagai Solusi.....	290
Gambar 4.81 <i>Sequence Diagram</i> Moderasi Forum <i>Training</i> .....	292
Gambar 4.82 Struktur Menu untuk Karyawan CTI .....	293
Gambar 4.83 Struktur Menu untuk Peserta dan Pemateri <i>Training</i> .....	293
Gambar 4.84 Struktur Menu untuk PIC <i>Training</i> .....	294
Gambar 4.85 Struktur Menu untuk Karyawan HC .....	294
Gambar 4.86 Struktur Menu untuk Kadiv HC .....	294
Gambar 4.87 Struktur Menu untuk Kepala Departemen .....	295
Gambar 4.88 Struktur Menu untuk Kadiv Departemen .....	295
Gambar 4.89 Tampilan Halaman Awal Aplikasi (HomeView).....	296
Gambar 4.90 Tampilan <i>Login</i> Aplikasi berupa <i>Popup</i> ( <i>Login Popup</i> ).....	296
Gambar 4.91 Halaman Awal JP Setelah Karyawan Melakukan <i>Login</i> ( <i>MainJobProfileView</i> ).....	297
Gambar 4.92 Halaman Profil Pengguna ( <i>ProfileView</i> ) .....	297
Gambar 4.93 Halaman Manajemen Master Data ( <i>MasterDataView</i> ) .....	298

Gambar 4.94 Halaman Manajemen Data Karyawan (EmployeeManagementView) .....	298
Gambar 4.95 Halaman awal <i>E-Learning</i> (ElearningView).....	299
Gambar 4.96 Tampilan Penggerjaan Modul <i>Training</i> dengan menu <i>sidebar</i> utama terbuka (DoLearning).....	299
Gambar 4.97 Tampilan Penggerjaan Modul <i>Training</i> dengan menu <i>sidebar</i> utama tertutup (DoLearning) .....	300
Gambar 4.98 Halaman Awal Pengajuan <i>Training</i> Baru .....	300
Gambar 4.99 Formulir Pengajuan <i>Training</i> Baru (FormPTBView) .....	301
Gambar 4.100 Tampilan Pembuat Modul <i>Training</i> (PemanduModulTrainingView) .....	301
Gambar 4.101 Tampilan Persetujuan PTB (PersetujuanPTBView) .....	302
Gambar 4.102 Tampilan Pengelolaan Modul <i>Training</i> (KelolaModulTrainingView) .....	302
Gambar 4.103 Tampilan Pratinjau Konten <i>Training</i> (KelolaModulTrainingView) .....	303
Gambar 4.104 Tampilan Pengelolaan Modul <i>Training</i> Bertipe Kuis (KelolaModulTrainingView) .....	303
Gambar 4.105 Tampilan Pengelolaan Modul <i>Training</i> Bertipe PDF (KelolaModulTrainingView) .....	304
Gambar 4.106 Tampilan Pengelolaan Modul <i>Training</i> Bertipe <i>Live Training</i> Daring dan Luring (KelolaModulTrainingView).....	304
Gambar 4.107 Tampilan Pengelolaan PIC <i>Training</i> (KelolaModulTrainingView) .....	305
Gambar 4.108 Tampilan Pengelolaan Peserta <i>Training</i> (KelolaModulTrainingView) .....	305
Gambar 4.109 Tampilan Laporan <i>Training</i> (KelolaModulTraining) .....	306
Gambar 4.110 Tampilan Forum Diskusi <i>Training</i> (ForumElearningView) .....	306
Gambar 4.111 Tampilan Posting Forum Diskusi <i>Training</i> (ForumElearningView) .....	307
Gambar 4.112 <i>Popup</i> Pembuatan Posting Baru .....	307

Gambar 4.113 Beberapa <i>issues</i> yang penulis buat untuk mengumpulkan kebutuhan pengembangan <i>e-learning</i> .....	309
Gambar 4.114 Tampilan awal sebelum penulis menyetel atribut <i>field</i> informasi untuk kebutuhan manajemen proyek. ..	310
Gambar 4.115 Daftar tugas yang telah diberi informasi pada <i>field Status</i> dan <i>Estimate</i> .....	311
Gambar 4.116 Tampilan tab <i>epic</i> yang memuat satu <i>issue</i> .....	311
Gambar 4.117 Tampilan tab <i>Planning</i> yang memuat 2 iterasi pengembangan ..	312
Gambar 4.118 Penulis menambahkan informasi di Github <i>Issues</i> untuk kebutuhan teknis pengembangan ..	313



## DAFTAR TABEL

Tabel 2.1 Tipe Diagram UML .....	35
Tabel 2.2 Contoh Spesifikasi <i>Use Case</i> .....	39
Tabel 2.3 Notasi-notasi pada <i>Activity Diagram</i> .....	41
Tabel 2.4 Daftar <i>visibility</i> .....	45
Tabel 2.5 Operator-operator umum yang digunakan pada <i>interaction frame</i> .....	61
Tabel 2.6 Contoh CRC <i>Card</i> .....	62
Tabel 2.7 Contoh tabel kontak pada basis data.....	67
Tabel 2.8 Contoh Spesifikasi Basis Data .....	76
Tabel 2.9 Contoh <i>Test Case</i> pada <i>Integration Testing</i> .....	93
Tabel 4.1 Spesifikasi Server Produksi .....	132
Tabel 4.2 Spesifikasi komponen aplikasi HC Portal.....	133
Tabel 4.3 Spesifikasi perangkat pengembangan aplikasi.....	134
Tabel 4.4 Isian formulir pembuatan repositori baru.....	137
Tabel 4.5 Identifikasi Aktor pada <i>e-Learning</i> dan Forum HC Portal .....	155
Tabel 4.6 Identifikasi <i>Use Case</i> pada <i>e-learning</i> dan forum HC Portal .....	156
Tabel 4.7 Spesifikasi <i>Use Case</i> Buka Halaman Utama .....	159
Tabel 4.8 Spesifikasi <i>Use Case Login</i> .....	160
Tabel 4.9 Spesifikasi <i>Use Case Logout</i> .....	161
Tabel 4.10 Spesifikasi <i>Use Case</i> Kelola Profil Akun .....	162
Tabel 4.11 Spesifikasi <i>Use Case</i> Kelola Data Karyawan .....	163
Tabel 4.12 Spesifikasi <i>Use Case</i> Akses Modul Belajar <i>Induction</i> .....	164
Tabel 4.13 Spesifikasi <i>Use Case</i> Isi Formulir Pengajuan <i>Training</i> Baru (PTB)	165
Tabel 4.14 Spesifikasi <i>Use Case</i> Persetujuan <i>Training</i> Baru .....	167
Tabel 4.15 Spesifikasi <i>Use Case</i> Kelola Modul <i>Training</i> /Penambahan Modul <i>Training</i> Baru.....	169
Tabel 4.16 Spesifikasi <i>Use Case</i> Kelola Modul <i>Training</i> /Pengarsipan Modul <i>Training</i> .....	170
Tabel 4.17 Spesifikasi <i>Use Case</i> Kelola Modul <i>Training</i> /Pengeditan Konten Modul <i>Training</i> Bertipe Tekstual dan Gambar.....	172
Tabel 4.18 Spesifikasi <i>Use Case</i> Kelola Modul <i>Training</i> /Pengeditan Konten Modul <i>Training</i> Bertipe Video .....	175

Tabel 4.19 Spesifikasi <i>Use Case</i> Kelola Modul <i>Training/Pengeditan Konten Modul training</i> Bertipe PDF (presentasi dek atau berkas .pdf lainnya) .....	178
Tabel 4.20 Spesifikasi <i>Use Case</i> Kelola Modul <i>Training/Pembuatan Konten Kuis</i> .....	180
Tabel 4.21 Spesifikasi <i>Use Case</i> Penjadwalan <i>Training Luring</i> .....	183
Tabel 4.22 Spesifikasi <i>Use Case</i> Penjadwalan <i>VLT (Virtual Live Training)</i> ....	186
Tabel 4.23 Spesifikasi <i>Use Case</i> Kelola Akses PIC Modul <i>Training</i> .....	189
Tabel 4.24 Spesifikasi <i>Use Case</i> Kelola Modul <i>Training/Kelola Akses Peserta Modul Training</i> .....	190
Tabel 4.25 Spesifikasi <i>Use Case</i> Akses Laporan <i>Training</i> .....	192
Tabel 4.26 Spesifikasi <i>Use Case</i> Akses Modul <i>Training</i> .....	193
Tabel 4.27 Spesifikasi <i>Use Case</i> Akses Forum <i>Training/Menampilkan Posting Forum</i> .....	196
Tabel 4.28 Spesifikasi <i>Use Case</i> Akses Forum <i>Training/Balasan Posting Forum</i> .....	196
Tabel 4.29 Spesifikasi <i>Use Case</i> Akses Forum <i>Training/Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul Training</i> .....	199
Tabel 4.30 Spesifikasi <i>Use Case</i> Akses Forum <i>Training/Pemberian Voting dan Penandaan Balasan Forum Training sebagai Solusi</i> .....	201
Tabel 4.31 Spesifikasi <i>Use Case</i> Moderasi Forum <i>Training</i> .....	202
Tabel 4.32 Spesifikasi Tabel <i>E-learning</i> Modul <i>Training</i> .....	242
Tabel 4.33 Spesifikasi Tabel <i>E-learning</i> Konten.....	242
Tabel 4.34 Spesifikasi Tabel <i>E-learning</i> Kuis Konten .....	244
Tabel 4.35 Spesifikasi Tabel <i>E-learning</i> Kuis <i>User Data</i> .....	245
Tabel 4.36 Spesifikasi Tabel <i>E-learning</i> <i>Text-based</i> Konten .....	246
Tabel 4.37 Spesifikasi Tabel <i>E-learning</i> PDF File Konten .....	246
Tabel 4.38 Spesifikasi Tabel <i>E-learning</i> <i>Live Training</i> Konten.....	247
Tabel 4.39 Spesifikasi Tabel <i>E-learning</i> <i>Progress</i> .....	249
Tabel 4.40 Spesifikasi Tabel <i>E-learning</i> <i>Pengajuan Training</i> .....	250
Tabel 4.41 Spesifikasi Tabel <i>E-learning</i> <i>User Access</i> .....	251
Tabel 4.42 Spesifikasi Tabel <i>E-learning</i> <i>Posting Forum</i> .....	253
Tabel 4.43 Spesifikasi Tabel <i>E-learning</i> <i>Balasan Posting Forum</i> .....	254

Tabel 4.44 Spesifikasi Tabel Master <i>Users</i> .....	256
Tabel 4.45 Spesifikasi Tabel Master Positions .....	257
Tabel 4.46 Spesifikasi Tabel Master Divisions.....	259
Tabel 4.47 Spesifikasi Tabel Master Departments .....	259
Tabel 4.48 Spesifikasi Tabel Master Entities.....	260
Tabel 4.49 Spesifikasi Tabel Master Work Location.....	260
Tabel 4.50 Spesifikasi Tabel Master Employee Status.....	261
Tabel 4.51 Spesifikasi Tabel Master Job Level .....	261
Tabel 4.52 Spesifikasi Tabel User Menu .....	262
Tabel 4.53 Spesifikasi Tabel Menu Accesses.....	262
Tabel 4.54 Spesifikasi Tabel Sub Menu .....	263
Tabel 4.55 Spesifikasi Tabel Sub Menu Accesses.....	263
Tabel 4.56 Spesifikasi Tabel User Role .....	264
Tabel 4.57 Spesifikasi Tabel User Role Appadmins .....	265
Tabel 4.58 Spesifikasi Tabel User Role Superadmins .....	265
Tabel 4.59 Spesifikasi Tabel User Role Special .....	266
Tabel 4.60 Spesifikasi Tabel User Token .....	267
Tabel 4.61 Pengujian <i>Use Case Lihat Halaman Awal</i> .....	314
Tabel 4.62 Pengujian <i>Use Case Login</i> .....	314
Tabel 4.63 Pengujian <i>Use Case Logout</i> .....	315
Tabel 4.64 Pengujian <i>Use Case Kelola Profil Akun</i> .....	315
Tabel 4.65 Pengujian <i>Use Case Kelola Data Karyawan</i> .....	316
Tabel 4.66 Pengujian Akses Modul Belajar <i>Induction</i> .....	317
Tabel 4.67 Pengujian Isi Formulir Pengajuan <i>Training Baru</i> (PTB) .....	317
Tabel 4.68 Pengujian Persetujuan <i>Training Baru</i> .....	318
Tabel 4.69 Pengujian Kelola Modul <i>Training</i> /Penambahan Modul <i>Training</i> Baru .....	318
Tabel 4.70 Pengujian Kelola Modul <i>Training</i> /Pengarsipan Modul <i>Training</i> .....	319
Tabel 4.71 Pengujian Kelola Modul <i>Training</i> /Pengeditan Konten Modul <i>Training</i> Bertipe Tekstual, Gambar, dan Video .....	319
Tabel 4.72 Pengujian Kelola Modul <i>Training</i> /Pengeditan Konten Modul <i>training</i> Bertipe PDF (presentasi dek atau berkas .pdf lainnya) .....	320

Tabel 4.73 Pengujian Kelola Modul <i>Training</i> /Pembuatan Konten Kuis .....	321
Tabel 4.74 Pengujian Penjadwalan <i>Training</i> Luring .....	322
Tabel 4.75 Pengujian Penjadwalan VLT ( <i>Virtual Live Training</i> ) .....	323
Tabel 4.76 Pengujian Kelola Akses PIC Modul <i>Training</i> .....	323
Tabel 4.77 Pengujian Kelola Akses Peserta Modul <i>Training</i> .....	324
Tabel 4.78 Pengujian Akses Laporan <i>Training</i> .....	324
Tabel 4.79 Pengujian Akses Modul <i>Training</i> .....	325
Tabel 4.80 Pengujian Akses Forum <i>Training</i> /Menampilkan Posting Forum ....	325
Tabel 4.81 Pengujian Akses Forum <i>Training</i> /Balasan Posting Forum.....	326
Tabel 4.82 Pengujian Akses Forum <i>Training</i> /Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul <i>Training</i> .....	326
Tabel 4.83 Pengujian Akses Forum <i>Training</i> /Pemberian <i>Voting</i> dan Penandaan Balasan Forum <i>Training</i> sebagai Solusi .....	327
Tabel 4.84 Pengujian Moderasi Forum <i>Training</i> .....	328





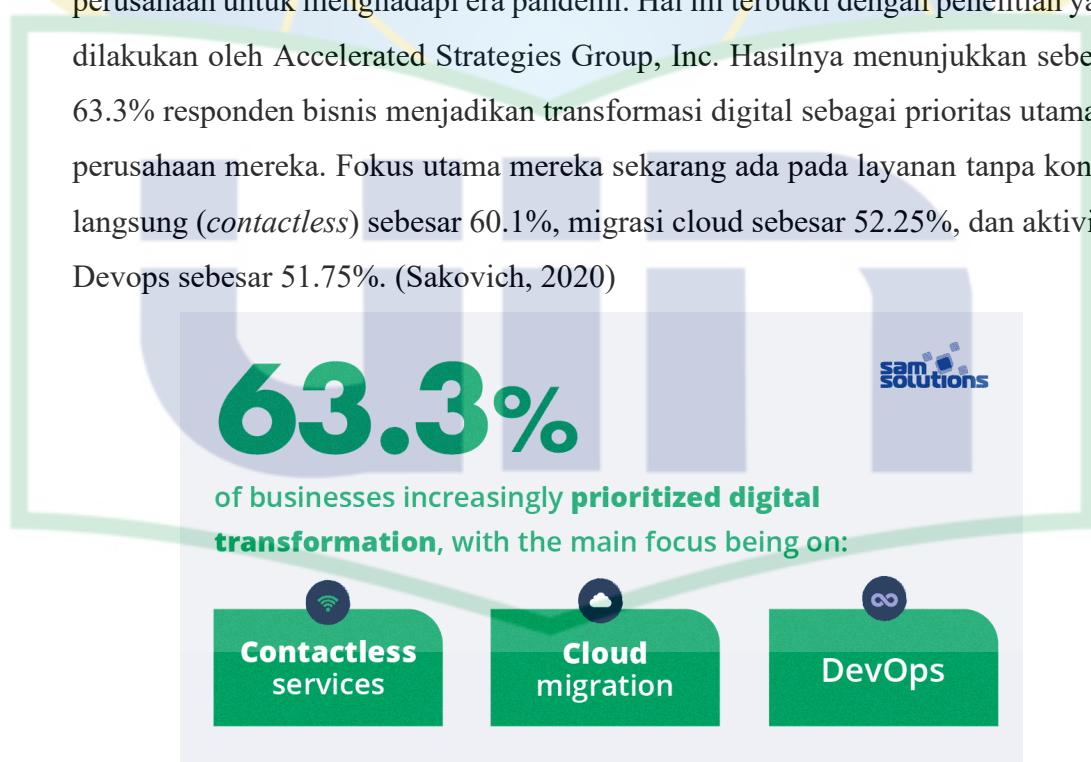
# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Era endemi menjadi sebuah awalan baru untuk menjalani kehidupan selanjutnya setelah pandemi yang melanda sekitar 2 tahun lamanya. Tahun 2022 juga menjadi akhir dari masa Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) di Indonesia. Ini akan membuat tahun 2023 sebagai sebuah era yang benar-benar baru. Banyak hal yang terjadi selama era pandemi khususnya pada aktivitas keseharian masyarakat, usaha bisnis, dan pekerjaan sehari-hari. Tidak sedikit usaha yang gulung tikar, namun beberapa lainnya mampu bertahan di era pandemi dengan mengelola kembali proses kerja mereka dan mengembangkan pendekatan baru untuk tetap bertahan di kondisi yang seperti itu. (Kemkes RI, 2022; Nalini, 2021)

Transformasi digital adalah salah satu strategi yang banyak diambil perusahaan untuk menghadapi era pandemi. Hal ini terbukti dengan penelitian yang dilakukan oleh Accelerated Strategies Group, Inc. Hasilnya menunjukkan sebesar 63.3% responden bisnis menjadikan transformasi digital sebagai prioritas utama di perusahaan mereka. Fokus utama mereka sekarang ada pada layanan tanpa kontak langsung (*contactless*) sebesar 60.1%, migrasi cloud sebesar 52.25%, dan aktivitas Devops sebesar 51.75%. (Sakovich, 2020)



Gambar 1.1 Persentase Fokus Utama Bisnis pada Transformasi Digital

Transformasi digital di era endemi sudah bukan lagi sebuah opsi, melainkan sudah menjadi keharusan agar bisnis usaha dapat terus berkembang. Ditambah dengan gaya hidup *hybrid* menjadi tren baru dalam berbagai bidang kehidupan.

Misalnya pada dunia pekerjaan, masyarakat sekarang dapat melakukan pekerjaannya secara daring (*online*) maupun luring (*offline*). Ini memerlukan dukungan teknologi informasi untuk masuk dalam tren tersebut. Namun, transformasi digital tidak akan terjadi tanpa adanya dukungan pimpinan digital (*digital leader*), literasi digital, dan budaya digital di organisasi. (Kemenkeu RI, 2022; Tulungen, Saerang, & Maramis, 2022)

Transformasi digital dapat dilakukan pada berbagai bidang industri, contohnya pada bidang pertanian. Pembuatan solusi digital berupa *Knowledge Management System* (KMS) untuk para petani sri rezeki (*Aglaonema*) dapat membantu mereka untuk meningkatkan pertumbuhan bisnis. Hal ini yang dilakukan oleh Sugiarti dkk dalam penelitian yang berjudul “*Knowledge Management System to Improve The Competence of Aglaonema Farmers*”. KMS yang dibuat dapat membantu para petani berbagi informasi berupa pengetahuan dan pengalaman mereka dalam pengolahan dan pemasaran tanaman hias. Hal ini dapat meningkatkan kompetensi para petani sri rezeki. Inilah yang dapat meningkatkan pertumbuhan bisnis untuk para petani sri rezeki. (Sugiarti, Suroso, Hermadi, Sunarti, & Broer, 2021)

Centratama Teknologi Indonesia (CTI) merupakan salah satu perusahaan yang mulai menerapkan transformasi digital. Perusahaan ini bergerak di bidang infrastruktur jaringan. CTI telah melakukan berbagai pengembangan solusi digital untuk mengolah informasi, seperti aplikasi memo pekerjaan e-memo, aplikasi pengelolaan perjalanan bisnis e-travel, aplikasi penilaian kinerja pegawai eKPI, dan masih banyak lagi. CTI terus melakukan inovasi dengan melakukan berbagai pengembangan *software* untuk mendukung proses kerja mereka.

Proses kerja yang tidak menggunakan solusi digital menyebabkan data dan informasinya tersebar di beberapa tempat penyimpanan seperti di komputer lokal karyawan atau di tempat penyimpanan *cloud*. Hal ini mengakibatkan proses pembuatan laporan hasil kerjanya jadi sulit untuk dilakukan. Inilah yang masih terjadi pada proses *training* karyawan di CTI.

Proses *training* karyawan merupakan tanggung jawab yang dipegang oleh divisi Human Capital (HC). Divisi ini merupakan salah satu dari tujuh divisi kerja di CTI. HC memiliki tugas dalam bidang pengelolaan *Human Resource*

*Development* (HRD). Proses *training* karyawan biasanya dilakukan dengan mengadakan seminar luring maupun daring. Seminar luring dilaksanakan di ruangan aula perusahaan, sedangkan seminar daring dilakukan dengan menggunakan platform konferensi video.

Seminar luring memerlukan banyak waktu dan biaya untuk persiapannya. Hal ini membuat *training* dengan seminar luring jarang untuk diadakan. Tidak jarang beberapa agenda *training* karyawan secara luring yang telah direncanakan pun pada akhirnya ditunda. Inilah yang memunculkan untuk pengadaan *training* secara daring. Jika dibandingkan dengan pengadaan *training* secara luring, pengadaan *training* secara daring lebih murah dan cepat untuk biaya dan waktu persiapannya. Karyawan divisi HC tidak perlu menyiapkan ruangan, *hard-copy* materi, ataupun mencetak lembar kuis untuk mengetes kemampuan karyawan yang mengikuti *training*. Mereka memanfaatkan platform konferensi video, mengirimkan berkas *pdf* materi kepada peserta *training*, dan menggunakan platform pembuat formulir untuk mengadakan kuis.

Sayangnya dari kacamata karyawan divisi HC, tidak semua karyawan dapat mengikuti dengan baik proses *training* secara daring. Selalu ada saja beberapa karyawan yang terlihat tidak aktif. Karyawan HC beranggapan mungkin pada saat pengadaan *training* secara daring, peserta mungkin tidak memiliki koneksi internet yang stabil. Namun jika dikaitkan dengan tren *hybrid* yang terjadi saat ini khususnya proses *training* yang dilakukan secara daring, mereka dapat melakukannya di mana saja termasuk di lingkungan yang kurang kondusif. Inilah yang mungkin dialami oleh mereka sehingga terlihat tidak aktif dalam proses *training* secara daring.

Konsep bekerja secara daring mulai merebak sejak dimulainya pandemi COVID-19. Saat itu perusahaan mau tidak mau harus menerapkan konsep *Work From Home* (WFH) demi menjaga kesehatan karyawannya. WFH telah memberikan dampak positif dan negatif bagi produktivitas karyawan. Beberapa dampak negatif yang dirasakan oleh karyawan seperti diharuskan melakukan *multitasking* yang mana membuat mereka harus berpindah fokus dari perkerjaan mereka ke keluarga, kurangnya motivasi kerja karena mereka terbiasa memiliki pola pikir untuk menyelesaikan perkerjaan di tempat mereka bekerja bukan di

rumah, serta gangguan yang datang dari keluarga di rumah. Namun dibalik dampak negatif tersebut, *Work From Home* (WFH) memberikan keluasan kepada karyawan untuk bekerja secara fleksibel dalam hal waktu. Mereka dapat dengan bebas merencanakan waktu untuk kehidupan sosial mereka dan waktu untuk bekerja. (Mustajab dkk., 2020)

Untuk dapat memanfaatkan fleksibilitas ini, HC memerlukan solusi baru yang mendukungnya. Proses *training* secara daring dilaksanakan sesuai dengan jadwal yang ditentukan. Hal ini tidak mendukung fleksibilitas waktu yang karyawan perlukan saat melaksanakan *training* secara daring. Dengan kondisi yang dialami karyawan CTI, mereka perlu solusi yang bisa memenuhi fleksibilitas waktu karyawan saat tidak berada di tempat kerja.

Dari segi pengelolaan data dan informasi, pengembangan *e-learning* dapat membantu proses *training* karyawan dengan mengelola berbagai media pembelajaran pada satu platform seperti penelitian yang dilakukan Mongi dan Hendry. Mereka melakukan pengembangan *e-learning* dengan platform Moodle. Platform Moodle merupakan sebuah *software open-source* yang menyediakan fitur-fitur untuk melakukan pembelajaran (*learning*). Fitur yang ditekankan pada penelitian ini yaitu materi, kuis, dan konten interaktif dengan h5p. Mereka juga membuat kuesioner untuk menilai penerapannya. Hasilnya penilaian keseluruhan menunjukkan di angka 82.25% yang berarti penerapan *e-learning* untuk proses *training* karyawan berada pada kategori baik. Pengembangan *e-learning* ini kelebihannya pada fitur yang lengkap untuk mendukung *learning* karyawan dan tampilan yang mudah digunakan di perangkat *Desktop* maupun *Mobile*. (Mongi & Hendry, 2021)

Penggunaan *e-learning* dapat meningkatkan kreativitas dan pemahaman karyawan saat melaksanakan *training*, serta dapat meningkatkan produktivitas karyawan. Kalimat tersebut dibuktikan pada penelitian yang dilakukan oleh Rijal dan Sholihah. Mereka melakukan penelitian deskriptif kualitatif penerapan *e-learning* pada suatu perusahaan. Beberapa responden yang menjadi sampel penelitian memberikan pendapatnya terkait topik yang dibahas pada penelitian ini. Responden tersebut merupakan karyawan yang bekerja pada perusahaan yang menjadi studi kasus penelitian. Mereka mengatakan fitur forum pada *e-learning*

dapat meningkatkan pemahaman terkait materi. Ini karena para peserta dapat menerapkan apa yang mereka pelajari dengan berdiskusi di forum tersebut dan ini dapat meningkatkan pemahaman materi *training*. Mereka saling bertukar informasi dengan peserta lain, menanyakan sesuatu yang masih menjanggal, menjawab pertanyaan peserta lain, ataupun saling bertukar pikiran. Peningkatan kreativitas dapat terjadi karena peserta yang tertarik untuk mencoba berbagai fitur yang disediakan *e-learning*. Satu peserta *training* dengan yang lainnya memiliki cara yang berbeda untuk memahami materi, di sinilah fitur-fitur beragam *e-learning* berperan jika dibandingkan dengan proses *training* dengan seminar langsung. Peningkatan produktivitas tentu dapat terjadi bila menggunakan *e-learning* karena kemudahan dan ketersediaan *e-learning* kapan saja dan di mana saja. (Rijal & Sholihah, 2022)

Penelitian di atas memberikan gambaran kepada penulis terkait masalah lain yang terjadi pada *training* secara daring di CTI. Hal ini menjadi kemungkinan faktor lain karyawan yang mengikuti peserta *training* secara daring terlihat tidak aktif karena mereka kesulitan dalam memahami materi. Ini juga bisa jadi disebabkan oleh pembawa materi yang mungkin tidak mampu dalam membuat peserta paham akan materi yang disampaikan. Dalam paragraf di atas, penelitian tersebut menunjukkan bahwa pengembangan *e-learning* membantu untuk meningkatkan pemahaman peserta terkait materi *training*. Mereka tidak harus selalu bergantung pada pembawa materi untuk memahami materi *training*. Tapi, mereka juga dapat memahami materi *training* dengan memanfaatkan fitur-fitur dari *e-learning*.

*E-learning* membantu dalam pelaksanaan *training* karyawan di suatu perusahaan dan proses pembelajaran siswa pada bimbingan belajar. *Training* karyawan biasanya dilakukan dengan mengadakan seminar dan memberikan *hardcopy* buku materi, dengan cara ini mentor *training* berhasil untuk menyampaikan materi bahan *training*. Namun, mentor tidak dapat mengukur pemahaman peserta, di sini *e-learning* membantu mentor untuk mengukur pemahaman peserta *training*. (Yusriyanto, 2019) *E-learning* juga membantu proses pembelajaran pada tempat bimbingan belajar. *E-learning* membantu dengan mendaftarkan siswa yang ingin mengikuti program bimbingan belajar,

mendapatkan modul materi *online*, dan mengunggah tugas ke mentor kapan saja dan di mana saja. (Effendey, 2018)

*E-learning* tidak hanya membantu sebagai media untuk membaca materi pelajaran secara daring dan media untuk memfasilitasi kegiatan tes kemampuan saja. *E-learning* dapat memenuhi 4 model pada pendekatan metode *Computer Assisted Instruction* (CAI), yaitu *drills and practices, tutorials, simulations*, dan *games*. *E-learning* dengan menggabungkan metode ini dapat membuat proses pembelajaran menjadi dua arah, di mana para peserta dapat mempresentasikan langsung solusi dari jawaban yang diberikan atau mereka dapat mengisi tutorial secara langsung pada tes yang interaktif. Penerapan *e-learning* dengan pendekatan metode CAI ini membuat proses pembelajaran jadi lebih mudah dan menambah daya tarik peserta *e-learning* untuk mengikuti materi yang disediakan. (Agustin, Anggraini, & Syarifah, 2018)

Berdasarkan pembahasan di paragraf sebelumnya, penulis membuat solusi *e-learning* untuk HC CTI. Namun, penulis tidak dapat mengembangkannya menjadi satu aplikasi. Seperti yang telah dibahas sebelumnya, CTI mengembangkan satu solusi digital untuk setiap proses kerja yang membutuhkannya sedangkan satu divisi di CTI saja seperti HC memiliki berbagai proses kerja yang berbeda. Hal ini merumitkan karyawan karena setiap kali mereka ingin berpindah melakukan proses kerja lain, mereka harus berpindah dari satu aplikasi ke aplikasi lainnya. Ditambah setiap aplikasi yang berjalan sendiri ini memiliki tempat penyimpanan datanya masing-masing. Hal ini membuat pengelolaan solusi digital yang telah dibuat CTI menjadi rumit. Inilah sebabnya HC membutuhkan pengembangan solusi *e-learning* pada solusi digital yang telah mereka miliki.

HC telah memiliki solusi digital yang mencakup beberapa proses di lingkungan kerja mereka, yaitu HC Portal. Aplikasi ini akan terus dikembangkan seiring bertambahnya kebutuhan di lingkungan kerja. Mereka ingin memusatkan seluruh proses kerja HC dalam satu aplikasi ini. *Training* karyawan adalah salah satu proses kerja yang ingin mereka tambahkan ke dalam HC Portal.

HC juga memiliki kebutuhan agar pengembangan aplikasi HC Portal bisa terus dilanjutkan. Kebutuhan ini bertujuan agar aplikasi dapat terus memenuhi kebutuhan di lingkungan kerja, dan pengembangan solusi digital selanjutnya tidak

harus mengembangkan satu aplikasi sendiri. Hal ini masih berkaitan dengan masalah sebelumnya. Pengembangan yang berkelanjutan ini memerlukan metode pengembangan *software* dan *tools* yang tepat untuk mendukungnya. (Jadhav, 2020)

Konsep *Agile Development* mungkin adalah jawaban yang tepat untuk memenuhi pengembangan *software* yang berkelanjutan (*Continuous Software Development*). Hal ini karena tahapan iterasi yang terdapat pada *Agile Development*. Berbeda dengan *waterfall*, tahapan-tahapan besar yang ada di *waterfall* menjadi tahapan-tahapan kecil di *Agile*. Satu iterasi pada pengembangan *Agile* mengembangkan satu fitur *software*. Fitur *software* dikembangkan sedikit demi sedikit dalam iterasi demi iterasi hingga menjadi suatu *software* yang utuh. Inilah yang membuat *Agile Development* mampu menjawab tantangan pengembangan *software* yang berkelanjutan. (Agilealliance, 2015; Black, 2018)

Sayangnya penulis tidak dapat menerapkan konsep pengembangan *software* ini ke dalam penelitian akademis. Hal ini karena konsep pengembangan tersebut tidak memperhatikan dokumentasi untuk pengembangan *software*. Konsep pengembangan ini hanya dapat dilakukan untuk melaksanakan proyek pengembangan *software* saja tanpa melakukan analisis sistem lebih lanjut terlebih dahulu. (Wells, 2013) Penulis memerlukan metode pengembangan yang dapat digunakan untuk penelitian akademis dan dapat memenuhi pengembangan *software* yang berkelanjutan (*Continuous Software Development*).

Dari penjelasan kedua paragraf di atas, penulis dapat menyimpulkan bahwa untuk melaksanakan pengembangan *software* yang berkelanjutan memerlukan metode pengembangan *software* yang mengedepankan iterasi dalam pengembangan *software*. Tidak hanya itu, penulis juga memerlukan metode pengembangan yang tidak melupakan analisis sistem untuk keperluan penelitian akademis. Penulis menemukan metode pengembangan yang tepat untuk kebutuhan ini, yakni *iterative model*.

*Iterative Model* sebenarnya adalah versi mini dari metode *Waterfall*. Perbedaannya terletak dari bagaimana pendefinisian sebuah *software* untuk dikembangkan. *Waterfall* mendefinisikan *software* menjadi sebuah produk yang besar dan utuh. Itulah mengapa *Waterfall* memerlukan tahapan perencanaan, analisis, dan desain sistem yang sangat kompleks. *Iterative Model* mendefinisikan

kebutuhan *software* dari yang paling dibutuhkan terlebih dahulu. Ini karena adanya kebutuhan agar *software* dapat cepat masuk ke dalam lingkungan produksi dan bisa dicoba oleh para pengguna. Pengembangan dengan *Iterative Model* memungkinkan developer *software* untuk membagi pengembangannya dalam beberapa iterasi. Setiap iterasi yang telah selesai, developer dapat melakukan reviu dan menerima umpan balik dari pengguna untuk iterasi pengembangan fitur selanjutnya. (Alshamrani & Bahattab, 2015; Yaseen, Ibrahim, & Mustapha, 2019)

Seperti yang telah dijelaskan sebelumnya, penulis juga memerlukan *tools* untuk melakukan pengembangan *software* yang berkelanjutan. Penggunaan Github untuk membuat repositori digital dapat memenuhi kebutuhan ini. Repositori digital bukan sekedar berupa tempat penyimpanan artefak kode, melainkan juga sebuah lingkungan pengembangan untuk mencatat segala perubahan terhadap *software* ataupun berdiskusi dengan developernya secara langsung. Github menggunakan teknologi Git untuk mencatat segala perubahan yang terjadi pada pengembangan *software*. Setiap kali developer *software* melakukan perubahan, dia akan melakukan *commit* untuk mencatat tugas apa yang telah dia lakukan. Fitur *issues* dan *pull-request* juga membantu mencatat kebutuhan dan log pengembangan *software*. Developer *software* yang akan melanjutkan pengembangannya dapat mempelajari proses pengembangan sebelumnya melalui catatan pada fitur-fitur ini. Inilah yang membantu pengembangan *software* yang berkelanjutan. (Github, 2020; Kalliamvakou, Damian, Blincoe, Singer, & German, 2015)

Penulis melakukan rancang bangun *e-learning* dan membuat repositori digital dengan Github. Metode pengembangan *software* yang penulis ambil untuk mengembangkan *e-learning* ini adalah *iterative model*. Metode ini sejalan dengan kebutuhan HC untuk melakukan pengembangan HC Portal secara terus menerus dengan menambahkan fitur baru setiap kali ada proses baru yang ingin ditambahkan. Kebutuhan ini difasilitasi dengan iterasi yang terdapat pada *Iterative Model*. Repositori digital yang dibuat dalam penelitian ini juga bertujuan untuk mendukung kebutuhan HC akan pengembangan aplikasi HC Portal secara terus menerus. Inilah yang membuat penulis mengangkat judul penelitian ini yaitu **“Rancang Bangun e-Learning Dan Repositori Digital dengan Platform Github”**.

## 1.2 Identifikasi Masalah

Identifikasi masalah berikut dijabarkan berdasarkan pada pembahasan latar belakang. Ada beberapa masalah yang ditemukan:

1. Karyawan HC kesulitan membuat laporan hasil *training* karena data dan informasi terkait proses *training* tersebar di beberapa tempat penyimpanan seperti di komputer lokal karyawan dan penyimpanan *cloud*.
2. Sulitnya melaksanakan proses *training* secara luring karena persiapannya membutuhkan banyak waktu dan biaya. Juga, tidak semua karyawan CTI dapat mengikuti proses *training* secara daring dengan baik karena selalu saja ada karyawan yang tidak berada di lingkungan yang kondusif saat mengikuti proses *training* secara daring.
3. Sulitnya mengelola solusi-solusi digital yang telah dibuat oleh CTI karena pengembangan satu solusi digital di CTI hanya mencakup satu proses kerja sedangkan di satu divisi saja seperti HC memiliki berbagai proses kerja yang berbeda.

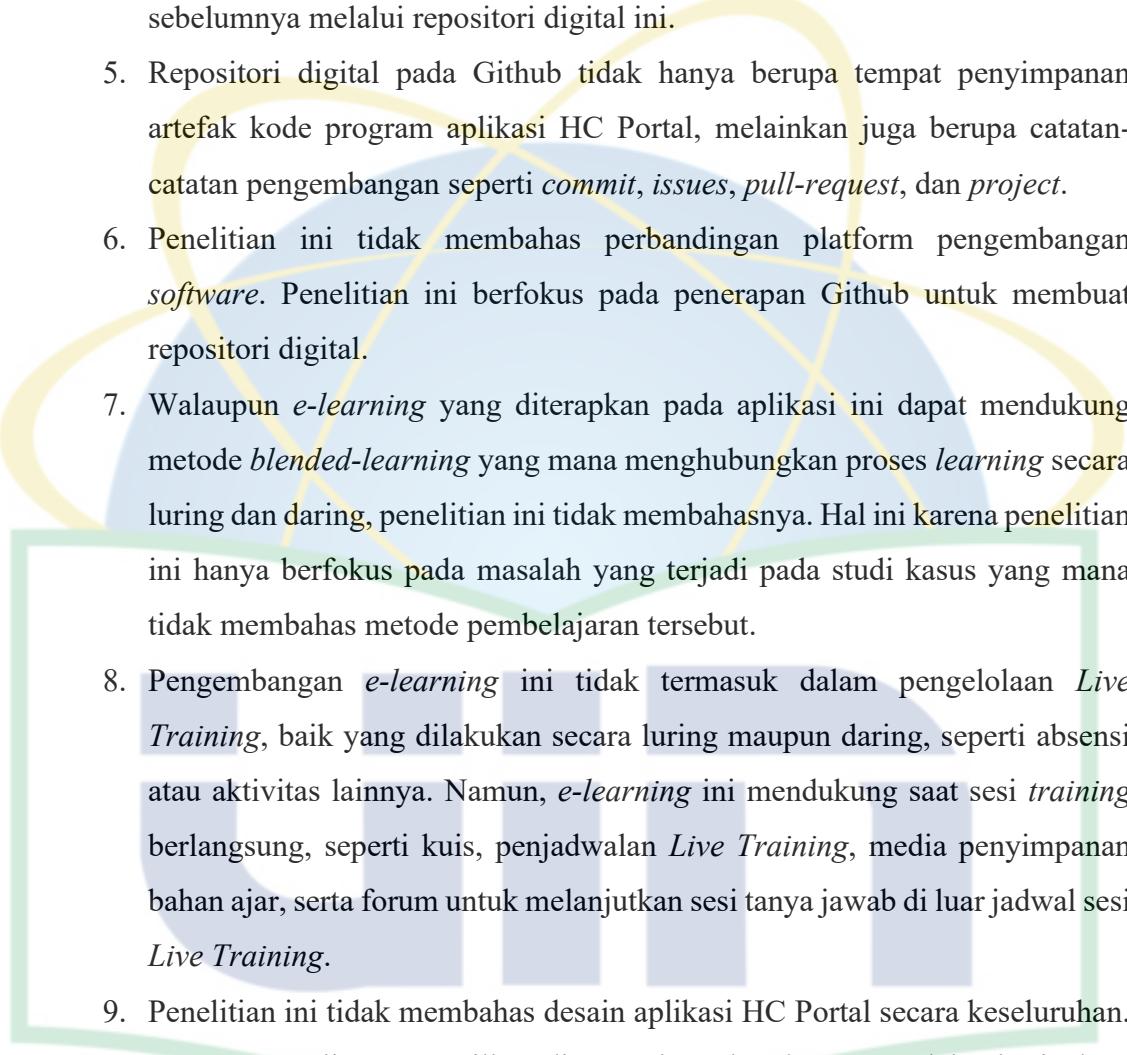
## 1.3 Rumusan Masalah

Rumusan masalah dijabarkan berdasarkan identifikasi masalah di atas. Adapun rumusan masalah pada penelitian ini, yaitu bagaimana merancang dan membangun *e-learning* pada aplikasi HC Portal dan membuat repositori digital dengan platform Github?

## 1.4 Batasan Masalah

Batasan masalah berikut dijabarkan untuk menghindari perluasan masalah yang dibahas pada penelitian ini, adapun batasan masalah penelitian ini:

1. Penelitian ini dilakukan di CTI tepatnya di divisi HC.
2. Penulis melakukan pengembangan aplikasi *e-learning* sebagai modul pada aplikasi web HC Portal.
3. Metode pengembangan *software* yang digunakan pada penelitian ini adalah *iterative model*. Metode pengembangan *software* ini terdiri dari iterasi per iterasi. Penggunaan metode ini bertujuan untuk mendukung pengembangan *software* yang berkelanjutan. CTI dapat menjalankan iterasi pengembangan selanjutnya dengan melihat cara penulis melakukannya di penelitian ini.

- 
4. Penulis menggunakan Github untuk membuat repositori digital pengembangan HC Portal. Hal ini digunakan juga untuk mendukung pengembangan *software* yang berkelanjutan pada aplikasi HC Portal dengan mencatat segala perubahan yang terjadi saat pengembangan berlangsung. Developer HC Portal selanjutnya dapat mempelajari pengembangan sebelumnya melalui repositori digital ini.
  5. Repositori digital pada Github tidak hanya berupa tempat penyimpanan artefak kode program aplikasi HC Portal, melainkan juga berupa catatan-catatan pengembangan seperti *commit*, *issues*, *pull-request*, dan *project*.
  6. Penelitian ini tidak membahas perbandingan platform pengembangan *software*. Penelitian ini berfokus pada penerapan Github untuk membuat repositori digital.
  7. Walaupun *e-learning* yang diterapkan pada aplikasi ini dapat mendukung metode *blended-learning* yang mana menghubungkan proses *learning* secara luring dan daring, penelitian ini tidak membahasnya. Hal ini karena penelitian ini hanya berfokus pada masalah yang terjadi pada studi kasus yang mana tidak membahas metode pembelajaran tersebut.
  8. Pengembangan *e-learning* ini tidak termasuk dalam pengelolaan *Live Training*, baik yang dilakukan secara luring maupun daring, seperti absensi atau aktivitas lainnya. Namun, *e-learning* ini mendukung saat sesi *training* berlangsung, seperti kuis, penjadwalan *Live Training*, media penyimpanan bahan ajar, serta forum untuk melanjutkan sesi tanya jawab di luar jadwal sesi *Live Training*.
  9. Penelitian ini tidak membahas desain aplikasi HC Portal secara keseluruhan. Namun, penulis menampilkan diagram konteks *class*, pemodelan basis data, serta beberapa *screenshot* kondisi HC Portal sebelum penulis melakukan pengembangan *e-learning* di aplikasi tersebut. Desain aplikasi yang penulis buat pada penelitian ini hanya mencakup pada modul aplikasi *e-learning* saja.
  10. Penelitian ini tidak membahas keamanan sistem pada *software* yang telah dikembangkan pada penelitian ini.
  11. *Tools* yang digunakan untuk mendesain sistem yaitu UML. Diagram yang digunakan pada UML meliputi *Use Case Diagram*, *Activity Diagram*, *Class*

*Diagram*, dan *Sequence Diagram*. Penulis juga membuat Skema Basis Data untuk mendesain basis data yang digunakan pada aplikasi ini. Penulis menggunakan aplikasi daring draw.io untuk membuat diagram tersebut.

12. *Tools* yang digunakan untuk merancang tampilan aplikasi (*User Interface*) yaitu Figma.
13. Penulis menggunakan *framework* CodeIgniter versi 3.1.11 untuk integrasi *e-learning* dengan aplikasi HC Portal yang menggunakan *framework* tersebut. Penulis menggunakan MySQL versi 7.5.22 sebagai basis data aplikasi. Penulis juga menggunakan JQuery versi 3.5.1 untuk membuat tampilan aplikasi ini menjadi lebih interaktif.
14. Waktu yang terbatas untuk mengerjakan penelitian ini membuat penulis tidak dapat mengimplementasikan semua fitur *software* dari semua desain UML yang telah dibuat.

## 1.5 Tujuan Penelitian

Secara umum, penelitian ini bertujuan untuk merancang dan membangun *e-learning*, serta membuat repositori digital dengan platform Github. Sedangkan secara khusus, penelitian ini bertujuan untuk:

1. Mempermudah proses pembuatan laporan hasil *training* dengan membuat platform untuk mewadahi proses *training* karyawan.
2. Mempermudah persiapan *training* karyawan dan membuat solusi baru *training* karyawan yang lebih fleksibel dalam hal waktu.
3. Mengembangkan solusi penelitian ini pada HC Portal, serta melakukan pengembangan *software* berbasis iterasi dengan *iterative model* dan membuat repositori digital dengan platform Github untuk mendukung kebutuhan akan pengembangan HC Portal yang berkelanjutan.

## 1.6 Manfaat Penelitian

Penelitian ini memiliki manfaat untuk kepentingan akademik dan untuk menyelesaikan permasalahan yang terjadi pada CTI. Adapun manfaat penelitian ini dijabarkan sebagai berikut:

1. Menambah wawasan dan pengetahuan penulis serta sebagai arsip akademik untuk universitas pada ilmu pengetahuan yang membahas tentang rancang

- bangun *e-learning* dan membuat solusi pengembangan *software* yang berkelanjutan dengan pembuatan repositori digital pada platform Github.
2. Menyelesaikan permasalahan *training* karyawan di CTI dan mendukung proses pengembangan HC Portal yang berkelanjutan.
  3. Berperan untuk mengembangkan ilmu pengetahuan terkait rancang bangun *software* serta mendukung pengembangannya yang berkelanjutan dengan pembuatan repositori digital pada platform Github.

## 1.7 Metode Penelitian

### 1.7.1 Metode Pengumpulan Data

#### 1. Observasi

Penulis melakukan observasi dengan melaksanakan program magang dari HC CTI yang terhitung dari tanggal 19 Maret sampai 18 Desember 2020. Penulis melaksanakan program magang tersebut di hari kerja tepatnya di hari Senin – Jumat.

#### 2. Wawancara

Penulis melakukan wawancara dengan HCIS *Specialist*. HCIS *Specialist* merupakan posisi kerja di bawah naungan divisi HC. Posisinya langsung membawahi kepala divisi, sementara untuk hierarkinya sebagai asisten kepala divisi. Penulis melakukan wawancara dengan menanyakan kebutuhan dan urgensi dari pengembangan *e-learning* untuk aplikasi portal HC. Penulis juga membuat draf wawancara untuk menghindari kesalahan pemahaman kebutuhan dalam mengembangkan aplikasi ini. Penulis melaksanakan wawancara ini pada tanggal 7 April 2021 tepatnya pada jam 19.57 – 22.13

#### 3. Studi Literatur

Penulis melakukan pencarian dan pembelajaran dari berbagai macam teori, literatur, internet, buku, dan jurnal relevan yang menunjang penyejaan penelitian ini khususnya yang berkaitan dengan rancang bangun *e-learning* dan pembuatan repositori digital dengan platform Github.

### 1.7.2 Metode Pengembangan *Software*

Penulis menggunakan metode pengembangan *Iterative Model* dengan mengedepankan iterasi dalam pengembangan *software*. Metode ini sesuai untuk

pengembangan *software* yang bertahap dan terus menerus. Metode ini merupakan salah satu dari Siklus Hidup Pengembangan Sistem (*System Development Life Cycle* yang disingkat SDLC). Metode ini sebenarnya adalah versi mini dari metode Waterfall dengan ditambahkannya tahapan masukan (*feedback*) dari satu fase iterasi. Masukan ini menyediakan informasi yang penting untuk proses desain pada fase iterasi berikutnya. Metode ini menghasilkan produk perangkat lunak di akhir dari setiap tahapan iterasi dan dapat langsung menuju ke lingkungan sistem yang sebenarnya sebagai rilis tambahan (*Incremental Releases*). (Alshamrani & Bahattab, 2015)

Metode ini memiliki dua fase yaitu fase *requirement* (kebutuhan) dan *iteration* (iterasi). Fase *iteration* terdiri dari empat tahapan, yaitu (1) *Analysis & Design* (Analisis & Desain), (2) *Implementation/Development* (Implementasi/ Pengembangan), (3) *Testing & Revise* (Testing & Revisi), dan (4) *Production* (Produksi). (Alshamrani & Bahattab, 2015; Yaseen dkk., 2019)

Penulis menggunakan UML (*Unified Modelling Language*) diagram untuk memperlihatkan aliran proses dan data, serta aktor yang akan menggunakan masing-masing modul aplikasi.

## 1.8 Sistematika Penulisan

Penulis menyusun laporan penelitian ini dalam lima bab yang akan diuraikan sebagai berikut,

### BAB I PENDAHULUAN

Bab ini berisi penjelasan latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat penelitian, metode penelitian yang akan dipakai, dan sistematika penulisan.

### BAB II LANDASAN TEORI

Bab ini membahas mengenai dasar-dasar teori, istilah, dan penjelasan terkait pengetahuan yang digunakan pada penelitian ini.

### BAB III METODE PENELITIAN

Bab ini membahas tentang metode yang digunakan dalam penelitian ini. Penjelasan metode yang dibahas yaitu metode pengumpulan data dan metode pengembangan aplikasi (*software*).

## **BAB IV HASIL DAN PEMBAHASAN**

Bab ini membahas hasil penelitian yang telah dilakukan oleh penulis dengan membahas hasil data yang diperoleh dari metode pengumpulan yang dilakukan. Bab ini juga menguraikan proses perancangan dan pengembangan aplikasi pada penelitian ini.

## **BAB V PENUTUP**

Bab ini berisi kesimpulan yang berkenaan dengan hasil pemecahan masalah serta beberapa saran untuk pengembangan.

## **DAFTAR PUSTAKA**





## **BAB II**

### **LANDASAN TEORI**

Bab ini membahas teori-teori yang digunakan oleh penulis untuk melakukan penelitian. Penulis dapat melaksanakan penelitian dengan tepat apabila memahami bagaimana teori pada topik ini. Pemilihan teori pada bab ini disesuaikan dengan pembahasan pada topik yang di bahas.

#### **2.1 Sistem Informasi**

Sebelum membahas definisi dari Sistem Informasi, kita akan membahas definisi Sistem dan Informasi terlebih dahulu. Kemudian baru kita akan membahas tentang Sistem Informasi itu sendiri dan komponennya.

##### **2.1.1 Pengertian Sistem**

Sistem merupakan kumpulan berbagai macam komponen yang membentuk kesatuan. Komponen-komponen ini saling mendukung satu sama lain untuk mencapai suatu tujuan tertentu. Sebuah organisasi dan sistem informasi adalah contoh dari sistem sosial dan fisik. Sistem memiliki batasan-batasan tertentu dalam pengoperasiannya. Hal ini bertujuan agar komponen-komponen yang berada di dalam sistem dapat berjalan sesuai dengan definisi dari tujuan sistem itu dibuat, sehingga proses yang berada pada sebuah komponen harus diatur dalam aturan tertentu. Komponen yang berada pada sistem saling terkait satu sama lain. Setiap komponen bisa saja memiliki sub komponen yang lebih kecil untuk membantu dalam melakukan proses besar yang harus ditangani dalam satu induk komponen.

Penentuan batasan sistem harus dilakukan secara seksama dan hati-hati. Pembuatan suatu sistem tidak akan lepas dengan yang namanya komponen, sehingga perlu untuk menentukan komponen-komponen apa yang akan mencakup pada suatu tujuan sistem. Penentuan komponen ini akan menemukan batasan sistem yang akan membantu untuk mencegah pelebaran tujuan awal dari sistem dibuat. Pembuatan suatu sistem juga memerlukan pemikiran secara berulang terkait tujuan atau sasaran yang ingin dicapai sistem, seperti tipe transaksi apa yang didukung. Pada akhirnya akan muncul keputusan apa yang akan masuk dan keluar dari suatu sistem. (Tyoso, 2016)

### 2.1.2 Pengertian Informasi

Informasi merupakan suatu data yang telah diolah dan ditujukan untuk seseorang, organisasi, atau siapa saja yang membutuhkannya. Pengolahan suatu informasi dapat terjadi apabila ada objek yang melakukan permintaan atas informasi tersebut. Permintaan informasi ini kemudian memicu beberapa proses pengolahan data yang dimulai dari proses pengumpulan, penyortiran, penggabungan, pemisahan, hingga ke proses penyimpulan untuk memberikan suatu informasi atas data yang telah didapatkan. (Mulyani, 2017)

Informasi dalam artian lain merupakan suatu pertambahan ilmu pengetahuan yang menyumbangkan konsep kerja yang umum dan penjelasan atas fakta-fakta yang ditemukan. Sebagai contoh, kata “lima” bukan berarti Anda hanya ingin mengetahui kalau itu lima ribu rupiah, tetapi itu juga bisa berarti hal lain seperti rata-rata nilai pada suatu kelas, urutan bilangan ganjil kelima, dan masih banyak lagi. Hal ini bergantung pada konteks dan pengetahuan umum si penerima informasi tersebut. (Tyoso, 2016)

Suatu informasi dapat bernilai untuk seseorang, namun belum tentu berarti bagi orang lain. Hal ini sangat bergantung pada kualitas informasi tersebut. Jika kita menerima informasi yang tidak berkaitan dengan kita maka informasi tersebut akan menjadi tidak berguna. Apabila kita menerima informasi yang sudah kita tidak kita butuhkan saat ini, sementara kita membutuhkan informasi ini sekitar beberapa hari yang lalu, maka informasi yang kita terima sudah usang (*outdated*). Dan apabila kita tidak dapat mengerti informasi apa yang kita terima, maka informasi itu menjadi tidak bernilai. Ada tiga dimensi ukuran untuk menentukan suatu kualitas informasi: (Romindo dkk., 2020)

1. Akurat, suatu informasi harus tepat, tidak bias, bebas dari kesalahan, tepat mewakili objek, dan kebenarannya tidak diragukan.
2. Tepat pada waktunya, berkaitan dengan kapan informasi diperoleh dan disajikan. Informasi harus tersedia dengan waktu yang dibutuhkan, tidak boleh terlambat karena hal ini akan terkait dengan kebaharuan informasi.
3. Relevan, informasi sesuai dengan keadaan yang terjadi sehingga akan memberikan manfaat bagi penerimanya. Hal ini berkaitan dengan level ataupun fungsional penerima informasi.

### **2.1.3 Pengertian Sistem Informasi**

Sistem Informasi (SI) adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian pada fungsi operasi organisasi dengan kegiatan strategi organisasi. Tujuannya adalah untuk menyediakan informasi kepada pihak tertentu, baik internal maupun eksternal, untuk pengambilan keputusan. SI menyediakan informasi kepada seluruh tingkatan dalam organisasi kapan saja saat diperlukan. SI mengambil, menyimpan, mengubah tanpa menghilangkan arti dari informasi, mengolah, dan mengomunikasikan informasi yang diterima dengan berbagai macam komponen yang membentuk SI tersebut. (Anggraeni, 2017)

### **2.1.4 Komponen Sistem Informasi**

SI Sebagai salah satu penerapan sistem, pastinya memiliki komponen-komponen yang membentuknya. Adapun beberapa komponen yang terdapat pada suatu SI: (Anggraeni, 2017)

- **Perangkat Keras (*Hardware*)**

Sumber daya *hardware* merupakan semua peralatan fisik yang digunakan untuk memproses informasi. *Hardware* dapat bekerja dengan perintah yang sudah ditentukan sebelumnya, perintah ini disebut *instruction set*. Komputer secara fisik memiliki beberapa komponen yang saling bekerja sama menjadi suatu kesatuan. Komputer tidak akan berjalan dengan baik apabila ada satu komponen yang tidak bekerja. Perangkat keras komputer secara fungsinya dibagi menjadi Unit Masukan (*Input Device*; Keyboard, Mouse, Touchpen, Lightpen, dan Joystick), Unit Pemrosesan (*Process Device*; PSU (*Power Supply Unit*)), RAM (*Random Access Memory*), VGA (Kartu Grafis), Prosesor, dan Motherboard), Unit Keluaran (*Output Device*; Monitor, Printer, dan Sepiker/*Speaker*), Unit Penyimpanan (*Backing Storage*; HDD/*Hard Disk Drive*, SSD/*Solid State Drive*, Flashdisk, CD/DVD ROM, dan disket/*Floppy Drive*), Unit Tambahan (*Peripheral Device*; Kartu Suara (*Soundcard*), Modem (untuk koneksi internet), perangkat WIFI (untuk koneksi tanpa kabel), UPS/*Uninterrupted Power Supply*, CD Optik/*Optical Disk Drive*)

- **Perangkat Lunak (*Software*)**

Perangkat Lunak memiliki tiga jenis yang dibagi berdasarkan fungsinya, di antaranya yaitu Sistem Operasi (*Operating System*; seperti Microsoft Windows, Linux, dan Apple MacOS), bahasa pemrograman (PHP, Java, Pascal, Delphi, C, C++, C#, ADA, JavaScript, Dart, Go, Python, dll.), dan program aplikasi (pengolah kata, pengolah angka, pengolah grafis, multimedia, dll.). Berikut penjelasan masing-masing jenis perangkat lunak:

- **Prosedur**

Prosedur merupakan sekumpulan aturan yang digunakan untuk mewujudkan proses pengolahan informasi dalam suatu organisasi. Aturan yang dibuat meliputi aktivitas sistem input, pemrosesan keluaran (*output*), aturan dalam penyimpanan berkas, dan kontrol informasi. (Sudirman dkk., 2020)

- **Orang**

Sumber daya orang merupakan semua pihak yang terlibat dalam pengelolaan sistem informasi, pemrosesannya, dan penggunaan hasil dari proses pengolahan informasi. Sumber daya orang sebagai salah satu komponen sistem informasi meliputi Operator, Analis Sistem (*System Analyst*), Pengembang Aplikasi (*Software Engineer*), Administrator Basis Data (DBA), Teknisi Jaringan dan Perawatan Sistem, Web Master, dan Auditor PDE. Masing-masing dari sumber daya orang ini memiliki tugas yang berbeda-beda, walaupun begitu mereka memiliki tujuan yang sama yaitu membuat sistem informasi yang berguna untuk membantu dalam melakukan pekerjaan di dalam organisasi.

- **Basis Data (*Database*)**

Sumber daya basis data merupakan kumpulan informasi yang disimpan di dalam komputer secara sistematik. Penyimpanan data dalam Basis Data biasanya disimpan dalam bentuk kolom dan baris. Kolom menyimpan jenis informasi yang disimpan pada satu baris data. Setiap baris data memiliki identitas unik (ID) yang digunakan untuk membedakan baris data satu dengan lainnya. Basis data memiliki perangkat lunak yang digunakan untuk mengelolanya yang disebut Sistem Manajemen Basis Data (DBMS). Pengelolaan pada basis data menggunakan bahasa

instruksi yang disebut SQL (*Structured Query Language*). Contoh DBMS yaitu MySQL, OracleDB, Microsoft SQL Server, dan masih banyak lagi.

- **Jaringan Komputer (*Networking*) dan Komunikasi Data**

Sumber daya jaringan merupakan sistem yang terdiri dari beberapa perangkat komputer yang saling berbagi sumber daya, berkomunikasi, dan berbagi akses informasi satu sama lain. Manfaatnya apabila menggunakan jaringan komputer yaitu memudahkan untuk berbagi sumber daya, pertukaran data antar komputer, berkomunikasi satu dengan yang lain, akses informasi bersama, dan menjaga informasi agar selalu terkini (*up-to-date*). Jenis-jenis jaringan komputer dibagi berdasarkan jangkauannya (LAN, MAN, dan WAN), fungsinya (*client-server* dan *peer-to-peer*), dan Media Transmisinya (berkabel/*wired* dan nirkabel/*wireless*).

## 2.2 Rancang Bangun

Perancangan merupakan proses awal dalam menyusun desain solusi suatu permasalahan. Solusi yang dibuat ini tentu saja merupakan sebuah sistem. Sistem ini dapat membantu meringankan masalah dalam pekerjaan atau sekedar meringankan pekerjaan mereka sehingga mereka dapat melakukan pekerjaan dengan lebih efektif untuk mencapai target yang ditetapkan dalam lingkup organisasi. Biasanya proses perancangan melibatkan beberapa grup pengguna yang terlibat dalam permasalahan tersebut. Pelaksanaan proses perancangan sering kali menyebabkan konflik di setiap langkahnya, karena setiap grup pengguna memiliki keinginan yang berbeda. Apabila solusi dari permasalahan dibuat secara buru-buru dan tidak ada kesepakatan antara pengguna sistem, maka solusi sistem yang dibuat pun tidak akan berguna dan akan terus melanjutkan masalah. (Greenbaum & Kyng, 2020)

Perancangan tidak hanya akan memberikan desain solusi sistem, tetapi juga akan mengubah cara kerja karyawan yang terlibat dalam penerapan solusi sistem tersebut. Perancangan kerja perlu juga dilakukan saat membuat perancangan sistem dengan merencanakan bagaimana untuk melakukan pelatihan penggunaan sistem baru nantinya, mengetahui bagaimana motivasi dan ketertarikan karyawan agar penerapan solusi sistem dapat berjalan dengan maksimal. Perancangan sebuah sistem tidak hanya melibatkan perancangan perangkat lunak, tetapi juga melibatkan

berbagai komponen-komponen lain dalam organisasi. Bisa jadi organisasi akan melakukan pengadaan komponen baru, mengubah komponen yang sudah ada, ataupun menghilangkan komponen lain yang tidak digunakan. (Munawar, 2018)

Pembangunan perangkat lunak yang biasa disebut dengan pengembangan perangkat lunak (*software development*) merupakan tahapan berikutnya setelah melakukan perancangan. Para manajer yang terlibat dalam perancangan sistem membuat desain level tinggi (*high level design*) terkait dengan desain keseluruhan hingga ke fungsi-fungsi dari sistem yang akan dibuat, sedangkan *programmer* mulai membuat desain level rendah (*low level design*). *Programmer* membuat desain yang lebih detail dari *high level design*. Mereka terus melakukan desain secara detail sampai mereka tahu bagaimana caranya untuk mengimplementasikan desain tersebut menjadi kode program dengan mudah. *Programmer* kemudian melakukan pengembangan dengan melakukan beberapa tahapan dalam pengembangan perangkat lunak (*Software Development Methods*) setelah menyelesaikan desain *low level*. Mereka terus melakukan debug terhadap kesalahan (*bug*) program se bisa mungkin, pada kenyataannya tidak ada program yang terbebas dari kesalahan. (Stephens, 2015)

### 2.3 Perangkat Lunak (*Software*)

Perangkat lunak membantu kita untuk membuat, mengakses, dan memvisualisasikan informasi dengan cara yang tidak terbayangkan sebelumnya, dengan kemampuan ini banyak perusahaan mulai berbondong-bondong untuk mengembangkan perangkat lunak baik untuk membantu pekerjaan mereka ataupun menggantikan proses kerja yang lama dengan melibatkan perangkat lunak untuk pemrosesan informasi. Perangkat lunak pada dasarnya dirancang untuk membantu dalam menyelesaikan suatu permasalahan yang terjadi pada proses kerja yang konvensional. Pengembangan perangkat lunak tidak perlu untuk dilakukan apabila proses kerja saat ini dirasa masih cukup efektif dan dapat membantu mencapai target kerja karena pengembangan perangkat lunak memerlukan sumber daya tambahan, konsepnya seperti investasi untuk masa depan. Perkembangan perangkat lunak yang sangat pesat di berbagai sektor industri ini juga mendorong akan kebutuhan pengembang perangkat lunak (*software developer*). (Munawar, 2018)

Terdapat 2 jenis perangkat lunak dalam dunia teknologi komputer, yaitu perangkat lunak sistem (*Software/System Software*) dan perangkat lunak aplikasi (*Application Software*). Perangkat lunak (*Software*) awalnya hanya tersedia pada perangkat keras (*hardware*) tertentu. Perangkat lunak terdiri dari berbagai instruksi kode untuk *hardware* agar fungsionalitas pada perangkat keras tersebut dapat berjalan. *Hardware* akan menjadi sia-sia secanggih apapun apabila tidak ada perangkat lunak yang menginstruksikan bagaimana ia harusnya berjalan. Seiring dengan berkembangnya teknologi komputer, perangkat lunak yang langsung terhubung dengan *hardware* bisa membuat sebuah produk yang lebih spesifik lagi, yaitu perangkat lunak aplikasi (*application software/application*). Aplikasi bisa menjalankan tugas spesifik sampai untuk keuntungan suatu individu, organisasi, ataupun bisnis. Program ini tidak langsung terhubung dengan *hardware*, melainkan berjalan di atas perangkat lunak lain. Kedua perangkat lunak ini kemudian dibedakan menjadi perangkat lunak sistem (*system software*) yang bisa berjalan pada *hardware* tertentu, dan perangkat lunak aplikasi (*application software*). (Kravchuk, 2021)

### 2.3.1 Perangkat Lunak Aplikasi (*Application Software*)

*Application software* didesain untuk menjalankan tugas spesifik sampai kepada keuntungan suatu individu, organisasi, ataupun bisnis seperti yang telah dijelaskan sebelumnya. Beberapa *software* memiliki fungsi yang cukup umum yang dapat digunakan untuk berbagai kalangan, ada juga yang memiliki fungsi khusus untuk beberapa pengguna tertentu. *Application software* yang umum (*generic application software*) seperti program pengolah kata (*word processing*), pengolah angka (*spreadsheet*), basis data (*databases*), server email, dan tipe program lain yang bisa diakses secara masal. Sementara itu, *Application software* yang dibuat khusus (*custom application software*) memiliki tugas yang lebih spesifik untuk pengguna ataupun kelompok tertentu yang memiliki kebutuhan unik, tentunya hasil (*output*) dari aplikasi ini juga disesuaikan dengan kebutuhan unik pengguna tersebut. (Kravchuk, 2021)

### 2.3.2 Modul Aplikasi (*Application Module*)

Modul Aplikasi mengandung fungsi kecil pada satu aspek kerja aplikasi. Modul aplikasi melakukan beberapa operasi spesifik untuk menghasilkan keluaran

(*output*) sesuai dengan input yang diberikan. Hasil *output* ini dapat digunakan sebagai input pada modul lain. Modul aplikasi mendirikan elemen pemrosesan data pada aplikasi yang berbeda. Mereka saling berbagi sumber daya untuk melakukan pemrosesan informasi. Pembagian fungsi aplikasi menjadi modul ini efektif untuk melakukan pengembangan aplikasi dalam skala besar, sama seperti konsep membagi fungsi aplikasi menjadi komponen-komponen yang lebih kecil. Pengembangan aplikasi akan jauh lebih mudah dengan menerapkan modul aplikasi, tidak hanya itu aplikasi pun akan menjadi lebih kokoh terhadap kecacatan (*bugs*) dan mudah juga untuk melakukan perbaikan karena kita akan lebih mudah mengidentifikasi kerusakan dengan melihat modul yang mengalami kerusakan saja. (Mahmud, Ramamohanarao, & Buyya, 2018)

### 2.3.3 Aplikasi Web (*Web Application*)

Aplikasi Web merupakan aplikasi *software* yang dikembangkan menggunakan teknologi web dan memanfaatkan kemampuan peramban web (*web browsers*) untuk menjalankan berbagai tugas sebagaimana menggunakan aplikasi *software* yang berjalan di atas *system software* melalui internet. Pengembangan aplikasi *software* yang berjalan langsung di atas sistem operasi sudah sangat cukup untuk dapat memberikan solusi sistem yang menjanjikan. Namun, teknologi web memungkinkan untuk perusahaan menggunakan sumber daya internet yang sangat efektif dalam penggunaan biaya. Perusahaan menggunakan internet untuk melakukan transaksi informasi, daripada membuat aplikasi yang terpisah yang berjalan sendiri pada sistem, kenapa tidak langsung membuat aplikasi dengan teknologi web yang menggunakan internet. (Gibb, 2016)

Aplikasi web dapat berjalan menggunakan beberapa kombinasi bahasa pemrograman, bahasa *scripting*, hingga *database query*. Pengembang aplikasi web (*web developer*) biasa membaginya menjadi tiga lapisan (*layer*). Pertama lapisan *frontend*, presentasi (*presentation layer*) merupakan bagian tampilan yang langsung berinteraksi dengan pengguna (*user interface*). Pengembang web (*web developer*) harus memiliki kemampuan untuk memahami beberapa bahasa pemrograman untuk dapat mengembangkan aplikasi web. Pengembangan aplikasi web dalam skala besar biasanya melakukan pembagian tugas atas ketiga lapisan tersebut seperti

pengembang *frontend* (*frontend developer*), *backend* (*backend developer*), dan basis data (*database engineer*). (Oliver, 2020)

Lapisan *frontend* dikembangkan menggunakan HTML dan CSS, serta menggunakan JavaScript yang disematkan untuk membuat aplikasi menjadi interaktif dan menambahkan logika untuk aksi pada sisi klien. Lapisan *frontend* pada pengembangan web mencakup pada pengembangan tampilan pengguna (*user interface*) dan *user experience*. (Oliver, 2020) *User Experience* membahas tentang bagaimana pengalaman pengguna dalam menggunakan aplikasi, ketimbang melihat estetis keindahan *user interface*. (Albert & Tullis, 2013) *Frontend developer* bertugas untuk melakukan pengembangan lapisan ini.

Lapisan *backend* merupakan bagian yang mengatur logika aplikasi web pada sisi server. *Backend* merupakan otak dari aplikasi web yang mengandung berbagai logika bisnis. Logika bisnis ini merupakan proses bagaimana informasi diolah oleh aplikasi. *Backend* bisa menggunakan bahasa *scripting* misalnya PHP, ataupun menggunakan bahasa berorientasi objek (OOP) misalnya Java atau Ruby. *Backend developer* merupakan orang yang bertanggung jawab pada lapisan ini. *Backend Developer* tidak jarang juga menangani basis data karena mereka berhubungan langsung dengan server logika bisnis dan juga mengurus server tempat aplikasi web berjalan, baik itu server *frontend*, *backend*, basis data, jaringan, serta hal lainnya untuk membuat aplikasi web bisa berjalan dengan baik. (Vuorimaa, Laine, Litvinova, & Shestakov, 2016)

## 2.4 *E-Learning*

Kata *e-learning* atau pembelajaran elektronik saat ini digunakan pada berbagai cara penyajiannya. Aktivitasnya dilakukan dengan komputer melalui berbagai cara, terkadang melalui dunia *World Wide Web* (WWW) dan terkadang menggunakan lingkungan pembelajaran virtual (*Virtual Learning Environment* /VLE) atau melalui Jejaring Sosial (*Digital Social Network/DSN*). (Rice & McKendree, 2013)

Pengembangan *e-learning* sejalan dengan perkembangan web yang dimulai dari era web 1.0. Era ini merupakan masa di mana konten-konten informasi yang terdapat pada buku fisik dimasukkan ke halaman web yang terstruktur. Semua orang yang membutuhkan materi pembelajaran dapat mengakses web tersebut.

Selang berapa saat muncul istilah *Computer-aided learning*. Kemunculan istilah ini membuat *e-learning* berkembang dengan penambahan fungsi latihan yang interaktif, tidak hanya mengandalkan informasi teks dan gambar saja. Pelajar disuguh dengan latihan yang terstruktur dan berulang seiring dengan materi informasi yang diberikan. *E-learning* dengan teknik simulasi juga dikenalkan pada era ini. Simulasi dilakukan dengan memberikan contoh dalam kehidupan nyata, kemudian contoh tersebut dituangkan dalam latihan. (Rice & McKendree, 2013; Rubens, Kaplan, & Okamoto, 2014)

Era web 2.0 merupakan era bagi semua orang untuk dapat saling berkomunikasi dan berkolaborasi. Era ini memunculkan banyak platform media sosial. Beberapa di antaranya yang terkenal yaitu Facebook dan Twitter. Dunia *e-learning* juga mengalami perkembangan. Hal ini ditandai dengan munculnya teknologi yang memudahkan untuk setiap orang membagikan pengetahuan melalui sebuah platform yang disebut blog (seperti WordPress, Blogger, Hashnode, dll.). Pelajar diminta untuk memberikan masukan secara langsung melalui fitur komentar yang disediakan pada artikel, dengan begitu proses pembelajaran dapat berjalan tanpa batasan tempat. Pelajar dapat berkonsultasi melalui diskusi forum apabila menemukan kesulitan dalam memahami materi. Tentunya diskusi forum ini memiliki aturan agar setiap orang yang terlibat tidak membahas topik di luar dari apa yang dibahas pada forum tersebut. Administrator dapat menghapus pembahasan topik yang tidak relevan.

Era web 2.0 juga memunculkan berbagai teknologi *e-learning* lainnya. Wiki memberikan keleluasaan untuk saling berkontribusi pada satu topik pengetahuan, contohnya yang terkenal Wikipedia untuk segala pengetahuan dunia nyata dan Fandom untuk pengetahuan dalam dunia hiburan (*entertainment*). Pengetahuan pada Wiki didapatkan dari kumpulan pengguna yang terkadang tidak memiliki rujukan referensi yang akurat. *Social Bookmarking* merupakan halaman web individu termasuk blog untuk menyimpan *bookmark* halaman web dan membagikannya dengan orang lain. *Social Bookmarking* memberikan pelajar untuk dapat berbagi informasi dan berbagai halaman web yang menarik untuk mereka. *Media Sharing* memungkinkan orang-orang untuk mengunggah media visual pada

sebuah halaman web, seperti Tumblr untuk berbagi animasi ataupun gambar dan Youtube untuk berbagi video.

Pengembangan *e-learning* baru-baru ini yaitu *Massive Open Online Course* (MOOC). Tujuan pengembangan *e-learning* ini untuk menyediakan pembelajaran kepada siapa saja yang ingin mengakses sumber pembelajaran kapan saja di kehidupan mereka, memberdayakan siapa saja yang ingin membagi apa yang mereka tahu dan mencari siapa saja yang ingin mempelajarinya, dan pada akhirnya menyediakan kepada semua orang yang ingin mengangkat sebuah isu ke publik dengan peluang untuk membuat tantangan yang mereka punya diketahui. Udemy merupakan salah satu penyedia layanan *e-learning* dengan konsep MOOC. Udemy memberikan kesempatan kepada semua orang mengajar ataupun belajar sesuatu di platform ini. Selain itu ada juga platform *e-learning* yang bekerja sama dengan institusi pendidikan seperti Udacity, Edx dan Coursera. (Rice & McKendree, 2013)

Melihat dari perkembangan web di atas yang mempengaruhi perkembangan *e-learning* memberikan gambaran bagaimana *e-learning* secara umum dibuat. Hal ini membuat penulis untuk memiliki gambaran terkait *e-learning* ini. Namun, penulis tetap menggunakan batas penelitian agar pembahasan *e-learning* ini tetap berada pada lingkup pembahasan. Tentunya pembahasan *e-learning* pada penelitian ini yaitu terkait dengan pelatihan karyawan.

CTI memerlukan *e-learning* yang dapat membantu proses pembelajaran karyawan terkait pengetahuan perusahaan. Proses pembelajaran ini dimulai dari karyawan baru masuk dan ikut terlibat dalam melaksanakan proses bisnis perusahaan. Mereka mengikuti proses pengenalan perusahaan yang disebut dengan *Induction*. Proses ini terdiri dari pengenalan visi misi perusahaan, pengetahuan produk, dan aturan yang berlaku. Pembelajaran tidak hanya dilakukan oleh karyawan baru saja, melainkan juga karyawan lama. Pengetahuan yang didapatkan oleh karyawan lama bisa berupa prosedur dalam melaksanakan proses bisnis maupun pelatihan untuk meningkatkan kemampuan tertentu yang diperlukan untuk menunjang suatu pekerjaan.

## 2.5 Pemrograman berbasis Objek (*Object Oriented Programming/OOP*)

Dunia ini penuh dengan berbagai macam objek, mulai dari kursi yang Anda duduki sampai laptop yang penulis gunakan untuk membuat skripsi ini, semua itu adalah objek. Objek memiliki sesuatu yang dapat diidentifikasi, baik itu sebuah bentuk, sebuah tekstur, maupun beberapa perilaku (*behaviour*) tertentu yang menyusunnya. Dunia ini penuh dengan objek jika berkaca pada ide “Objek” di kehidupan sehari-hari, bahkan manusia adalah objek. Manusia dapat diidentifikasi dengan unik, memiliki sifat dan perilaku. Manusia melakukan komunikasi antara satu dengan yang lain. Manusia tidak dapat meraih apapun di dunia tanpa adanya komunikasi. Mereka perlu melakukan komunikasi untuk memandu perilaku dari objek individu/manusia untuk mencapai tujuan mereka. Sama halnya dengan dasar dari ide pemrograman berbasis objek (OOP), bagaimana kita membuat objek yang dapat saling berkomunikasi satu dengan yang lain untuk meraih tujuan tertentu. (Ege, 2014)

Sebuah objek memiliki beberapa komponen penyusunnya, seperti keadaan sesaat (*state*) dan perilaku (*behaviour*). *State* merupakan kondisi dari sebuah objek yang menjelaskan informasi terkait identifikasi sebuah objek. State dinyatakan dalam *attribute/properties*. Perilaku merupakan definisi apa yang dapat dilakukan suatu objek berupa suatu tindakan/operasi dan memberikan reaksi dari tindakan yang dilakukan. Perilaku sebuah objek dinyatakan dalam *operation/behaviour*. *Properties* dan *behaviour* membentuk suatu fitur pada sebuah objek. Fitur nantinya merupakan suatu proses untuk menyelesaikan permasalahan bisnis. (Ege, 2014; Munawar, 2018)

Nama Objek	<b>Kucing</b>
Attribute/Property	nama warnaBulu berat jenisKelamin umur
Operation/Behaviour	<i>makan()</i> <i>tidur()</i> <i>meow()</i> <i>berjalan()</i>

Gambar 2.1 Sebuah kelas memiliki *property* dan *behaviour*

Gambar di atas menunjukkan bentuk kelas yang digambarkan dalam sebuah tabel satu kolom dan tiga baris. Kelas merupakan himpunan dari objek-objek yang sejenis. Ketiga baris pada kolom ini memberikan penjelasan terkait komponen yang terdapat pada sebuah objek, yaitu nama objek, *attribute/property*, dan *operation/behaviour*. Contoh kelas pada gambar di atas yaitu Kucing. Kelas Kucing memiliki *property* nama, warnaBulu, berat, jenisKelamin, dan umur yang digunakan untuk mengidentifikasi objek kucing. Kelas Kucing juga memiliki *operation/behaviour* yang digunakan untuk menggambarkan apa saja perilaku yang dapat kucing lakukan. Biasanya setiap objek kucing memiliki nilai *property* berbeda. (Munawar, 2018)

Objek adalah contoh/*instance* pada sebuah kelas. Contoh pada kelas kucing yaitu kucing anggora, kucing kampung, kucing kaos kaki, dan masih banyak lagi. Objek tersebut memiliki nilai *property* yang berbeda, tetapi memiliki *behaviour* yang sama. Kelas Kucing berkontribusi pada pengelompokan makhluk hidup di kehidupan sehari-hari. Objek tidak dapat berdiri sendiri, mereka memiliki aspek-aspek penting yang menyusunnya seperti Abstraksi (*Abstraction*), Pewarisan (*Inheritance*), Polimorfisme (*Polymorphism*), *Encapsulation*, *Message Sending*, *association*, dan *aggregation*. (Munawar, 2018) Semua ini dibahas pada subbab berikutnya.

### 2.5.1 Abstraksi (*abstraction*)

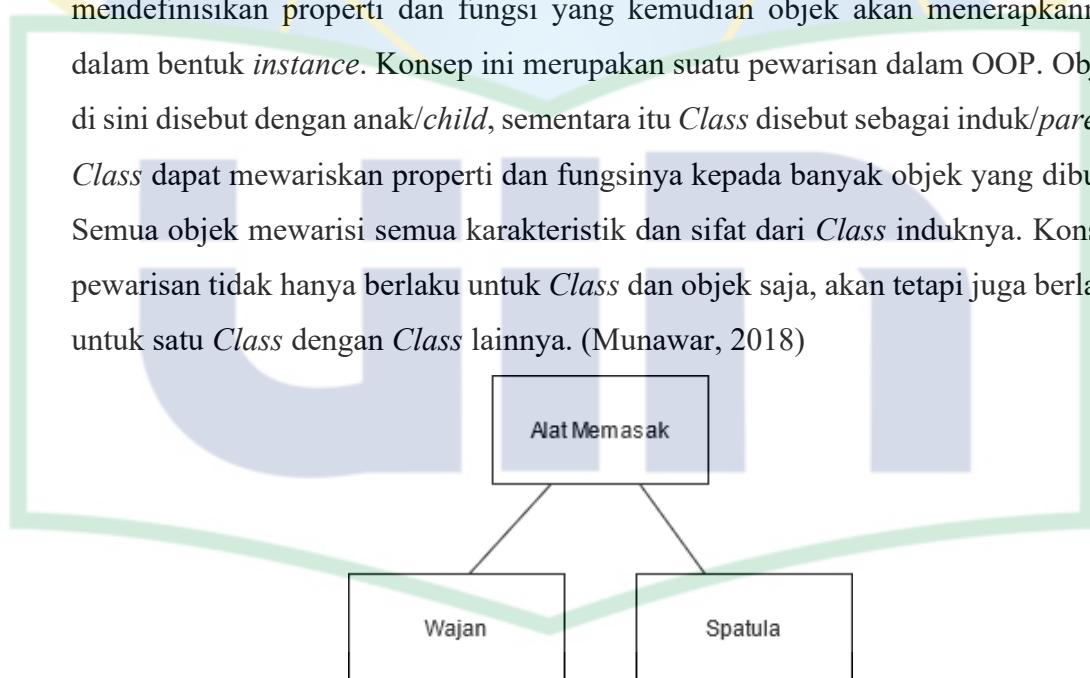
Abstraksi dalam OOP bertujuan untuk menyaring *property/properti* dan *behaviour/fungsi* apa saja yang tersedia pada sebuah objek. Abstraksi berguna untuk menyembunyikan fungsi internal yang terjadi di dalam sebuah objek. Objek

lain tidak perlu tahu bagaimana proses yang berjalan pada sebuah objek, dengan begitu maka objek-objek hanya akan melakukan interaksi pada *property* dan *behaviour* yang diperlukan saja. Hal ini akan mengurangi kompleksitas fungsi dari sebuah objek jika hendak mengakses fungsi pada objek lain. Abstraksi bekerja dengan memfilter *property* dan *behaviour* pada suatu objek. (Hillar, 2015; Munawar, 2018)

Misalnya kita akan membuat program untuk mengendalikan TV. Program ini dapat mengubah saluran, mengubah volume, hingga mengatur waktu tidur. Setiap TV memiliki *property* nomor serial yang tidak berguna untuk digunakan pada program yang ingin kita kembangkan, karena hanya menyangkut proses pada satu TV saja. Namun jika kasusnya pada transaksi penjualan TV, kita memerlukan nomor serial tersebut untuk membedakan antara satu TV dengan TV lain.

### 2.5.2 Pewarisan (*inheritance*)

Objek merupakan suatu *instance* atau contoh dari sebuah *Class/Kelas*. *Class* mendefinisikan properti dan fungsi yang kemudian objek akan menerapkannya dalam bentuk *instance*. Konsep ini merupakan suatu pewarisan dalam OOP. Objek di sini disebut dengan anak/*child*, sementara itu *Class* disebut sebagai induk/*parent*. *Class* dapat mewariskan properti dan fungsinya kepada banyak objek yang dibuat. Semua objek mewarisi semua karakteristik dan sifat dari *Class* induknya. Konsep pewarisan tidak hanya berlaku untuk *Class* dan objek saja, akan tetapi juga berlaku untuk satu *Class* dengan *Class* lainnya. (Munawar, 2018)



Gambar 2.2 Wajan dan spatula merupakan bagian dari *Class* alat memasak

### 2.5.3 Polimorfisme (*polymorphism*)

Polimorfisme merupakan konsep OOP yang menjelaskan suatu objek yang memiliki nama fungsi yang sama dengan objek lain, namun cara kerja dari fungsi tersebut berbeda antar objek. Misalkan fungsi “load” pada bongkar muat barang

yang berfungsi untuk memuat barang dari truk kontainer, aplikasi emulator permainan konsol yang berfungsi untuk memuat berkas permainan ke dalam emulator, ataupun aplikasi perpustakaan yang berfungsi untuk mengambil informasi atau buku. Polimorfisme sangat handal untuk pengembangan perangkat lunak pada sub sistem yang berbeda. Konsepnya pada pemrograman dengan menerapkan *interface* antar *Class*. (Munawar, 2018)

Contoh lain penerapan Polimorfisme misalnya fungsi “nyalakan” pada aplikasi manajemen peralatan rumah tangga, terdapat banyak jenis peralatan rumah tangga yang bisa dikendalikan melalui aplikasi seperti lampu dan mesin cuci. Fungsi “nyalakan” pada lampu akan menghubungi lampu yang telah didaftarkan pada aplikasi dan kemudian menjalankan fungsi pada lampu tersebut untuk mengaktifkan LED pada lampu. Sedangkan fungsi nyalakan pada mesin cuci akan menghubungi mesin cuci yang telah didaftarkan pada aplikasi dan kemudian menjalankan berbagai instruksi untuk menyelesaikan pencucian baju.

#### 2.5.4 Penyembunyian Informasi (*encapsulation*)

Konsep ini didasarkan pada fakta di dunia nyata yang menyatakan bahwa semua hal tidak perlu diperlihatkan, sebagai contoh sebuah *smartphone*. Pengguna *smartphone* tidak perlu harus mengetahui bagaimana suatu benda elektronik yang berbentuk kotak persegi panjang ini dapat bekerja. *Smartphone* sebenarnya memiliki serangkaian instruksi agar dapat bekerja dan tersusun dari berbagai komponen kecil seperti layar, baterai, *speaker*, *mic*, dan lainnya. Pengguna hanya perlu memakai *smartphone* dengan tampilan (*interface*) yang sudah disediakan seperti layar sentuh, tombol daya, *fingerprint*, dan komponen lainnya yang dapat berinteraksi dengan pengguna. Konsep ini juga digunakan pada dunia pemrograman untuk memudahkan dalam menemukan kesalahan/error dengan fokus pada objek yang bermasalah saja tanpa perlu memperhatikan objek lain. (Munawar, 2018)

*Encapsulation* juga berguna untuk melindungi dan menyembunyikan data. Kita ingin membuat sebuah *Class* yang hanya menyediakan informasi yang diperlukan saja. Sementara itu, kita tidak ingin informasi yang digunakan untuk proses internal diubah. Sebuah *Class* memiliki *setter* dan *getter*. *Setter* berguna untuk memberikan nilai pada informasi yang ada di dalam *Class*. Sedangkan *getter*

berguna untuk mengambil informasi yang ada di dalam *Class*. Penggunaan *setter* dan *getter* ini bermanfaat untuk mengatur restriksi informasi yang bisa diakses pada sebuah *Class*. Suatu variabel yang ada di dalam *Class* memiliki akses *read-only* jika hanya memiliki *getter*. Sebagai contoh pada pendaftaran ulang mahasiswa baru, mereka tidak diizinkan untuk mengubah nomor induk mahasiswa (NIM). Tetapi mereka bisa mengambil informasi tentang nomor NIM yang bisa mereka gunakan untuk mengikuti berbagai aktivitas perkuliahan di kampus. NIM merupakan informasi penting yang tidak dapat diubah, di sini *encapsulation* bekerja untuk menyembunyikan bagaimana pembuatan NIM yang hanya bisa dilakukan oleh staf akademik saja. (Hillar, 2015)

### 2.5.5 Pengiriman Informasi (*message sending*)

Komunikasi pesan/informasi dapat terjadi pada OOP. Fungsi dari setiap objek yang dibuat dapat saling membutuhkan dengan fungsi dari objek lain. Hal ini yang menyebabkan proses komunikasi pesan/informasi terjadi. Suatu objek mengirimkan pesan kepada objek lain dengan menjalankan satu fungsi, kemudian objek yang menerima pesan/informasi tersebut mengirimkan respons hasil menjalankan fungsi padanya. Contoh pada dunia nyata yaitu tombol pada *remote TV* yang memberikan perintah kepada TV untuk menyalakan saluran ataupun mengatur volume suara. (Munawar, 2018)

Contoh lain pada pemrograman yaitu apabila ingin mengecek tagihan SPP mahasiswa. Pada kasus ini objek Mahasiswa memiliki fungsi *cekTagihan* dan *getInfo*. Fungsi *getInfo* memberikan informasi seputar mahasiswa seperti nama, NIM, dan program studi berdasarkan NIM yang dimasukkan. Sedangkan *cekTagihan* berfungsi untuk mengecek berapa tagihan SPP mahasiswa yang harus dibayar pada semester saat ini. Kemudian ada objek Kampus yang memiliki fungsi *getSPP*. Fungsi ini digunakan untuk mengecek tagihan SPP berdasarkan informasi program studi yang dimasukkan. Fungsi *cekTagihan* apabila dijalankan akan memanggil *getInfo* untuk mendapatkan informasi program studi dari NIM mahasiswa yang telah dimasukkan. Kemudian fungsi *cekTagihan* akan memanggil *getSPP* dari objek Kampus untuk mendapatkan tagihan SPP dari mahasiswa tersebut. Pertukaran informasi terjadi antara fungsi *cekTagihan* dengan *getInfo* pada objek Mahasiswa yaitu NIM untuk mendapatkan informasi program studi dari

mahasiswa. Kemudian pertukaran informasi juga terjadi antara cekTagihan pada objek Mahasiswa dengan getSPP pada objek Kampus yaitu informasi program studi untuk mendapatkan tagihan dari mahasiswa tersebut.

### 2.5.6 Asosiasi (*Association*)

Asosiasi adalah hubungan antar objek yang saling membutuhkan. Objek yang saling membutuhkan ini tentunya melakukan pertukaran informasi untuk menjalankan fungsi tertentu. Hubungan objek bisa jadi satu arah ataupun atau lebih. Contoh hubungan satu arah misalnya fungsi konversiNilai yang melakukan pengubahan nilai ke bentuk IPK rentang 4 dan fungsi saveNilai yang melakukan penyimpanan nilai tersebut. Hubungan pada kedua fungsi tersebut terjadi secara satu arah di mana konversiNilai memerlukan saveNilai untuk melakukan penyimpanan dari hasil konversi nilai, tetapi saveNilai tidak akan melakukan apa-apa jika tidak mendapatkan hasil dari konversiNilai dan dia tidak masalah dengan hal itu. Contoh hubungan dua arah misalnya fungsi getInfoMahasiswa yang memerlukan getSPP untuk mendapatkan tagihan SPP mahasiswa. Fungsi getSPP juga memerlukan informasi dari getInfoMahasiswa yaitu program studi untuk mendapatkan info tagihan SPP mahasiswa.

*Multiplicity* merupakan aspek penting pada asosiasi. Konsepnya yaitu objek dapat memiliki hubungan asosiasi satu ke satu objek (*one-to-one*) dan objek dapat juga memiliki hubungan asosiasi satu objek ke banyak objek lain (*one-to-many*). Contoh hubungan asosiasi *one-to-one* yaitu satu mata kuliah yang hanya bisa diampu oleh satu dosen. Sedangkan contoh hubungan asosiasi *one-to-many* yaitu dosen yang dapat mengampu beberapa mata kuliah pada satu semester di angkatan kelas yang berbeda. (Munawar, 2018)

### 2.5.7 Agregasi (*aggregation*)

Agregasi merupakan bentuk khusus asosiasi dengan mendeskripsikan bahwa seluruh bagian suatu objek merupakan bagian objek lain. Sebagai contoh sebuah komputer yang memiliki layar monitor, CPU, tetikus (*mouse*), dan papan kunci (*keyboard*). Mouse dan *keyboard* memiliki komponen lagi yang menyusunnya, komponen-komponen ini menyusun bagian dari objek komputer.

*Composition* merupakan salah satu bentuk agregasi yang melibatkan hubungan yang erat antar objek yang saling berhubungan. Poin penting pada

*composition* yaitu sebuah komponen objek hanya bisa terbentuk apabila ada objek tertentu yang diperlukan. Sebagai contoh pada kasus komputer, sebuah komputer tidak akan dikatakan komputer jika hanya terdiri dari mouse atau *keyboard* saja. Mereka memerlukan komponen lain yang menyusunnya agar dapat dikatakan objek komputer. (Munawar, 2018)

### 2.5.8 Manfaat dan Keterbatasan OOP

Penggunaan konsep OOP pada pengembangan aplikasi memiliki beberapa manfaat seperti memudahkan untuk memecahkan masalah dengan mengubah objek yang ada di dunia nyata menjadi objek yang sama pada pemrograman. Sebuah objek pada dunia nyata memiliki sifat/keadaan sesaat (*property/state*) dan perilaku (*behaviour*) sama halnya dengan objek pada OOP yang memiliki dua hal tersebut untuk menyusunnya. Bentuk objek yang sama antara kasus di dunia nyata dengan di pemrograman ini membuat OOP memberikan informasi yang jelas dalam konteks sistem.

Pemikiran berorientasi objek tidak hanya pada pemrograman saja, melainkan juga dapat digunakan pada analisis, desain, dan perancangan UI (*user interface*). Selain itu objek lain di dunia nyata juga dapat digunakan untuk memodelkan sistem seperti model sosial, ekonomi, ataupun kondisi sosial. Pengembangan berorientasi objek menggunakan model yang ada di dunia nyata untuk mendeskripsikan objek yang ada pada pemrograman. Hal ini membuat objek-objek tersebut menjadi sebuah konsep sentral saat pengembangan sistem. Pemikiran berorientasi objek juga memberikan penjelasan objek secara natural tanpa memberikan batasan dalam pandangan teknis saja.

OOP juga memiliki keterbatasan untuk beberapa kasus pengembangan aplikasi seperti pengembangan aplikasi yang sangat berorientasi ke *database* dan aplikasi yang melibatkan perhitungan besar dan kompleks. Aplikasi yang sangat berorientasi ke *database* sangat tidak cocok untuk dikembangkan dengan OOP karena konsep relasional pada RDBMS (*Relational Database Management System*) akan hilang. RDBMS memiliki *query* untuk melakukan pemanggilan ataupun penyimpanan data. Sedangkan fungsi pada objek dikembangkan dengan cara statis yang hanya menjalankan satu *query* untuk kasus tertentu. Kemudian aplikasi yang melibatkan perhitungan yang besar dan kompleks juga sangat tidak cocok apabila

dikembangkan dengan OOP karena aplikasi ini akan memerlukan proses yang sangat banyak dengan melibatkan banyak logika algoritma dalam satu fungsi. (Munawar, 2018)

## 2.6 Metode Pengumpulan Data

Pengumpulan data merupakan suatu proses untuk mendapatkan bukti/fakta dengan menggali lebih lanjut terkait sesuatu yang dibahas. Pembahasan terkait pengumpulan data bisa digunakan untuk penyelesaian suatu permasalahan, seperti penelitian akademik dan pembuatan solusi bisnis. Metode pengumpulan data membahas tentang teknik atau cara bagaimana Penulis melakukan pengumpulan data. Data yang dikumpulkan dalam penelitian digunakan untuk membuktikan suatu hipotesis atau merancang suatu rumusan masalah. Kemudian data ini digunakan sebagai dasar dalam pengambilan keputusan untuk melakukan sebuah penelitian ataupun digunakan untuk membuat kesimpulan. Metode pengumpulan data adalah langkah awal dari setiap penelitian. Suatu penelitian tidak akan dilakukan apabila tidak dilandasi oleh data yang menyimpulkan bahwa perlunya melakukan penelitian tersebut. (Gultom, 2020; Makbul, 2021)

### 2.6.1 Observasi

Observasi adalah salah satu teknik untuk mengumpulkan data dengan cara mengamati langsung berbagai perilaku atau interaksi sosial yang terjadi di tempat lingkup penelitian. Observasi membuat data yang dikumpulkan menjadi lebih baik dan lebih valid ketika dilengkapi dengan analisis perilaku dan konteks subjek dan objek penelitian. Data yang sebelumnya telah dimiliki oleh Penulis bisa divalidasi lebih lanjut dengan pihak terkait. Penulis juga bisa mendapatkan data baru dengan melaksanakan teknik pengumpulan data ini. Data yang diperoleh yaitu berupa pola-pola (*open-ended data*) atau konfirmasi pola-pola tertentu (*closed & coded*). Observasi dilakukan dengan cara partisipatif (Penulis diterima sebagai orang yang mengikuti kegiatan rutin organisasi) maupun non-partisipatif (Penulis merupakan orang luar yang melakukan pengamatan langsung tanpa berinteraksi ataupun terlibat dalam rutinitas organisasi). (Makbul, 2021; Rachmawati, 2017)

### 2.6.2 Wawancara

Wawancara adalah salah satu teknik yang dapat digunakan untuk mengumpulkan data penelitian. Prosesnya dengan cara penulisan angket secara

lisan dan langsung terhadap para narasumber. Wawancara melibatkan pewawancara/penulis dan pemberi sumber informasi/narasumber yang saling berkomunikasi. Proses komunikasi dua arah atau lebih ini menghasilkan suatu data yang dapat digunakan untuk melakukan studi pendahuluan, menemukan masalah, hingga mendefinisikan rumusan masalah. Proses pengumpulan data dengan wawancara dapat memberikan hal-hal yang lebih mendalam dari responden. Pengambilan sampel beberapa responden membuat hal-hal yang terjadi pada suatu permasalahan akan terlihat lebih jelas dengan menarik kesimpulan dari jawaban-jawaban responden tersebut.

Wawancara setidaknya harus dilakukan dengan baik agar mendapatkan data-data yang lebih mendalam yang tidak mungkin didapatkan dari angket. Jawaban-jawaban pada angket dapat ditanyakan kembali untuk mendapatkan wawasan dan menilai keadaan responden pemberi informasi. Pewawancara/Penulis yang melakukan wawancara bertindak sebagai pemimpin dalam proses wawancara. Pewawancara harus merencanakan bagaimana wawancara akan dilaksanakan, menyusun petunjuk wawancara, menyiapkan alat tulis untuk merekam jawaban atau alat perekam, hingga menyusun pertanyaan yang akan diajukan dalam wawancara. Pewawancara dapat merencanakan kapan wawancara akan dilakukan ataupun menyesuaikannya dengan jam ketersediaan narasumber. (Makbul, 2021)

### 2.6.3 Studi Literatur

Studi literatur merupakan salah satu cara teknik pengumpulan data penelitian terakhir yang digunakan penulis pada penelitian ini. Prosesnya yaitu menghimpun data-data atau sumber-sumber yang berhubungan dengan topik penelitian penulis. Sumber-sumber ini bisa berupa artikel dari jurnal, buku, skripsi lain, ataupun dokumen lain yang dapat mendukung penelitian. Sumber-sumber yang telah didapatkan kemudian dianalisis dengan metode analisis deskriptif. Analisis deskriptif dilakukan dengan cara mendeskripsikan fakta-fakta, kemudian diuraikan dan dianalisis dengan memberikan pemahaman dan penjelasan secukupnya. Penyusunan pustaka literatur bisa menggunakan bantuan perangkat lunak seperti Zotero, Mendeley, Qiqqa, Endnote, dan masih banyak lagi. Perangkat lunak ini tidak hanya memberikan kemudahan untuk mengelola pustaka saja, melainkan juga

dapat membuat daftar pustaka secara otomatis mengikuti templat tertentu. (Habsy, 2017; Makbul, 2021)

## 2.7 Unified Modelling Language (UML)

Pengembangan sistem tidak dapat dilakukan semata-mata secara langsung tanpa perencanaan. Setiap pemangku kepentingan (*stakeholder*) yaitu orang-orang yang terlibat secara langsung maupun tidak langsung terhadap pengembangan sebuah sistem pastilah memiliki ide yang berbeda-beda untuk sistem yang akan mereka buat. Mereka memerlukan sebuah alat yang dapat menampung ide mereka dan mengubahnya menjadi bentuk yang baku, mudah dimengerti, serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan ide rancangan mereka. Di sinilah UML berperan sebagai alat bantu yang sangat handal untuk menampung ide-ide tersebut.

UML (*Unified Modelling Language*) menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas ide yang mereka miliki. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, OMT (*Object Modelling Technique*), dan OOSE (*Object Oriented Software Engineering*). Tentunya UML merupakan alat bantu pengembangan sistem yang berorientasi objek (OOP). Visualisasinya pun menggunakan istilah-istilah yang ada dalam OOP seperti penggambaran kelas menggunakan *Class Diagram*, menggambarkan hubungannya, dan masih banyak lagi. UML membuang elemen-elemen yang tidak praktis pada metode Booch, OMT, dan OOSE ditambah dengan elemen-elemen dari metode lain yang lebih efektif serta elemen-elemen baru dari metode terdahulu. Hal ini membuat UML memiliki sifat yang lebih ekspresif dan seragam ketimbang metode lainnya.

UML terdiri dari beberapa diagram yang dapat digunakan untuk membantu dalam memvisualisasikan sistem dengan berbasis objek. Diagram ini memiliki fungsinya masing-masing dan saling mendukung satu sama lainnya. Tabel berikut merupakan daftar tipe diagram yang ada di UML. (Munawar, 2018)

Tabel 2.1 Tipe Diagram UML

Diagram	Tujuan
Activity	Perilaku prosedural & pararel

Class	Kelas, Fitur dan relasinya
Communication	Interaksi antara objek, lebih menekankan ke hubungan (link)
Component	Struktur dan koneksi dari komponen
Composite structure	Dekomposisi sebuah kelas saat berjalan (runtime)
Deployment	Penyebaran/Instalasi ke klien
Interaction Overview	Gabungan antara diagram Activity dan Sequence
Object	Contoh konfigurasi instance
Package	Struktur hierarki saat kompilasi
Sequence	Interaksi antar objek, lebih menekankan pada urutan
State Machine	Bagaimana suatu kejadian (event) dapat mengubah objek
Timing	Interaksi antar objek, lebih menekankan pada waktu
Use Case	Bagaimana pengguna (user) berinteraksi dengan sebuah sistem

UML sudah cukup banyak memberikan diagram yang dapat membantu dalam memvisualisasikan ide dan menuangkannya sebagai cetak biru pengembangan sistem. Namun, kita tidak diharuskan untuk menggunakan semua diagram UML atau hanya menggunakan diagram-diagram yang ada pada UML saja. Kita dapat menggunakan diagram UML yang sesuai dengan kebutuhan kita dan menggunakan diagram lain selain UML atau kita dapat mengombinasikan keduanya. Penulis pada penelitian ini hanya akan menggunakan diagram Use Case, Activity, Class, dan Sequence sesuai dengan kebutuhan penulis untuk memenuhi kebutuhan penelitian. Diagram-diagram ini akan dibahas di subbab berikutnya.

### 2.7.1 Use Case Diagram

Pada dasarnya sebuah sistem memiliki fitur-fitur yang bertugas untuk melaksanakan satu atau beberapa tugas. Tugas-tugas ini merupakan langkah-langkah untuk menyelesaikan suatu pekerjaan yang disebut sebagai skenario. Skenario menjadi cikal bakal pembuatan sebuah fitur pada solusi perangkat lunak.

Tujuan dari skenario adalah untuk menggambarkan apa yang dibutuhkan oleh perangkat lunak, sebagai desain dasar pembuatan perangkat lunak, dan membatasi serangkaian persyaratan yang dapat divalidasi ketika perangkat lunak dibangun. Penggambaran skenario pada UML menggunakan diagram *Use Case*.

Diagram *Use Case* merupakan deskripsi fungsi yang ada pada sistem dari perspektif pengguna. Cara kerjanya yaitu dengan mendeskripsikan interaksi antara pengguna (*user*) dengan sistem yang ia gunakan melalui sebuah skenario bagaimana sistem itu dipakai. *Use Case* menggambarkan analisis kebutuhan dari sistem level atas melalui penggambaran interaksi antara fungsionalitas sistem dan para pengguna sistem yang disebut aktor. *Use Case* dibuat berdasarkan keperluan dari aktor dan harus merupakan “Apa” yang akan dikerjakan oleh perangkat lunak aplikasi, bukan “Bagaimana” perangkat lunak aplikasi menjalankannya.

Tujuan dari *Use Case* kurang lebih hampir sama seperti skenario. Namun, secara spesifik penggunaan diagram *Use Case* bertujuan untuk mendapatkan pandangan dari luar sistem, mengidentifikasi faktor yang mempengaruhi sistem baik internal maupun eksternal, dan menunjukkan interaksi dari para aktor dari sistem.

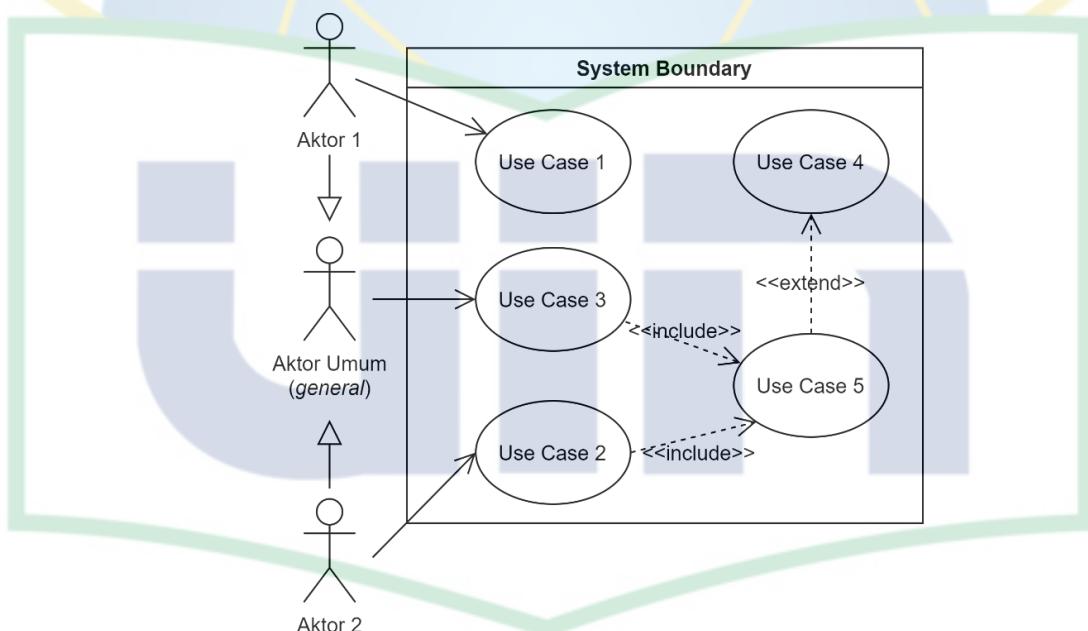
*Use Case* sebagai sebuah diagram memiliki dua jenis notasi dalam penggunaannya, yaitu notasi utama dan notasi *stereotype*. Notasi utama pada *Use Case* meliputi: (Munawar, 2018)

- Aktor, merupakan entitas yang berinteraksi dengan sistem. Entitas ini tidak selalu melambangkan orang walaupun digambarkan dengan bentuk orang, melainkan bisa berupa organisasi atau proses lain yang bertindak sebagai pengguna dan berinteraksi dengan sistem.
- *Use Case*, merupakan notasi yang menggambarkan fungsionalitas sebuah sistem. Notasi ini menjadi simbol fungsi sistem yang berinteraksi dengan aktor.
- Sistem/ *Sub System Boundary*, merupakan notasi yang menggambarkan batasan sebuah sistem. *Use Case* digambarkan dalam sebuah *boundary* sistem agar terlihat jelas fungsi-fungi apa saja yang ada di dalamnya. Notasi ini juga digunakan untuk menggambarkan batasan fungsi sub sistem.

Notasi utama menggambarkan interaksi yang ada di sebuah sistem. Ada pula notasi untuk menggambarkan Sedangkan, notasi *stereotype* pada *Use Case* meliputi:

- <<include>> digunakan untuk menggambarkan bahwa satu *Use Case* keseluruhan fungsionalitasnya adalah dari *Use Case* lain.
- <<extend>> digunakan untuk menggambarkan bahwa satu *Use Case* merupakan tambahan fungsional dari *Use Case* lain jika syarat atau kondisi tertentu dipenuhi.

Selain kedua jenis notasi di atas, ada satu hal lagi yang dapat digunakan untuk membuat diagram *Use Case*, yaitu Generalisasi Aktor. Maksudnya adalah bagaimana kita menggambarkan dua atau lebih spesialisasi aktor yang dapat berinteraksi dengan satu *Use Case* menjadi satu aktor saja yang lebih umum (*general*). Perhatikan gambar 2.4 untuk mengetahui bentuk notasi diagram *Use Case* berikut:



**Gambar 2.3 Notasi-notasi diagram *Use Case***

Gambar 2.4 menunjukkan notasi diagram *Use Case* yang memiliki dua aktor spesialis, satu aktor umum, lima *Use Case*, dua notasi <<include>>, satu notasi <<extend>>, dan sebuah *System Boundary*. Aktor baik aktor spesialis maupun aktor umum digambarkan dengan notasi bentuk orang, sedangkan *Use Case* digambarkan dengan bentuk elips. Aktor spesialis yang memiliki hubungan dengan aktor umum

dihubungkan menggunakan anak panah dengan bentuk kepala panah kosong. Aktor umum (*general*) merupakan generalisasi dari Aktor 1 dan Aktor 2 sehingga dapat diartikan bahwa *Use Case* 3 bisa diakses oleh kedua aktor. Masing-masing dari aktor yang memiliki akses ke *Use Case* dihubungkan dengan anak panah yang mengarah dari bentuk aktor ke bentuk elips pada *Use Case*. Penggambaran <<include>> dan <<extend>> menggunakan anak panah putus-putus yang mengarah ke *Use Case* dituju dengan memberikan teks pada anak panah tersebut untuk membedakan keduanya. Terakhir, *System Boundary* digambarkan dengan kotak lebar yang berisi semua *Use Case* di dalamnya dan diberi nama sistem pada bagian atasnya.

Penggunaan diagram *Use Case* tidak cukup hanya dengan notasi-notasi di atas. Setiap notasi *Use Case* yang dibuat memerlukan teks yang menjelaskan tujuan dan fungsinya ketika dijalankan. Teks ini disebut dengan **Spesifikasi Use Case** (*Use Case Specification*). Spesifikasi *Use Case* dibuat dengan iterasi demi iterasi. Iterasi pertama biasanya hanya berupa deskripsi singkat tentang langkah-langkah yang diperlukan untuk mencapai keberhasilan suatu *Use Case*. Kemudian pada iterasi berikutnya kita melakukan analisis lebih lanjut spesifikasi *Use Case* dan membuatnya menjadi lebih detail bagi *Use Case* yang memerlukannya. Iterasi kedua akan terus berlanjut seiring berjalannya diskusi hingga pada akhirnya kita menambahkan kondisi pengecualian apabila dibutuhkan. Berikut adalah contoh spesifikasi *Use Case*. (Munawar, 2018; Sugiarti, 2013)

**Tabel 2.2 Contoh Spesifikasi *Use Case***

<b>Use Case Name</b>	Login	
<b>Actor(s)</b>	Karyawan, Approver, dan Karyawan HC (Admin)	
<b>Description</b>	Use Case ini mendeskripsikan proses autentikasi aktor agar dapat mengakses fungsi aplikasi.	
<b>Pre-Condition(s)</b>	Aktor telah berada di halaman utama aplikasi HC Portal.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b> 1. Aktor melakukan klik tombol <i>login</i> yang tersedia pada halaman utama.	<b>Application Response</b> 2. Menampilkan modal (jendela <i>popup</i> ) <i>login</i> .

	3. Mengisi formulir <i>login</i> .	4. Memvalidasi format input pengguna.
	5. Klik tombol “ <i>Login</i> ”.	6. Aplikasi memvalidasi input pengguna berupa nomor nik dan kata sandinya.
		7. Melakukan autentikasi pengguna dan menampilkan halaman awal <i>Job Profile</i> .
<i>Alternative Course(s)</i>	Alt 7. Pengguna akan diarahkan ke halaman awal dengan jendela popup login yang juga menampilkan pesan eror atas kegagalan autentikasi. Pengguna dapat menjalankan ulang use case ini ke langkah 1 kembali.	
<i>Conclusion</i>	Use Case berakhir saat pengguna telah masuk ke halaman awal <i>Job Profile</i> .	
<i>Post-Condition(s)</i>	Pengguna masuk ke halaman awal <i>Job Profile</i> .	
<i>Implementation Constraint(s) and Specification(s)</i>	Halaman awal <i>Job Profile</i> ..	

### 2.7.2 Activity Diagram

*Activity Diagram* menggambarkan aspek dinamis dari sistem berupa logika prosedural, proses bisnis, dan aliran kerja. *Activity Diagram* memiliki kesamaan peran sama halnya dengan *Flowchart*, tetapi yang membedakannya adalah *Activity Diagram* mendukung penggambaran perilaku pararel. Pembuatan *Activity Diagram* bertujuan untuk menggambarkan aliran aktivitas sistem, menggambarkan urutan sistem dari satu aktivitas ke aktivitas lainnya, dan menggambarkan percabangan aliran konkuren dari sistem. Penggunaan *Activity Diagram* tidak hanya untuk memodelkan aliran aktivitas pada suatu sistem saja, melainkan juga mencakup pemodelan aliran proses kerja bisnis, kebutuhan bisnis, dan tingkat tinggi fungsionalitas sistem. (Munawar, 2018)

Sebelum membuat *Activity Diagram*, kita perlu memahami tentang elemen apa yang akan kita gunakan di *Activity Diagram*. Elemen ini berupa aktivitas-

aktivitas/fungsi yang dilakukan oleh sistem. Elemen ini bisa didapatkan setelah kita membuat diagram *Use Case*. *Use Case* pada diagram *Use Case* merupakan fungsi yang ada pada sistem sehingga kita bisa mengambil elemen *Use Case* untuk membuat sebuah diagram *Activity*. (Munawar, 2018; Sugiarti, 2013)

*Activity Diagram* memiliki notasi-notasi yang sering digunakan dalam pembuatannya. Notasi-notasi tersebut dijabarkan pada tabel berikut:

Tabel 2.3 Notasi-notasi pada *Activity Diagram*

Notasi	Keterangan
●	Titik Awal
○	Titik Akhir
[ ]	<i>Activity</i> (Aktivitas)
◇	Pilihan untuk pengambilan keputusan; kondisi
—	<i>Fork</i> ; menunjukkan aktivitas secara pararel; menggabungkan dua aktivitas pararel menjadi satu
⊕	<i>Rake</i> ; menunjukkan adanya dekomposisi apabila ingin membuat <i>sub-activity</i> dari sebuah <i>activity</i>
×	Tanda waktu
→	Tanda Pengiriman ( <i>Send sign</i> )
←	Tanda penerimaan ( <i>Receive sign</i> )
⊗	Aliran akhir ( <i>Flow final</i> )

Notasi-notasi tersebut disusun secara berurutan dan dihubungkan dengan anak panah mengikuti alur proses kerja atau bisnis. *Activity Diagram* dimulai dengan membuat tanda Titik Awal dan diakhiri dengan Titik Akhir. Tanda Aliran Akhir (*Flow Final*) menunjukkan akhir dari sebuah aliran (*flow*) tertentu tanpa

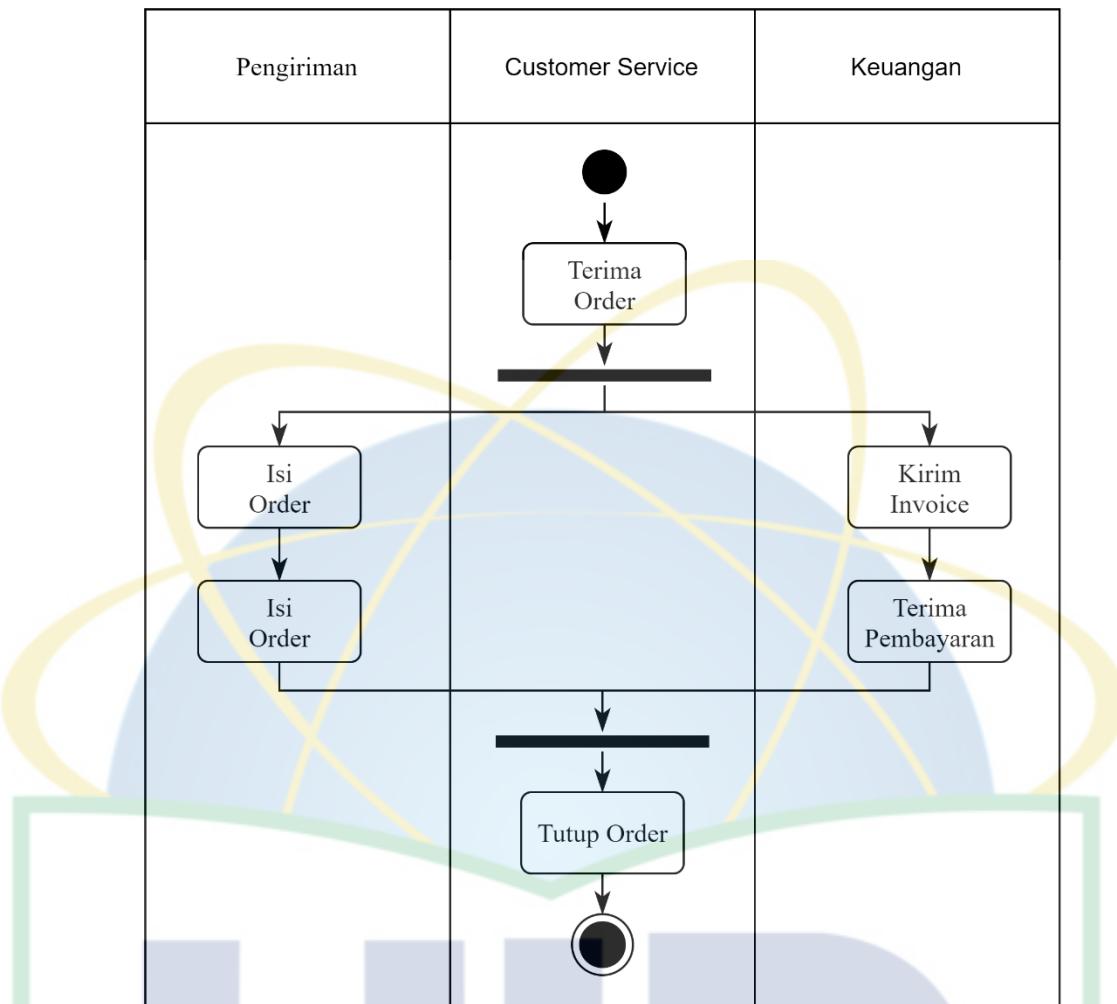
mengakhiri seluruh *activity*. Tanda Pengiriman menunjukkan bahwa sebuah *activity* mengirim sebuah *event* ke proses luar, sedangkan Tanda Penerimaan menerima sebuah *event* dari proses luar. Tanda Penerimaan membuat sebuah *activity* tertahan (*hold*) sebelum ia dapat melanjutkan ke proses berikutnya. Biasanya Tanda Penerimaan digunakan setelah menggunakan tanda *fork* untuk menjalankan proses lain secara pararel. Proses pararel tersebut biasanya menggunakan Tanda Waktu yang menunjukkan *activity* akan ditahan dalam jangka waktu tertentu sebelum dilanjutkan ke aliran berikutnya.

Pembuatan *Activity Diagram* bertujuan untuk menunjukkan apa yang terjadi, tetapi tidak menunjukkan siapa yang melakukan kejadian itu. Hal ini tidak terlalu penting karena biasanya pembuatan *Activity Diagram* hanya berfokus pada proses apa yang terjadi pada satu *Use Case*. (Munawar, 2018)

*Activity Diagram* bisa dibagi dalam bagian-bagian yang disebut *partition*. *Partition* menunjukkan siapa yang melakukan proses spesifik pada *activity* tertentu. UML versi 1 menyebut pendekatan ini sebagai ***Swim Lane***. (Grobelny & Grobelna, 2015; Munawar, 2018)

*Swim Lane* membuat konsultasi antara *engineers* dan *customers* menjadi lebih mudah. Di satu sisi, hal ini menunjukkan aktor yang melakukan input dan *output*. Di sisi lain, hal ini menunjukkan aksi yang saling terhubung dari aktor pada *activity* tertentu. Oleh karena itu, desainer dan *customer* dapat memastikan bahwa kepentingan mereka terhadap sistem ditemukan dan informasinya sesuai dengan kesepakatan bersama. Karenanya, pembuatan *Activity Diagram* yang menggunakan pendekatan *Swim Lane* dapat meningkatkan keterbacaan diagram dan kualitas dari produk final. (Grobelny & Grobelna, 2015)

Penulis menerapkan *Swim Lane* dalam pembuatan *Activity Diagram*. Penulis membuatnya dengan menggambarkan interaksi antara para pengguna dan sistem. Para pengguna melakukan aksi pemicu untuk mendapatkan respons tertentu dari sistem. Aksi-aksi dan respons ini dijabarkan dalam urutan sehingga satu *Use Case* dapat diaplikasikan. Perhatikan gambar 2.5 untuk memahami lebih lanjut.



Gambar 2.4 Contoh *Activity Diagram*

### 2.7.3 Class Diagram

Setelah membuat *Activity Diagram*, biasanya dilanjutkan dengan pembuatan *Class Diagram*. Tetapi sebelum itu, kita perlu memahami lebih dalam terkait *Class*. *Class Diagram* menunjukkan koleksi *Class* (pada sebuah *Class* menggambarkan atribut, *visibility* atribut, dan *operation/behavior*), *interface* (antarmuka), asosiasi, kolaborasi, dan *constraints*. Kita harus memahami komponen-komponen tersebut agar kita dapat membangun sebuah *Class Diagram*. Kumpulan *Class* pada *Class Diagram* menunjukkan hubungan satu sama lain. Hal ini melibatkan asosiasi, *multiplicity*, generalisasi dan pewarisan (*inheritance*), serta agregasi dan *composite*.

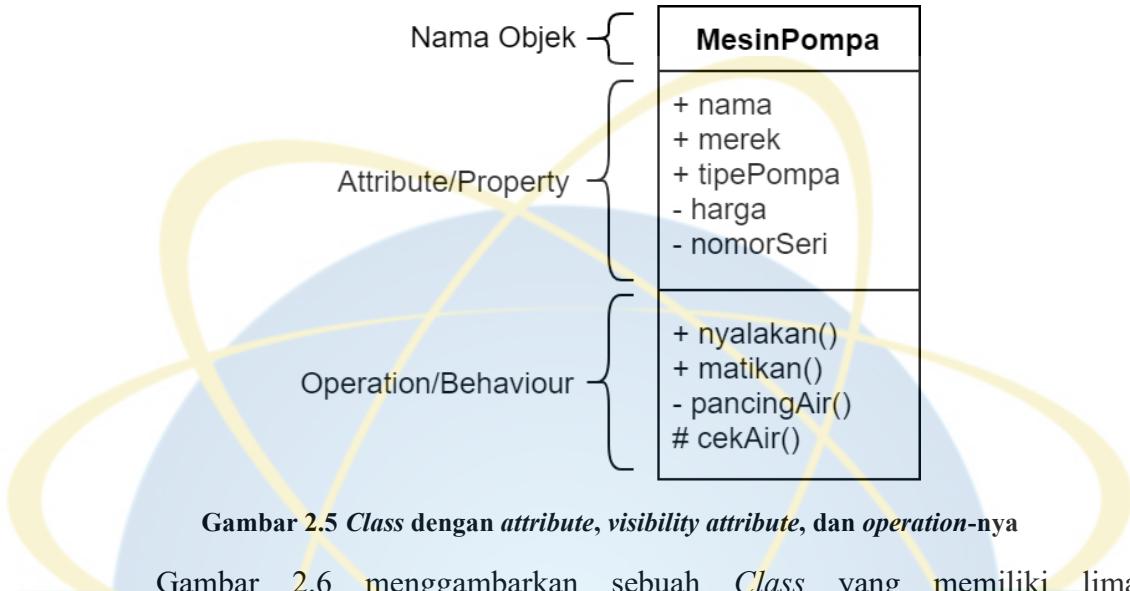
*Class* merupakan kumpulan objek yang memiliki atribut yang sama. Sedangkan objek merupakan *instance* (contoh) dari *Class*. Pengelompokan objek dalam dunia nyata telah dilakukan pada dunia ilmu pengetahuan, seperti

pengelompokan objek hewan Mamalia yang merupakan sebuah *Class* dalam makhluk hidup. *Instance* dari *Class* Mamalia terdiri dari manusia, gajah, sapi, kuda, singa, dll. Mereka memiliki kesamaan yaitu sama-sama hewan menyusui, inilah yang disebut dengan *Class*. Entitas bisnis akan menjadi sebuah model *Class* dengan menggunakan kata benda dalam dunia nyata. Proses bisnis akan menjadi *operation* pada *Class* yang dimodelkan. Identifikasi entitas bisnis menjadi atribut *Class*-nya. (Munawar, 2018)

Penulisan kata untuk penamaan benda atau variabel pada dunia pemrograman menggunakan konsep **CamelCase**, Konsep ini menghilangkan spasi pada dua suku kata atau lebih, karena setiap bahasa pemrograman menggunakan spasi untuk memisahkan logika kode mereka dan hal ini membuat penamaan objek tergambar dalam satu objek utuh. Terdapat dua jenis konsep CamelCase, pertama **UpperCamelCase** atau PascalCase atau DromedaryCase, kemudian yang kedua **lowerCamelCase**. UpperCamelCase menggunakan huruf kapital di setiap awal suku katanya, contoh mesin cuci menjadi MesinCuci, las listrik menjadi LasListrik, botol air menjadi BotolAir, dll. Sedangkan lowerCamelCase menggunakan huruf kecil di awal suku katanya dan suku kata berikutnya diawali dengan huruf kapital, contoh nomor seri menjadi nomorSeri, status pesanan menjadi statusPesanan, dll. (Porter, 2014)

*Class* digambarkan dengan tabel tiga baris, di mana pada baris teratas ke bawah berisi nama *Class*, atribut, dan *operation*. Nama *Class* ditulis dengan huruf kapital di awalnya, apabila nama *Class* terdiri dari dua suku kata atau lebih maka penamaan *Class* menggunakan konsep UpperCamelCase, contoh penamaan *Class* seperti Mamalia, MamaliaBerkakiEmpat, Kucing, KucingRumahan, dll. Baris kedua pada tabel *Class* berisi atribut yang ditulis dengan huruf kecil di awal kalimatnya, jika nama atribut terdiri dari dua suku kata atau lebih maka penulisannya menggunakan konsep lowerCamelCase, contoh penamaan atribut seperti nomorSeri, statusPesanan, merek, model, dll. Atribut sebuah *Class* bisa memiliki nol atribut atau lebih dari itu. Baris ketiga dan terakhir dari tabel *Class* berisi *operation* menggunakan penulisan yang sama seperti atribut dengan menambahkan tanda kurung di akhir kalimat untuk menyatakan bahwa kata tersebut merupakan sebuah fungsi dari *operation*, contohnya seperti

nyalakanMesin(), kunciPintu(), sedotAir(), dll. *Operation* merupakan hal yang bisa dilakukan oleh sebuah *Class*. *Operation* ini nantinya berinteraksi dengan pengguna atau sistem lain untuk menjalankan sebuah *event*. Perhatikan gambar berikut untuk memahaminya lebih dalam.



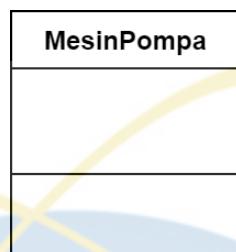
**Gambar 2.5 Class dengan attribute, visibility attribute, dan operation-nya**

Gambar 2.6 menggambarkan sebuah *Class* yang memiliki lima properti/*attribute* dan empat fungsi/*operation/behavior*. Masing-masing memiliki tanda visibilitas/*visibility* sebelum nama properti dan fungsi untuk memperlihatkan lingkup akses. Hal ini berguna untuk menetapkan apakah sebuah properti atau fungsi dapat diakses di lingkup system OOP. Ada empat tanda *visibility* yang bisa dipakai pada *Class* sebagaimana dijabarkan pada tabel 2.4.

**Tabel 2.4 Daftar visibility**

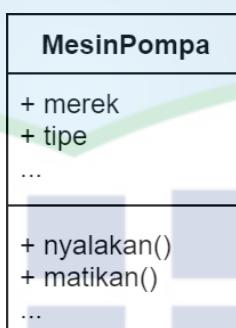
Simbol	Nama <i>Visibility</i>	Keterangan
+	<i>Public</i>	Fitur (sebuah properti atau fungsi) bisa diakses oleh <i>Class</i> apapun
-	<i>Private</i>	Fitur hanya boleh digunakan oleh <i>instance</i> dari <i>Class</i>
#	<i>Protected</i>	Fitur hanya boleh digunakan oleh <i>instance Class</i> dan anak-anaknya ( <i>children Class</i> )
~	<i>Package</i>	Fitur hanya boleh diakses langsung oleh <i>instance</i> sebuah <i>Class</i> pada <i>package</i> yang sama

Visualisasi atribut dan fungsi/*operation* pada praktiknya sering kali tidak selalu ditampilkan untuk menggambarkan sebuah *Class*. Hal ini terjadi apabila fokus dari penggambaran *Class Diagram* adalah menunjukkan hubungan/asosiasi antar *Class* saja.



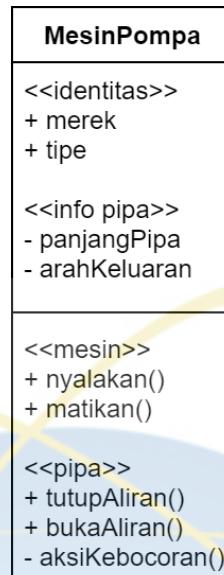
Gambar 2.6 *Class* tanpa atribut dan fungsi

Terkadang kita hanya perlu menampilkan beberapa item saja dan ini malah membuat *Class Diagram* menjadi lebih mudah terbaca. Penampilan sebagian item atribut atau fungsi pada sebuah *Class* memerlukan tanda titik tiga “...” atau yang disebut tanda ellipsis.



Gambar 2.7 *Class* dengan tanda ellipsis

Penggambaran sebuah *Class* terkadang memiliki atribut dan fungsi yang sangat banyak. Kita bisa menggunakan tanda stereotip (*stereotype*) yang diwakili dengan sepasang tanda “<<” dan “>>” untuk membantu agar diagram tetap mudah terbaca.



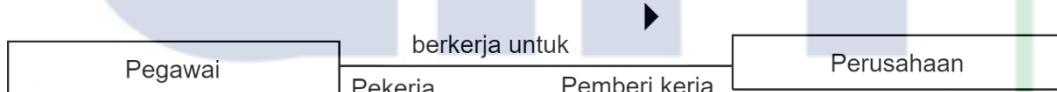
Gambar 2.8 *Class dengan tanda stereotype*

**Asosiasi/association** *Class* terjadi antara dua *Class* atau lebih yang saling berhubungan dengan konsep sistem tertentu. Dua *Class* dihubungkan dengan sebuah garis, kemudian hubungan tersebut diberi nama, dan diberikan bentuk segitiga diarsir untuk menunjukkan arah hubungan.



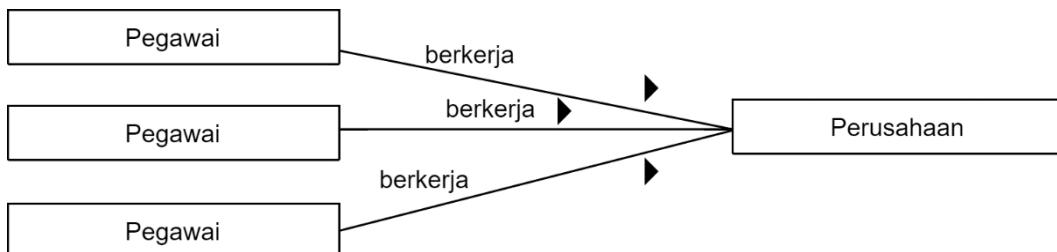
Gambar 2.9 *Asosiasi dua Class*

Ketika sebuah *Class* memiliki hubungan dengan *Class* lain, mereka memiliki peran pada hubungan tersebut. Hal ini bisa ditunjukkan dengan memberikan teks pada masing-masing ujung garis asosiasi.



Gambar 2.10 *Peran pada asosiasi dua Class*

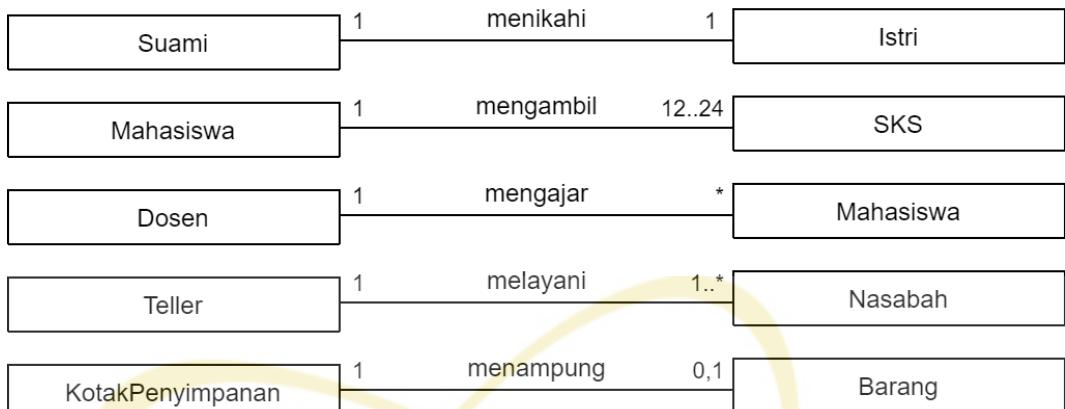
Asosiasi juga dapat melibatkan beberapa *Class* terhubung dengan satu *Class*. Hal ini membuat asosiasi menjadi lebih kompleks. Biasanya *Class* yang memiliki hubungan seperti ini adalah *Class* yang memiliki jenis asosiasi yang sama dengan *Class* yang dituju. (Munawar, 2018)



Gambar 2.11 Asosiasi beberapa *Class* dengan sebuah *Class*

Asosiasi *Class* pada UML mengenal istilah *cardinality* dan *multiplicity*. *Cardinality* merupakan nilai spesifik dari jumlah elemen yang bisa dimiliki oleh satu *Class*. Sedangkan, *Multiplicity* mendefinisikan kemungkinan jangkauan *cardinality* yang bisa dimiliki oleh *Class*. *Multiplicity* menggambarkan dengan pertanyaan berapa banyak objek dari satu *Class* dapat berhubungan dengan satu objek yang diberikan pada *Class* lain. (Zhou, Li, Lv, & Zhang, 2013)

Ada beberapa jenis *multiplicity* yang dapat dipakai dalam *Class Diagram*. Sebuah *Class* bisa memiliki hubungan satu ke satu (*one-to-one*), satu ke banyak (*one-to-many*), satu ke satu atau lebih (*one-to-one or more*), satu ke nol atau satu, satu ke sebuah interval tertentu (misalnya satu ke sepuluh sampai dengan dua puluh), dan satu ke sejumlah n. UML memberikan notasi untuk menggambarkan *multiplicity*. Penulisan angka pada *multiplicity* menunjukkan jumlah *cardinality* yang bisa dimiliki sebuah *Class* apabila memiliki asosiasi dengan *Class* lain yang dituju. Tanda bintang (\*) merujuk pada kata “banyak”. Tanda titik dua (..) merujuk pada kata “atau”, contohnya 1..\* yang berarti “satu atau lebih”. Kata “atau” bisa juga dinotasikan dengan tanda koma (,), contohnya seperti 1,\* yang berarti juga “satu atau lebih”. Ketika misal *Class A* dan *Class B* memiliki asosiasi, kedua *Class* memiliki *cardinality* 1 untuk *Class A* dan *cardinality* nol (0) untuk *Class B*, maka *Class B* menjadi pilihan (*optional*) bagi *Class A*. Perhatikan gambar 2.13 untuk memahami macam-macam *multiplicity*. (Munawar, 2018; Zhou dkk., 2013)



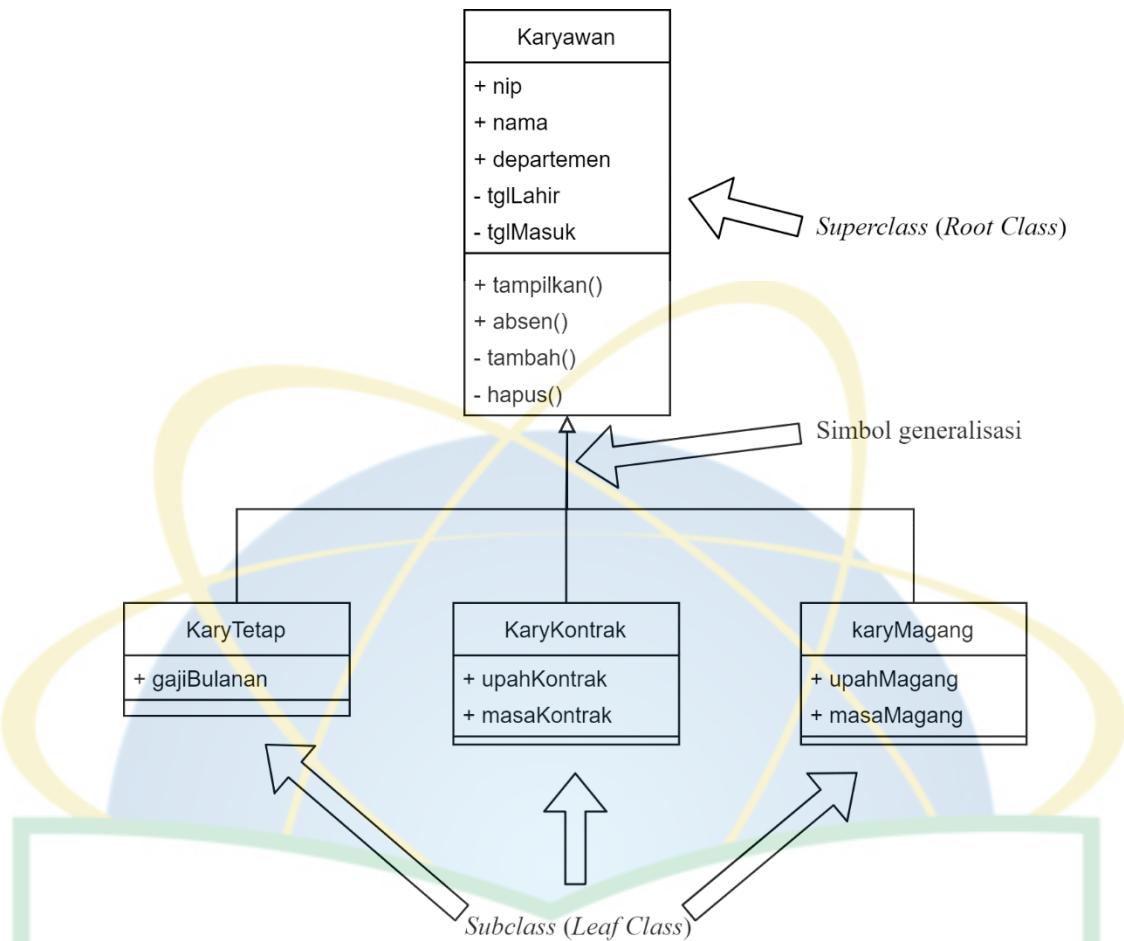
**Gambar 2.12 Contoh Multiplicity**

**Generalisasi/generalization** menjelaskan hubungan kesamaan antar *Class* dengan mengurnya secara hierarkis. Penyusunan secara hierarki ini memperlihatkan persamaan dan perbedaan antar *Class*. Penyusunan dengan cara ini juga memudahkan kita untuk menambah *Class* baru tanpa mengubah *Class* yang sudah ada. Hal ini dalam pengembangan sistem memudahkan untuk menambahkan fitur baru dari sistem yang sudah dikembangkan sebelumnya. Kita bisa membuat sebuah *Class* baru untuk satu fitur baru yang akan kita kembangkan, kemudian mencari *Class* yang memiliki kesamaan dengan *Class* baru tersebut.

Implementasi generalisasi disebut dengan mekanisme *Inheritance/pewarisan*. Ketika dua *Class* atau lebih dihubungkan dengan mekanisme *inheritance*, maka satu *Class* yang lebih umum disebut dengan *superclass*, sedangkan yang lainnya merupakan *Class* yang lebih umum disebut dengan *subclass*. *Inheritance* memiliki aturan sebagai berikut:

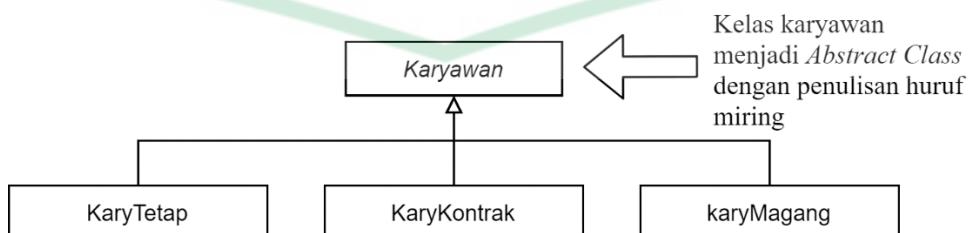
- *Subclass* selalu mewarisi semua sifat dari *superclass*-nya, dalam hal ini berarti *subclass* mewarisi semua properti/atribut dan fungsi/*operation* *superclass*-nya.
- *Subclass* selalu memiliki setidaknya satu detail perbedaan (properti atau fungsi) yang tidak diturunkan dari *superclass*-nya.

*Inheritance* dan generalisasi memiliki hubungan yang sangat dekat. Generalisasi menjelaskan hubungan logis antar *Class* yang memiliki karakteristik yang sama, sedangkan *inheritance* menjelaskan mekanisme berbagi (*sharing*) karakteristik antar *Class*. Perhatikan gambar 2.14 untuk memahami mekanisme ini lebih lanjut.



**Gambar 2.13 Mekanisme Inheritance**

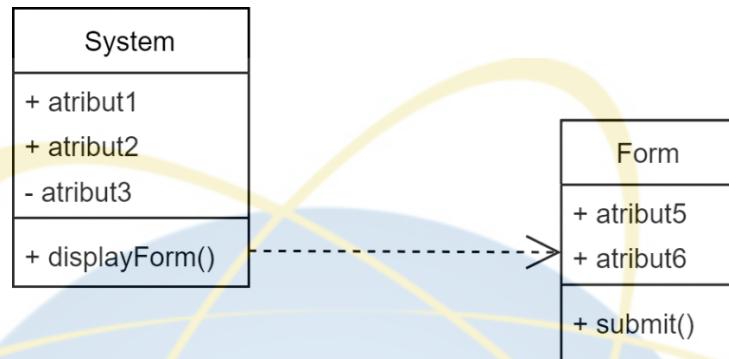
*Superclass* biasanya digunakan untuk membuat definisi fitur yang sering dipakai bersama oleh beberapa *Class* yang akan menjadi anaknya (*subclass*). hal ini membuat *superclass* sering kali tidak memiliki objek (*instance*) sama sekali. *Superclass* yang tidak memiliki objek atau tidak akan memiliki objek disebut sebagai ***abstract class***. UML memberikan penulisan *abstract class* dengan menuliskan nama *Class* dengan huruf miring. Perhatikan gambar 2.15.



**Gambar 2.14 Abstract Class karyawan karena tidak memiliki instance**

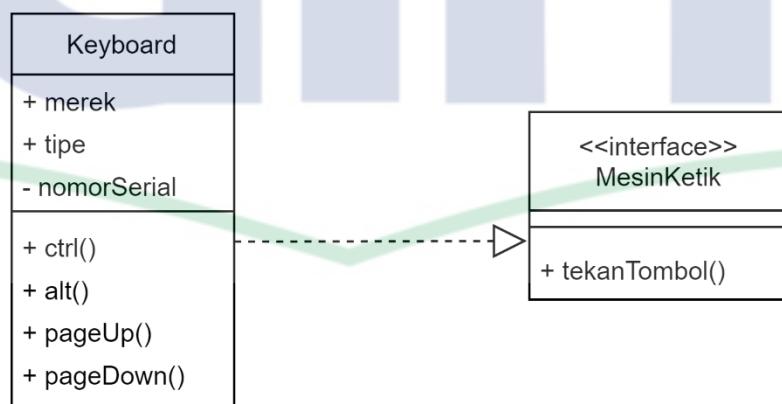
***Dependency Class*** kadang diperlukan untuk membuat sebuah *Class* yang memiliki fungsi dengan cara menggunakan *Class* lain. Penggunaan *Class* lain

diperlukan agar fungsi pada *Class* tersebut dapat diselesaikan. Biasanya *dependency* digunakan apabila satu proses dapat menyelesaikan beberapa kasus masalah. Pembuatan *Dependency Class* dapat mengurangi penulisan dan pemeliharaan kode yang sama berulang kali (*redundant*).



Gambar 2.15 Dependensi class *Form* terhadap class *System*

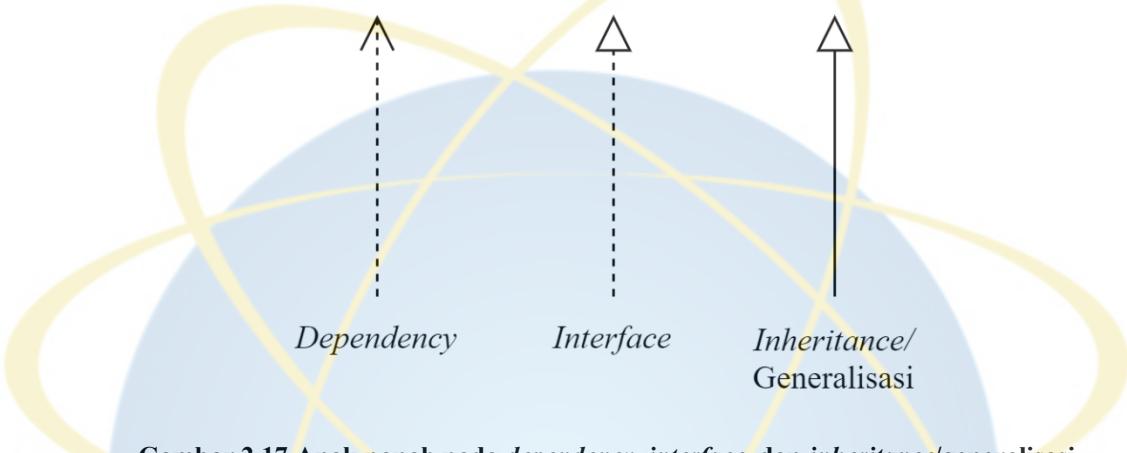
**Interface** merupakan satu set operasi yang memberikan spesifikasi aspek dari perilaku dan operasi dari satu *Class* ke *Class* yang lain. *Interface* dapat menjadi *subset* dari operasi *Class* tetapi tidak semuanya. Pemodelan *interface* sama seperti pemodelan *Class*, namun *interface* tidak memiliki atribut/properti. Relasi antara *Class* dan *interface* disebut dengan **Realization**. Misalnya *keyboard* pada komputer yang merupakan *interface* yang bisa dicopot pasang dan dipakai ulang. Tombol-tombol operasi dan penempatan pada *keyboard* sebenarnya berasal dari mesin ketik. Namun, ada beberapa fungsi yang tidak ada pada mesin ketik, seperti *control*, *alt*, *page up*, *page down*, dll.



Gambar 2.16 Contoh *Interface*

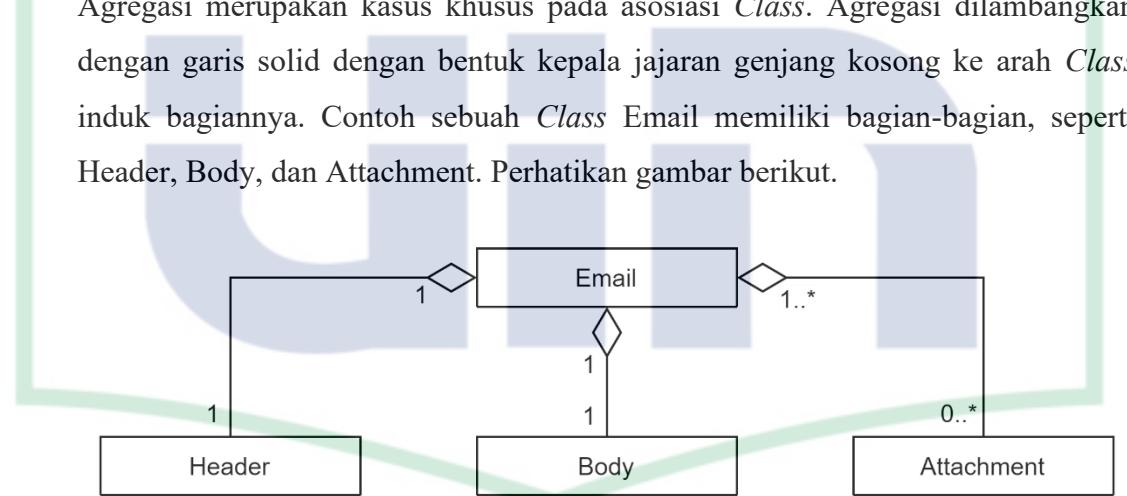
Pemodelan relasi *dependency*, *interface*, dan *inheritance* pada UML sama-sama menggunakan garis anak panah. Pembeda dari ketiganya terlihat dari bentuk

garis anak panah yang mereka gunakan. *Dependency* menggunakan garis panah putus-putus dengan bentuk kepala panah merupakan garis segitiga tanpa alas. *Interface* menggunakan garis panah yang sama dengan *Dependency*, hanya saja bentuk kepala panah merupakan bentuk segitiga tanpa isi sama seperti bentuk kepala panah *inheritance*. Sedangkan *inheritance* menggunakan garis solid tidak putus-putus. Perhatikan gambar berikut untuk memahami lebih lanjut.



Gambar 2.17 Anak panah pada *dependency*, *interface*, dan *inheritance/generalisasi*

**Agregasi** menjelaskan suatu *Class* adalah bagian (*part of*) dari *Class* lain. Agregasi merupakan kasus khusus pada asosiasi *Class*. Agregasi dilambangkan dengan garis solid dengan bentuk kepala jajaran genjang kosong ke arah *Class* induk bagiannya. Contoh sebuah *Class* Email memiliki bagian-bagian, seperti Header, Body, dan Attachment. Perhatikan gambar berikut.

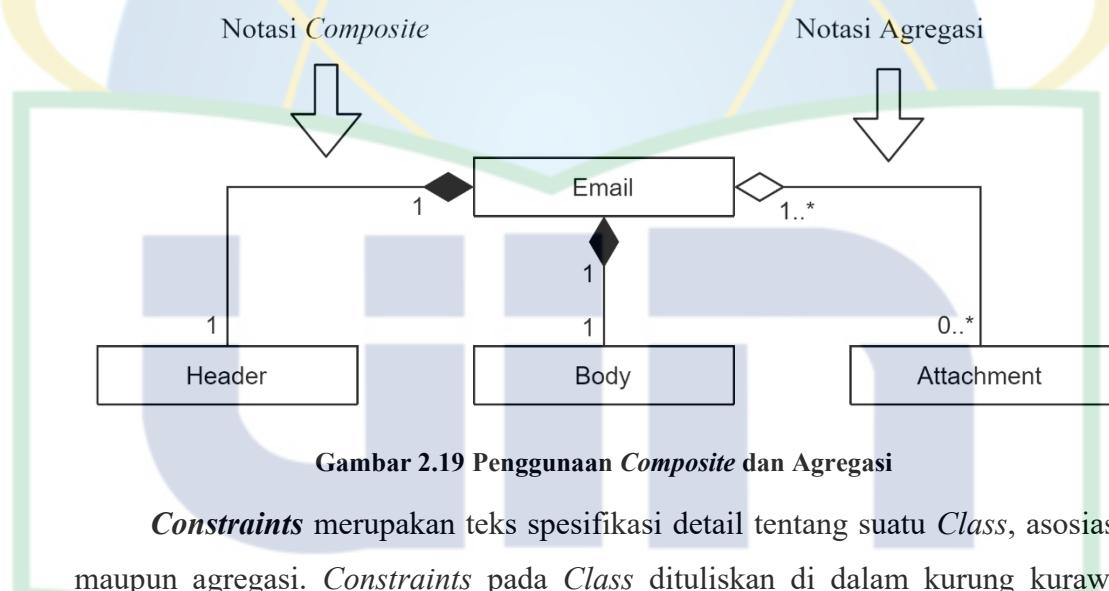


Gambar 2.18 Contoh Agregasi

Agregasi bisa menunjukkan *multiplicity* layaknya pada asosiasi biasa. Gambar di atas menunjukkan bahwa sebuah email setidaknya terdiri dari Header, Body, dan Attachment. Tidak seperti Header dan Body, Email dapat tidak memiliki atau memiliki banyak Attachment. Begitu pula Attachment dapat disisipkan pada satu atau banyak Email. Apabila Attachment tidak disisipkan ke satu pun Email,

maka hal tersebut tidak bisa disebut sebagai Attachment, melainkan menjadi *file* (berkas) biasa.

*Composite* merupakan tipe agregasi yang menjelaskan hubungan kuat bagian suatu *Class* dengan *Class* lain. Perbedaannya dengan agregasi adalah apabila *Class* induk yang memiliki hubungan dihapus, maka *Class* bagiannya akan terhapus juga. Notasinya pada UML yaitu berupa garis asosiasi dengan bentuk kepala jajar genjang berisi (solid). Contohnya pada kasus Email sebelumnya, sebuah Email harus memiliki Header dan Body. Apabila *Class* Email dihapus, maka Header dan Body juga akan terhapus, sebaliknya Attachment tidak akan terhapus. Kasus tersebut menggambarkan hubungan *composite* antara Email dengan Header dan Email dengan Body. Sedangkan hubungan antara Email dengan Attachment adalah Agregasi. Perhatikan gambar berikut untuk memahami lebih lanjut.

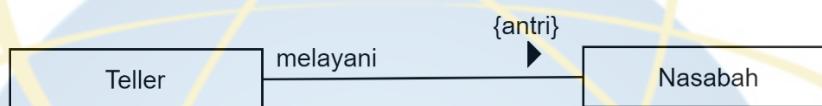


Gambar 2.19 Penggunaan *Composite* dan Agregasi

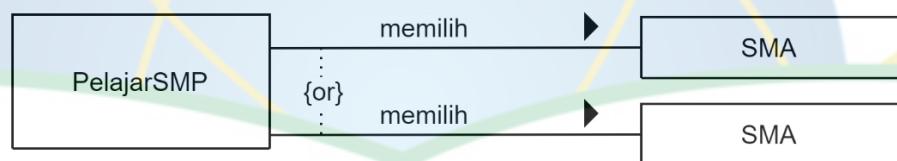
*Constraints* merupakan teks spesifikasi detail tentang suatu *Class*, asosiasi, maupun agregasi. *Constraints* pada *Class* dituliskan di dalam kurung kurawal ditaruh di sebelah atribut atau fungsi yang memerlukan penjelasan tentang aturan tertentu. *Constraints* pada asosiasi menunjukkan aturan asosiasi tertentu. *Constraints* pada asosiasi diletakkan di dekat garis asosiasi di dalam kurung kurawal. *Constraints* pada agregasi menunjukkan aturan agregasi dengan menjelaskan pada hubungan agregasi, biasanya hal ini dilakukan untuk menjelaskan bahwa hubungan agregasi *Class* memiliki aturan relasi OR untuk 2 atau lebih *Class* yang berhubungan. Perhatikan gambar berikut untuk memahami lebih lanjut.



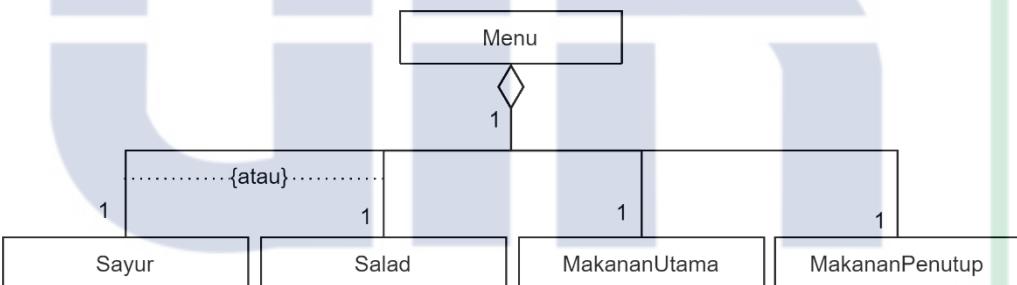
**Gambar 2.20 Constraints pada atribut *Class*; atribut kapasitas memiliki tiga kemungkinan nilai, yaitu 7 kg, 12 kg, ataupun 15 kg**



**Gambar 2.21 Constraints pada asosiasi *Class***



**Gambar 2.22 Relasi OR pada dua asosiasi dalam sebuah *Constraints***



**Gambar 2.23 Constraints pada agregasi untuk menunjukkan komponen mana yang akan dipakai dengan relasi “atau”**

Kita telah memahami pembahasan terkait *Class* sampai kepada relasi relasinya, dengan begitu kita siap untuk membuat *Class Diagram*. Seperti yang telah dijelaskan sebelumnya, *Class Diagram* menunjukkan koleksi *Class*, *interface* (antarmuka), asosiasi, kolaborasi, dan *constraints*. *Class Diagram* merupakan satu-satunya diagram struktural UML yang dapat dipetakan langsung ke bahasa pemrograman berorientasi objek. Kumpulan *Class* pada *Class Diagram* di UML

digunakan sebagai cetak biru (*blueprint*) pembuatan *Class* pada tahap pengembangan (*development*) juga dan sangat dekat sekali hubungannya. Hal ini yang membuat *Class Diagram* paling disukai oleh kalangan pengembang sistem (*developer*). Pembuatan *Class Diagram* bertujuan untuk menggambarkan pandangan statis sistem dan kolaborasi elemen-elemennya, menjelaskan tanggung jawab atau fungsi suatu sistem, dan konstruksi aplikasi menggunakan bahasa OOP. *Class Diagram* juga dapat digunakan untuk membuat aplikasi dengan teknik *reverse* maupun *backward engineering*. (Munawar, 2018)

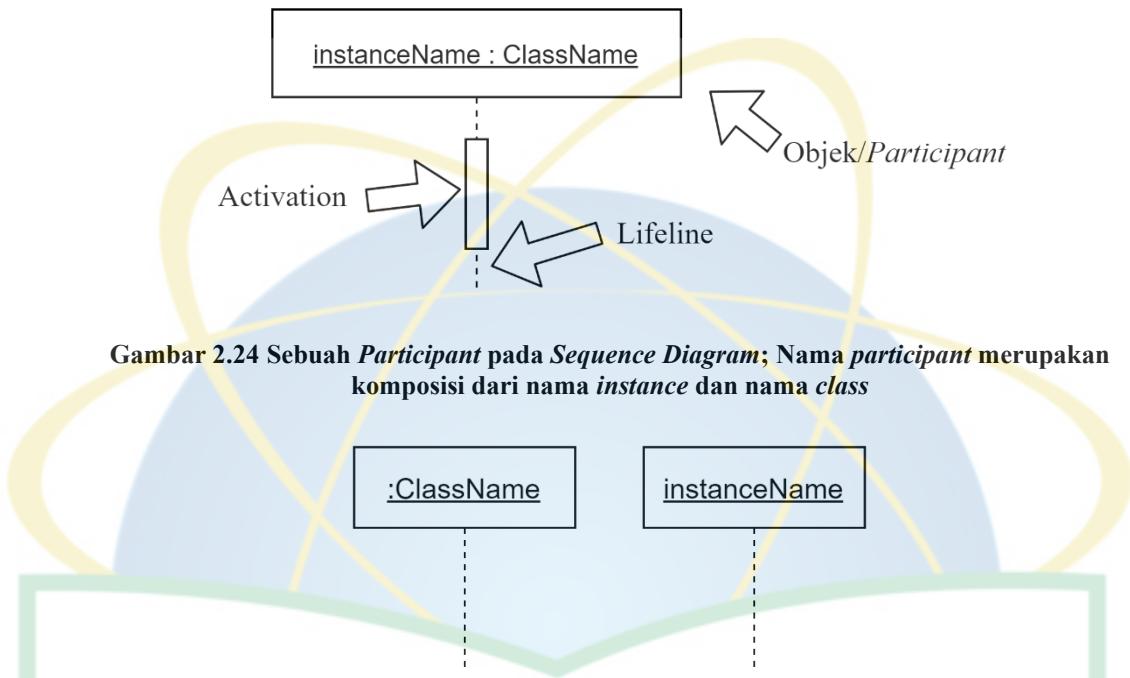
Ada beberapa hal yang perlu diperhatikan sebelum membuat *Class Diagram*, di antaranya sebagai berikut:

- Pembuatan *Class Diagram* pada tahapan desain sistem hanya dilihat dalam konteks level atas, walaupun nantinya perlu mempertimbangkan banyak properti yang harus ditambahkan.
- Nama *Class Diagram* harus memiliki makna yang menggambarkan aspek sistem.
- Setiap elemen, asosiasi, atribut, dan fungsi *Class* harus diidentifikasi dengan jelas.
- Jumlah minimum properti perlu ditentukan untuk setiap *Class*, agar tidak menyulitkan pembacaan diagram karena terdapat properti yang tidak perlu.
- Gunakan catatan untuk menjelaskan beberapa aspek diagram apabila diperlukan, namun tetap harus dimengerti oleh pengembang (*developer*).
- Kumpulan *Class Diagram* merepresentasikan keseluruhan sistem, pastikan *Class Diagram* benar-benar menggambarkan keadaan riil dari sistem.

#### 2.7.4 Sequence Diagram

*Sequence Diagram* menggambarkan perilaku pada sebuah skenario dengan menunjukkan sejumlah contoh objek yang saling berinteraksi untuk bertukar *message* (pesan). Pembuatan *Sequence Diagram* bertujuan untuk menunjukkan urutan waktu aliran pesan dari satu objek ke objek lain. *Sequence Diagram* digunakan untuk mengetahui perilaku pada satu *Use Case*. Apabila *Activity Diagram* menunjukkan interaksi antara pengguna dan sistem, *Sequence Diagram* menunjukkan interaksi antara objek-objek sistem di dalamnya secara lebih spesifik dan menunjukkan aliran pesan yang terjadi. (Munawar, 2018)

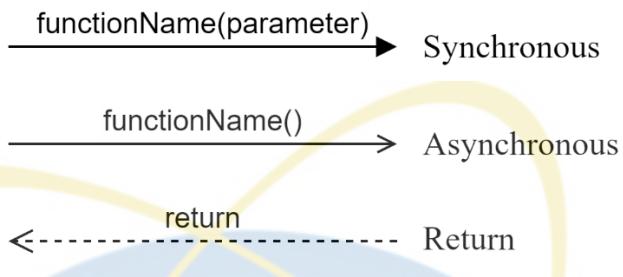
Ada beberapa notasi yang digunakan untuk membuat *Sequence Diagram*, yaitu *participant/objek*, *message*, *recursive*, menghapus dan menambahkan *participant*, dan *frame*. Pembahasan pertama terkait *participant* digambarkan dengan gambar berikut ini.



**Gambar 2.25** *Participant* dengan hanya nama *class* (kiri) dan *participant* dengan nama objek dari *class* yang tidak diketahui (kanan)

Gambar di atas menunjukkan sebuah objek/participant pada *Sequence Diagram*. Penggambaran **objek/participant** menggunakan bentuk kotak persegi panjang. UML 1 memberikan istilah objek, sedangkan istilah objek pada UML 2 diganti menjadi *participant*. Pemberian nama *participant* bisa dilakukan dengan tiga cara, yaitu pertama dengan memberi nama *instance* dan *class*-nya dengan dipisah titik dua (“*InstanceName : ClassName*”), kedua bisa dengan memberi nama *class*-nya saja (“*:ClassName*”), dan ketiga bisa dengan memberikan nama *instance*-nya saja apabila *class*-nya belum diketahui (“*InstanceName*”). Setiap *participant* memiliki garis putus-putus ke bawah yang disebut **Lifeline** berfungsi untuk menunjukkan masa aktif/hidup objek selama skenario digambarkan. Setiap *lifeline* terdapat bentuk kotak berada di atas *lifeline* yang disebut **activation** berfungsi untuk menunjukkan waktu operasi pada aliran tertentu. (Munawar, 2018; Torlak, 2015)

*Sequence Diagram* menggambarkan aliran *message* antara satu *participant* ke *participant* lain. Notasi berikut menunjukkan macam-macam aliran *message* pada *Sequence Diagram*.

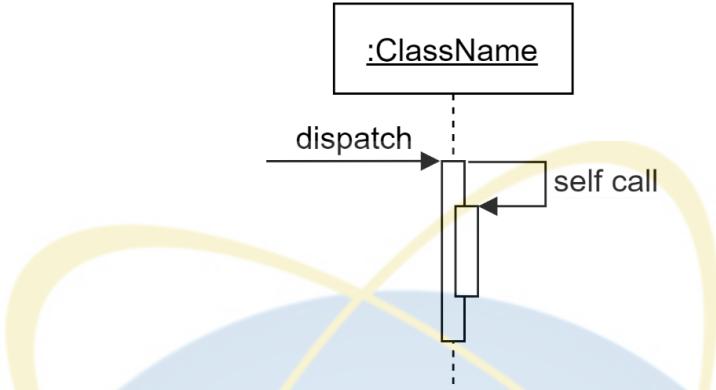


Gambar 2.26 Notasi-notasi *Message*

Notasi **message** berfungsi untuk menunjukkan aliran pesan terjadi antara satu *participant* ke *participant* lain. Ada empat notasi untuk penggambarannya, yaitu **synchronous**, **asynchronous**, dan **return**. Notasi **message synchronous** berfungsi untuk mengirimkan sebuah *message*, jika notasi ini digunakan maka jawaban dari *message* tersebut akan ditunggu sebelum menjalankan proses berikutnya. Notasi **message asynchronous** berfungsi untuk mengirimkan *message* secara pararel, dengan begitu proses berikutnya akan langsung dijalankan tanpa menunggu jawaban dari *message* pertama. Notasi **return** menunjukkan jawaban dari pesan yang dikirim oleh *participant* yang mengirim *message*.

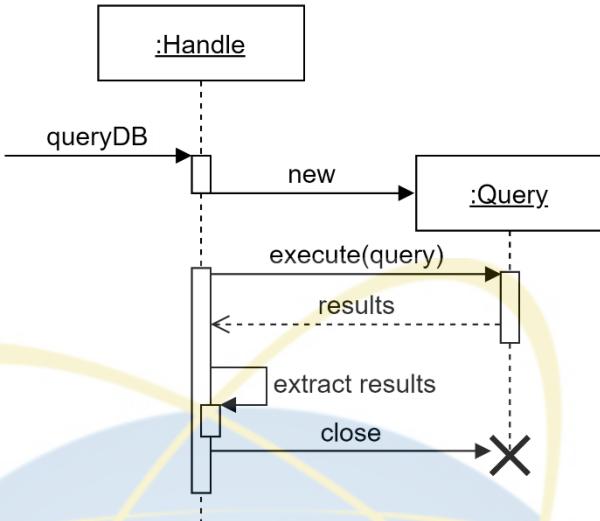
Sebuah *participant* bisa mengirimkan pesan kepada dirinya sendiri yang disebut dengan **Recursive**. Hal ini berguna untuk menggambarkan sebuah fungsi yang terdapat pada *participant* itu sendiri. Konsep OOP mengajarkan untuk membuat objek yang banyak tetapi memiliki fungsi yang lebih sedikit ketimbang satu objek memiliki fungsi yang lebih banyak. Hal ini membuat dua konsep aliran kontrol yang berbeda, yaitu kontrol tersentralisasi dan terdistribusi. Kontrol tersentralisasi membuat sebuah *participant* memungkinkan untuk memiliki notasi **recursive** untuk menjalankan fungsi yang ada pada dirinya sendiri. Sedangkan, kontrol terdistribusi membuat sebuah *participant* akan selalu menggunakan fungsi dari *participant* lain sehingga kontrol *participant* terbagi ke berbagai *participant* lainnya. **Recursive** digambarkan dengan anak panah berbalik arah ke **nested activation** yang menunjukkan aliran berbalik ke arahnya. **Nested activation** merupakan notasi *activation* yang diletakkan bertumpuk dengan notasi utama.

Gambar berikut menjelaskan notasi *recursive* pada *Sequence Diagram*. (Munawar, 2018; Torlak, 2015)



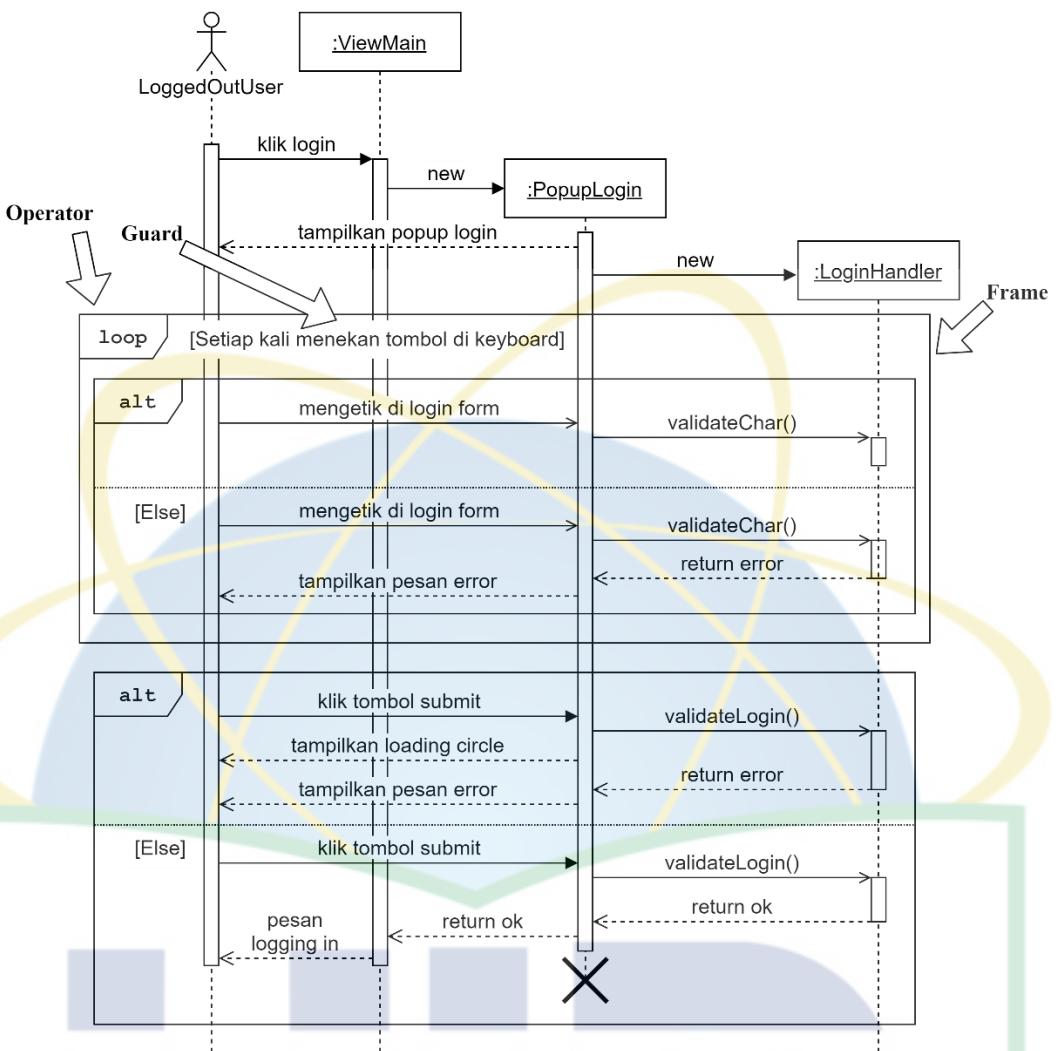
**Gambar 2.27 Recursive pada Sequence Diagram**

Terkadang dalam OOP kita perlu menghapus dan membuat objek baru untuk menunjukkan keterlibatannya dalam sistem. *Sequence Diagram* dapat menggambarkan hal ini dengan membuat panah *message* mengarah secara langsung ke kotak *participant* untuk menunjukkan bahwa itu *participant* baru yang telah dibuat. Biasanya panah *message* diberi nama “new”, namun sebenarnya pencantuman nama *message* tidak diwajibkan. Sedangkan, penghapusan *participant* bisa menggunakan notasi X besar di akhir *lifeline participant*. Apabila sebuah panah mengarah ke notasi X tersebut maka sebuah *participant* menghapus *participant* lainnya. Apabila notasi X ada di akhir *lifeline* dan tidak ada tanda panah apapun yang mengarah kepadanya, maka dapat dikatakan bahwa *participant* tersebut menghapus dirinya sendiri. Perhatikan gambar berikut untuk memahami lebih lanjut cara penggunaannya.



**Gambar 2.28 Menghapus dan membuat *participant***

Kontrol aliran (*control flow*) pemrograman tidak selalu menunjukkan jalan yang lurus saja, melainkan akan ada di mana aliran akan terpisah dengan kondisi tertentu. Hal ini dikenal dengan perulangan (*looping*) dan persyaratan (*conditional*). Jika kita kembali lagi ke awal, tujuan dari *Sequence Diagram* adalah untuk menunjukkan interaksi objek saja. Pemodelan logika bisa digambarkan dengan *Activity Diagram* atau kode pemrograman langsung. Namun, *Sequence Diagram* tetap memberikan notasi untuk menggambarkan hal tersebut menggunakan *interaction frame*. Perhatikan gambar berikut.



Gambar 2.29 Penggunaan *interaction frame* pada *Sequence Diagram*

Setiap *frame* memiliki operator yang diletakkan di pojok kiri atasnya yang berisi **guard**. Contoh gambar tersebut menunjukkan penggunaan *frame* pada operasi perulangan dan kondisional. Operasi perulangan menggunakan operator *loop*, sedangkan operasi kondisional menggunakan operator *alt*. *Frame* dengan operator *alt* memiliki dua fragmen untuk menunjukkan operasi yang berbeda, hanya fragmen yang memiliki kondisi “*true*” yang akan dijalankan. Ada beberapa jenis *guard* lain yang dapat digunakan, selengkapnya dijabarkan pada tabel berikut ini. (Munawar, 2018)

**Tabel 2.5 Operator-operator umum yang digunakan pada *interaction frame***

Operator	Keterangan
alt	Alternatif dari beberapa fragmen; fragmen yang kondisinya bernilai “True” yang akan dijalankan.
opt	Opsional; sama seperti alt hanya saja terdiri dari satu fragmen.
par	Pararel; setiap fragmen dijalankan bersamaan.
loop	Perulangan ( <i>Looping</i> ); fragmen dijalankan berulang kali; <i>guard</i> menunjukkan basis iterasinya.
region	<i>Critical Region</i> ; fragmen hanya memiliki satu <i>thread</i> untuk dijalankan.
neg	<i>Negative</i> ; fragmen menunjukkan <i>interaction</i> yang salah.
ref	<i>Reference</i> ; fragmen menunjukkan sebuah <i>interaction</i> yang didefinisikan pada diagram yang lain; <i>frame</i> ini diarahkan dengan anak panah <i>message</i> dari <i>participant</i> dan memberikan <i>return</i> pula.
sd	<i>Sequence Diagram</i> ; digunakan sebagai batas keseluruhan <i>sequence diagram</i> jika diperlukan.

**CRC Card (Class Responsibility Collaboration)** berguna untuk membantu menemukan interaksi antar objek. Fokus CRC Card adalah pada perilaku bukan pada data. CRC Card memberikan gambaran tanggung jawab sebuah *class* dan melihat kolaborasi di setiap tanggung jawab tersebut terhadap *class* lain. Tanggung jawab ini merupakan fitur dari sebuah *class*, apa yang bisa dia lakukan dalam sebuah sistem, aksi apa yang harus dijalankannya pada kondisi tertentu. CRC Card berguna untuk mempermudah penentuan bagaimana sebuah *class* diimplementasikan. Kartu-kartu di CRC Card bisa langsung dipindahkan ke *Sequence Diagram* yang menggambarkan interaksi *message* dan interaksi antar objek/*participant*.

Cara membuat CRC Card cukup mudah. CRC Card terdiri dari tiga bagian, yaitu nama *class*, *responsibility/tanggung jawab*, dan *collaboration/kolaborasi*. Nama *class* ditaruh di atas kartu, di sebelah kiri kartu merupakan daftar tanggung jawab *class*, dan di sebelah kanan berisi daftar *class* yang berkolaborasi dengan

*class* yang didefinisikan. Definisi tanggung jawab setidaknya ringkas namun jelas dan tidak melebihi ukuran kartu agar memudahkan dalam membaca CRC *Card* yang dibuat. Perhatikan tabel berikut untuk memahami bentuk CRC *Card*. (Keller dkk., 2019; Munawar, 2018)

**Tabel 2.6 Contoh CRC *Card***

PinjamBuku	
Cek apa buku ada	ArsipPerpus
Tentukan waktu peminjaman	
Cek status peminjaman buku	Mahasiswa
Pengembalian buku	
Bayar denda keterlambatan	KasPerpus

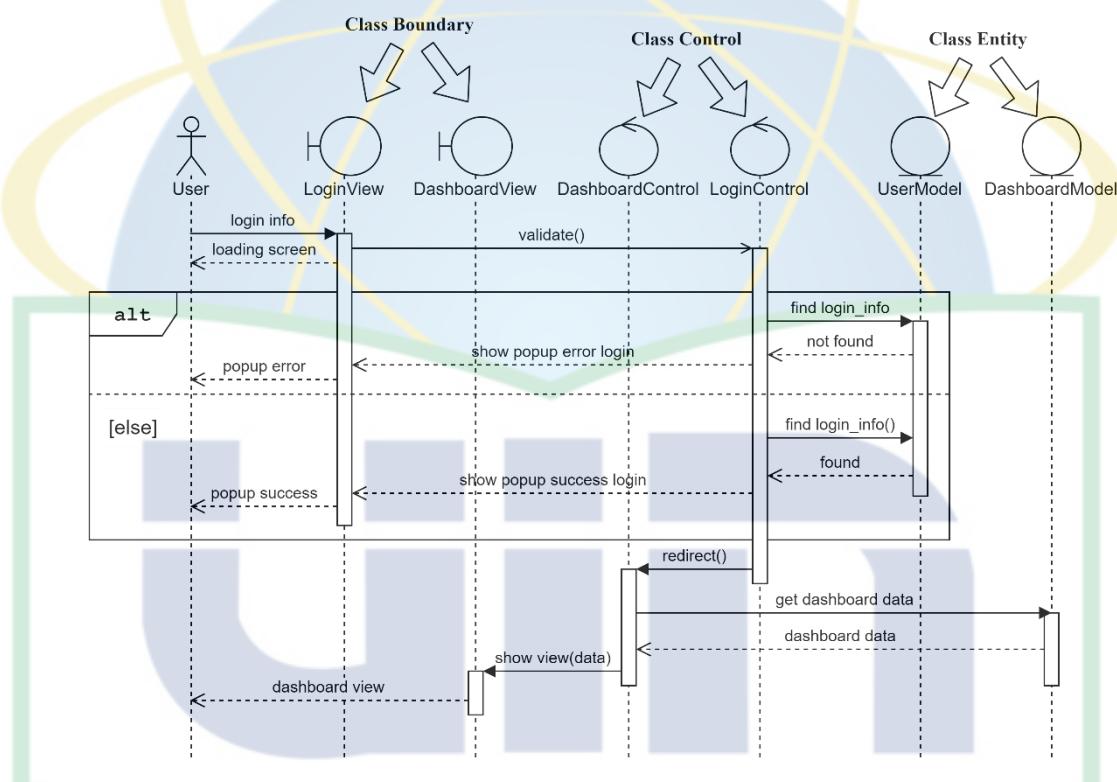
Pembahasan *class* dalam *Sequence Diagram* bisa menggunakan notasi **Stereotype Class**. Notasi ini meliputi beberapa aturan dalam beberapa jenis *class*, seperti *boundary*, *control*, dan *entity*. Hal ini berguna agar *Sequence Diagram* dapat menjelaskan bentuk *class* yang berbeda dengan lebih jelas.

*Boundary* merupakan *class* yang berhubungan dengan antarmuka (*user interface*). *Boundary* tidak hanya terkait antarmuka dengan aktor manusia, melainkan juga meliputi sistem lain ataupun perangkat lain. *Boundary* yang menghubungkan dengan sistem lain mencegah agar tidak mengubah proses logika kontrol di dalam suatu sistem. Hal ini membuat batasan antara satu sistem dengan sistem yang lain. *Boundary* juga menggambarkan hubungan suatu sistem dengan perangkat lain, seperti sensor dan printer. Hal ini memperlihatkan komunikasi yang terjadi di *port interface* yang terhubung.

*Control* merupakan *class* yang berhubungan dengan aliran kontrol aplikasi. *Control* melakukan koordinasi tugas dari objek lain yang sedang menjalankan suatu fungsi dan mendelegasikan pekerjaan tersebut ke objek yang bertanggung jawab berikutnya. *Control* dapat berpartisipasi dalam beberapa *use case*. Selain itu, beberapa *Control* yang berbeda dapat berpartisipasi dalam satu *use case*. Namun, tidak semua *use case* memerlukan *Control*.

*Entity* merupakan *class* yang mengatur perubahan informasi jangka panjang pada aplikasi. Perubahan informasi jangka panjang ini biasanya pada basis data yang ditujukan sebagai tempat penyimpanan informasi dari hasil pengolahan oleh *class Control*. (Munawar, 2018)

Gambar berikut menjelaskan *use case* Login pada sebuah aplikasi. Penggunaan notasi *stereotype* memudahkan kita untuk lebih memahami jenis dari *class* dan memetakan penempatannya. Penelitian ini menggunakan *Sequence Diagram* dengan notasi *stereotype* untuk mendapatkan manfaat tersebut. Perhatikan gambar *Sequence Diagram* dengan notasi *stereotype* berikut.



Gambar 2.30 Sequence Diagram dengan notasi stereotype

## 2.8 Iterative Model

Metode ini merupakan salah satu dari Siklus Hidup Pengembangan Sistem (*System Development Life Cycle* yang disingkat SDLC). Metode ini sebenarnya adalah versi mini dari metode Waterfall dengan ditambahkannya tahapan masukan (*feedback*) dari satu fase iterasi. Masukan ini menyediakan informasi yang penting untuk proses desain pada fase iterasi berikutnya. Metode ini menghasilkan produk perangkat lunak di akhir dari setiap tahapan iterasi, kemudian perangkat lunak

menuju ke lingkungan sistem yang sebenarnya sebagai rilis tambahan (*Incremental Releases*). (Alshamrani & Bahattab, 2015; Okesola dkk., 2020)

### 2.8.1 Tahapan-tahapan Iterative Model

*Iterative Model* merupakan metode yang mengedepankan iterasi dalam pengembangan aplikasi. Metode ini tidak memerlukan desain yang lengkap untuk keseluruhan aplikasi, namun desain dapat diperbarui seiring dengan penambahan fitur aplikasi. Metode ini memiliki dua fase, adapun berikut fase dalam pengembangan dengan *Iterative Model*: (Alshamrani & Bahattab, 2015; Yaseen dkk., 2019)

#### 1. *Requirement (Kebutuhan)*

Fase ini membahas persiapan awal untuk mengembangkan aplikasi, seperti menyiapkan kebutuhan perangkat keras (*hardware*) untuk keperluan pengembangan, perangkat lunak untuk membantu pengembangan sistem (*software tools*), dan komponen penting lainnya. Fase ini juga melakukan analisis terhadap lingkungan produksi agar bisa menentukan teknologi apa yang bisa digunakan. Fase ini kemudian akan diulas setelah tahapan terakhir *iteration* apabila ingin memulai iterasi berikutnya.

#### 2. *Iteration (Iterasi)*

Fase ini merupakan aktivitas untuk mengembangkan aplikasi. Fase ini dilakukan secara berulang terus menerus seiring adanya ide fitur baru. Aplikasi nantinya dikembangkan secara perlahan untuk menambahkan fitur baru. Fase ini memiliki beberapa tahapan untuk memenuhi pengembangan aplikasi:

##### a. *Analysis & Design (Analisis & Desain)*

Tahapan ini merupakan tahapan untuk melakukan analisis dan desain aplikasi. Analisis aplikasi bisa dilakukan dengan klien atau pemilik proyek untuk mengidentifikasi kebutuhan dan melakukan desain sederhana dari fitur aplikasi yang akan dibuat. Pertimbangan *requirement* juga dilakukan di tahapan ini untuk melihat apakah fitur aplikasi yang akan dibuat masih sesuai dengan spesifikasi kebutuhan sistem. Tentunya kebutuhan sistem ini sudah didefinisikan sebelumnya di iterasi sebelumnya.

### **b. Implementation/Development (Implementasi/Pengembangan)**

Tahapan ini merupakan tahapan pembuatan kode program. Prosesnya dilakukan dengan melihat desain yang telah dibuat. Tahapan ini juga memperbarui desain aplikasi secara berkala untuk membuat desain menjadi lebih detail.

### **c. Testing & Revise (Testing & Revisi)**

Developer melakukan proses testing aplikasi setelah selesai membuat kode program aplikasi. Testing bisa dilakukan dengan berbagai cara. Pembahasan testing dibahas pada subbab Pengujian Perangkat Lunak (*Software Testing*) di bawah. Hasil dari testing biasanya akan menemukan beberapa eror yang pastinya akan selalu ada. Eror bisa berupa kesalahan penulisan sintaksis kode program, kesalahan prosedur proses pengolahan informasi, sampai kepada tampilan aplikasi. Eror-eror yang ditemukan tersebut kemudian dicatat dan direvisi. Tahapan ini dilakukan secara berulang sampai eror benar-benar tidak ditemukan atau siap untuk memasuki lingkungan sistem yang sebenarnya (*Production Server*).

### **d. Production (Produksi)**

Developer menyiapkan server produksi untuk memasang aplikasi. Developer melakukan pembaruan fitur apabila tahapan ini dilakukan pada iterasi selanjutnya setelah melakukan iterasi awal. Tahapan ini biasanya berupa instalasi/pemasangan, konfigurasi, pengoperasian, dan membuat perubahan lainnya yang diperlukan.

#### **2.8.2 Kelebihan dan Kekurangan *Iterative Model***

*Iterative model* layaknya metode pengembangan lainnya memiliki kelebihan dan kekurangan. Mungkin tidak semua orang cocok dengan model pengembangan *software* ini. Berikut kelebihan *iterative model*: (Budi, Siswa, & Abijono, 2016)

- Umpulan balik dari pemilik proyek dapat terus menerus didapatkan. Hal ini dapat meningkatkan kualitas proyek, karena keterlibatan pengguna membuat proyek jadi lebih sesuai dengan kebutuhan dan keinginan pemilik proyek.
- Pelaksanaan revisi dapat dilaksanakan di keseluruhan aplikasi maupun fungsi yang spesifik.
- Penyampaian pekerjaan disampaikan di awal proyek. Pemilik proyek menjelaskan detail pekerjaan di awal proyek. Walaupun begitu developer pun tetap bisa menanyakan lebih lanjut detail proyek saat pengembangan aplikasi.

Kekurangan *iterative model* dijabarkan pada uraian berikut: (Budi dkk., 2016; Nugroho, Waluyo, & Hakim, 2017)

- Setiap perulangan pada *iteration model* merupakan struktur kaku yang menyerupai proyek kecil *waterfall*.
- Bisa menyebabkan masalah yang tidak terduga pada desain sistem atau arsitektur. Hal ini bisa terjadi karena *iteration model* melakukan desain sistem seminimal mungkin untuk mempercepat waktu pengerjaan. Akibatnya hal ini bisa meningkatkan waktu pengerjaan proyek secara keseluruhan dan biaya proyek. Hal ini bisa diatasi dengan menambahkan beberapa iterasi baru, namun sayangnya penyelesaian satu masalah bisa memerlukan lebih dari satu iterasi.
- Keterlibatan pemilik proyek di keseluruhan tahapan pengembangan aplikasi dapat menghambat proses pengerjaan proyek. Hal ini bisa terjadi apabila pemilik proyek yang seharusnya aktif dalam proses pengembangan tidak melaksanakan perannya dengan baik. Hal ini ditambah dengan desain sistem seminimal mungkin yang menyebabkan ketergantungan terhadap *feedback* langsung dari pemilik proyek. Hal ini bisa diatasi dengan menjalin komunikasi yang baik dengan pemilik proyek sehingga pelaksanaan proyek tidak terhambat.

### 2.8.3 Alasan Penggunaan *Iterative Model*

Penggunaan *iterative model* cocok untuk penulis gunakan pada penelitian ini berdasarkan dengan alasan-alasan berikut:

- Penulis memerlukan metode pengembangan *software* yang melibatkan umpan balik dari pemilik proyek langsung. Ini diperlukan karena pemilik proyek tidak dapat menyampaikan kebutuhan aplikasi secara teknis. Hal ini yang membuat penulis memerlukan umpan balik langsung dari pemilik proyek.
- Pemilik proyek ingin melihat langsung bagaimana konsep aplikasi yang ada di pikirannya ini berjalan pada sebuah aplikasi. Pengembangannya memerlukan metode yang mengedepankan pengembangan inkremental. Iterasi pada *iterative model* memberikan solusi atas masalah ini. *Iterative*

*model* melakukan pengembangan dengan membaginya menjadi beberapa iterasi agar *software* bisa langsung dilihat oleh pemilik proyek.

- Pemilik proyek tidak membutuhkan prototipe *software*. Mereka ingin langsung mencoba *software* yang sudah dikembangkan. Hal ini membuat metode pengembangan dengan RAD tidak relevan dengan objek penelitian ini.

Pemilik proyek ingin melakukan pengembangan *software* secara berkelanjutan. Mungkin *Agile Development* bisa lebih membantu dari sisi proyek pengembangan *e-learning* ini. Namun, penulis tetap memerlukan tahapan analisis dan desain untuk membuat perancangan sistem agar memenuhi penelitian akademis. *Iterative model* memiliki kriteria tersebut. Pengembangan *software* pada *Iterative Model* terdiri dari iterasi per iterasi. Developer baru dapat membuat iterasi baru apabila ingin mengembangkan fitur baru dengan mudah. Ini karena proses iterasi telah terdokumentasi di penelitian ini. Developer dapat mempelajari iterasi pengembangan *software* di dalam penelitian ini.

## 2.9 Basis Data (*Database*)

Basis data memainkan peran penting dalam penggunaan komputer yang terus berkembang. Basis data merupakan koleksi data yang dikumpulkan dan disusun secara terstruktur. Struktur penyimpanan data pada basis data berupa tabel yang terdiri dari atribut dan *record*. Atribut merupakan kunci data sebagai indeks dari informasi yang disimpan dan dinyatakan dalam kolom, misalnya dalam sebuah buku telepon menyimpan nama, nomor telepon, dan tanggal lahir. Ketiga item pada buku telepon tersebut merupakan atribut, kemudian atribut-atribut ini diisi dengan informasi yang sesuai disebut dengan *record*. *Record* merupakan baris data yang mengisi informasi setiap atribut yang telah disediakan. (Elmasri & Navathe, 2016)

**Tabel 2.7 Contoh tabel kontak pada basis data**

<b>id</b>	<b>telepon</b>	<b>nama</b>	<b>ulang_tahun</b>
CJ-001	6285289675777	Jodi	1998-05-04
CJ-001	6221791872723	Emma	1997-05-22

Tabel di atas menunjukkan bentuk tabel kontak di basis data. Kolom “id” merupakan kunci utama (**Primary Key**) menyimpan identitas unik untuk setiap *record* yang dicatat dan diperlukan oleh sistem untuk menghapus dan memperbarui *record*. (Artemiou, 2018) Atribut telepon bisa saja digunakan untuk menjadi *primary key*, hanya saja bisa jadi ada kemungkinan satu nomor telepon dapat dimiliki oleh dua orang atau lebih.

Basis data merupakan dasar dalam penyediaan dan pengolahan informasi dalam sebuah sistem komputer. Penggunaan basis data membantu dalam menentukan kualitas informasi yang lebih akurat, mendapatkan informasi tepat pada waktunya, dan tentunya informasi yang didapatkan bisa lebih relevan apabila dibandingkan dengan cara biasa. Hal ini karena sistem pada basis data yang menyimpan informasi secara terstruktur memudahkan kita dalam melakukan pencarian dan pengolahan data. Manfaat yang kita dapatkan apabila menggunakan basis data yaitu mengurangi duplikasi data (*redundancy*), meningkatkan hubungan antar data, mengurangi pemborosan ruang penyimpanan, dan penyimpanan data menggunakan basis data memiliki sifat integritas, independen, dan konsistensi yang selalu terjaga. (Latief, 2012)

### 2.9.1 Sistem Manajemen Basis Data (DBMS)

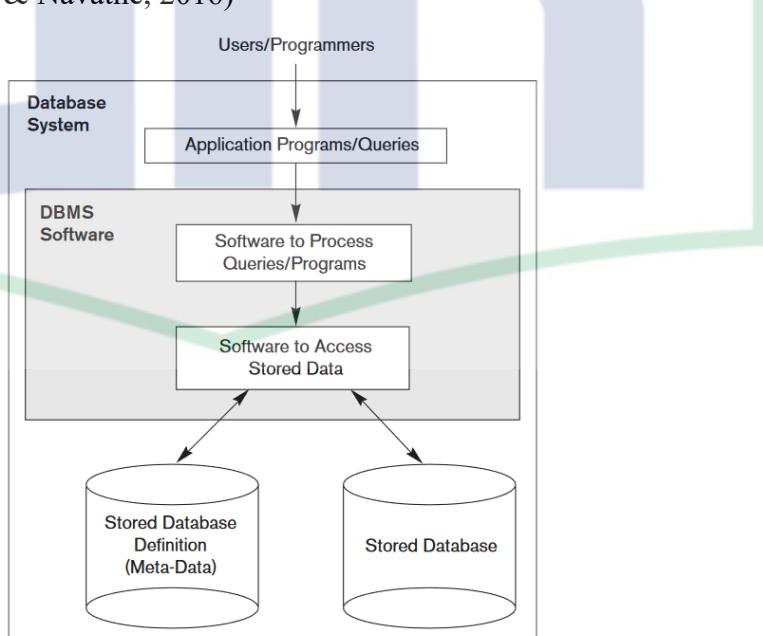
DBMS merupakan sebuah perangkat lunak yang di desain untuk menyediakan cara untuk menyimpan, mengambil, dan mengatur informasi yang terdapat pada sistem basis data. Penggunaan DBMS membuat pengelolaan informasi di sistem basis data menjadi lebih aman. Hal ini karena DBMS memberikan manajemen pengaksesan basis data dengan menerapkan *multi-user* dan mengatur pengaksesan agar sistem basis data tetap aman walaupun dipakai bersama. (C. A. Pamungkas, 2017)

Pengolahan data di DBMS menggunakan bahasa skrip yang disebut dengan **Query**. Terdapat dua jenis bahasa *query* pada basis data, yaitu *Data-Definition Language* (DDL) dan *Data-Manipulation Language* (DML). DDL dan DML pada dasarnya adalah bukan bahasa yang berbeda, melainkan hanya pengelompokan perintah bahasa *query*. DDL berfungsi untuk membuat skema basis data. DML berfungsi untuk melakukan pemasukan, pengolahan, pengambilan, dan perbaruan

data pada skema basis data yang telah dibuat dengan DDL. (Widodo & Kurnianingtyas, 2017)

Program aplikasi mengakses basis data dengan mengirimkan *query* ke DBMS untuk melakukan pemrosesan data. Program aplikasi bisa mendapatkan informasi ataupun menulis informasi ke dalam sistem basis data dengan cara ini. Program aplikasi ini berupa aplikasi yang dibuat oleh developer perangkat lunak secara khusus untuk melakukan pemrosesan data tertentu. Ada juga program aplikasi yang sengaja dibuat hanya untuk melakukan pengolahan basis data secara umum. Biasanya program aplikasi seperti ini menggunakan tampilan antarmuka (*user interface*) untuk menampilkan data dilengkapi dengan berbagai fungsi tombol untuk menjalankan berbagai macam *query* basis data yang sering dilakukan. Hal ini membuat proses untuk melakukan pengolahan data menjadi lebih mudah.

Gambar berikut menggambarkan struktur bagaimana sistem basis data diakses mulai dari pengguna yang mengakses aplikasi berantarmuka atau menjalankan *query* langsung ke perangkat lunak DBMS. Perangkat lunak DBMS memiliki dua bagian lagi di dalamnya, yaitu perangkat lunak untuk memproses *query*/perintah pada aplikasi program dan perangkat lunak untuk mengakses data yang tersimpan. Baru kemudian perangkat lunak ini pada akhirnya bisa mengakses definisi basis data yang tersimpan (*metadata*) dan data yang tersimpan di sistem basis data. (Elmasri & Navathe, 2016)



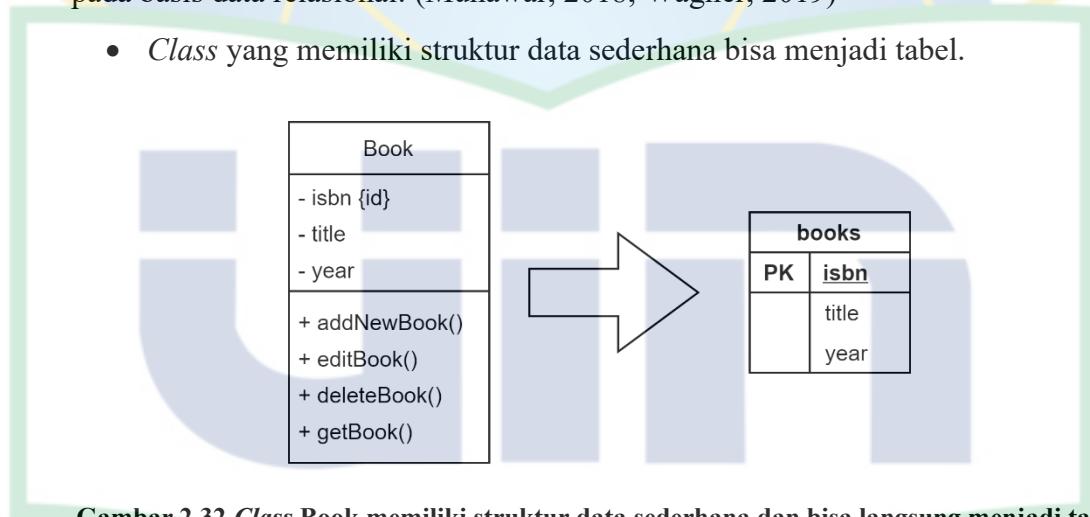
Gambar 2.31 Lingkungan Sistem Basis Data

## 2.9.2 Skema Basis Data dan Pemodelannya dari *Class Diagram*

Skema basis data merupakan sebuah koleksi metadata yang mendeskripsikan beberapa hubungan antara objek dan informasi di dalam satu basis data. Cara termudah untuk memahaminya yaitu dengan membayangkan sebuah kotak yang menyimpan beberapa tabel, prosedur tersimpan (*stored procedures*), tampilan (*views*), dan asset data terkait. Sebuah skema mendefinisikan infrastruktur dari kotak ini. Basis data yang dibuat pada aplikasi DBMS disebut dengan Skema Basis Data. Setelah membuat skema basis data barulah kita bisa mengisinya sesuai dengan struktur skema basis data yang dibuat. (Chapple, 2021)

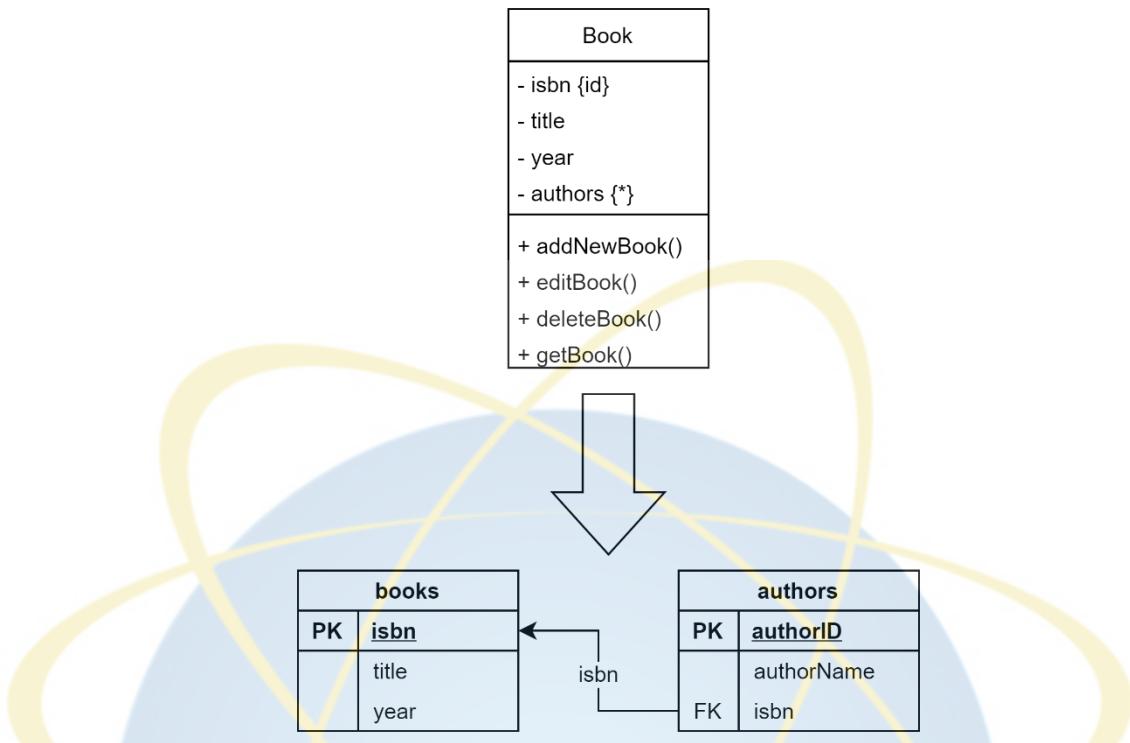
Pembuatan skema basis data sebelumnya didahului dengan membuat perancangan basis data relasional. Penggunaan *Class Diagram* khususnya hal terkait *multiplicity* bisa digunakan untuk memetakan perancangan basis data relasional. Namun, kita tidak bisa sembarangan mengubah satu *Class* menjadi sebuah tabel. Berikut pedoman untuk melakukan pemetaan *Class* menjadi tabel pada basis data relasional: (Munawar, 2018; Wagner, 2019)

- *Class* yang memiliki struktur data sederhana bisa menjadi tabel.



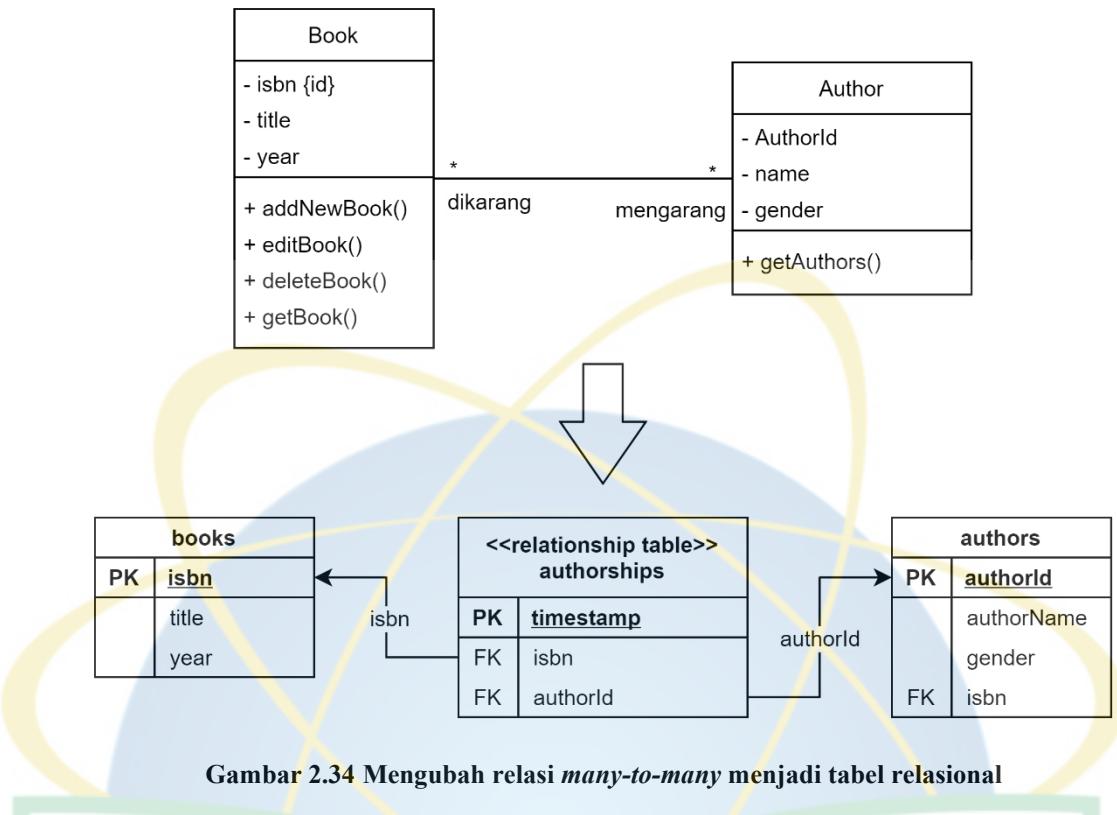
Gambar 2.32 *Class Book* memiliki struktur data sederhana dan bisa langsung menjadi table

- *Identifier* (id) objek bisa menjadi *primary key* di tabel relasional, seperti pada gambar di atas isbn yang merupakan id dari objek *Book* menjadi *primary key* di tabel *books*.
- *Class* yang memiliki *collection* perlu dibuatkan sendiri tabel untuk menampung *collection*-nya dan masing-masing diberikan id.

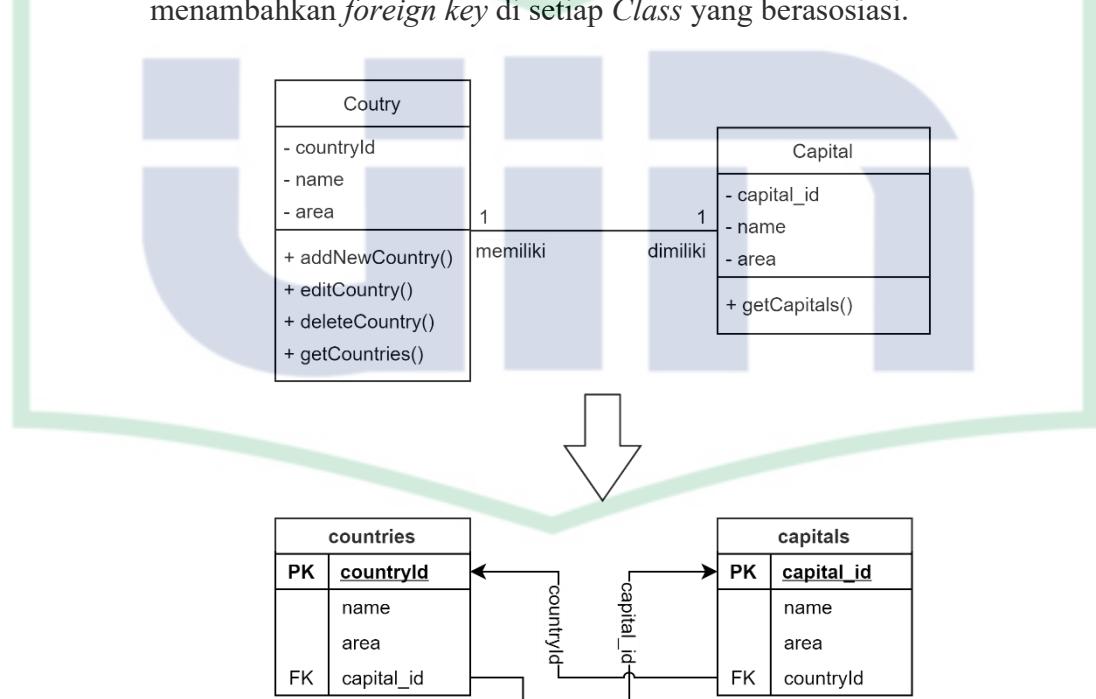


Gambar 2.33 *Collection* pada *Class* dibuatkan tabel terpisah dan diberikan id sendiri

- Asosiasi *one-to-many* (satu ke banyak) bisa menggunakan cara yang sama seperti *collection*.
- Asosiasi *one-to-many* (banyak ke banyak) perlu dipisahkan menjadi tabel tersendiri menjadi dua tabel. Kemudian di antara kedua tabel itu, perlu membuat sebuah tabel *identifier* yang berisi dua kolom. Kolom pertama berisi id dari tabel pertama yang memiliki asosiasi. Kolom kedua berisi id dari tabel kedua asosiasi yang dituju. Hal ini seperti memiliki 2 *collection*. Perhatikan gambar berikut.



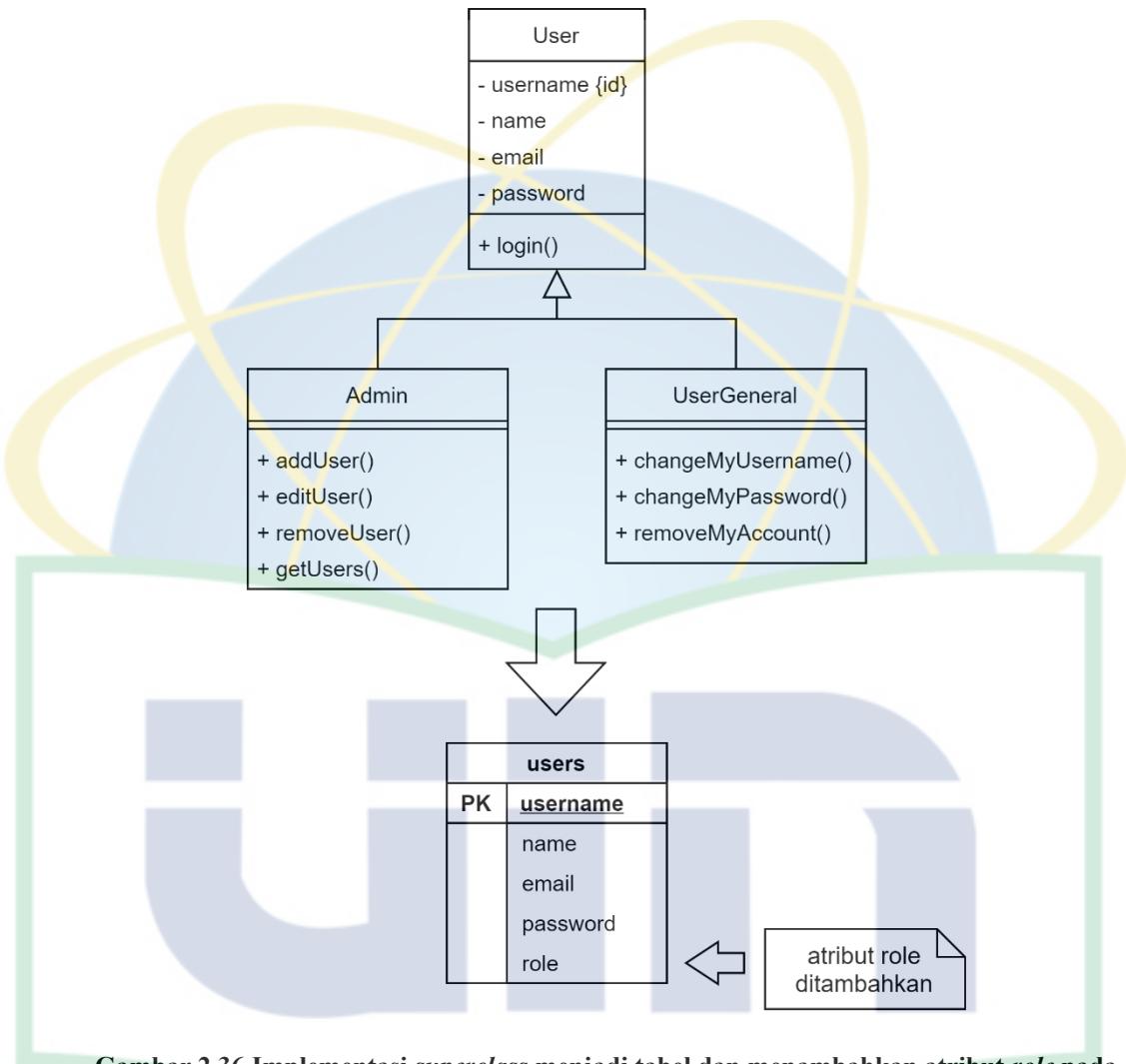
- Asosiasi *one-to-one* (satu ke satu) bisa diimplementasikan dengan menambahkan *foreign key* di setiap *Class* yang berasosiasi.



**Gambar 2.35 Mengubah relasi one-to-one menjadi tabel relasional**

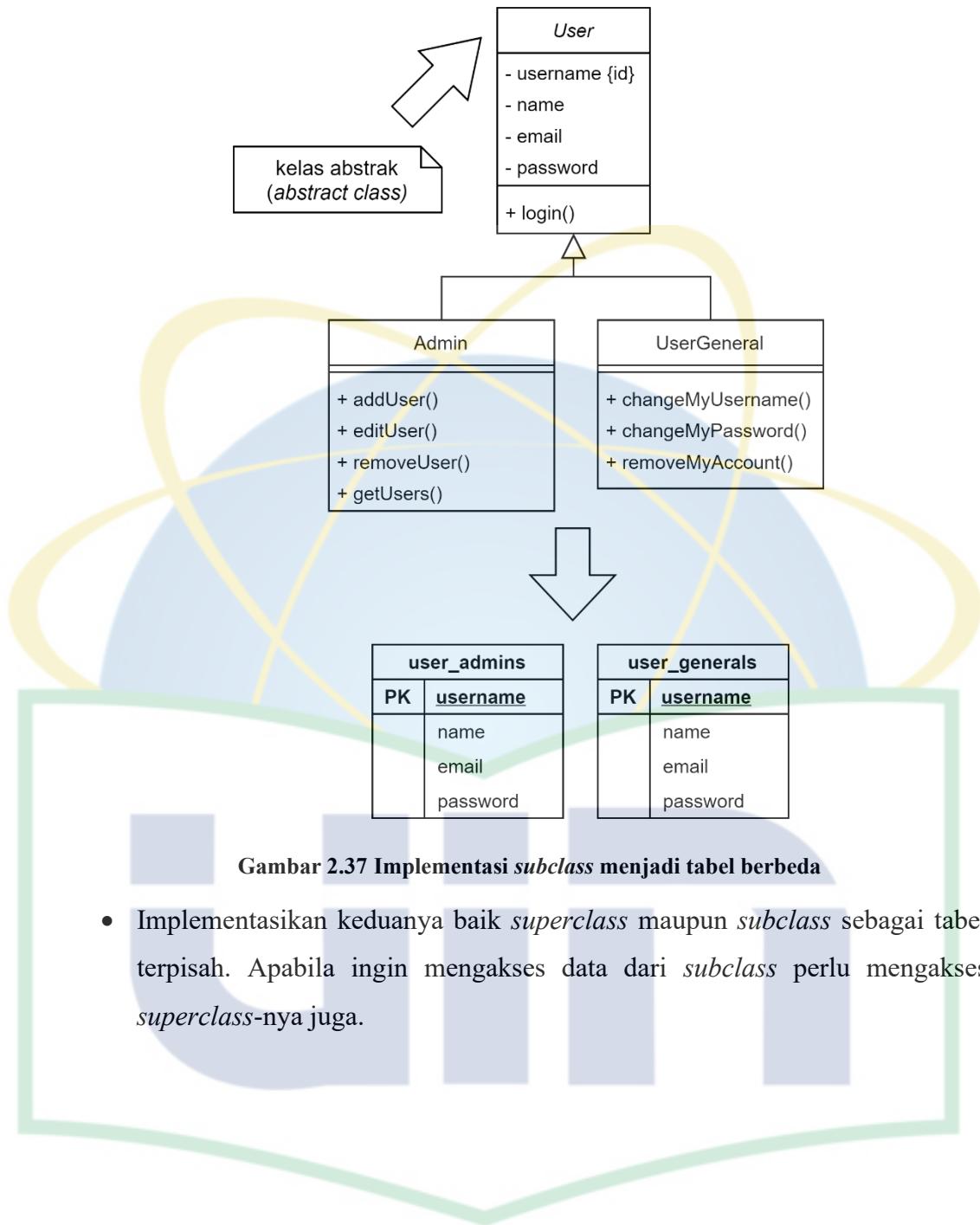
Kemudian untuk mengubah pewarisan (*inheritance*) *Class* menjadi tabel pada basis data, ada tiga cara yang dapat dilakukan:

- Hanya mengimplementasikan *superclass* sebagai tabel. Pendekatan ini cocok pada *subclass* yang hanya memiliki *behaviour* yang berbeda dari induknya.



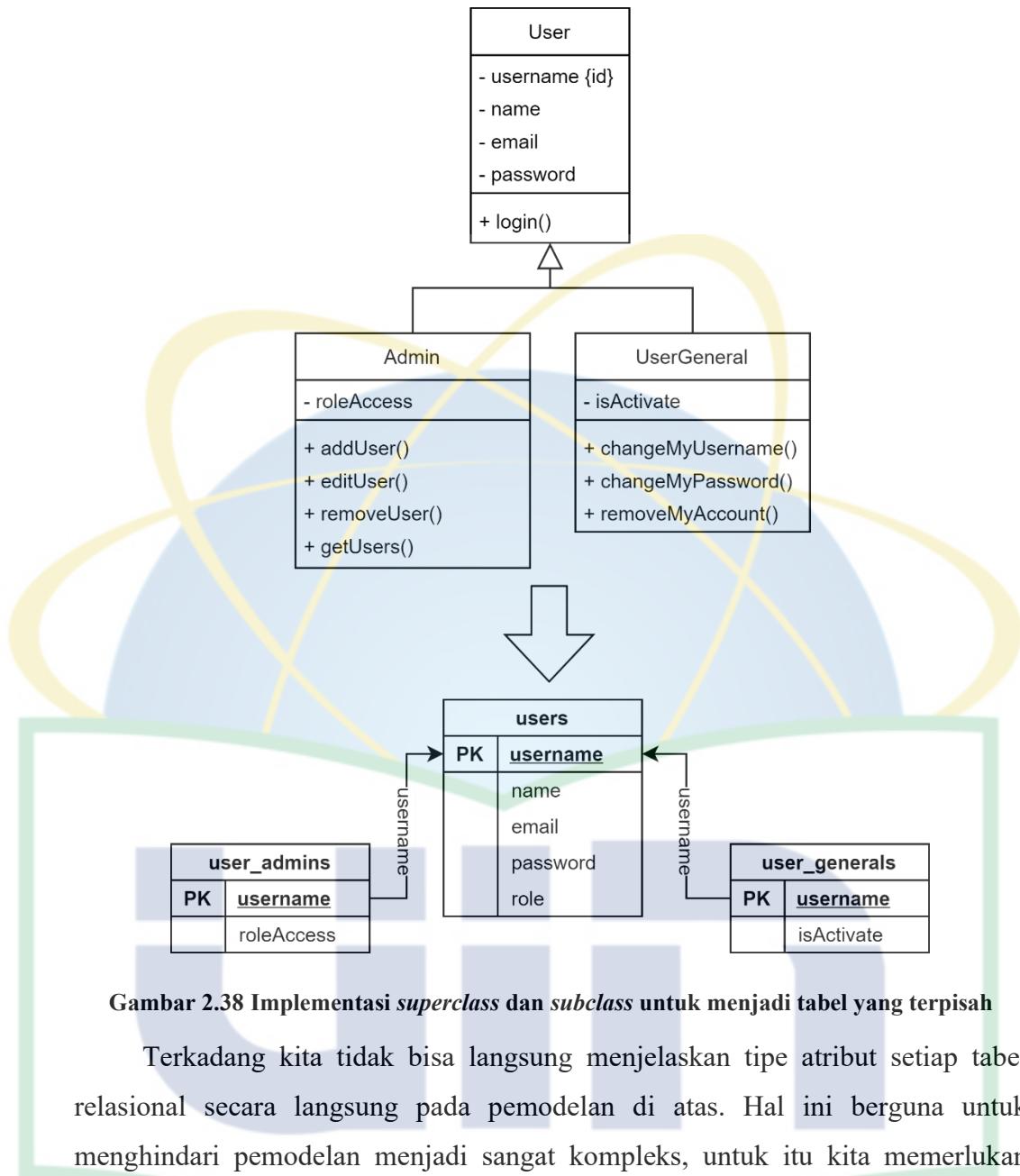
**Gambar 2.36 Implementasi *superclass* menjadi tabel dan menambahkan atribut *role* pada tabel relasional untuk membedakan kedua jenis *user***

- Hanya mengimplementasikan *subclass* sebagai tabel dengan memasukkan atribut *superclass* ke tabel *subclass*. Pendekatan ini cocok pada *superclass* yang berbentuk abstrak (*abstract class*).



Gambar 2.37 Implementasi *subclass* menjadi tabel berbeda

- Implementasikan keduanya baik *superclass* maupun *subclass* sebagai tabel terpisah. Apabila ingin mengakses data dari *subclass* perlu mengakses *superclass*-nya juga.



Gambar 2.38 Implementasi *superclass* dan *subclass* untuk menjadi tabel yang terpisah

Terkadang kita tidak bisa langsung menjelaskan tipe atribut setiap tabel relasional secara langsung pada pemodelan di atas. Hal ini berguna untuk menghindari pemodelan menjadi sangat kompleks, untuk itu kita memerlukan Tabel Spesifikasi Data. Tabel spesifikasi basis data memenuhi hal ini dengan menjelaskan atribut setiap tabel lengkap dengan tipe, ukuran, dan keterangan. Atribut pada tabel relasional disebut dengan *field*. Tabel berikut memberikan contoh bagaimana membuat spesifikasi basis data. (Fikry, 2019)

- User

Nama tabel: users

*Primary Key:* username

*Foreign Key:* roleId

Jenis Tabel: Master

**Tabel 2.8 Contoh Spesifikasi Basis Data**

No.	Field	Tipe	Keterangan
1.	username	VARCHAR(16)	Id pengguna
2.	name	VARCHAR(64)	Nama pengguna
3.	email	VARCHAR(128)	Alamat email pengguna
4.	password	TEXT()	Kata sandi pengguna
5.	roleId	BOOLEAN	Hak akses pengguna 0 = Admin 1 = Pengguna biasa

## 2.10 Git dan Github

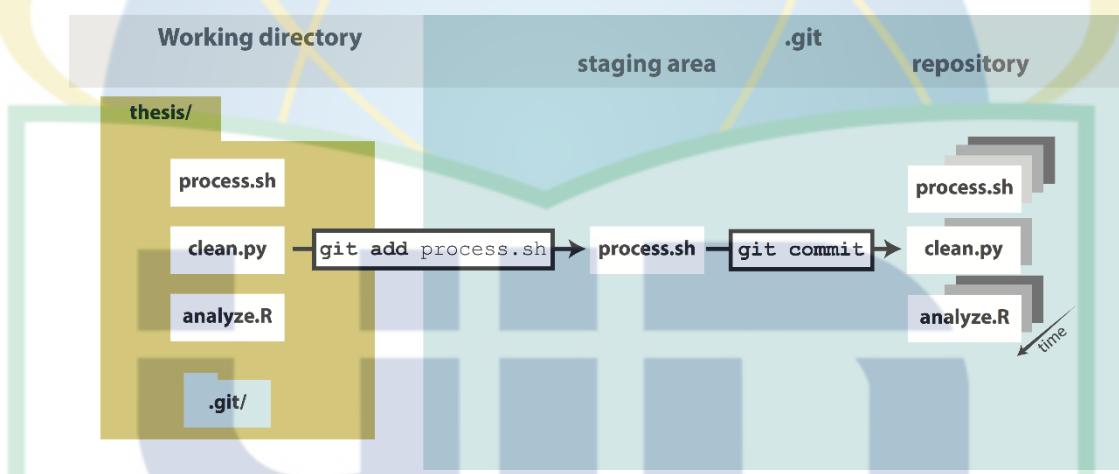
Github pada penelitian ini digunakan sebagai pembuatan lingkungan pengembangan *software*. Sebelum membahas lebih lanjut terkait Github, kita perlu mengenal sistem yang membentuk Github sebagai *social coding site* yaitu Git.

### 2.10.1 Pengenalan Git

Git merupakan sebuah sistem pengendali versi (*Version Control System* /VCS) yang digunakan untuk merekam perubahan yang dibuat pada sebuah berkas atau satu set berkas pada periode waktu tertentu. Perubahan ini disimpan dalam sebuah *tag* (penanda) nantinya di masa depan kita dapat menarik kembali perubahan versi yang kita buat pada penanda waktu tersebut. Ada tiga jenis VCS yaitu *Local*, *Centralized*, dan *Distributed*. (Somasundaram, 2013)

Proses pengembangan *software* dengan Git adalah dengan menyimpan setiap perubahan dari waktu ke waktu. Terdapat tiga fase status berkas saat mengembangkan dengan Git, yaitu *working directory*, *staging area*, dan *repository*. *Working directory* adalah folder tempat kita melakukan perubahan terhadap berkas.

Git akan memberikan status *unstaged changes* apabila mendeteksi perubahan pada satu atau beberapa berkas. *Unstaged changes* ini kemudian bisa ditinjau, kemudian developer akan memilih berkas mana kira-kira yang akan dimasukkan ke *staging area*. Area ini akan menjadi daftar berkas yang siap untuk menjadi sebuah *commit*, yaitu item yang mencatat satu perubahan dalam satu waktu dan satu pesan. *Commit* bertindak layaknya daftar riwayat pada peramban web. Setiap satu *commit* berisi satu pesan detail perubahan dan satu atau beberapa berkas yang berubah. *Commit* yang dibuat menandakan bahwa kita berhasil menyimpan perubahan pada berkas yang kita buat. Setelah itu developer bisa menyimpan *commit* ini pada repositori di layanan *social coding* seperti Github, Gitlab, Bitbucket, atau Source Forge. Perhatikan gambar berikut untuk memahami proses pengembangan *software* dengan Git lebih lanjut. (Blischak, Davenport, & Wilson, 2016; Somasundaram, 2013)



Gambar 2.39 Alur Perubahan Berkas pada Git

Gambar di atas menjelaskan alur perubahan berkas *process.sh* dari *working directory* sampai menuju *repository*. *Working directory* menunjukkan folder yang menjadi tempat kerja developer. Di sana terdapat daftar berkas kode program yang diperlukan untuk menyelesaikan sebuah tugas skripsi. Satu folder yang menjadi fokusnya yaitu folder *.git/*, di dalam sana terdapat berkas-berkas yang diperlukan oleh Git untuk melakukan tugasnya mencatat perubahan. Berkas *process.sh* ditambahkan ke *staging area* untuk menandakan bahwa berkas ini akan dijadikan *commit*. Akhirnya *commit* ini dapat kita simpan pada repositori di layanan *social coding*.

Gambar tersebut juga menjelaskan bagaimana cara kita menyimpan perubahan dari *working directory* ke *staging area*, lalu dari *staging area* ke *repository* dengan perintah *bash git add process.sh* dan *git commit*. Namun penulis tidak membahas penggunaan perintah ini di sini. Penulis menyarankan untuk melihat kegunaan fungsi perintah ini langsung pada laman web berikut <https://training.github.com/downloads/id/github-git-cheat-sheet/>. Penulis juga menyarankan untuk membaca langsung dokumentasi Git di laman web berikut <https://git-scm.com/book/en/v2>. Git saat ini bisa dipelajari juga dengan aplikasi web interaktif. Proyek aplikasi web di laman berikut <https://github.com/pcottle/learnGitBranching> memberikan kemudahan untuk mempelajari dasar-dasar pengembangan dengan Git. Bahasa yang digunakan pada aplikasi web itu adalah bahasa Inggris.

### 2.10.2 Kenapa Harus Menggunakan Git?

Git seperti yang sudah dijelaskan pada subbab sebelumnya berfungsi untuk menyimpan perubahan berkas dari waktu ke waktu. Sebenarnya hal ini bisa dilakukan dengan mengubah nama berkas sesuai tanggal perubahan berkas dan memberikan nama berkasnya seperti ini 2021-08-01\_skripsi.docx. Namun cara seperti itu sangat tidak disarankan, karena bisa rawan sekali terjadi kesalahan. Kita mudah sekali lupa di mana kita menyimpan berkas yang asli dan secara tidak sengaja menulis berkas yang salah atau menggandakan berkas yang tidak diinginkan. Hal ini yang membuat Git memudahkan kita untuk melacak perubahan *software*. Developer dapat mengembalikan ke perubahan sebelumnya dengan mudah apabila terjadi masalah saat pengembangan dilakukan. (Chacon & Straub, 2014; Perez-Riverol dkk., 2016)

Git tidak hanya dapat digunakan oleh developer untuk mengembangkan *software*, namun bisa digunakan oleh desainer web atau grafik. Mereka dapat menyimpan setiap perubahan berkas desain yang mereka buat tanpa harus menggandakannya di setiap perubahan. Mereka dapat mengembalikan perubahan berkas desain apabila terjadi masalah. Mereka juga dapat melihat siapa yang melakukan perubahan tersebut. (Chacon & Straub, 2014)

Pengaplikasian Git juga dapat digunakan pada penelitian ilmiah untuk menyimpan berbagai hasil penelitian. Contohnya seperti *datasets*, kode statistik,

gambar penelitian, catatan lab, dan naskah penelitian. Git memberikan cara yang ampuh untuk melacak dan membandingkan versi hasil penelitian tersebut, menelusuri kembali eror, mengeksplorasi pendekatan baru dengan cara yang terstruktur sambil mempertahankan jejak audit penelitian. Manfaat didapatkan oleh setiap individu yang melakukan penelitian. Hal ini membuat Git dapat meningkatkan untuk meningkatkan *reproducibility* dan meningkatkan transparansi dalam penelitian ilmiah. Peningkatan ini didapatkan dari fungsi Git yang mencatat setiap perubahan hasil penelitian.

Paragraf-paragraf sebelumnya memberikan wawasan kepada penulis kenapa penggunaan Git diperlukan untuk pengembangan *software*. Penggunaan Git tidak hanya dapat digunakan untuk pengembangan *software*, namun bisa juga digunakan untuk pembuatan laporan penelitian yang penulis lakukan di penelitian ini. Seharusnya penggunaan Git sudah diajarkan sejak bangku kuliah. Sayangnya penulis tidak mendapatkan pengalaman tersebut sejak menduduki bangku kuliah semester satu hingga saat ini. Penulis mempelajari Git secara otodidak melalui berbagai sumber di internet seperti yang dijelaskan pada subbab pengenalan Git. Cara ini berhasil untuk penulis bisa meyakinkan diri untuk menggunakan Git, namun tidak untuk semua orang. (Ram, 2013)

Individu perlu mananamkan literasi pengembangan *software* agar menyadari pentingnya penggunaan Git dalam aktivitas yang mereka lakukan. Tentunya hal ini tidak hanya bisa dilakukan pada individu yang memiliki latar belakang kuliah di jurusan yang berhubungan dengan ilmu komputer. Git bahkan dapat digunakan untuk pembuatan kursus akademik. Hasilnya cara ini memudahkan untuk mengatur manajemen kursus sambil melakukan pengajaran Git kepada mahasiswa. (Lawrance, Jung, & Wiseman, 2013; Ram, 2013)

Penelitian ini menggunakan Git yang berguna untuk mencatat segala perubahan yang terjadi pada kode program *software*. Ini akan menjadi salah satu sumber referensi untuk developer baru yang ingin mengembangkan aplikasi HC Portal. Hal ini sebagai harapan penulis untuk mendukung pengembangan *software* yang berkelanjutan.

### 2.10.3 Pengenalan Github

Github merupakan web sosial pemrograman (*social coding site*). Ia memberikan kesempatan kepada para developer untuk menampilkan hasil karya dan aktivitas mereka dalam satu platform. Tentunya hasil karya mereka akan tampil pada pengguna yang tertarik dengan apa yang developer *software* lakukan. Github menggunakan Git untuk mencatat aktivitas pengguna tentunya menggunakan *Distributed VCS*. Berkas-berkas dan proyek pengembangan *software* yang disimpan pada Github dibungkus dalam satu repositori. Pengguna dapat berkontribusi pada proyek orang lain ataupun membuat proyek mereka sendiri secara bebas dan gratis. Github saat ini menjadi tempat untuk 28 juta proyek yang dikelola oleh 11 juta pengguna. Github memberikan fungsionalitas lain untuk mendukung pengembangan *software*, seperti Wiki (halaman web untuk menjelaskan detail proyek *software*), pelacak isu (*issue tracking*), dan ulasan kode (*code review*). (Thung, Bissyandé, Lo, & Jiang, 2013; Wu, Kropczynski, Shih, & Carroll, 2014)

### 2.10.4 Fitur-fitur Github

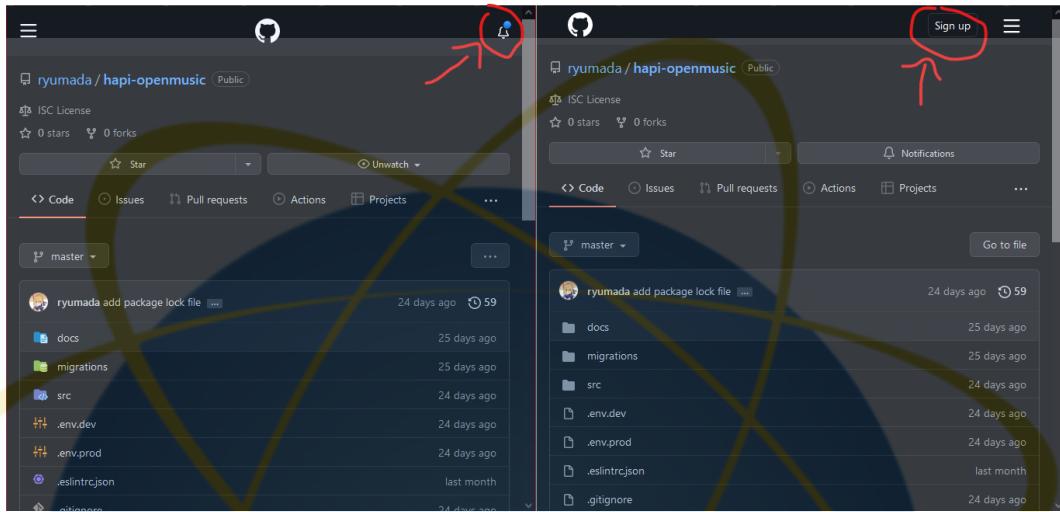
Github sebagai salah satu *social coding site* memiliki berbagai fitur untuk mendukung aktivitas developer untuk berkolaborasi dalam melaksanakan berbagai proyek *software*. Github tidak hanya memiliki fitur tersebut, namun terdapat fitur-fitur lain juga yang dapat meningkatkan produktivitas kerja developer serta mendukung komunitas proyek *software open-source*. Semua fitur ini mereka kelompokkan menjadi tujuh kelompok fitur. Berikut penjelasan kelompok fitur tersebut dan apa saja fitur yang ada di dalam kelompoknya: (Github, 2020)

- *Collaborative Coding*

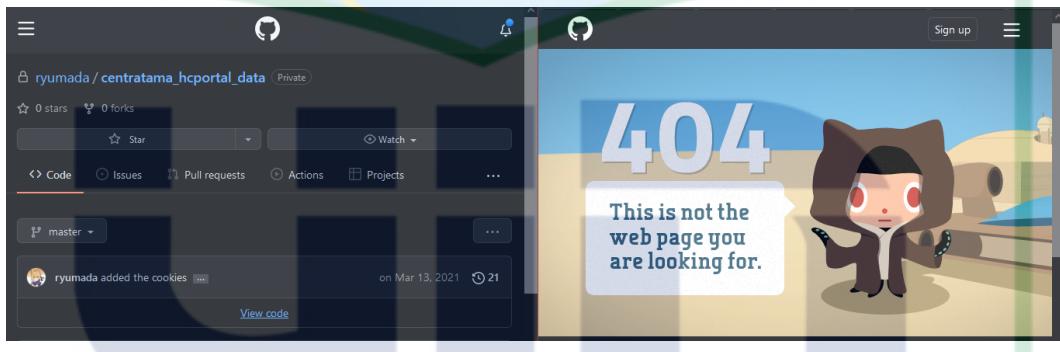
Github tidak hanya memiliki fitur-fitur dasar sebagaimana sebuah VCS. Ia tentunya memiliki fitur-fitur tambahan yang dapat digunakan untuk membuat sebuah platform *social coding*. Berikut fitur-fitur yang disediakan dalam kelompok fitur ini:

- *Public & Private Repositories*: merupakan sebuah tempat yang akan menyimpan semua berkas kode pekerjaan proyek *software*. Di sinilah lokasi utama developer untuk memulai mengembangkan *software*. Repositori Github terdiri dari dua jenis, yaitu *public* dan *private*. Pembuatan repositori

dengan atribut *public* memungkinkan semua orang dapat melihat semua sumber kode dan apapun yang ada di dalamnya secara terbuka. Sedangkan pembuatan repositori dengan atribut *private* memungkinkan hanya pemilik repositori dan kolaboratornya yang dapat melihatnya.



Gambar 2.40 Sebuah repositori publik yang dapat diakses dengan masuk ke akun (kiri) dan secara bebas tanpa masuk ke akun (kanan)

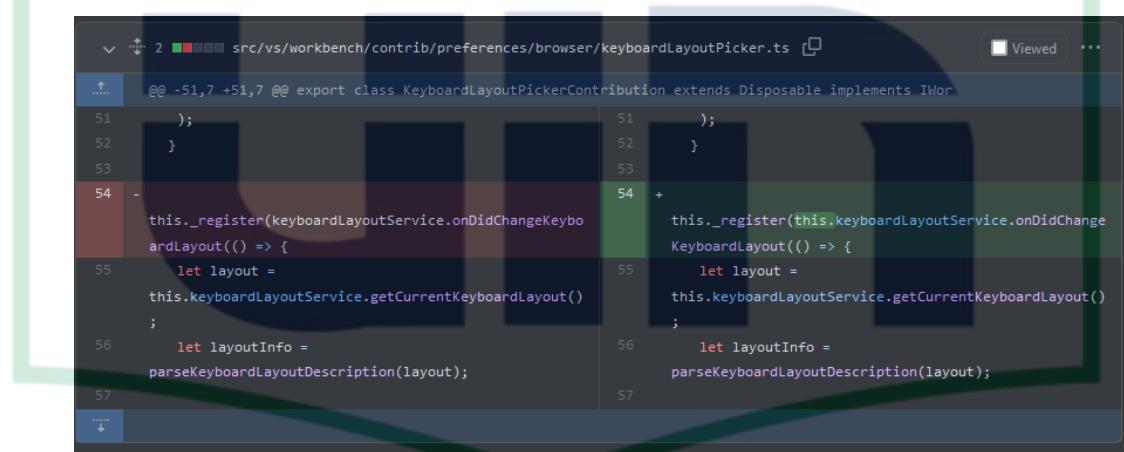


Gambar 2.41 Sebuah repositori *private* yang dapat diakses oleh pemilik repositori ataupun kontributor (kiri)

- *Pull Requests (PR)*: PR merupakan sebuah proses integrasi kode dari *branch* pengembangan fitur ke *branch* utama (*main*). Standar pengembangan dengan Github adalah melakukan PR saat ingin melakukan integrasi kode ke *branch main*. Tim pengembang menceritakan apa yang telah mereka kembangkan dalam *branch* yang berisi perubahannya. Si pemilik perubahan dapat menghubungkan tugas mereka yang tercatat dalam *issue*. Ini memudahkan klien untuk melihat bahwa *issue* yang mereka buat telah ditangani dengan adanya PR. Salah seorang atau beberapa anggota tim dapat melakukan *Code Review* saat PR berlangsung. Biasanya penerimaan PR

akan terjadi bila *Code Review* telah diterima. Jika belum *Code Review*, pengembang fitur di *branch* tersebut melakukan proses *coding* dan melakukan *commit* perubahan kembali.

- *Issues*: merupakan tempat untuk melacak ide, umpan balik, tugas-tugas, atau *bugs* di Github. *Issue* dapat dihubungkan dengan *pull request*. Developer juga dapat memanfaatkan *Issue* untuk membuat daftar tugas. *Issue* juga dapat diberikan label agar bisa dikelola dengan baik. Selain itu, terdapat fitur-fitur lainnya yang mendukung komunikasi, notifikasi, dan pembuatan templat *issue*.
- *Notifications*: fitur ini digunakan untuk mendapatkan informasi aktivitas terbaru dari proyek *software* yang developer ikuti. Fitur ini juga dapat digunakan untuk mengatur pemberitahuan yang ingin developer dapatkan, seperti menghapus notifikasi, berhenti langganan pemberitahuan terbaru sebuah proyek *software*, dll.
- *Code Review*: fitur ini digunakan untuk mengulas kode baru yang ditambahkan ke repositori, melihat perubahan visualnya, dan menggabungkannya tanpa ragu dengan status cek otomatis.



The screenshot shows a code editor displaying a diff (difference) between two versions of a file named `src/vs/workbench/contrib/preferences/browser/keyboardLayoutPicker.ts`. The interface highlights changes in red for deleted code and green for added code. The code itself is written in TypeScript, defining a class `KeyboardLayoutPickerContribution` that extends `Disposable` and implements `IWork`. The changes focus on registering a service observer for keyboard layout changes.

```
@@ -51,7 +51,7 @@ export class KeyboardLayoutPickerContribution extends Disposable implements IWork
 51   );
 52 }
 53
 54 - this._register(keyboardLayoutService.onDidChangeKeyboardLayout(() => {
 55   let layout =
 56     this.keyboardLayoutService.getCurrentKeyboardLayout();
 57
 58   let layoutInfo =
 59     parseKeyboardLayoutDescription(layout);
 60
 61   ...
 62 })
 63
 64 + this._register(this.keyboardLayoutService.onDidChangeKeyboardLayout(() => {
 65   let layout =
 66     this.keyboardLayoutService.getCurrentKeyboardLayout();
 67
 68   let layoutInfo =
 69     parseKeyboardLayoutDescription(layout);
 70
 71   ...
 72 })
```

Gambar 2.42 Contoh perubahan visual pada baris kode yang dihapus ditandai dengan warna merah dan ditambahkan ditandai dengan warna hijau; disebut tampilan *diff* yang berarti perbedaan

- *Code Review Assignment*: fitur ini berguna untuk memberi tugas kepada anggota tim yang harus mengulas sebuah *pull request*.
- *Draft Pull Request*: fitur yang berguna untuk berdiskusi dan berkolaborasi di sebuah *pull request* tanpa mengirimnya secara formal untuk menghindari penggabungan perubahan yang tidak diinginkan.

- *Protected Branches*: fitur ini berguna untuk menerapkan larangan pada bagaimana *branch* digabungkan, termasuk memerlukan ulasan atau membolehkan hanya kontributor tertentu yang boleh bekerja dengan sebuah *branch* tertentu.
- *Team Discussions*: fitur ini berjalan layaknya halaman web forum yang memungkinkan pemilik repositori dan kontributornya berdiskusi di dalam repositori secara langsung. Partisipan akan mendapatkan pembaruan terbaru dari forum melalui notifikasi yang dikirimkan ke akun mereka.
- *Team Reviewer*: fitur ini berguna untuk meminta tim dari sebuah organisasi untuk mengulas perubahan pada *pull request*.
- *Multiple Assignees*: fitur ini berfungsi untuk memberikan tugas sampai kepada 10 orang untuk bekerja pada *issue* atau *pull request* yang diberikan. Fitur ini memudahkan developer untuk melacak siapa yang bekerja dengan apa.
- *Multiple Reviewers*: fitur ini berfungsi untuk meminta ulasan dari beberapa kontributor. Pengulas yang diminta akan menerima notifikasi kalau developer memerlukan ulasan mereka.
- *Multi-line Comments*: fitur ini berfungsi untuk menjelaskan ulasan kode dengan mereferensikan beberapa baris dalam satu komentar di sebuah tampilan perbedaan (*diff view*) *pull request*.
- *Codespaces*: merupakan layanan yang memungkinkan para developer untuk melakukan pengodean (*coding*) *software* di aplikasi teks editor *visual studio code* yang berjalan pada sebuah mesin virtual secara *cloud*. Layanan ini diklaim oleh Github sendiri memiliki performa tinggi yang dapat dimulai hanya dalam beberapa detik.

- ***Automation & CI/CD***

Github memiliki fitur untuk mengotomatiskan proses dari fitur-fitur kolaboratif di atas. Developer bisa mengotomatiskan CI/CD, testing, perencanaan, manajemen proyek, dan masih banyak lagi. Automasi dengan Github tidak hanya untuk keperluan proses pengembangan, tapi bisa juga untuk melakukan standarisasi kode dengan mengikuti praktik terbaik (*best practice*), keamanan, dan kepatuhan (*compliance*) pada suatu organisasi. Semua ini bisa dilakukan dengan **Github**

*Actions* yang merupakan sebuah mesin komputer virtual (VM). Github *Actions* bisa berjalan dengan daftar perintah yang didefinisikan dalam sebuah *jobs*. Di dalam *jobs* tugas-tugas didefinisikan dengan format penulisan *.yaml* secara berurut dari baris ke baris.

Perintah-perintah yang dapat dijadikan *jobs* pada Github *Actions* bisa berupa perintah pada sistem Linux, Windows, atau MacOs sesuai dengan VM yang dipilih untuk menjalankannya. Github *Actions* juga bisa dijalankan di mesin sendiri yang disebut dengan *self-hosted runners*, dengan begitu developer memiliki kendali lebih dengan mengatur label mesin, melakukan pengelompokan, dan membuat kebijakan sendiri. Aplikasi untuk menjalankannya pun *open-source*.

Github juga menyediakan fitur-fitur lainnya yang dapat menunjang Github *Actions*. Github *Marketplace* menyediakan ribuan templat aksi dan aplikasi dari komunitas untuk mempercepat pembuatan alur kerja *Actions* otomatis. *Secrets Management* yang digunakan untuk menyimpan variabel dengan informasi sensitif seperti kata sandi atau token untuk mengakses aplikasi yang didapat dari Github *Marketplace*. Ada banyak lagi fitur-fitur lain yang bisa dilakukan dengan Github *Actions* ini, penjelasan lebih lanjut tersedia pada halaman dokumentasinya di tautan berikut: <https://docs.github.com/en/actions>.

- ***Security***

Github menyediakan berbagai fitur yang dapat menunjang pengelolaan keamanan sebagaimana developer menulis kode dan menyimpannya di platformnya. Github dapat secara otomatis mengulas setiap perubahan yang tersimpan dan mencari tahu kerentanan apabila menerapkan *open source dependency* aplikasi pihak ketiga pada *software* yang sedang dikerjakan. Semua ini ditunjang dengan berbagai fitur keamanan untuk menjaga akun dan repositori developer, seperti 2FA, *Dependabot Alert*, *GPG Commit Signing Verification*, *Security Audit Log*, *Dependency Alert*, *Code Scanning*, dan masih banyak lagi. Github juga mendukung keamanan tambahan untuk perusahaan yang menggunakan layanan mereka seperti Github *Connect*, *Audit Log API*, dll.

- ***Client Apps***

Cara utama untuk mengakses Github adalah melalui halaman webnya. Namun, developer juga dapat mengakses layanan Github melalui aplikasi *client* yang bisa dipasang di perangkat desktop, *command line interface* (CLI), maupun perangkat seluler. Perangkat Android mendapatkan rilis aplikasi *client* Github pada 15 Februari 2020. Aplikasi *client* tersebut memberikan fungsi dasar layanan github, seperti notifikasi; membuat dan mengulas *pull-request*, *issues*, dan forum diskusi; sampai melihat kode yang ada di dalam repositori.

- ***Project Management***

Github juga memberikan kemudahan untuk mengatur proyek yang sedang dikerjakan dengan fitur *Kanban Board*. Fitur ini memberikan kemudahan untuk mengatur tugas yang perlu anggota tim atau developer kerjakan. Waktu pengerjaan (*timeline*) juga bisa ditambahkan ke setiap proyek yang dibuat, dengan begitu semua anggota tim paham kapan proyek di mulai dan berakhir. Github juga memberikan fitur lainnya untuk membantu pengelolaan proyek *software* dengan fitur *labelling* untuk menandai *pull-request* ataupun *issue*, *milestones*, berbagai grafik dan informasi lengkap terkait repositori dengan fitur *insight*, *wiki* untuk membuat dokumentasi, dan masih banyak lagi

- ***Team Administration***

Github tidak hanya memberikan kemudahan dalam pengelolaan repositori dan proyek saja. Github juga memberikan kemudahan pengelolaan administrasi dalam tim dengan menyederhanakan manajemen akses dan izin di seluruh proyek dan tim. Pemilik proyek dapat memperbarui izin, menambahkan pengguna baru, dan memberikan semua orang akses spesifik yang mereka perlukan untuk mengakses proyeknya. Tidak hanya itu, Github juga dapat menyinkronisasi pengaturan administrasi ini dengan Okta dan *Azure Active Directory*. Github mengakomodasi ini semua dengan fitur-fitur seperti *Organizations*, *Custom Roles*, *Domain Verifications*, *Team Sync*, *Notification Restrictions*, dll.

- **Community**

Github mendukung komunitas developer *software* dengan menyediakan layanan tambahan seperti Github *Marketplace*, Github *Sponsors*, dan Github *Learning Lab*. Github *Marketplace* tidak hanya digunakan untuk mendapatkan templat dan aplikasi Github *Actions*, melainkan developer juga bisa membagikan templat milik mereka sendiri di platform ini. Github *Sponsors* digunakan untuk mendapatkan dukungan keuangan pada proyek *open-source* pihak ketiga yang developer gunakan. Developer yang menggunakan proyek tersebut dapat memberikan dukungan dengan pembayaran satu kali ataupun berulang di rentang waktu tertentu. Github *Learning Lab* dapat digunakan sebagai sarana pembelajaran yang dapat diakses langsung di dalam Github dengan dipandu oleh sebuah *Learning Lab Robot*. Github juga mendukung komunitas developer dengan membuat teks editor milik mereka sendiri yaitu bernama Atom. Github juga mendukung pembuatan aplikasi desktop di berbagai platform. Hal ini Github lakukan dengan membuat Electron.js, sebuah *framework* yang menyediakan *runtime* untuk menjalankan aplikasi web tanpa menggunakan peramban web.

#### 2.10.5 Kenapa Harus Menggunakan Github?

Github memberikan kemudahan untuk berkolaborasi dalam membuat proyek *software*. Berkolaborasi adalah kemampuan seseorang untuk dapat bekerja dalam sebuah tim. Ini merupakan salah satu kemampuan kerja yang paling dicari oleh pemberi kerja/perusahaan dalam dunia *software engineering*. Kemampuan tersebut yaitu kreativitas, kemampuan bekerja dalam tim, dan berpikir kritis. Pembelajaran kolaboratif (*Collaborative Learning*) adalah alternatif terbaik untuk mengembangkan ketiga kemampuan tersebut.

Penggunaan Github dalam proses pembelajaran di institusi pendidikan dapat memudahkan mahasiswa untuk belajar tiga kemampuan yang paling dicari tersebut. Sebuah penelitian yang dilakukan di *Department of Electronics and Computers, Transilvania University* membuktikan hal ini dapat dilakukan. Mereka menggunakan Github untuk melaksanakan proses pembelajaran selama satu semester. Hasilnya mahasiswa di departemen tersebut memiliki peningkatan kemampuan dalam berkolaborasi, bahkan mahasiswa sampai berinteraksi dengan bahan ajar pula. Mereka menunjukkan bagian materi pembelajaran yang eror,

kemudian mereka menawarkan materi baru melalui fitur *pull-request*. (Kertész, 2015; Ram, 2013)

Pengembangan *software* yang dilakukan dengan *workflow* Github dapat memberikan beberapa manfaat, seperti dapat meningkatkan kolaborasi yang lebih terbuka (*Open Collaboration*), meningkatkan penggunaan ulang kode program (*increase code reuse*) yang membuat proyek pengembangan *software* menjadi lebih mudah, serta mendorong berbagi pengetahuan (*knowledge sharing*) melintasi tim mereka.

Pendekatan *Open Source-Style* (OSS) yang biasanya digunakan untuk proyek-proyek pengembangan *software open-source* (sumber terbuka), digunakan pula pada proyek pengembangan *software* di lembaga-lembaga komersial. Beberapa manfaat apabila proyek pengembangan *software* menggunakan pendekatan OSS, yaitu sebagai berikut: (Kalliamvakou dkk., 2015)

- **Pengurangan Komunikasi (*reduced communication*)**

Penggunaan Github dapat mengurangi keperluan untuk komunikasi. Hal ini bisa terjadi karena pembahasan terkait proyek pengembangan *software* di Github dapat dilihat oleh keseluruhan tim. Tim juga dapat melakukan pembahasan lebih lanjut terkait proyek pengembangan *software* pada aplikasi IM (*instant messaging*) yang dapat terintegrasi dengan Github.

- **Bekerja dengan lebih independen (*more independent work*)**

Github mendorong developer untuk melaksanakan pengembangan *software* secara lebih mandiri. Mereka dapat memilih *issue* yang telah dibuat pada fitur *Github Issue Tracker* untuk mereka kerjakan. Manajer proyek dapat memilih *issue* mana yang memiliki prioritas lebih untuk dikerjakan terlebih dahulu. Ini dilakukan agar developer dapat mengerjakan *issue* yang lebih penting untuk keberlangsungan *software* di tahap pengembangan.

- **Organisasi diri (*Self-Organization*)**

Github mendorong developer untuk bisa melakukan organisasi diri. Developer dapat memilih pekerjaan mana yang dapat mereka lakukan, tentunya melalui pertimbangan dari petunjuk label *issue* prioritas pengembangan. Manajer

proyek juga dapat menentukan *issue* mana yang harus dikerjakan oleh developer. Namun biasanya developer diberikan kebebasan untuk memilih sesuai dengan kemampuannya.

Manfaat-manfaat tersebut bisa kita dapatkan lebih banyak di Github ketimbang platform *social coding* lainnya seperti Gitlab, Bitbucket, dll. Hal ini karena saat ini para developer *software* memanfaatkan Github untuk membagikan portofolio miliknya berupa proyek kecil pribadi pengembangan *software* atau kontribusinya terhadap proyek-proyek *open-source*. Kontribusi dan cara kerja developer dapat ditayangkan secara jelas di Github. Hal ini membuat mereka dapat tampil menjadi seorang developer yang memiliki nilai lebih dibandingkan dengan yang lain. Ini menunjukkan pengalaman seorang developer *software*. Github juga menjadi sebuah platform yang paling banyak dipercaya oleh para pengembang *software open-source*. Hal ini membuat kita dapat lebih mudah mempelajari bagaimana konsep pengembangan *software* menggunakan platform *social-coding* dengan melihat bagaimana para pengembang *software open-source* ini melakukannya di Github. (Sharma, 2022)

#### 2.10.6 Berkas README.md untuk Repositori Github

**README.md** merupakan berkas yang akan menyediakan segala informasi terkait proyek dan bagaimana orang-orang dapat berpartisipasi dalam pengembangannya. Kontributor di sini bisa berupa developer lain ataupun rekan kerja yang akan ikut bersama mengembangkan proyek *software*. Pembuatan berkas README.md harus seinformatif mungkin agar dapat menarik kontributor developer lain untuk proyek sumber terbuka (*open-source*) atau membantu rekan kerja baru yang baru akan mengikuti pengembangan proyek.

Nyakundi seorang *technical writer* memberikan langkah-langkah dalam membuat berkas README.md agar dapat memenuhi tujuan tersebut. Ia mengatakan bahwa berkas README.md seharusnya dapat menjawab pertanyaan **apa, kenapa, dan bagaimana**. Berikut beberapa pertanyaan yang dapat kita tanyakan pada diri sendiri untuk membantu dalam menyusun berkas README.md: (Nyakundi, 2021)

- Apa motivasi kamu dalam membuat proyek ini?
- Apa yang kamu bangun di proyek ini?

- Apa masalah yang kamu selesaikan dengan proyek ini?
- Apa yang kamu pelajari?
- Apa yang membuat proyek ini lebih menonjol dari yang lain?

Dari beberapa pertanyaan tersebut kemudian bisa kita susun ke dalam beberapa bagian pembahasan, yaitu:

1. Judul proyek
2. Deskripsi proyek
3. Daftar isi
4. Bagaimana cara menyiapkan dan menjalankan proyek
5. Bagaimana cara menggunakan proyek
6. Sertakan kredit untuk para kontributor proyek
7. Tambahkan lisensi proyek
8. Tambahkan lencana status proyek agar lebih terlihat menarik
9. Bagaimana berkontribusi ke dalam proyek
10. Tambahkan contoh tes kode dan bagaimana cara menjalankannya

Penjelasan terkait bagian pembahasan tersebut dapat dilihat di sumber referensi aslinya yang penulis gunakan. (Nyakundi, 2021) Selain dari referensi tersebut, Github juga membuat dokumentasi dalam membuat berkas README.md juga di tautan berikut:

<https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>

Pembuatan bagian README pada poin 4 dan 5 akan lebih mudah jika kita sekaligus menjalankannya. Hal ini karena kita tidak perlu mengingat-ingat kembali langkah apa yang perlu kita lakukan untuk menyiapkan ataupun menjalankan proyek *software*. Langkah-langkah yang kita susun pun jadi tervalidasi dapat dilakukan oleh orang lain karena kita telah melakukan langkah-langkahnya secara langsung.

## 2.11 Repotori Digital

Repositori digital terdiri dari dua kata, yaitu repositori dan digital. Bagian ini akan membahas mengapa pada penelitian ini menggunakan kata Repotori Digital. Hal pertama yang perlu dibahas terkait penggunaan katanya, yakni Repotori dan

Digital. Pembahasan pada paragraf berikutnya akan membahas terkait kata digital. Kemudian, paragraf selanjutnya akan membahas terkait kata repositori. Terakhir, penulis memberikan kesimpulan dari penjelasan kedua paragraf sebelumnya.

Kata digital mencakup pada suatu hal yang dianggap berhubungan dengan teknologi komputer atau internet. Hal ini mencakup artefak yang dihasilkan dari teknologi yang disebut sebagai artefak digital. Artefak digital ini bisa berupa data angka atau teks yang menunjukkan informasi atau sesuatu yang dihasilkan oleh sistem perhitungan ataupun pengolah informasi. (KBBI, 2016a; Sauro & Zourou, 2019)

Menurut KBBI, repositori merupakan suatu tempat penyimpanan. Di Github, repositori menyimpan semua berkas kode proyek pengembangan *software* dan masing-masing riwayat perubahannya. Tidak hanya itu, satu repositori di Github juga mencakup beberapa fitur untuk membantu pengembangan *software*, seperti *Issues*, *Pull-request*, *Wiki*, *Discussions*, dan masih banyak lagi. Repositori di Github berguna untuk membantu dalam melakukan pengembangan *software* secara kolaboratif. Fitur Git yang menjadi sistem yang menjadi fondasi pengelolaan berkas di Github. Semua perubahan berkas yang disimpan dalam repositori Github dicatat dalam satuan *commit*. (Github, 2022; KBBI, 2016b) Kata repositori tidak hanya mencakup pada tempat penyimpanan artefak kode. Kata repositori akan bermakna luas jika kita hilangkan pembahasan Github. (Lutfiani, 2020)

Kata repositori tidak dapat berdiri sendiri pada suatu pembahasan topik penelitian karena akan bermakna sangat luas. Itulah mengapa penulis membuat kata repositori digital dengan platform Github. Repositori digital mencakup pada pembuatan repositori untuk menyimpan artefak digital, yakni artefak pengembangan *software*. Kemudian platform Github membahas bahwa repositori ini tidak hanya mencakup tempat penyimpanan artefak kode saja, namun juga mencakup fitur-fitur yang disediakan oleh Github untuk mendukung pengembangan *software*.

## 2.12 Pengujian Perangkat Lunak (*Software Testing*)

*Software testing* adalah sebuah proses untuk mengevaluasi fungsionalitas aplikasi dengan maksud untuk menemukan apakah *software* yang telah dikembangkan memenuhi kebutuhan klien atau tidak. *Software testing* juga

berfungsi untuk menemukan kesalahan/kecacatan yang dapat menurunkan kualitas *software*. Saat ini *software testing* tidak hanya dilakukan dengan cara manual, tetapi juga dilakukan dengan cara otomatis dengan satu atau beberapa alat *testing*. (Hamilton, 2021; Rajkumar, 2015a)

Kembali ke tujuan *Software testing* yaitu bertujuan untuk menemukan kesesuaian antara apa yang dibutuhkan oleh klien dan apa yang sudah dikembangkan, serta menemukan kesalahan/kecacatan pada *software*. Hal ini yang menyebabkan *software testing* penting untuk dilakukan. Pasalnya, apabila ada kesalahan atau kecacatan pada *software* yang tidak teridentifikasi, *software* yang sudah jalan di lingkungan pelanggan harus ditarik lagi dan kemudian perusahaan akan memulai ulang proses pengembangan *software*. Akibatnya perusahaan harus menerima kerugian yang sangat besar baik sumber daya maupun keuangan. (Oliver, 2020)

Terdapat **dua tipe testing** yang dapat dilakukan untuk pengujian *software*, yaitu **Manual Testing** dan **Automatic Testing**. *Manual testing* dilakukan dengan tangan manusia. *Manual Testing* bermodalkan dokumen *requirement* dan perspektif pengguna untuk menguji apakah aplikasi yang sudah dikembangkan sesuai dengannya dan mencari apakah ada yang tidak bekerja. Perencanaan *Manual Testing* bisa dilakukan dengan berbagai cara, seperti menggunakan **dokumen test cases**, memberikan tester langkah-langkah pengujian dan hasil yang diharapkan, serta melalui panduan tingkat tinggi (*High-Level Guide*) yang mengendalikan sesi testing. Sedangkan *Automatic Testing* dilakukan dengan menggunakan alat tester otomatis (*Testing Automation Tool*) untuk mencari kesalahan dan kecacatan pada produk. Proses testing ini menggunakan *testing script*. Tester mengeksekusi *script* tersebut dan menghasilkan *test result* secara otomatis menggunakan alat tester otomatis. Contoh alat tester otomatis yang terkenal yaitu Selenium dan Katalon Studio.

Terdapat **dua metode testing** yang dapat dilakukan untuk pelaksanaan testing, yaitu **Verification** dan **Validation**. *Verification* dalam *software testing* merupakan proses untuk memastikan apakah kita telah membangun *software* dengan benar. *Verification* dilakukan dengan cara mengecek dokumen dan berkas lainnya yang dibuat di tahapan awal pengembangan, kemudian memastikan apakah

kita mengembangkan *software* sesuai dengan dokumen tersebut. *Validation* dalam *software testing* merupakan proses untuk memastikan apakah kita telah membangun *software* yang tepat. *Validation* dilakukan dengan melakukan pengujian *software* dengan cara testing *White Box* maupun *Black Box*. (Rajkumar, 2015a)

### 2.12.1 *Black Box Testing* dan *White Box Testing*

Ada dua pendekatan yang biasa dilakukan dalam melakukan *testing* yaitu *black box testing* dan *white box testing*. ***Black Box Testing*** juga disebut sebagai *Behavioural*, *Specification-Based*, atau *Input-Output Testing*. Ia melakukan *testing* dengan melakukan uji fungsionalitas tanpa melihat struktur dalam kode yang dibuat. ***White Box Testing*** juga disebut sebagai *Glass Box*, *Clear Box*, atau *Structural Testing*. Ia melakukan *testing* dengan melihat perspektif internal sistem, serta melihat sebaik apa struktur kode yang dibuat. (Munawar, 2018; Rajkumar, 2015a)

Ada beberapa teknik untuk membuat desain *test case* dengan pendekatan *Black Box Testing*, di antaranya: *Equivalence partitioning*, *Boundary Value Analysis* (BVA), *Decision Table*, *State Transition*, *Functional testing*, *Non-Functional Testing* (Testing Performa Aplikasi). (Rajkumar, 2015a)

*White Box Testing* juga memiliki teknik untuk melakukan testing, tidak hanya *Black Box Testing*, yaitu *Statement Coverage*, *Decision Coverage*, dan *Path Coverage*. *Statement Coverage* melakukan testing dengan memastikan satu persatu *line-of-code* berhasil dieksekusi dan menghasilkan *output* yang diharapkan. *Decision Coverage* melakukan testing dengan memastikan kode yang menangani keputusan (*decision/if-else/true-false*) berhasil dieksekusi dan menghasilkan *output* yang diharapkan. *Path Coverage* melakukan testing dengan melihat rute aliran eksekusi kode, memastikan rutunya dieksekusi dengan tepat, dan menghasilkan *output* yang diharapkan. (Rajkumar, 2015a)

Teknik-teknik testing *black box* dan *white box* di atas digunakan untuk menyusun dokumen *test cases*. Nantinya dokumen tersebut menjadi pedoman untuk melakukan testing aplikasi.

## 2.12.2 Level Testing

Pengujian testing digunakan untuk memastikan dan meningkatkan kualitas produk *software* yang telah dikembangkan. Peningkatan produk *software* memerlukan proses testing. Pelaksanaan testing terdiri dari 4 level yang mana setiap level ini memiliki tujuannya masing-masing. Berikut penjelasan masing-masing level testing tersebut: (Rajkumar, 2015b)

- ***Unit Testing***

*Unit testing* dilakukan oleh developer *software*. Pengetesan ini dilakukan dalam lingkungan developer dengan menguji satu persatu modul *software* yang telah dibuat apakah modul tersebut bekerja dengan baik. *Unit testing* disebut juga sebagai *Module Testing* atau *Component Testing*.

- ***Integration Testing***

*Integration testing* dilakukan untuk mengecek keterhubungan dan transfer data antar modul aplikasi. Setiap modul *software* harus sudah dites pada proses *unit testing* agar *integration testing* dapat dijalankan dengan mulus. *Integration testing* dilakukan dengan menggunakan dokumen *test plan*. Dokumen ini nantinya memberikan gambaran secara jelas bagaimana alur prosesnya dan menghindari kekacauan saat *integration testing* dijalankan. Tentunya setiap proses testing memerlukan *test case*, begitu juga dengan *integration testing*.

Contoh pembuatan *test case*-nya misal ada tiga modul *software*, yaitu halaman *login*, kotak masuk (*inbox*), dan hapus *email*. Ketika menuliskan *test case* untuk *integration testing* kita perlu fokus pada komunikasi antar modul, bukan fokus pada fungsionalitas modul *software*. Seperti yang telah dijelaskan sebelumnya, testing fungsionalitas *software* dapat dilakukan pada *unit testing*. Fokus kita pada akhirnya adalah berfokus dengan “Bagaimana halaman *login* terhubung dengan halaman *inbox*” dan “Bagaimana halaman *inbox* terhubung dengan modul hapus *email*”. (Rajkumar, 2018)

Tabel 2.9 Contoh *Test Case* pada *Integration Testing*

ID	Objective	Description	Expected Result
----	-----------	-------------	-----------------

1	Verifikasi integrasi antara modul <i>login</i> dan <i>inbox</i>	Masukkan kredensial untuk <i>login</i>	Halaman <i>inbox</i> harus ditampilkan
2	Verifikasi integrasi antara modul <i>inbox</i> dan hapus <i>email</i>	Pilih satu atau lebih <i>email</i> , kemudian klik tombol hapus	<i>Email</i> yang dihapus menghilang dari <i>inbox</i> dan muncul di kotak sampah <i>email</i>

*Integration testing* dibagi menjadi beberapa pendekatan, yaitu *Big Bang Integration Testing*, *Top-Down Integration Testing*, *Bottom-Up Integration Testing*, (Rajkumar, 2018)

- ***System Testing/Functional Testing (End-to-End Testing)***

Testing ini merupakan *Black Box Testing* yang mana dilakukan dengan memanggil *test scenario*. Tujuannya adalah untuk memastikan *software* berjalan sebagaimana mestinya. Hal ini dilakukan dengan memverifikasi *output* yang diharapkan dari setiap input yang dimasukkan. Testing ini juga melibatkan tes *user experience software*.

- ***Acceptance Testing***

Testing ini dilakukan untuk mendapatkan verifikasi dari pengguna langsung dengan mendapatkan tanda tangan yang menyatakan bahwa *software* siap untuk didistribusi dan mendapatkan pembayarannya. Testing ini memiliki beberapa tipe sebagai berikut: (Rajkumar, 2015b)

- *Alpha Testing*, testing ini seringnya seperti *Usability Testing* yang melibatkan orang-orang yang ada di dalam tim pengembangan *software*. Terkadang testing ini dilakukan oleh klien atau orang luar dari developer atau tester.
- *Beta Testing*, testing ini dilakukan dengan beberapa klien terbatas sebelum pendistribusian *software*. Klien memberikan umpan balik (*feedback*) apabila ada masalah atau bila ingin mengubah sesuatu pada *software*.
- *Gamma Testing*, testing ini dilakukan ketika *software* siap dirilis dengan spesifikasi yang ditentukan. Testing ini dilakukan melewati testing oleh tim developer.

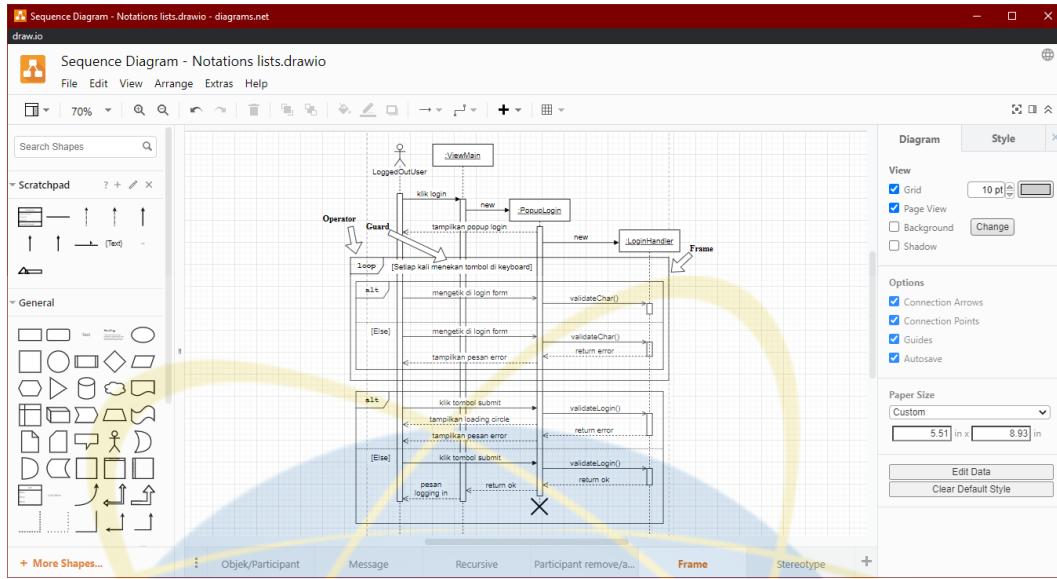
## 2.13 Tools Pengembangan Aplikasi

Penelitian ini menggunakan beberapa *tools* yang berupa *software* untuk mengembangkan objek penelitian. Subbab berikut menjelaskan beberapa *tools* yang digunakan pada penelitian ini.

### 2.13.1 Draw.io (diagrams.net)

Draw.io menyebut dirinya sebagai *open-source technology stack* (seperangkat teknologi sumber terbuka yang membeberkan kode sumber pembuatan aplikasinya secara bebas) untuk aplikasi membuat diagram. Aplikasinya banyak digunakan secara global. Hal ini karena aplikasi tersebut gratis untuk digunakan dan tersedia di berbagai *platform* mulai dari peramban web, perangkat desktop, integrasi ke beberapa aplikasi desktop dalam bentuk *add-in* (semacam aplikasi tambahan untuk satu aplikasi spesifik), hingga ke berbagai layanan SaaS (layanan *cloud* yang menyediakan perangkat lunak untuk digunakan layaknya *software* PC berbayar) seperti Microsoft Teams, Google Workspace, dan Jira Confluence.

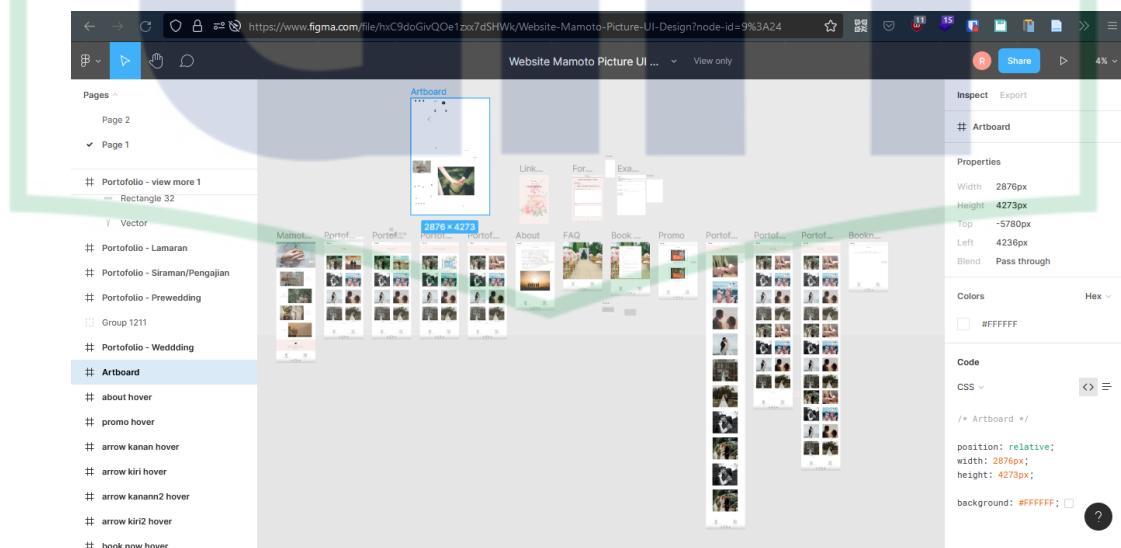
Draw.io dapat digunakan untuk membuat diagram manajemen bisnis dan proyek (seperti *Kanban boards*, *Project Planning Diagram*, *BPMN Diagram*, dll), diagram pengembangan *software* (seperti diagram UML, diagram *Gitflow*, diagram ERD dan basis data, dll), serta diagram infrastruktur IT (seperti *Rack Diagram*, diagram jaringan dan infrastruktur, hingga diagram infrastruktur AWS Cloud). Draw.io dapat digunakan untuk berbagai keperluan dalam pembuatan desain sistem. Penelitian ini menggunakan Draw.io untuk membuat diagram UML yang digunakan sebagai desain *software* yang dikembangkan. (diagrams.net, 2021)



**Gambar 2.43 Tampilan aplikasi draw.io berjalan pada platform desktop dengan sistem operasi Windows 10**

### 2.13.2 Figma

Figma merupakan aplikasi *open-source* pengolah grafis yang dikhususkan untuk membuat desain tampilan produk aplikasi berbasis web, *website*, maupun aplikasi *mobile*. Sama seperti Draw.io, Figma juga berjalan pada peramban web. Hal ini juga membuat aplikasi Draw.io dapat dijalankan di berbagai platform, hanya saja perangkat desktop yang didukung hanya Windows dan Mac. Walaupun begitu, perangkat Linux, ChromeOS, dan lainnya dapat menjalankan aplikasi ini melalui peramban web. (Figma, 2020)



**Gambar 2.44 Tampilan aplikasi Figma berjalan pada peramban web Firefox**

Figma dan Draw.io walaupun sama-sama merupakan aplikasi pengelola grafis, mereka memiliki fungsi khusus yang tidak dimiliki satu dengan yang lain. Hal ini yang membuat kedua aplikasi tersebut digunakan dalam hal yang berbeda.

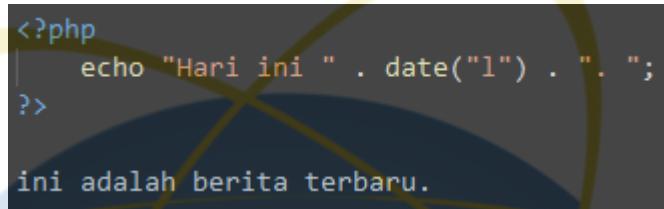
Figma memiliki *inspect sidebar* yang digunakan untuk melihat properti dari item grafis yang dibuat seperti ukuran panjang, lebar, dan properti lainnya. *Inspect sidebar* ini dapat dijadikan pedoman langsung oleh developer untuk membuat tampilan *software*. Figma juga memiliki fungsi untuk membuat prototipe *software*. Hal ini membuat Figma dapat digunakan untuk membuat simulasi bagaimana pengguna berinteraksi dengan *software*. Draw.io memiliki berbagai fungsi *connector* dan *automatic layout* yang digunakan untuk membantu membuat diagram. Semua hal terkait pembuatan tampilan aplikasi dan bagaimana pengguna berinteraksi dengan aplikasi dapat menggunakan Figma, sedangkan hal yang terkait dengan desain sistem dan pembuatan diagram dapat menggunakan Draw.io. (diagrams.net, 2021; Figma, 2020)

### 2.13.3 PHP

PHP (*Hypertext Pre-processor*) merupakan bahasa *scripting* yang digunakan untuk pengembangan web dinamis. Pengembangan awal dunia Web 1.0 yang tidak berlangsung lama kemudian muncul Web 1.1 dengan membawa beberapa peningkatan seperti Java, JavaScript, JScript, dan ActiveX. Sementara itu di sisi server juga membawa peningkatan dengan munculnya *Common Gateway Interface* (CGI) menggunakan bahasa *scripting* seperti Perl dan PHP. Program basis data MySQL digunakan untuk mengambil dan menyimpan data. PHP dapat dengan mudah terhubung dengan MySQL. Hal ini menjadikan halaman web menjadi lebih dinamis. Hal ini juga membuat PHP menjadi populer di kalangan developer web dan menjadi awal era Web 2.0. Era ini menjadi bagian penting yang dibutuhkan untuk membuat pengembangan *website* jejaring sosial (*Social Networking Sites*).

PHP tidak hanya dapat diintegrasikan dengan MySQL, tapi juga dengan JavaScript membentuk proses AJAX. Proses ini menangani pengolahan dan mengirim *request* ke server di latar belakang tanpa membuat pengguna menyadari bahwa proses ini terjadi. Proses AJAX menggunakan JavaScript untuk eksekusinya di peramban web pengguna. JavaScript juga digunakan untuk memanipulasi HTML dan CSS, sehingga membuat tampilan halaman web menjadi lebih interaktif.

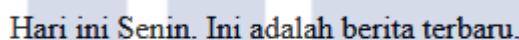
PHP menggunakan ekstensi berkas *.php* yang akan dikirimkan ke klien secara langsung sebagai HTML. Penggunaan bahasa *scripting* diawali dengan menggunakan *tag* pembuka `<?php` yang kemudian kode logika dapat dimasukkan setelahnya. Setelah kode logika selesai dibuat kemudian ditutup dengan *tag* `?>`. Gambar berikut menunjukkan contoh penggunaan bahasan *scripting* PHP. (Nixon, 2015)



```
<?php  
echo "Hari ini " . date("l") . ". ";  
?  
ini adalah berita terbaru.
```

Gambar 2.45 Contoh penggunaan bahasa *scripting* PHP

*Tag* pembuka `<?php` memberitahu server web untuk mengeksekusi kode program PHP sampai *tag* penutup `?>`. Di luar dari kedua *tag* tersebut semuanya akan dikirimkan langsung ke peramban web klien sebagai HTML. Jadi tulisan “*ini adalah berita terbaru*” akan menjadi keluaran ke peramban web. Sedangkan kode di dalam *tag* PHP akan menghasilkan tanggal pada sistem server web, sehingga kode di atas akan menghasilkan tampilan sebagai berikut.



Hari ini Senin. Ini adalah berita terbaru.

Gambar 2.46 Hasil dari eksekusi kode PHP sebelumnya

Pengembangan web dinamis dapat dilakukan dengan menggunakan keenam teknologi yaitu PHP, MySQL, JavaScript, CSS, dan HTML. PHP dapat dieksekusi oleh server web seperti Apache dan Nginx. Hasil eksekusinya dikirimkan ke peramban web klien menjadi HTML. PHP dapat berinteraksi dengan server basis data MySQL. Hal ini yang membuat awal mula web menjadi dinamis. Penggunaan JavaScript pada pengembangan web dinamis dapat menghasilkan aplikasi berbasis web. Hal ini karena JavaScript memiliki kemampuan untuk memanipulasi HTML dan CSS hingga mengeksekusi kode PHP secara langsung menggunakan AJAX pada peramban web klien sehingga membuat tampilan web menjadi lebih interaktif. (Nixon, 2015)

#### 2.13.4 PHP Framework

PHP *Framework* membantu developer untuk mengembangkan web dinamis dengan cepat dan mudah dengan menyediakan model dasarnya, lengkap dengan API (yang digunakan sebagai jembatan penghubung *framework* ke teknologi lain seperti ke server basis data), *libraries* (kumpulan kode sumber yang digunakan untuk menjalankan logika kompleks dengan sintaksis sederhana), dan *extensions* (sama halnya seperti *libraries* hanya saja kode telah di-*compile* menjadi program yang dapat dieksekusi dalam satu kesatuan). Fitur-fitur yang disediakan oleh PHP *framework* ini membantu para developer menjadi lebih produktif dengan mengurangi kode berulang pada sebuah proyek.

PHP mampu menjalankan *query* basis data dan menghasilkan *tag* HTML secara langsung. Kedua hal ini bisa digabungkan dalam satu kode logika. Pengembangan web yang lebih sederhana mungkin akan mendapatkan manfaat dari kelebihan ini, namun manfaat ini tidak akan didapatkan apabila proyek web yang dikembangkan menjadi lebih kompleks. Hal ini membuat pemeliharaan dan skalabilitas dengan mengembangkan fitur baru web menjadi sulit, itulah mengapa kita perlu menggunakan *framework* PHP. (Laaziri, Benmoussa, Khoulji, & Kerkeb, 2019)

PHP *framework* membahas bagaimana praktik terbaik untuk membuat aplikasi web PHP yang dinamis. PHP *framework* memberikan aturan bagaimana seharusnya pengembangan aplikasi dilakukan. Aturan ini bisa berupa struktur folder aplikasi, penempatan berkas kode program, sampai kepada aturan dalam memisahkan kode program. Terdapat banyak model *framework* PHP yang dapat digunakan sebagai acuan untuk membuat aplikasi web yang baik. Salah satunya yaitu MVC (*model, view, and controller*). Model ini merupakan sebuah model yang terbukti efektif untuk membuat aplikasi modular dan aplikasi yang terorganisir. MVC sebagai desain model memisahkan aplikasi menjadi beberapa lapisan yang terpisah. Lapisan-lapisan ini membuat analisis kesalahan (*error*) aplikasi dan terkadang diimplementasikan secara terpisah. (Laaziri dkk., 2019; Olanrewaju, Islam, & Ali, 2015)

MVC membantu mengurangi kompleksitas desain arsitektur serta meningkatkan fleksibilitas dan penggunaan ulang kode dengan melepaskan kode

*controller*, *model*, dan *view*. Ketiga lapisan ini memiliki fungsi yang berbeda, di mana *controller* berfungsi sebagai alur logika program, *model* berfungsi untuk menyimpan kode logika yang mengakses *query* ke basis data, dan *View* berfungsi untuk membuat tampilan web yang dinamis. (Olanrewaju dkk., 2015)

### 2.13.5 Codeigniter

Codeigniter merupakan salah satu *framework* PHP yang digunakan untuk mengembangkan web dinamis. Codeigniter memiliki berbagai fungsi yang dapat digunakan dengan mudah layaknya sebuah *framework* pada umumnya. Fungsi ini merupakan kumpulan kode disusun secara terorganisir sehingga kita dapat menggunakan dengan sintaksis yang sederhana. Hal ini membuat penulisan kode yang kita buat menjadi lebih singkat. (Enterprise, 2015)

Ada banyak *framework* PHP yang dapat digunakan untuk membuat web dinamis dengan mudah selain CodeIgniter, seperti Laravel, Symfony, CakePHP, Yii, dll. Beberapa *framework* tersebut memiliki banyak manfaat dan penggunaannya dapat disesuaikan dengan kebutuhan. *Framework* terbaik adalah *framework* yang memiliki fitur terbaik untuk mendukung kebutuhan proyek. Fayyaz dan Madiha membandingkan dan menganalisis performa pada Codeigniter dan CakePHP. Mereka menyimpulkan bahwa CakePHP cocok untuk membuat aplikasi dengan skala perusahaan dan aplikasi besar, sedangkan Codeigniter cocok untuk aplikasi skala kecil dan menengah. (Laaziri dkk., 2019)

Penulis menggunakan *framework* Codeigniter untuk mengembangkan aplikasi pada penelitian ini. Hal ini karena aplikasi sebelumnya, yaitu HC Portal, menggunakan *framework* Codeigniter untuk pengembangannya dan ini menjadi batasan penulis dalam memilih *framework* PHP. Walaupun hal ini menjadi batasan, kita kembali lagi seperti yang sudah dijelaskan pada paragraf sebelumnya yaitu penulis menggunakan *framework* ini sesuai dengan kebutuhan. Penulis tidak mungkin akan menggunakan *framework* lain dan membuat aplikasi ini dari awal karena itu akan memakan waktu yang lebih banyak untuk menyelesaikan penelitian ini.

### 2.13.6 MySQL dan PHPMyAdmin

MySQL merupakan sistem manajemen basis data terpopuler yang dikembangkan di pertengahan tahun 90an. Dewasa ini MySQL merupakan

teknologi basis data yang memberdayakan banyak laman web yang sering dikunjungi. SQL pada MySQL merupakan kepanjangan dari *Structured Query Language*. *Language* (bahasa) yang digunakan adalah bahasa Inggris yang didesain untuk memberikan perintah pada sistem basis data. Contoh penggunaannya bisa dilihat pada gambar berikut. (Nixon, 2015)

```
SELECT title FROM publications WHERE author = 'Charles Dickens';
```

Gambar 2.47 Contoh *query* SQL untuk memberikan perintah pada sistem basis data

*Query* SQL di atas jika di artikan memiliki makna yaitu “tampilkan judul dari tabel *publications* di mana *author*-nya adalah Charles Dickens”. *Query* SQL tersebut merupakan salah satu sintaksis bagaimana memberikan perintah untuk sistem basis data untuk mengolah data. Ada lima jenis *query* SQL yang sering digunakan yaitu, (Peterson, 2021)

- Data Definition Language (DDL), membantu developer untuk mendefinisikan struktur basis data atau skema.
- Data Manipulation Language (DML), memungkinkan developer untuk memodifikasi basis data dengan memasukkan, mengubah, dan menghapus datanya.
- Data Control Language (DCL), merupakan perintah yang berfungsi untuk memberikan dan menghapus otorisasi basis data. Perintah ini memberikan parameter untuk kontrol izin pengelolaan basis data.
- Transaction Control Language (TCL), perintah ini berhubungan dengan transaksi data yang terjadi pada basis data.
- Data Query Language (DQL), perintah ini berhubungan dengan pengambilan data dari basis data.

Masing-masing jenis *query* SQL tersebut berisi perintah yang dapat digunakan untuk mengeksekusi pengelolaan basis data. Gambar berikut menunjukkan *query* apa saja yang ada di masing-masing jenis *query* SQL tersebut:

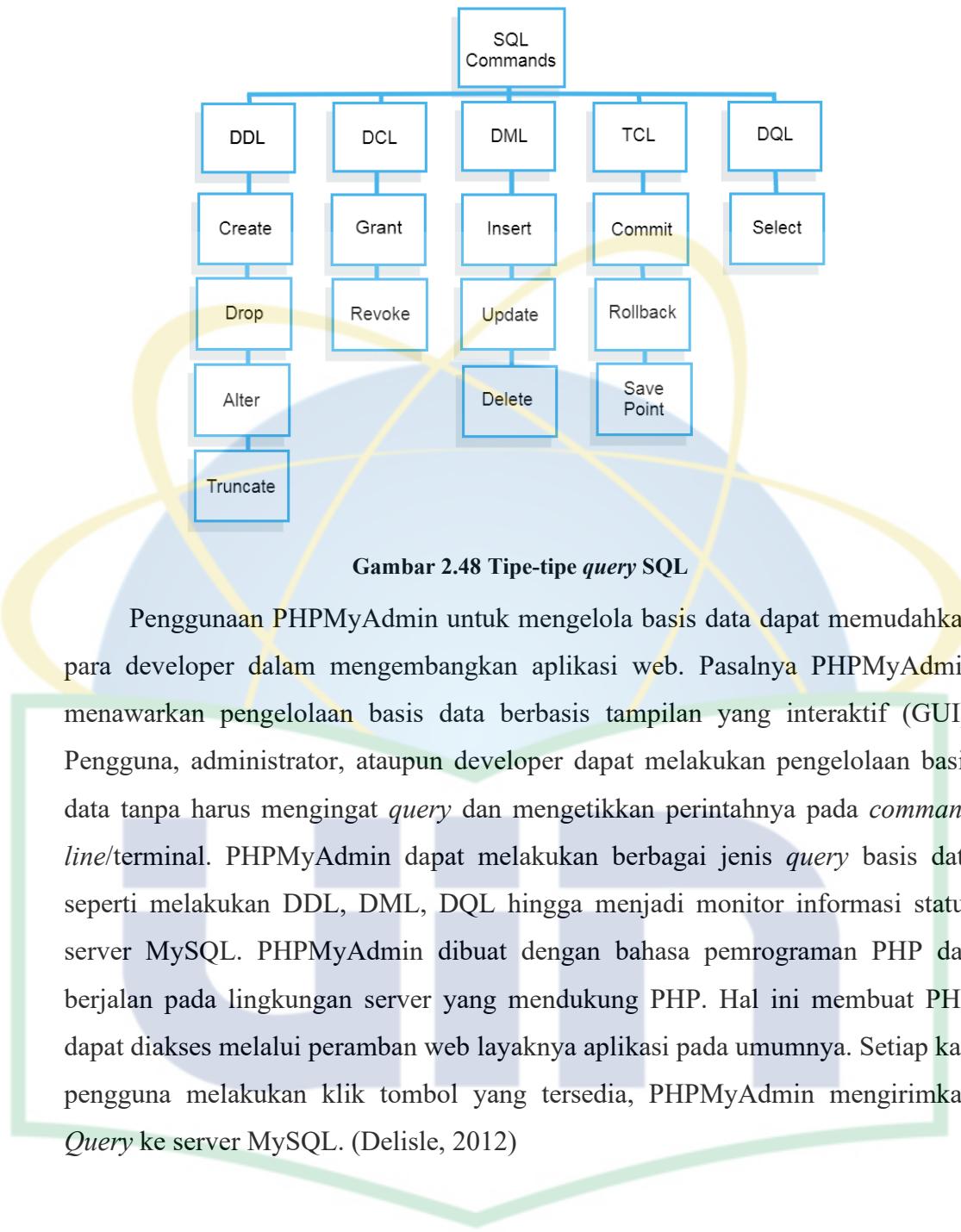


Table	Action	Rows	Type	Collation	Size	Overhead
actor	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	200	InnoDB	utf8_general_ci	32.0 Kib	-
actor_info	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Edit</a> <a href="#">Drop</a>	~0	MyISAM	utf8_general_ci	-	-
address	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	603	InnoDB	utf8_general_ci	96.0 Kib	-
category	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	16	InnoDB	utf8_general_ci	16.0 Kib	-
city	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	600	InnoDB	utf8_general_ci	64.0 Kib	-
country	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	109	InnoDB	utf8_general_ci	16.0 Kib	-
customer	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	599	InnoDB	utf8_general_ci	128.0 Kib	-
customer_list	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Edit</a> <a href="#">Drop</a>	~0	MyISAM	utf8_general_ci	-	-
film	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1,000	InnoDB	utf8_general_ci	272.0 Kib	-
film_actor	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	5,462	InnoDB	utf8_general_ci	272.0 Kib	-
film_category	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1,000	InnoDB	utf8_general_ci	80.0 Kib	-
film_list	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Edit</a> <a href="#">Drop</a>	~0	MyISAM	utf8_general_ci	317.8 Kib	-

Gambar 2.49 Tampilan PHPMyAdmin menampilkan tabel-tabel dari satu basis data

### 2.13.7 Bootstrap dan AdminLTE

Bootstrap merupakan sebuah *framework* untuk penataan gaya (*styling*) tampilan web. Tata letak tampilan sebuah halaman web ditangani oleh HTML, sedangkan untuk penataan gayanya ditangani oleh CSS. Ini memberikan tampilan HTML menjadi lebih cantik dengan mengatur berbagai gaya tampilan seperti penyesuaian ukuran huruf, warna, dan bentuk. CSS memberikan *styling* dengan merujuk pada *tag* HTML. Bootstrap memberikan kemudahan untuk melakukan ini dengan menyediakan berbagai properti *styling* dalam bentuk atribut *class* di HTML. Developer yang ingin menerapkan gaya Bootstrap pada laman web yang mereka buat bisa langsung memanggil nama *class* pada *tag* HTML. Beberapa penataan gaya juga telah diterapkan secara bawaan langsung pada *tag* HTML. (Spurlock, 2013)

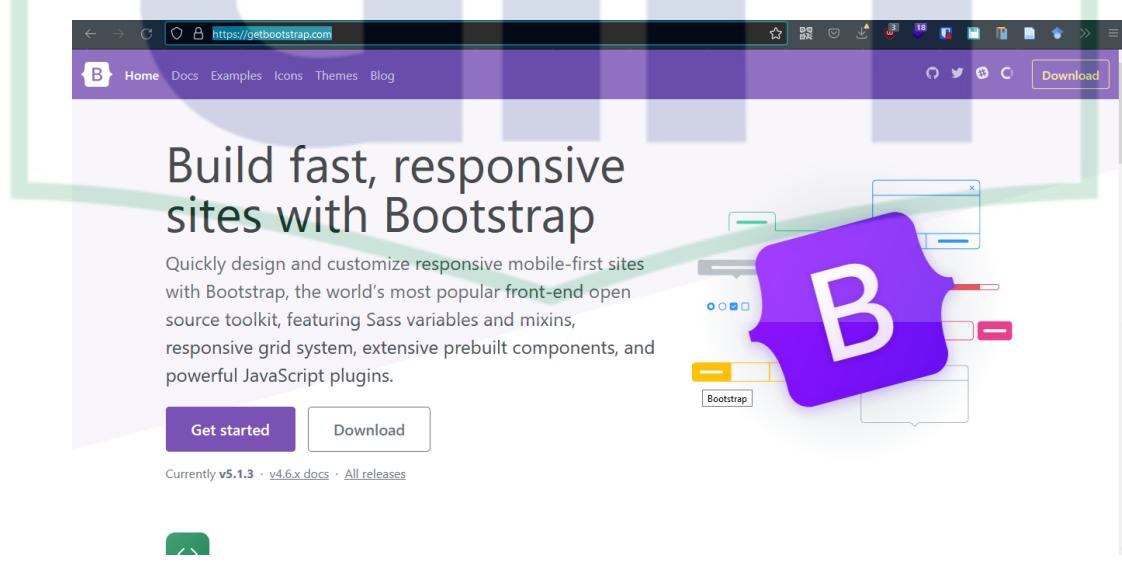
Bootstrap awal mulanya dibuat hanya untuk kepentingan internal perusahaan. Mark Otto dan Jacob Thornton adalah orang yang membuat produk *open source* ini. Mereka sebagai karyawan di Twitter memulai proyek ini untuk melakukan standarisasi alat *frontend* melintasi lingkungan perusahaan. Sebelumnya pengembangan *frontend* di lingkup perusahaan sering kali menemukan tidak adanya konsistensi. Itulah sebabnya *framework* ini dibuat. Kini Bootstrap menjadi *framework* CSS paling populer pada platform berbagi kode pemrograman Github (Shenoy & Sossou, 2014; Spurlock, 2013)

Bootstrap dibuat awalnya hanya untuk pengembangan laman web dan aplikasi berbasis web, namun saat ini karena perkembangan JavaScript yang kian pesat membuat halaman web dapat berjalan layaknya aplikasi di perangkat *smartphone*. Bootstrap dapat diterapkan pada *framework* JavaScript yang digunakan untuk pengembangan aplikasi berbasis *mobile*. Penggunaan *framework* seperti React Native, Ionic Framework, CapacitorJS, dan NativeScript membuat hal ini bisa dilakukan dengan mudah.

Bootstrap dapat digunakan dengan memasukkan berkas.css pada laman web. Sebelumnya berkas ini kita dapatkan dengan mengunduhnya dari halaman web resminya yaitu <https://getbootstrap.com/>. Kita dapat menyisipkan berkas yang telah kita unduh seperti kode HTML di bawah ini. (Spurlock, 2013)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap 101 Template</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Gambar 2.50 Kode HTML yang menerapkan Bootstrap



Gambar 2.51 Halaman web resmi untuk mengunduh Bootstrap

Sebenarnya sudah ada banyak *framework* CSS lainnya seperti Materialize CSS, Tailwind CSS, Semantic UI, dll. Namun, penulis lebih memilih untuk menggunakan Bootstrap karena ada satu templat yang dibuat dengan Bootstrap untuk membuat sebuah tampilan aplikasi Admin, yaitu AdminLTE. Penulis menggunakan templat ini untuk mengembangkan tampilan aplikasi.

AdminLTE dibuat oleh penyedia layanan tema Bootstrap gratis terbesar yaitu Colorlib. Mereka memiliki 700 lebih templat tersedia di halaman web resmi mereka. Colorlib diluncurkan pada tahun 2013 oleh Aigars Silkalns. Ia sering membuat tempat gratis berbasis *open source* untuk para developer yang ingin mengembangkan tampilan aplikasi web admin dengan mudah. AdminLTE menarik banyak minat developer web karena templat ini dibuat di atas *framework* Bootstrap. Hampir semua *class* untuk *styling* pada AdminLTE berdasarkan *framework* Bootstrap dengan penambahan beberapa komponen agar membuat tampilan aplikasi admin berbasis web menjadi lebih mudah. Kemudahan ini yang membuat penulis pada akhirnya menggunakan AdminLTE. (Silkalns, 2021)

## 2.14 Penelitian Sejenis

Penulis telah mempelajari sepuluh penelitian sejenis berupa jurnal-jurnal penelitian maupun skripsi lain. Pembahasan penelitian-penelitian sejenis tersebut penulis susun dalam beberapa paragraf.

Penelitian sejenis pertama berjudul “*Open Source-Style Collaborative Development Practices in Commercial Projects Using GitHub*” yang dilakukan oleh Kalliamvakou dkk. Penelitian ini bertujuan untuk memahami bagaimana penggunaan Github untuk proyek komersial pengembangan *software*. Pendekatan yang ingin dilihat di penelitian ini adalah *Open Source-Style* (OSS). Penelitian ini dilakukan dengan mengambil sampel yang terdiri dari 240 developer *software* untuk mengisi survei penelitian tersebut dan 30 orang pengguna Github untuk melakukan interview semi terstruktur. Hasilnya pendekatan OSS yang biasanya digunakan untuk proyek-proyek pengembangan *software open-source* (sumber terbuka), digunakan pula pada proyek pengembangan *software* di lembaga-lembaga komersial. Mereka juga mendapatkan beberapa manfaat lain apabila proyek pengembangan *software* menggunakan pendekatan OSS, seperti Pengurangan Komunikasi (*reduced communication*), bekerja dengan lebih independen (*more*

*independent work*), dan organisasi diri. (*Self-Organization*). (Kalliamvakou dkk., 2015)

Penelitian tersebut juga membahas sebuah praktis pengembangan *software* di Github. Selain itu mereka juga menyebutkan bagaimana penerapannya dengan fiturnya. Ini menjadi landasan penulis untuk menetapkan fitur Github yang dapat digunakan pada penelitian ini.

Penelitian sejenis kedua berjudul “*The Promises and Perils of Mining GitHub*” yang juga dibuat oleh Kalliamvakou di tahun sebelum penelitian sejenis pertama. Penelitian ini membahas apa saja yang kita dapatkan dan hal yang tidak diinginkan saat ingin melakukan penggalian di repositori Github. Penelitian ini merupakan sebuah studi empiris dengan melihat kondisi repositori terpilih, kemudian mereka mendokumentasikan hasilnya berdasarkan karakteristik repositori-repositori di Github dan melihat bagaimana para pengguna memanfaatkan fitur utama Github seperti *commits*, *pull requests*, dan *issues*. Mereka menjelaskan hasil penelitian ini dalam beberapa poin pembahasan, di antaranya membahas penggunaan fitur-fitur Github. Terlepas dari hasil penelitiannya, penulis mengambil penelitian sejenis ini untuk melengkapi pembelajaran penulis terkait penggunaan fitur-fitur Github berdasarkan studi penelitian yang sudah ada. (Kalliamvakou dkk., 2014)

Penelitian sejenis ketiga berjudul “*Requirement Management in Github with a Lean Approach*” yang dilakukan oleh Salo dkk. Penelitian ini membuat sebuah pedoman (*guideline*) untuk melaksanakan manajemen kebutuhan (*requirement management*) pada pengembangan *software* di Github dengan pendekatan Lean (*Lean Approach*). Pedoman yang mereka buat membahas pemakaian fitur Github untuk persiapan awal pengembangan Agile di Github. Hal yang mereka bahas di sini salah satunya yaitu aturan nama label (*label naming convention*) untuk membuat tugas secara hierarki (*hierarchy task*). Mereka juga membahas bagaimana cara menggunakan fitur *Issue Tracker* Github untuk membahas *User Story* atau *Story Card* dan menjelaskan langkah-langkahnya. Pedoman ini kemudian mereka terapkan pada proyek mahasiswa di jurusan *Information Science* di Universitas Tampere pada tahun 2013. Hasilnya, mereka mendapat respons positif dari mahasiswa yang melaksanakan proyek. Pedoman ini mampu membantu mereka

untuk menyusun *Requirement Management* untuk proyek Agile mereka. (Salo, Poranen, & Zhang, 2015)

Penelitian sejenis keempat berjudul “*HRM portal: Human resource management portal*” yang dilakukan oleh Kittivaraporn dkk. Penelitian ini menyebutnya sebagai Portal HRM, namun dalam penelitian ini bisa disebut dengan Portal HC. Tujuan penelitian ini untuk membuat sebuah aplikasi portal HRM yang tidak hanya memiliki fungsi untuk proses *recruitment*, tetapi juga untuk memberikan informasi kepada para calon penerima kerja terkait status penerimaan mereka, notifikasi melalui surel, sampai memberikan informasi dasar dan terkini seputar perusahaan. Mereka membagi aplikasi menjadi dua bagian, yaitu sistem *backend* dan *frontend*. Aplikasi *frontend* yang mereka maksud adalah modul aplikasi yang dapat diakses oleh pencari kerja, sedangkan Aplikasi *backend* adalah modul aplikasi yang dapat diakses oleh karyawan HC. (Kittivaraporn, Chokdeeanan, Yaophrukchai, & Sunetnanta, 2014)

Penelitian sejenis kelima berjudul “Rancang Bangun E-HRM (*Electronic-Human Resources Management*) (Studi Kasus: BPRS Harta Insan Karimah Kantor Pusat)” yang dilakukan oleh Surya. Penelitian ini bertujuan untuk membuat aplikasi e-HRM masih sama seperti penelitian sejenis sebelumnya. Permasalahannya membahas tentang proses manual yang menggunakan kertas. Hal ini membuat data-data menjadi jarang terorganisir dengan baik, khususnya apabila saat diperlukan. Penelitian ini menggunakan metode pengembangan *software RAD*. Desain sistem dibuat dengan empat diagram UML, yaitu *use case*, *activity*, *class*, dan *sequence*. Hasilnya aplikasi e-HRM memudahkan HRD dalam mengambil suatu keputusan. Hal ini bisa terjadi karena data-data yang dimasukkan maupun dihasilkan dalam proses HRD disimpan dalam sistem basis data. (Surya, 2015)

Penelitian sejenis keenam dan terakhir berjudul “Rancang Bangun Sistem Operasional Budidaya Tambak Ikan Kerapu Berbasis Android (Studi Kasus: Kelompok Tani Bhakti Usaha 2)”. Penelitian ini bertujuan untuk membangun Aplikasi Budidaya Tambak Ikan Kerapu Berbasis Android. Penelitian ini penulis ambil sebagai penelitian sejenis karena pembangunan aplikasi tersebut menggunakan metode *iterative model*. Penelitian ini menjadi satu contoh untuk

penulis bagaimana mengembangkan aplikasi dengan metode tersebut untuk diimplementasikan di dalam penelitian ini. (Wafa, 2019)

Penelitian sejenis ketujuh berjudul “Rancang Bangun *Website* dan *E-Learning* di TPQ Al-Fadhillah” yang dilakukan oleh Fadlilah. Tujuan penelitian ini yaitu membuat *website* dan *e-learning* untuk membantu proses belajar mengajar. Penelitian ini tidak menyebutkan kalau menggunakan spesifik metode penelitian ataupun kerangka kerja (*framework*) apapun dalam melakukan pengembangan aplikasi. Namun, hal yang dilakukan di penelitian ini adalah dengan melakukan penggambaran alur sistem belajar dengan DFD, lalu melakukan spesifikasi fitur yang akan dibuat ke dalam aplikasi. Spesifikasi fitur ini memberikan gambaran kepada penulis bagaimana struktur *e-learning* dari penelitian yang sudah ada. Fadlilah juga melakukan pengujian aplikasi dengan menyebarkan kuesioner kepada beberapa pengguna untuk menilai kualitas aplikasi yang telah ia buat. Hasilnya sesuai dengan harapan yang telah ia sebutkan di tujuan penelitian. (Fadlilah, 2015)

Penelitian sejenis kedelapan berjudul “Rancang Bangun Aplikasi Pelatihan *Online* dengan Metode *Blended Learning System* Berbasis *Website* (Studi Kasus)” yang dilakukan oleh Yusriyanto dkk. Tujuan penelitian ini adalah melakukan rancang bangun aplikasi pelatihan *online* untuk kementerian kesehatan. Sistem pembelajaran yang ia gunakan dalam mengembangkan aplikasi ini adalah *Blended Learning System*. Ini merupakan sebuah sistem pembelajaran yang menggabungkan sistem belajar *online* (daring) dan *offline* (luring). Metode pengembangan yang digunakan pada penelitian ini ada RAD. Desain yang dilakukan pada penelitian ini terdiri dari analisis sistem berjalan dengan *Rich Picture*, serta *Workshop Design* dengan membuat desain UML (terdiri dari diagram *Use Case*, *Activity*, *Class*, dan *Sequence*) dan tampilan aplikasi untuk pengguna (*User Interface*). Aplikasi ini mampu menyediakan sistem pembelajaran secara *Synchronous* (melalui seminar dan ujian daring) dan *Asynchronous* (melalui materi daring berupa berkas dokumen dan video). (Yusriyanto, Arham, & Nuryasin, 2018)

Penelitian sejenis kesembilan berjudul “Rancang Bangun Aplikasi *Smart Training* berbasis Android Menggunakan Flutter dengan Metode RAD” yang dilakukan oleh Luthfyana dan Sediyono. Tujuannya yaitu melakukan pembuatan aplikasi manajemen *training* yang mereka sebut sebagai *Smart Training*. Aplikasi

ini mampu untuk memanajemen seluruh karyawan yang akan melakukan *training*, menyediakan materi *training* dalam bentuk artikel dan video, memberikan tes awal dan akhir (*pre-test and post-test*), mengadakan *feedback* dan *polling* untuk karyawan, melakukan *chat* dengan karyawan, hingga memberikan hasil nilai dari *training* yang diikuti karyawan. Metode pengembangan *software* yang digunakan dalam penelitian ini adalah RAD. Mereka menggunakan UML untuk melakukan desain sistem. Mereka juga melakukan *usability testing* di akhir penelitian. Hasilnya menunjukkan bahwa aplikasi ini dapat menekan biaya operasional untuk pembelian kertas. Selain itu, aplikasi pun mendapatkan nilai *usability testing* sebesar 80,91% yang menunjukkan aplikasi memiliki nilai kegunaan yang sangat baik. (Luthfyana & Sediyono, 2021)

Penelitian sejenis terakhir berjudul “Rancang bangun modul pelatihan Karyawan untuk peningkatan SDM pada USP Swamitra Sumberpucung Malang” yang dilakukan oleh Pamungkas. Tujuan penelitian ini yaitu melakukan rancang bangun *website* pelatihan karyawan. Fitur yang disajikan dalam *website* pelatihan ini adalah halaman awal untuk memperkenalkan USP Swamitra Sumberpucung Malang, fitur materi pelatihan, fitur untuk mengerjakan soal tes, fitur hasil tes, serta fitur manajemen materi pelatihan khusus untuk admin. Metode pengembangan *software* yang digunakan adalah *waterfall*. Desain sistemnya menggunakan *flowchart*, DFD, dan ERD. Ia juga melakukan desain tampilan *website*. Hasilnya *website* cukup memenuhi kebutuhan *training* karyawan. Ini diambil dari hasil pengambilan survei kepada 10 pengguna yang merupakan karyawan di USP Swamitra Sumberpucung Malang. (F. L. Pamungkas, 2016)

Penelitian sejenis ketujuh sampai yang terakhir membahas proyek pengembangan *e-learning* dan *e-training* untuk karyawan. Penelitian-penelitian tersebut menjelaskan bagaimana mereka membangun *software* tersebut. Walaupun metode pengembangan *software* yang digunakan berbeda dari penelitian ini, penulis dapat mempelajari fitur yang biasanya dipakai dalam pengembangannya.



## **BAB III**

### **METODE PENELITIAN**

Bab ini menjelaskan metode penulis dalam melaksanakan penelitian ini. Pembahasan metode penelitian dibahas dalam tiga subbab, yaitu metode pengumpulan data, metode pengembangan perangkat lunak, dan kerangka penelitian.

#### **3.1 Metode Pengumpulan Data**

Penulis melakukan tiga teknik pengumpulan data, yaitu observasi, wawancara, dan studi literatur. Penulis melakukan ketiga metode pengumpulan ini untuk mendapatkan data dan informasi pendukung. Penulis melakukan observasi dengan mengikuti program magang yang diadakan perusahaan sekaligus penulis menyusun studi kasus untuk penelitian ini. Penulis juga melakukan wawancara awal dengan tujuan untuk mendapatkan latar belakang masalah yang konkret. Ini penulis lakukan setelah mendapatkan topik bahasan penelitian. Metode studi literatur bertujuan untuk mendapatkan informasi terkait topik penelitian. Metode ini digunakan untuk membantu penulis dalam menyusun solusi atas permasalahan yang ditemukan.

##### **3.1.1 Observasi**

Penulis melakukan observasi dengan melaksanakan program magang yang diadakan oleh divisi HC CTI. Penulis melaksanakan program magang ini terhitung dari tanggal 19 Maret sampai 18 Desember 2020. Penulis melaksanakan program magang tersebut di hari kerja tepatnya di hari Senin – Jumat. Penulis mengikuti jam kerja di CTI layaknya sebagai seorang karyawan di sana yang biasa disebut sebagai karyawan magang. Penulis menelusuri kebutuhan perusahaan atas pertukaran informasi yang terjadi di lingkungan perusahaan dan mempelajari bagaimana untuk menjadi bagian dalam satu perusahaan selagi mengerjakan tugas-tugas yang diberikan oleh mentor penulis.

Pekerjaan yang penulis lakukan di program magang tersebut adalah mengembangkan aplikasi web HC Portal. Aplikasi ini digunakan untuk mendukung proses internal yang ada di lingkungan perusahaan. Tentunya, aplikasi ini digunakan untuk semua proses yang melibatkan divisi HC. Aplikasi web itu

berhasil membuat divisi HC dapat memantau beberapa proses kerja yang sebelumnya sulit untuk dilakukan. Selain itu, aplikasi mampu mengurangi penggunaan kertas dan dapat mempercepat proses kerja karena prosesnya yang dilakukan secara terpusat dalam satu aplikasi. Karyawan HC dan karyawan di divisi lainnya yang terlibat dalam proses kerja HC terhubung dalam aplikasi web HC Portal itu.

Selang setelah beberapa lama mengikuti program magang, penulis mulai menanyakan akan kebutuhan baru yang diperlukan oleh divisi HC. Penulis menemukan beberapa hal, seperti:

- Divisi HC memerlukan cara baru dalam melakukan *training* karyawan karena terdapat dua kendala dalam pelaksanaan *training* saat ini:
  - Dokumen-dokumen informasi dan data terkait proses *training* yang tersebar di beberapa tempat penyimpanan seperti di komputer lokal karyawan HC maupun di penyimpanan *cloud* menyulitkan proses pencarian dan pengumpulan data saat ingin membuat laporan hasil proses *training*.
  - Proses *training* yang diadakan secara luring memerlukan banyak waktu dan biaya. Proses *training* secara daring tidak semua karyawan dapat mengikutinya, selalu saja ada karyawan yang tidak aktif saat proses *training* secara daring berlangsung.
- HC memerlukan agar *e-learning* ini dapat dikembangkan pada HC Portal, yaitu aplikasi web yang telah memiliki beberapa fungsi proses kerja. Hal ini agar semua proses kerja HC dapat diakses secara terpusat di aplikasi web tersebut. *Training* karyawan adalah salah satu proses kerjanya.
- Mereka juga memerlukan agar aplikasi HC Portal dapat terus dikembangkan mengikuti kebutuhan proses kerja di divisi HC. Masalah ini memberikan tantangan kepada penulis untuk dapat merumuskan solusi yang tidak hanya berupa produk *e-learning* saja. Namun, penulis juga perlu merumuskan solusi agar mendorong pengembangan yang terus berlanjut untuk HC Portal ini walaupun nanti penulis telah meninggalkan proyek ini. Hal ini juga untuk mendorong pengembangan HC Portal selanjutnya yang lebih mudah untuk dilakukan.

Penulis telah melakukan observasi server produksi saat mengerjakan tugas magang yaitu mengembangkan aplikasi HC Portal. Pengembangan aplikasi ini telah dilakukan oleh karyawan magang sebelumnya. Penulis melanjutkan pengembangannya dalam program magang tersebut. Kemudian, penulis mengangkat topik pengembangan *e-learning* dalam penelitian ini dengan beberapa kondisi permasalahan yang telah dibahas di atas.

### 3.1.2 Wawancara

Penulis melakukan wawancara untuk menggali lebih lanjut tentang urgensi dan kebutuhan pengembangan *e-learning* sebagai modul aplikasi HC Portal. Penulis melakukan wawancara ini dengan mentor magang penulis pada tanggal 7 April 2021 tepatnya pada jam 19.57 – 22.13. Mentor magang penulis merupakan seorang HCIS Specialist yang merupakan tangan kanan dari kepala divisi HC CTI. Beliau menampung ide ini dari rekan-rekan kerja yang bertugas untuk melakukan rekrutmen karyawan baru. Penulis membuat draf pertanyaan sebelum melakukan wawancara yang sesungguhnya. Ini penulis lakukan agar proses wawancara dapat berjalan dengan lebih efektif dan informasi yang didapatkan dapat memenuhi kebutuhan penulis untuk menyusun penelitian ini.

Wawancara ini menghasilkan berbagai informasi yang penulis perlukan dalam pengembangan *e-learning* ini. Berikut beberapa informasi yang penulis dapatkan dalam wawancara ini, di antaranya informasi terkait:

- Proses *learning* dan *training* yang sudah ada di CTI.
- Masalah yang terjadi pada proses *learning* dan *training* di CTI.
- Ekspektasi mereka terhadap *e-learning* ini.
- Platform pengembangan *e-learning*.
- Saran terkait proses *learning* dan *training* di *e-learning*.

### 3.1.3 Studi Literatur

Penulis melakukan studi literatur untuk mencari dan memahami lebih lanjut informasi yang mendukung penelitian ini. Penulis mencari informasi dari berbagai buku, halaman web, dan penelitian sejenis dari skripsi lain ataupun jurnal penelitian. Penulis mencari topik pembahasan tentang *e-learning*; serta pengembangan *software*, metodenya, dan *tools* yang mendukung pengembangan *software* yang berkelanjutan.

Penulis menemukan beberapa topik yang sesuai dengan kebutuhan penulis untuk penelitian ini. Topik tentang *e-learning* merupakan topik yang cukup umum dibahas dalam dunia penelitian. Penulis menemukan topik yang membahas tentang penerapan *e-learning* dan pengembangannya, serta beberapa penelitian yang membahas uji kegunaan (*usability testing*) dan menjelaskan manfaatnya pada satu lembaga atau instansi tertentu.

Penulis menemukan beberapa topik pembahasan tentang pengembangan *software* yang berkelanjutan secara bertahap. Penulis mengawalinya dengan menemukan bahwa untuk melakukan pengembangan *software* yang berkelanjutan diperlukan sebuah metode pengembangan secara iterasi dan *tools* untuk mendukungnya. Kemudian penulis mendalami konsep pengembangan *Agile Development* yang mengedepankan iterasi, dan memilih metode pengembangan XP (*eXtreme Programming*). Kemudian penulis memilih Github sebagai *tools* pengembangan *e-learning*.

Sayangnya, penulis tidak dapat menerapkan metode pengembangan XP ke dalam penelitian ini. Hal ini karena penulis memerlukan metode analisis dan perancangan sistem untuk kebutuhan penelitian akademis. Pada akhirnya, penulis memutuskan untuk menggunakan metode *iterative model*. Metode pengembangan *software* ini sesuai dengan kebutuhan pengembangan *software* yang berkelanjutan dan memenuhi kebutuhan penulis untuk melaksanakan analisis dan perancangan sistem. Penulis telah membahas lebih lanjut terkait metode penelitian ini pada [bab sebelumnya](#).

### **3.2 Metode Pengembangan *Software***

Metode pengembangan *software* pada penelitian ini merupakan metode yang penulis ambil dari *Iterative Model* berdasarkan referensi-referensi yang telah penulis temukan. Berikut fase dan tahapan dalam pengembangan *software* dengan *iterative model*:

#### **1. *Requirement (Kebutuhan)***

Penulis pada fase ini membahas gambaran besar aplikasi *e-learning* yang ingin dibangun. Penulis melakukan beberapa pengumpulan data untuk mendukung penelitian ini, di antaranya:

- Mengumpulkan dan menyusun informasi umum perusahaan CTI yang bertujuan untuk mengetahui profil, visi misi dan struktur organisasi perusahaan.
- Menganalisis masalah yang terjadi pada proses *training* karyawan di divisi HC CTI.
- Mengidentifikasi aplikasi web HC Portal dengan membuat gambaran besar kondisi aplikasi HC Portal saat ini.
- Menganalisis aplikasi usulan yang dapat menyelesaikan permasalahan.
- Menganalisis struktur aplikasi dan mengulas teknologi yang digunakan.
- Mengidentifikasi server produksi sebagai tempat pemasangan aplikasi portal HC.
- Melakukan persiapan awal dengan menetapkan komponen dalam membangun aplikasi, termasuk kebutuhan perangkat keras (*hardware*), dan perangkat lunak untuk membantu pengembangan sistem (*software tools*).

Selain itu, penulis membuat lingkungan pengembangan dengan membuat repositori digital di Github dan mengatur segala keperluannya untuk memenuhi kebutuhan pengembangan *e-learning*.

Penulis membuat templat format penulisan *Issues* untuk mencatat kebutuhan ke depannya. Selain pembuatan templat *issue*, penulis juga membuat templat PR (*pull request*). Ini digunakan untuk menceritakan apa yang telah dikembangkan dalam *branch* yang berisi perubahannya. PR dapat dihubungkan ke *issue*. Penulis membuat templat PR agar penulisannya jelas dan dapat dipahami segala perubahan yang telah dilakukan. Kedua templat ini penulis buat untuk memudahkan dalam mendokumentasikan pengembangan *software*. Ini nantinya dapat dipelajari oleh pengembang *software* di masa depan untuk melanjutkan proses pengembangan aplikasi HC Portal.

Fase ini nantinya akan diulang kembali pada fase *Iteration* untuk menyesuaikan dengan pengembangan inkremental aplikasi selanjutnya.

## 2. Iteration (Iterasi)

Penulis pada fase ini melakukan pengembangan iterasi pengembangan *e-learning*. Fase ini memiliki beberapa tahapan dalam melakukan pengembangan *software*, di antaranya sebagai berikut:

### a. Analysis & Design (Analisis & Desain)

Penulis pada tahap ini melakukan analisis sistem berjalan. Analisis dilakukan dengan membahas alur proses kerja HC. Penulis melakukan analisis ini dengan mentor penulis. Pada penelitian ini, penulis membahas alur proses *training* karyawan yang dimulai saat menjadi karyawan baru, dan proses *training* untuk karyawan lama. Penulis juga membahas bagaimana proses pengelolaan *training* yang dapat dilakukan oleh divisi lain dengan persetujuan dari divisi HC.

Penulis juga melakukan identifikasi masalah untuk menentukan solusi usulan yang dapat membantu proses tersebut. Penulis membuat desain modul *e-learning* untuk ditambahkan ke dalamnya. Penulis merancang aplikasi HC Portal dengan menggunakan beberapa diagram UML. Berikut perancangan-perancangan yang dibuat dalam penelitian ini:

- a. Analisis Sistem Berjalan dan Usulannya
- b. *Use Case Diagram*
- c. *Activity Diagram*
- d. *Class Diagram*
- e. Pemodelan skema basis data
- f. Spesifikasi basis data
- g. *Sequence Diagram*
- h. Struktur menu
- i. *User Interface (UI)*

### b. Implementation/Development (Implementasi/Pengembangan)

Penulis mengawali tahapan ini dengan mengumpulkan kebutuhan dengan mencatatnya menggunakan Github *Issues*. Kemudian, penulis melakukan manajemen tugas dengan Github Project. *Issues* yang telah dibuat oleh penulis masuk ke Github Project.

Penulis membuat kode program modul *e-learning* pada aplikasi HC Portal sesuai dengan perancangan yang telah dibuat di tahapan sebelumnya. Penulis juga melakukan pembaruan perancangan apabila terdapat perubahan dari mentor penulis ataupun terdapat logika yang mungkin bisa diubah mengikuti logika program aplikasi. Pembaruan perancangan ini dilakukan untuk membuatnya menjadi lebih detail. Penulis membangun aplikasi HC Portal dengan menggunakan salah satu

*framework* PHP yaitu CodeIgniter. Penulis menggunakan MySQL sebagai sistem basis datanya.

#### c. Testing & Revise (Revisi)

Penulis melakukan 2 level testing, yaitu *alpha testing* dan *beta testing*. Penulis melakukan *alpha testing* untuk memastikan semua proses dan fungsi pada fitur yang telah dibuat berjalan dengan lancar sebelum nantinya penulis melakukan *beta testing* dengan klien dan pengguna fitur (*user*).

Penulis menggunakan teknik *black-box testing* agar fokus pada keberhasilan fungsionalitas aplikasi. Penggunaan teknik testing ini dapat mempersingkat waktu testing karena fokusnya adalah pada fungsionalitas program ketimbang struktur kodennya. Penggunaan *black-box* testing juga memudahkan penulis untuk mendapatkan umpan balik dari pemilik proyek karena eror yang ditampilkan berhubungan dengan alur proses *training*.

Penulis juga melakukan revisi apabila terdapat kerusakan/kesalahan (*error*) pada fitur yang telah dibuat. Proses ini dilakukan secara berulang sampai fitur benar-benar siap untuk memasuki lingkungan sistem yang sebenarnya (*Production Server*).

#### d. Production (Produksi)

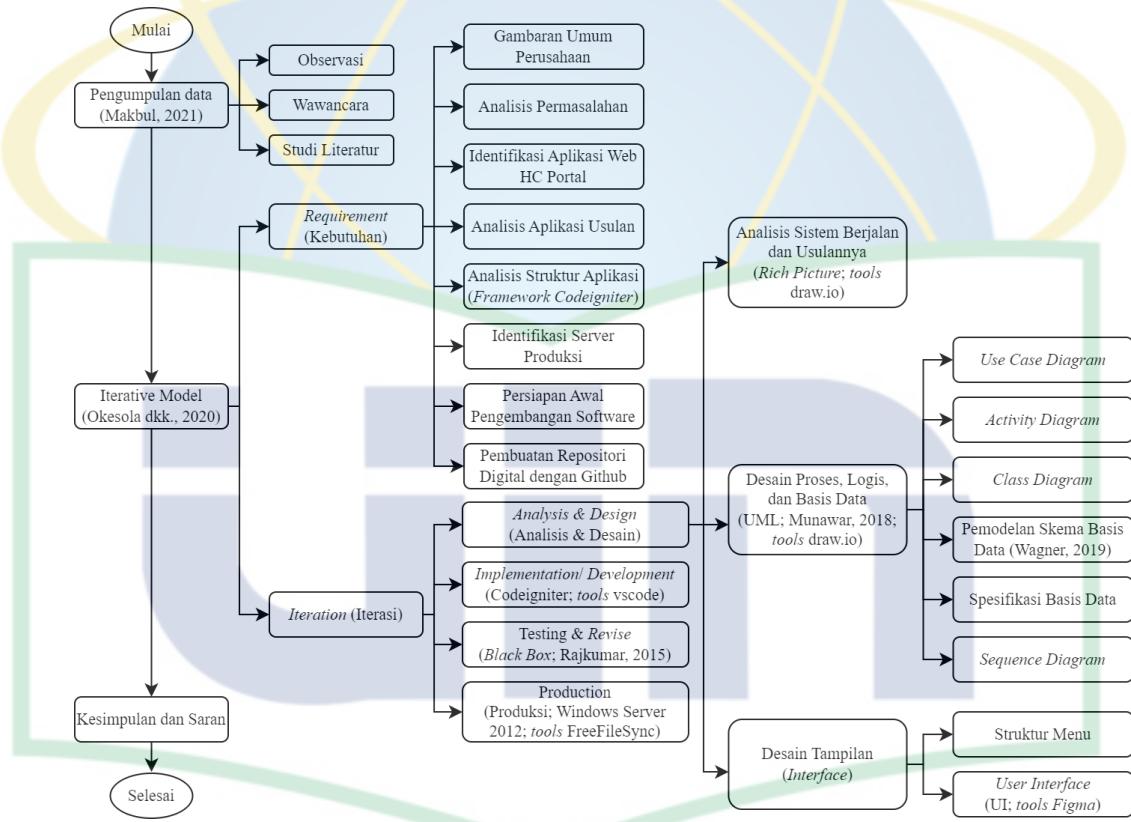
Penulis melakukan pembaruan fitur pada *Production Server*, tentunya setelah berhasil melalui tahapan sebelumnya, termasuk instalasi, konfigurasi, pengoperasian, dan membuat perubahan yang diperlukan. Penulis melakukan instalasi lingkungan pemrograman PHP, server MySQL, dan ekstensi-ekstensi lainnya agar aplikasi HC Portal dapat berjalan pada server produksi. Penulis kemudian melakukan konfigurasi server. Penulis melakukan konfigurasi server MySQL dengan mengimpor basis data yang telah dibuat saat melakukan pengembangan. Penulis juga mengubah konfigurasi lingkungan pemrograman dan domain server sampai aplikasi benar-benar bisa diakses melalui browser.

Tahapan ini merupakan tahapan akhir dalam satu iterasi pengembangan *software*. Pengembangan dinyatakan selesai apabila tidak ada iterasi pengembangan lagi. Hal ini membawa pengembang *software* ke status pemeliharaan (*maintenance*). Pengembang *software* seperti penulis menyerap umpan balik dari pengguna aplikasi HC Portal untuk melakukan penyempurnaan

aplikasi secara terus menerus. Umpulan balik ini dapat ditampung dalam *issues* di Github. Beberapa *issues* ini nantinya akan menjadi tugas dalam iterasi pengembangan baru untuk aplikasi HC Portal ke depannya. Ini mendukung kebutuhan divisi HC untuk melakukan pengembangan aplikasi HC Portal secara berkelanjutan. Pengembangan *software* akan diulang kembali ke tahapan Analisis dan Desain apabila terdapat permintaan kebutuhan untuk menambahkan modul baru ke aplikasi HC Portal.

### 3.3 Kerangka Penelitian

Kerangka penelitian merupakan alur kegiatan penulis dalam melakukan penelitian ini. Berikut kerangka penelitian ini secara keseluruhan:



Gambar 3.1 Kerangka Penelitian



## BAB IV

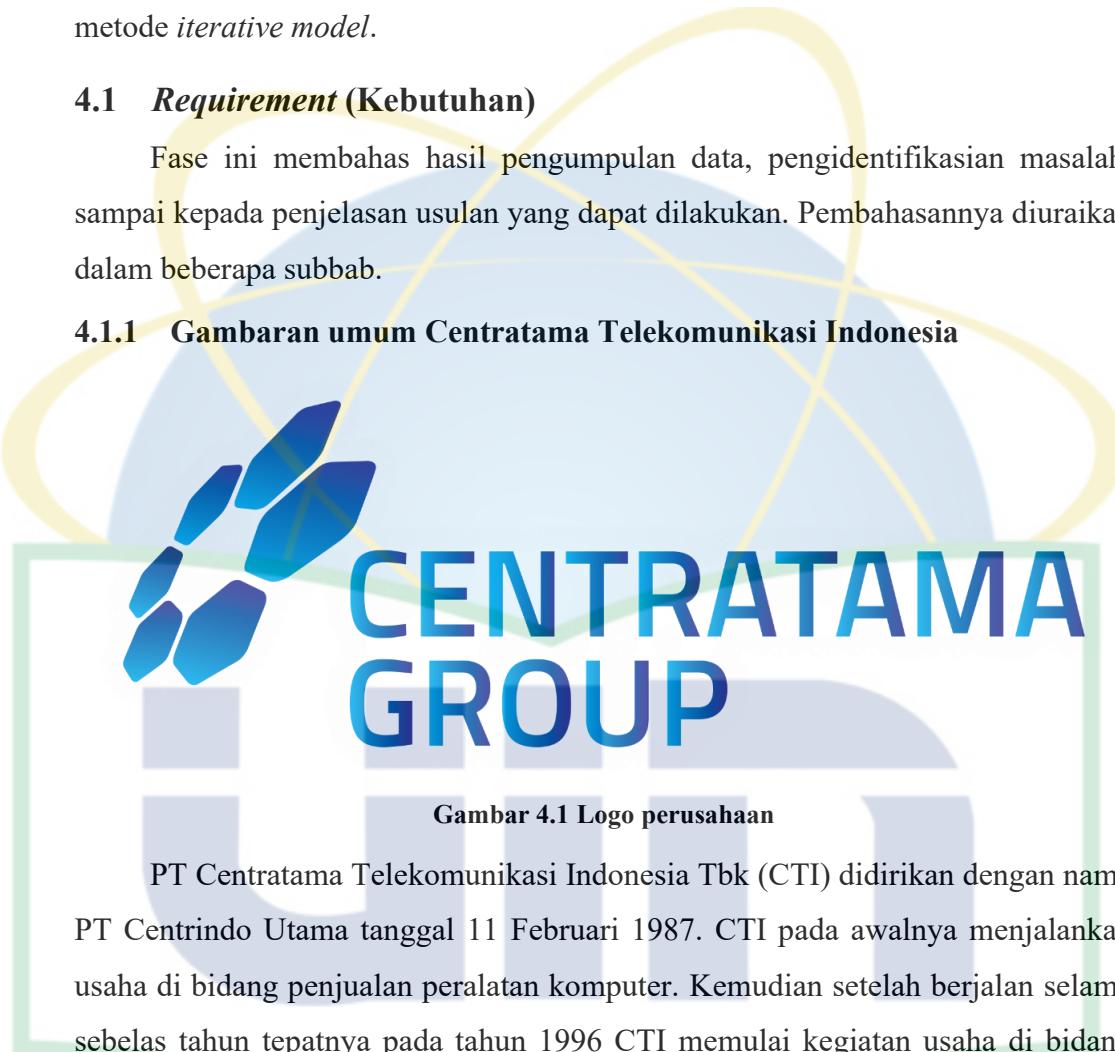
### HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil penelitian yang dilakukan oleh penulis. Penjelasan dalam bab ini diuraikan dalam setiap tahapan pengembangan *software* dengan metode *iterative model*.

#### 4.1 **Requirement (Kebutuhan)**

Fase ini membahas hasil pengumpulan data, pengidentifikasi masalah, sampai kepada penjelasan usulan yang dapat dilakukan. Pembahasannya diuraikan dalam beberapa subbab.

##### 4.1.1 Gambaran umum Centratama Telekomunikasi Indonesia



Gambar 4.1 Logo perusahaan

PT Centratama Telekomunikasi Indonesia Tbk (CTI) didirikan dengan nama PT Centrindo Utama tanggal 11 Februari 1987. CTI pada awalnya menjalankan usaha di bidang penjualan peralatan komputer. Kemudian setelah berjalan selama sebelas tahun tepatnya pada tahun 1996 CTI memulai kegiatan usaha di bidang Penyelenggara Jasa Internet. CTI telah mengalami berbagai pengembangan usaha sejak saat itu, seperti membuat beberapa anak perusahaan, mengakuisisi sebagian atau seluruh saham dari beberapa perusahaan lain, sampai menyinergikan proyek-proyek anak perusahaan dengan kegiatan utama perusahaan yang bergerak dibidang investasi secara maksimal. CTI saat ini dikenal sebagai penyedia infrastruktur jaringan di Indonesia dengan bisnis utamanya melakukan kegiatan usaha dalam bidang perdagangan, jasa, dan melakukan investasi atau penyertaan pada perusahaan lain.

CTI memiliki empat anak perusahaan yaitu PT Centratama Menara Indonesia (CMI), PT Network Quality Indonesia (NQI), PT MAC Sarana Djaya, dan PT Fastel Sarana Indonesia. CMI bergerak pada bidang penyediaan dan pemeliharaan infrastruktur menara telekomunikasi. NQI bergerak pada bidang sarana dan prasarana jaringan telekomunikasi, serta pemborong bidang telekomunikasi. PT MAC Sarana Djaya bergerak pada penyediaan layanan jaringan *Distributed Antenna System* (DAS) di dalam ruangan (*in-building DAS*). Anak perusahaan ini menyediakan infrastruktur yang dibutuhkan oleh operator seluler yang menjalankan jaringan 2G, 3G, dan 4G di dalam ruangan sebuah gedung seperti gedung perkantoran, kompleks apartemen, rumah sakit, universitas, pusat perbelanjaan (mal), pusat bisnis (*business centre's*), kompleks *mixed-used*, dan masih banyak lagi. PT Fastel Sarana Indonesia memiliki bisnis utama dalam penyediaan jasa akses internet data *dedicated* atau *shared* dengan kuota tanpa batas melalui serat fiber optik. Perusahaan ini juga memiliki dua anak perusahaan, yaitu PT. Sentra Inovasi Prima dan Clink Digital Media. Masing-masing perusahaan ini bergerak pada layanan solusi kebutuhan hunian dalam aplikasi yang mereka beri nama Yipy dan layanan periklanan *Out-of-Home* dengan penempatan di beberapa mal, toko retail, apartemen, perkantoran, dan tempat nongkrong bareng (*hangout places*).

#### A. Visi dan Misi

CTI sebagai perusahaan yang dikenal sebagai penyedia infrastruktur jaringan di Indonesia memiliki visi yaitu menjadi pemain utama dalam bisnis penyediaan infrastruktur telekomunikasi dan internet berkecepatan tinggi di industri ICT (*Information and Communication Technology*/Teknologi Informasi dan Komunikasi (TIK)) di Indonesia.

Visi tersebut mereka realisasikan dengan membuat misi sebagai berikut:

- a. Memberikan nilai tambah bagi para pihak yang berkepentingan.
- b. Mengembangkan landasan dan proses bisnis modern yang dapat diandalkan untuk terciptanya performa organisasi yang memiliki tingkat efektivitas tinggi.
- c. Berkomitmen untuk membangun infrastruktur yang paling efisien dan berkualitas melalui perencanaan dan manajemen proyek yang matang.

- d. Berkomitmen untuk menjaga jaringan dan standar layanan yang tinggi guna terus meningkatkan kepuasan pelanggan.
- e. Terus memajukan produk dan layanan.

## B. Struktur Organisasi Perusahaan

CTI memiliki lima divisi kerja dan dua divisi Eksekutif lainnya. Divisi-divisi kerja ini sebagai komponen utama dalam keberlangsungan perusahaan.



**Gambar 4.2 Struktur Organisasi PT Centratama Telekomunikasi Indonesia (CTI)**

## C. Produk dan Jasa

CTI memiliki beberapa produk dan jasa sebagai hasil dari pengelolaan sumber daya yang mereka lakukan dalam melakukan bisnis di bidang TIK. Berikut beberapa produk dan jasa tersebut:

- *New Build (B2S) Micro & Macro*: CTI membangun menara jaringan dengan ketinggian sampai dengan 20-meter dari permukaan tanah untuk menara mikro dan dengan ketinggian lebih dari 20-meter untuk menara makro. Pembangunan ini dilakukan untuk memenuhi kebutuhan pelanggan operator seluler guna memiliki jangkauan jaringan yang luas.
- *Internet Service/Triple Play*: CTI menyediakan layanan akses internet menggunakan jaringan serat optik dan melayani baik pelanggan retail maupun korporat. Layanan ini memiliki *Network Operation Center* yang memberikan pelayanan selama 24/7 dan terintegrasi dengan *Helpdesk* atau *Customer Care System* sehingga dapat melayani keluhan atau permintaan pelanggan sepanjang waktu, serta didukung oleh tim pemeliharaan (*maintenance*) yang berpengalaman.
- *Base Transmission Station (BTS) Hotel*: BTS Hotel adalah sebuah infrastruktur telekomunikasi memfasilitasi komunikasi nirkabel antara piranti komunikasi dan jaringan operator. Layanan ini disentralisasi pada suatu tempat yang melayani beberapa *remote site* yang tersebar dan terhubung dengan serat optik. Konsep BTS Hotel merupakan solusi alternatif tatkala

pembangunan menara makro di area-area tertentu sudah tidak diizinkan karena kepadatan yang tinggi. Layanan ini bekerja dengan membangun beberapa menara mikro berbentuk tiang (*pole*) yang terhubung pada satu BTS Hotel. Layanan ini menjadi pilihan tepat sekaligus efisien untuk memperkuat jaringan operator seluler di area dengan kepadatan yang tinggi.

- Jaringan Penguat Sinyal Dalam Gedung (*Distributed Antenna System – Das*): CTI merupakan penyedia jaringan DAS terbesar di Indonesia. Layanan ini untuk digunakan bersama oleh beberapa operator di dalam gedung sehingga sinyal operator seluler dapat menjangkau pelanggan mereka di dalam gedung. Operator menyewa jaringan DAS yang dimiliki oleh CTI, kemudian menghubungkan perangkat BTS mereka ke jaringan DAS.
- Jaringan Wi-fi: CTI menyediakan akses jaringan Wi-Fi di area publik atau *hot spot* di dalam gedung. Layanan ini memberikan akses Wi-Fi kepada penghuni gedung dan manajemen gedung.
- *Digital Signage*: Layanan ini berfungsi untuk melakukan Pengiriman konten visual digital. Layanan ini dikelola dan dikendalikan secara terpusat melalui jaringan internet. Layanan ini bisa diterapkan pada periklanan, promosi, pengumuman, dan produk lainnya yang menggunakan layar digital media. Teknologi pada layanan ini menghilangkan biaya percetakan dan pengiriman. Hal ini membuat layanan ini lebih ramah lingkungan daripada produk cetak/fisik karena mengurangi penggunaan kertas.

IP Transit: Layanan ini didukung oleh jaringan yang saling terhubung satu sama lain di dunia. CTI menawarkan jaringan dengan kualitas tinggi, terukur, dan terjangkau. Layanan ini mengombinasikan pilihan lokasi layanan yang luas dengan koneksi yang luar biasa ke akses utama dan jaringan konten di seluruh dunia. CTI memastikan kualitas pekerjaan yang bersifat global dengan jaringan khusus dan aman yang menghubungkan seluruh dunia.

#### 4.1.2 Analisis Permasalahan

Penulis melakukan analisis permasalahan utama pada HC CTI setelah melakukan pengumpulan data. Penulis menemukan beberapa poin masalah yang

dapat penulis jadikan acuan dalam membangun aplikasi ini. Poin-poin tersebut telah dibahas pada subbab 3.1.1. Berikut penjabaran lengkapnya:

### A. Cara baru untuk melaksanakan *training*

Seperti yang telah dijelaskan pada bab sebelumnya, HC memiliki kendala dengan proses *training* yang ada saat ini, yaitu masalah pengelolaan data dan informasi terkait *training* yang tersebar di beberapa tempat penyimpanan, sulitnya menyiapkan proses *training* secara luring, dan tidak semua karyawan dapat mengikuti proses *training* secara daring dengan baik. Untuk itu, penulis pada penelitian ini membuat aplikasi *e-learning* untuk memberikan solusi atas permasalahan tersebut.

Saat melakukan wawancara, penulis mendapatkan informasi berupa harapan fitur utama untuk *e-learning* ini seperti,

- **Pre-test**, para mentor bisa melakukan penilaian awal terlebih dahulu untuk menilai sejauh mana pengetahuan *trainee* sebelum mengikuti materi.
- **Post-test**, para *trainee* bisa menilai sejauh mana pengetahuannya terhadap materi yang telah diikuti dan bisa langsung melihat hasilnya.
- **Learning Track**, para *trainee* bisa melihat sejauh mana materi yang telah diikuti dan materi yang akan dia ikuti selanjutnya, dan mentor dapat memantau progres para *trainee*.
- **Quiz**, para *trainee* bisa mengisi kuis di setiap akhir modul pembelajaran untuk mengetahui kualitas pemahaman pada satu modul materi.
- **Forum**, fitur ini digunakan untuk melakukan proses tanya dan jawab dengan mentor dan peserta lain. Para *trainee* bisa mencari jawaban atas materi yang diikuti maupun menanyakannya langsung. Para *trainee* bisa saling bertukar pikiran juga di fitur ini.

Selain kelima fitur tersebut, mereka memerlukan fleksibilitas untuk menonaktifkan beberapa fitur yang mungkin tidak digunakan dalam satu *training*. Mereka juga memerlukan agar fitur tes juga diberi fitur *timer* (penghitung waktu mundur). Jadi ketika waktu telah habis dalam penggerjaan tes, pengguna akan dipaksa untuk diarahkan ke pesan waktu telah habis dan mereka tidak dapat melanjutkan penggerjaan tes. Kemudian untuk jenis materi pembelajaran, mereka

memerlukan agar setiap modul pembelajaran dapat mendukung materi dalam bentuk Teks modul dan Gambar, Video, dan Presentasi Powerpoint.

## B. Pembuatan *e-learning* pada HC Portal

HC memerlukan agar pengembangan *e-learning* ini bisa menjadi modul pada aplikasi web HC Portal. Aplikasi ini merupakan aplikasi internal milik HC yang berisi berbagai proses kerjanya. Inilah yang membuat aplikasi ini merupakan aplikasi sentral untuk divisi HC. Setiap modul yang ada pada aplikasi web HC Portal memuat satu proses kerja HC. Beberapa proses kerja yang telah ditambahkan ke dalam HC Portal yaitu pengelolaan *Job Profile*, pengelolaan dokumen surat keluar masuk, pengelolaan pemintaan perekrutan karyawan baru (*Employee Requisition*), penilaian kontrak karyawan, dan pelaksanaan survei.

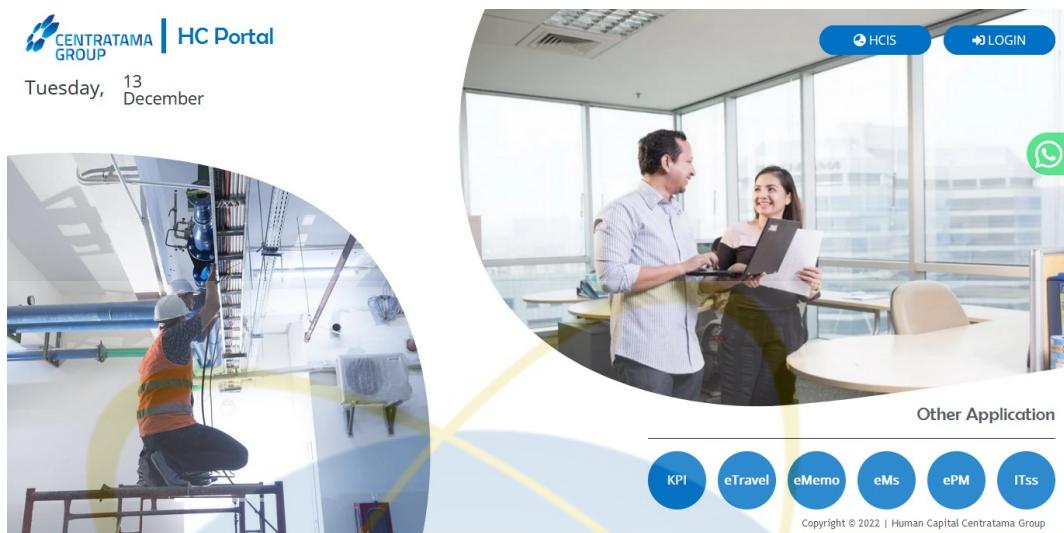
## C. Pengembangan *software* yang berkelanjutan

HC memiliki kebutuhan agar aplikasi HC Portal dapat terus dikembangkan mengikuti kebutuhan proses kerja di divisinya. Penulis perlu menyusun solusi untuk mendukung kebutuhan tersebut. Pengembangan yang berkelanjutan dimaksudkan untuk developer baru yang akan melanjutkan pengembangan aplikasi web HC Portal walaupun penulis telah meninggalkan pengembangannya.

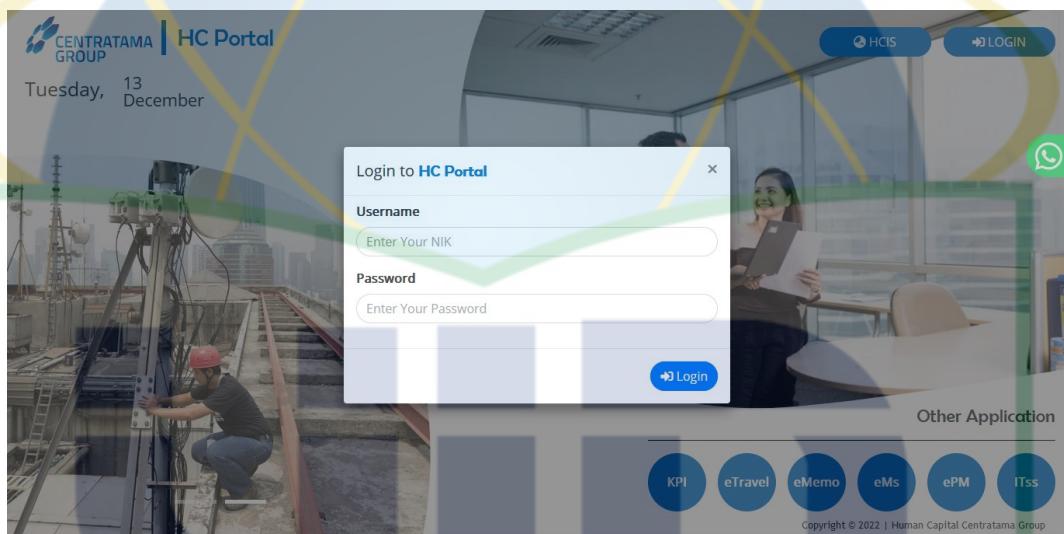
### 4.1.3 Identifikasi Aplikasi web HC Portal

Penulis melakukan identifikasi aplikasi web HC Portal dengan menggunakan aplikasinya langsung. Kemudian penulis membuat konteks diagram *Class* dan pemodelan basis data pada penelitian ini untuk menjelaskan bagaimana desain sistemnya.

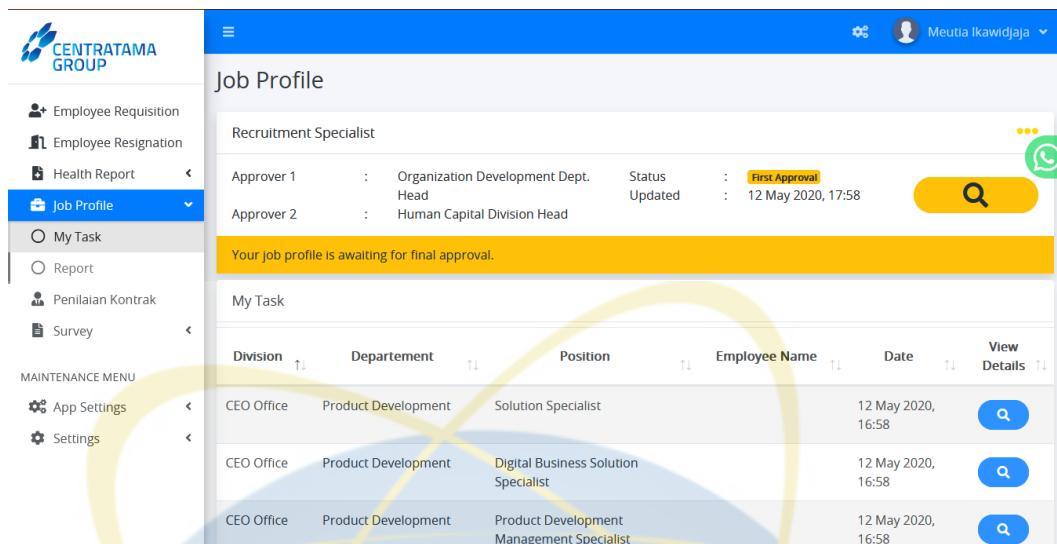
Saat penulis membuka aplikasi HC Portal, penulis disuguh dengan halaman depan aplikasi. Halaman ini memiliki tombol untuk menampilkan jendela *popup Login*. Setelah berhasil masuk, aplikasi menampilkan halaman *Job Profile* yang memuat detail posisi karyawan dan status profil kerjanya. *Job Profile* merupakan salah satu proses kerja divisi HC yang telah ditambahkan menjadi satu modul aplikasi di HC Portal. Di sebelah kiri tampilan aplikasi, terdapat *sidebar* (bilah samping) untuk melakukan navigasi ke modul aplikasi lainnya.



**Gambar 4.3 Tampilan halaman awal HC Portal; tombol *Login* di pojok kanan atas akan memunculkan jendela *popup Login* ketika diklik.**



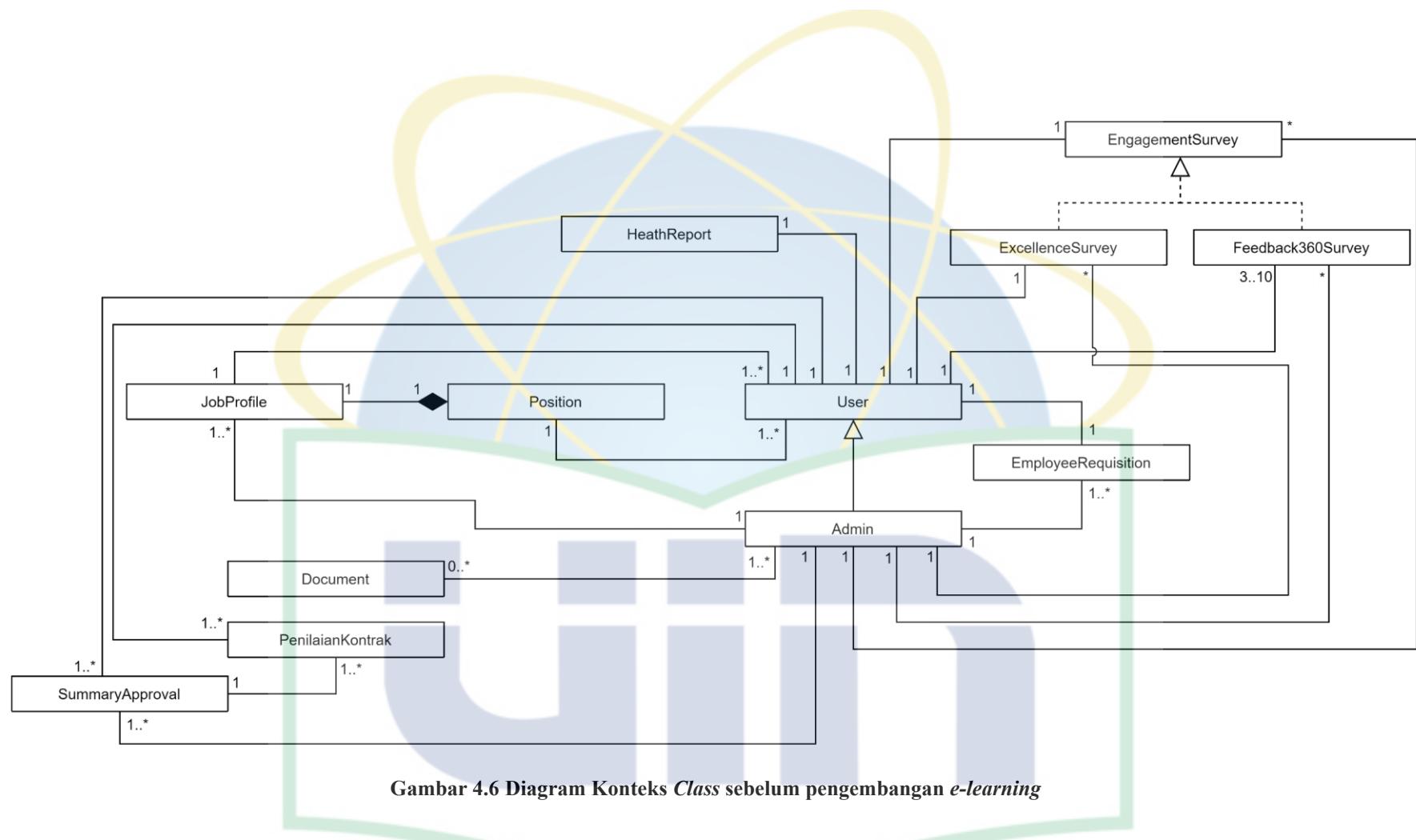
**Gambar 4.4 Tampilan jendela *popup Login* aplikasi HC Portal.**



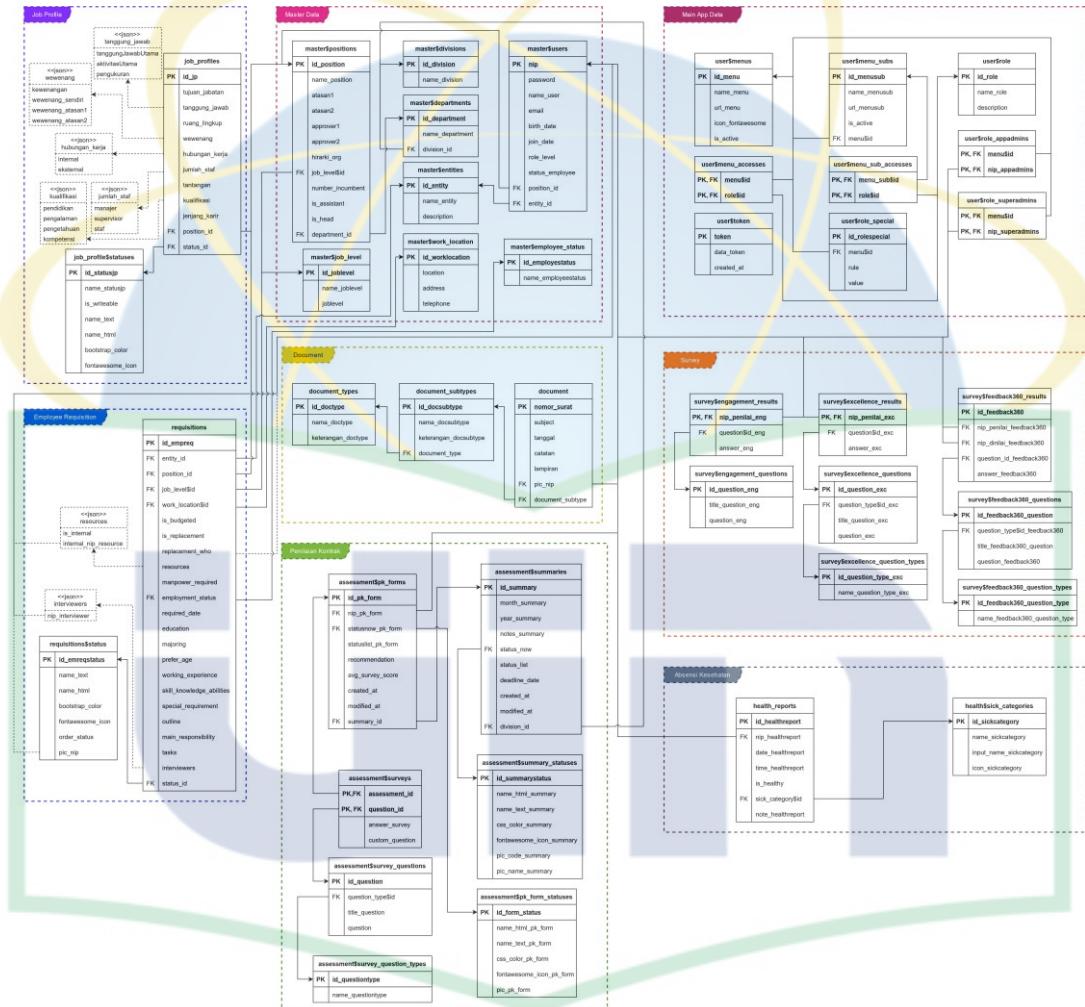
**Gambar 4.5 Tampilan modul aplikasi *Job Profile* aplikasi HC Portal; sebelah kiri gambar merupakan sidebar untuk navigasi ke modul aplikasi lainnya.**

Setelah melihat sekilas tampilan aplikasi HC Portal, penulis kemudian membuat diagram konteks *Class* berdasarkan *Class Diagram* yang penulis buat saat melaksanakan program magang di CTI. Diagram ini menggambarkan objek yang terdapat pada aplikasi HC Portal berjumlah 12 objek. Di antaranya terdapat beberapa objek yang merupakan modul aplikasi HC Portal. Satu modul aplikasi bisa memiliki satu objek atau lebih. Beberapa modul aplikasi yang memiliki satu objek di antaranya adalah *Job Profile*, *Employee Requisition*, *Document*, dan *Health Report*. Satu objek tersebut menggambarkan satu modul aplikasi. Selain itu, ada juga modul aplikasi yang terdiri dari dua sampai 3 objek, seperti modul penilaian kontrak karyawan yang memiliki objek *PenilaianKontrak* dan *SummaryApproval*, kemudian modul Survei yang memiliki objek *EngagementSurvey*, *ExcellenceSurvey*, dan *Feedback360Survey*. Terakhir untuk tiga objek lainnya merupakan objek untuk pengguna aplikasi seperti *User*, *Admin*, dan *Position* (untuk menjelaskan posisi kerjanya).

Pemodelan basis data dibuat dengan mengelompokkan tabel-tabelnya dalam beberapa modul aplikasi. Dari enam modul aplikasi yang dikelompokkan, dua di antaranya merupakan pengelompokan untuk Master Data yang berisi data pengguna dan detail posisi kerjanya, serta Main App Data untuk keperluan aplikasi.



Gambar 4.6 Diagram Konteks Class sebelum pengembangan e-learning



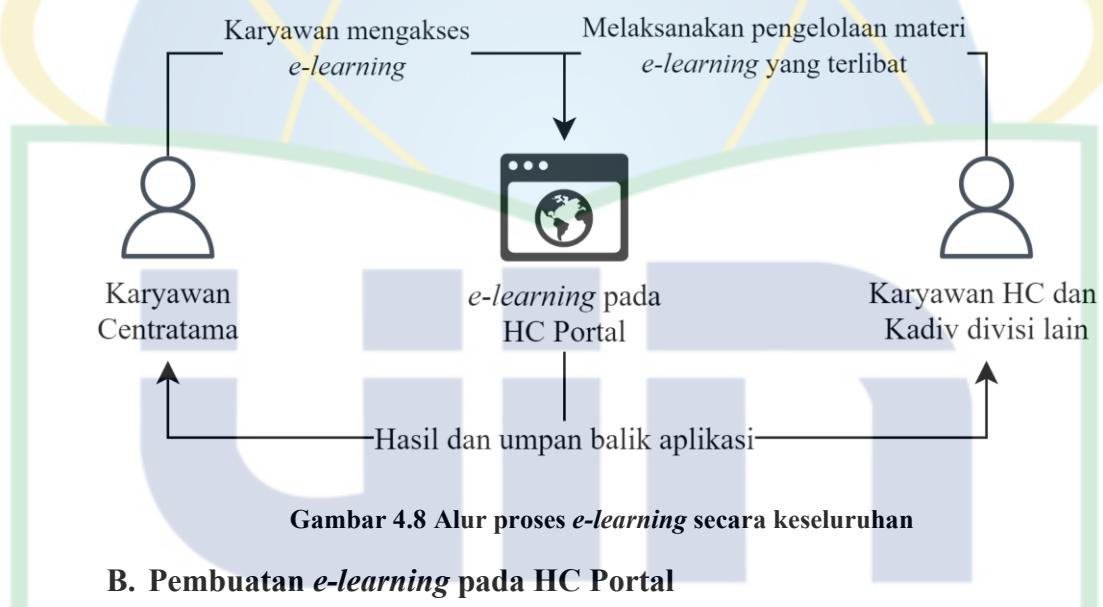
Gambar 4.7 Pemodelan Basis Data sebelum pengembangan *e-learning*

#### 4.1.4 Analisis Aplikasi Usulan

Penulis menjawab kebutuhan CTI untuk pembuatan *e-learning* pada HC Portal dengan melakukan penelitian ini. Penulis juga membuat lingkungan pengembangan HC Portal agar pengembangannya bisa terus dilakukan oleh developer baru di masa depan. Berikut penjabarannya dalam tiga poin pembahasan sesuai dengan poin permasalahan yang telah dijelaskan sebelumnya.

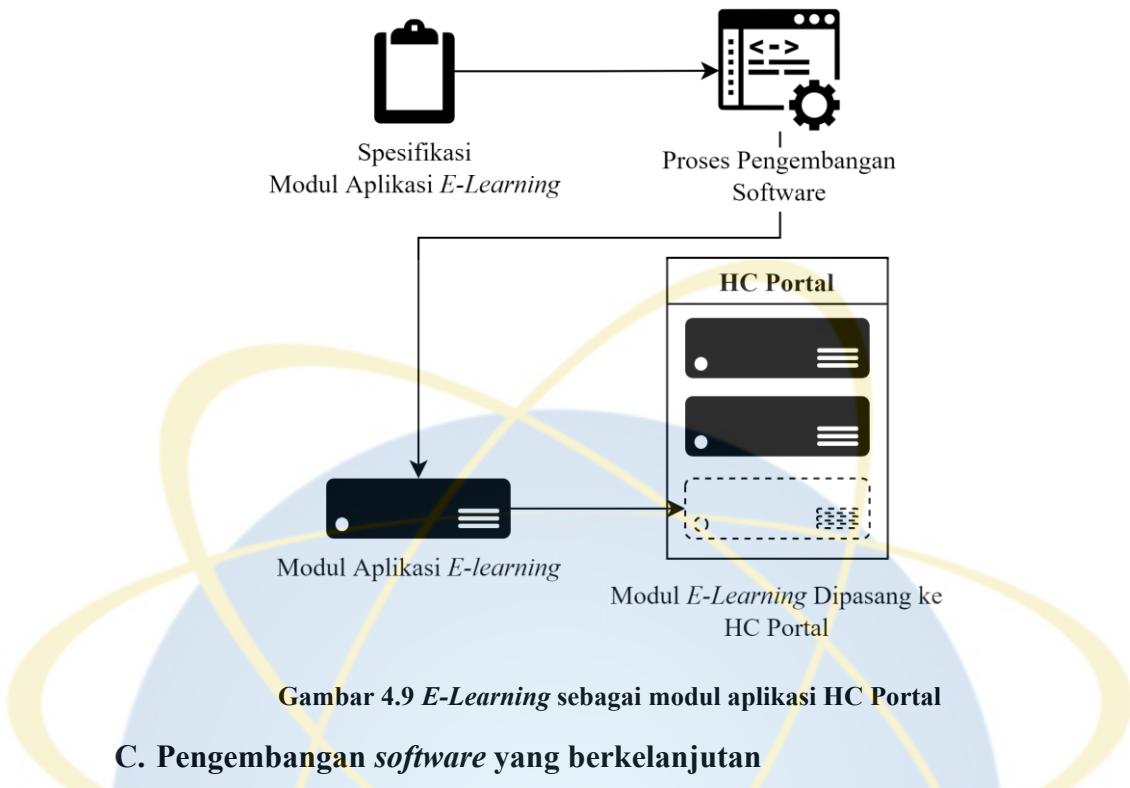
##### A. Cara baru untuk melaksanakan *training* dengan *e-learning*

Penulis perlu membuat aplikasi *e-learning* sesuai dengan kebutuhan fitur yang telah disebutkan sebelumnya. Penulis harus fokus pada pengembangan fitur-fitur tersebut agar HC dapat benar-benar memanfaatkan hasil pengembangan *software* pada penelitian ini. Pembahasan terkait alur *e-learning* dibahas pada fase iterasi. Berikut gambar alur proses *e-learning* secara keseluruhan:



##### B. Pembuatan *e-learning* pada HC Portal

Penelitian-penelitian sejenis yang telah penulis temukan melakukan pengembangan *e-learning* menjadi satu aplikasi yang berdiri sendiri. Pembahasannya telah penulis bahas pada [bab sebelumnya](#). Pengembangan *e-learning* pada penelitian ini akan dilakukan pada aplikasi web yang sudah ada, yakni HC Portal. *E-learning* akan menjadi modul aplikasi HC Portal. Berikut proses pengembangan *e-learning* secara garis besar pada penelitian ini.



Gambar 4.9 *E-Learning sebagai modul aplikasi HC Portal*

### C. Pengembangan *software* yang berkelanjutan

Penulis pada permasalahan poin ini perlu menggunakan platform pengembangan *software*, yakni Github. Ini berfungsi untuk menyimpan sumber kode pengembangan aplikasi HC Portal ke dalam platform tersebut. Konsep pengembangan dengan memanfaatkan sistem Git yang juga tersedia pada Github dapat membantu pengembangan *software* yang berkelanjutan. Developer baru dapat mempelajari bagaimana pengembangan sebelumnya melalui catatan:

- *Commit* memuat catatan perubahan yang terjadi pada kode sumber aplikasi HC Portal,
- *Issues* yang memuat pernyataan akan kebutuhan untuk pengembangan fitur aplikasi.
- *Pull-Request* yang memuat fitur yang telah dikembangkan.

Dengan begitu, developer baru memiliki lebih banyak referensi untuk melanjutkan pengembangan aplikasi HC Portal, dan memudahkan untuk melakukan pengembangan selanjutnya. Mereka dapat mempelajari proses pengembangan *software* secara langsung dalam repositori aplikasi HC Portal. Mereka dapat mempelajari apa kebutuhan yang perlu dipenuhi melalui fitur *Issues*,

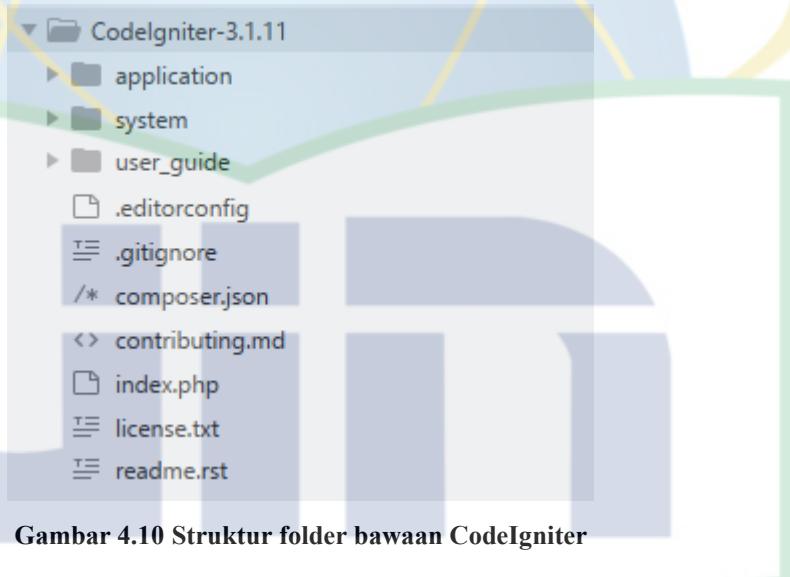
mempelajari bagaimana pengembangannya melalui catatan *Commit*, dan melihat hasil fitur yang dihasilkan untuk menjawab *Issue* pada *Pull-Request*.

#### 4.1.5 Analisis Struktur Aplikasi

Penulis membuat struktur aplikasi agar aplikasi usulan yang dikembangkan bisa menjawab ketiga permasalahan yang telah dibahas pada subbab 4.1.2. Penulis perlu membuat beberapa struktur ini karena aplikasi ini akan menjalankan beberapa modul aplikasi berbeda di dalamnya. Hal ini akan sangat menyulitkan apabila tidak membuat cetak biru struktur aplikasi terlebih dahulu. Penulis membahas struktur aplikasi yang terdiri dari struktur folder, penamaan berkas kode program, dan penamaan tabel di basis data.

##### a. Struktur Folder

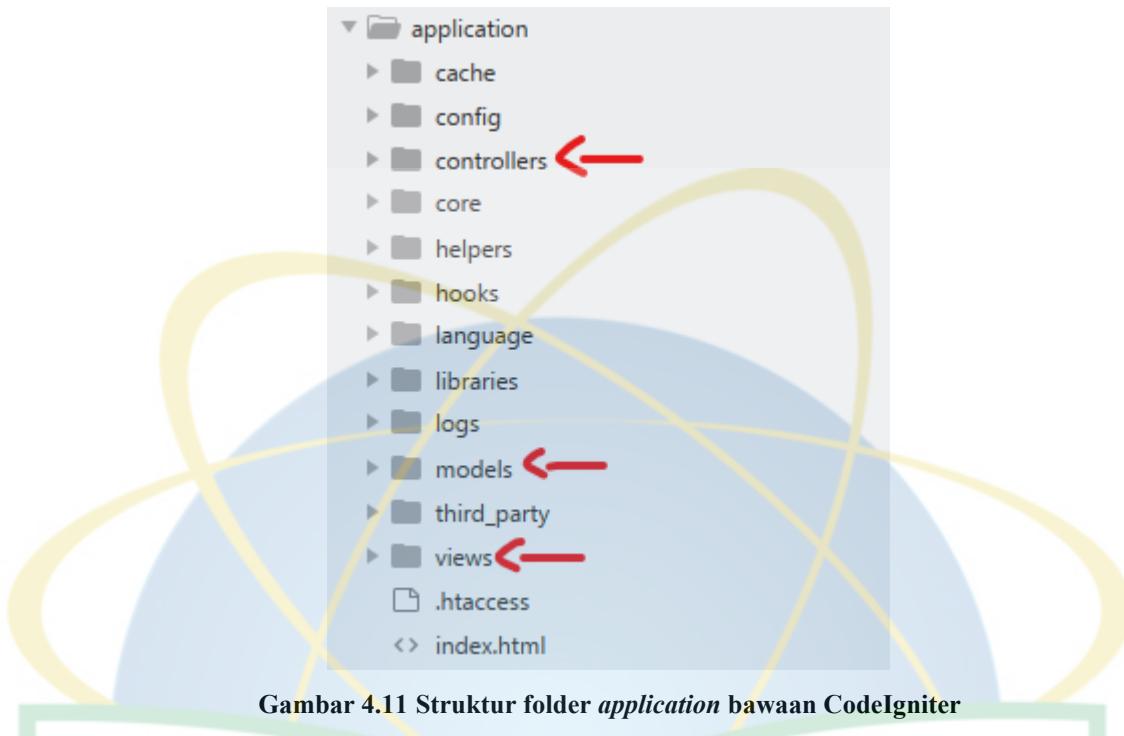
CodeIgniter memberikan struktur folder bawaan untuk memulai pengembangan aplikasi web. Berikut struktur foldernya:



Gambar 4.10 Struktur folder bawaan CodeIgniter

Terdapat satu folder yang perlu diperhatikan yaitu folder *application*. Folder inilah yang akan menjadi tempat untuk membuat kode program aplikasi. Folder *system* menyimpan kode dasar *framework* ini, untuk itu pengubahan berkas ataupun kode di folder ini sangat tidak disarankan. CodeIgniter juga memberikan semacam buku manual berupa halaman dokumentasi yang disimpan dalam folder *user\_guide*. Selain itu terdapat beberapa folder pendukung lainnya untuk mendukung pengembangan aplikasi web. Berkas *index.php* merupakan berkas penting yang menjadi pintu utama untuk mengakses aplikasi web dan berkas ini juga sangat

direkomendasikan untuk tidak diubah. Berikut struktur folder *application* bawaan CodeIgniter:



Gambar 4.11 Struktur folder *application* bawaan CodeIgniter

Folder ini memiliki banyak sub folder yang memiliki fungsinya masing-masing. Fokus saat ini adalah pada folder yang diberi tanda panah warna merah. Tanda tersebut menunjukkan di mana peletakan kode program MVC. Masing-masing lapisan MVC ini disimpan pada satu folder.

### b. Penamaan Berkas Kode Program

Setelah memahami struktur folder CodeIgniter berikutnya adalah membuat aturan penamaan berkas. Penamaan berkas *controller* mengikuti standar aturan bawaannya yaitu menggunakan penulisan “UpperCamelCase” contohnya “JobProfile.php”. Kemudian penamaan berkas model menggunakan huruf besar di awal serta menambahkan akhiran “\_m” atau “\_model”, contohnya “jobprofile\_m.php”. Hal ini bertujuan untuk memudahkan proses pengembangan dengan membedakan penamaannya dengan *controller*. Terakhir penamaan berkas *view* menggunakan huruf kecil di awal serta menambahkan akhiran “\_v” atau “\_view” apabila diperlukan, contohnya “jobprofile\_v.php”.

### c. Penamaan Tabel di Basis Data

Aturan penamaannya yaitu dengan menambahkan prefiks nama modul lalu dilanjutkan dengan penggunaan tanda garis bawah/*underscore* “\_” dan nama tabel.

Contohnya yaitu “master\_position” berarti tabel menyimpan data modul aplikasi utama yang menyimpan data posisi, dalam hal ini data posisi karyawan. Aturan ini berguna apabila satu modul aplikasi memiliki beberapa tabel. Hal ini membuat proses pengembangan modul aplikasi menjadi lebih mudah dengan hanya fokus pada awal nama tabel saja yang menggambarkan bahwa tabel tersebut dimiliki oleh satu modul aplikasi.

#### 4.1.6 Identifikasi Server Produksi

Penulis melakukan identifikasi server produksi sebagai tempat aplikasi dijalankan dan bisa diakses oleh seluruh karyawan CTI. Penulis mendapatkan spesifikasi server produksi sebagai berikut:

**Tabel 4.1 Spesifikasi Server Produksi**

<b>Sistem Operasi</b>	Windows Server 2012 R2 Standard
<b>Tipe Sistem</b>	Sistem Operasi 64-bit, prosesor x64
<b>Prosesor</b>	Intel Xeon CPU E5-2620 v4 @2.10GHz (16 CPUs)
<b>Ram</b>	16GB
<b>Media Penyimpanan</b>	<i>Hard disk</i>
<b>Ukuran Penyimpanan</b>	1 TB
<b>Teknologi Keluaran</b>	Microsoft RemoteFX; untuk berinteraksi dengan server melalui tetikus, <i>keyboard</i> , dan layar monitor server melalui koneksi <i>remote</i> .

Spesifikasi tersebut memberikan petunjuk bahwa penulis dapat memanfaatkan *Internet Information Services* (IIS) sebagai web server untuk menjalankan aplikasi HC. IIS merupakan web server bawaan sistem operasi Windows yang dibuat oleh Microsoft. IIS mendukung beberapa protokol internet untuk mengakses sumber daya aplikasi web pada sebuah server, seperti HTTP, HTTP/2, HTTPS, FTP, FTPS, SMTP dan NNTP. Versi IIS yang telah terpasang di server produksi ini yaitu versi 8.5. IIS telah menjadi bagian yang tidak terpisahkan dari keluarga sistem operasi Windows NT sejak versi Windows NT 4.0. Informasi selengkapnya bisa dilihat pada laman web IIS langsung di <https://www.iis.net/>.

#### 4.1.7 Persiapan Awal Pengembangan *Software*

Persiapan awal dilakukan dengan mengidentifikasi berbagai kebutuhan untuk pengembangan aplikasi HC Portal. Kebutuhan aplikasi ini berupa komponen yang diperlukan untuk membangun aplikasi, fungsionalitas dasar yang diperlukan yang nantinya dapat memenuhi pengembangan iterasi ke depannya, serta perangkat keras dan perangkat lunak apa saja yang dibutuhkan untuk mengembangkan aplikasi. Berikut uraian lengkapnya:

##### a. Identifikasi Komponen Aplikasi

Sebuah aplikasi terdiri banyak komponen-komponen di dalamnya. Komponen-komponen ini berupa teknologi yang digunakan sebagai penyusun utama aplikasi. Tabel spesifikasi berikut menguraikan komponen-komponen yang diperlukan.

Tabel 4.2 Spesifikasi komponen aplikasi HC Portal

<b>Teknologi</b>	Aplikasi Web dengan protokol HTTP/s
<b>Bahasa Pemrograman</b>	PHP versi 7.x, JavaScript
<b>Bahasa markup; untuk user interface</b>	HTML, CSS, JavaScript
<b>Sistem Basis Data</b>	MySQL versi 5.6
<b>Pengelola Basis Data</b>	PHPMyAdmin
<b>Framework PHP</b>	CodeIgniter versi 3.1.11
<b>Framework CSS</b>	Bootstrap dengan templat AdminLTE versi 3.x

##### b. Identifikasi Kebutuhan Fungsional Dasar Aplikasi

Fungsional dasar merupakan kerangka fungsi dasar yang menyusun setiap modul aplikasi HC Portal. Berikut beberapa kebutuhan fungsional dasar yang diperlukan:

- Aplikasi perlu memiliki pemisahan kepentingan berdasarkan hierarki dari karyawan CTI, seperti Karyawan HC, Karyawan selain HC, dan Kadiv.

- Aplikasi memiliki tampilan yang rapi apabila dibuka di berbagai peramban web dan perangkat lainnya seperti peramban web di perangkat *desktop* dan *mobile*. Hal ini bisa dilakukan dengan menerapkan *Responsive Web Design* dengan memanfaatkan fitur Bootstrap pada templat AdminLTE.

### c. Identifikasi Kebutuhan Perangkat Keras

Pengembangan aplikasi pastinya memerlukan perangkat komputer atau laptop sebagai alatnya. Perangkat keras untuk lingkungan produksi aplikasi telah dijelaskan pada subbab 4.1.6 yang menjelaskan identifikasi server produksi aplikasi HC Portal, sedangkan lingkungan pengguna hanya perlu menyiapkan perangkat komputer atau laptop yang terhubung dengan koneksi internet. Lingkungan produksi ini telah ada sebelum pengembangan aplikasi ini. Penulis akan memasang aplikasi HC Portal ini pada lingkungan tersebut. Perangkat keras untuk lingkungan pengembangan tentunya juga memerlukan spesifikasi tertentu. Tabel berikut menguraikan spesifikasi perangkat keras yang penulis gunakan dalam pengembangan aplikasi.

**Tabel 4.3 Spesifikasi perangkat pengembangan aplikasi**

<b>Sistem Operasi</b>	Linux KDE Neon User Edition: 20221222, based on Ubuntu 22.04 & Plasma 5.26.4
<b>Tipe Sistem</b>	Sistem Operasi 64-bit, prosesor x64
<b>Prosesor</b>	AMD A6 6310 @1.8 – 2.1 GHz (2 CPUs/4 Threads)
<b>Ram</b>	8GB
<b>Media Penyimpanan</b>	<i>Hard disk</i>
<b>Ukuran Penyimpanan</b>	500GB
<b>Input Devices</b>	Tetikus ( <i>mouse</i> ) dan <i>keyboard</i>
<b>Output Device</b>	Layar monitor LCD dengan resolusi 1366x768
<b>Koneksi</b>	Wi-Fi Rumahan

#### **d. Identifikasi Kebutuhan Perangkat Lunak**

Identifikasi kebutuhan terakhir yaitu perangkat lunak untuk tempat pengembangan, tempat menjalankan, dan tempat mengakses aplikasi HC Portal. Penulis melakukan pengembangan aplikasi pada laptop penulis sebagai tempat pengembangan. Aplikasi HC portal dipasang pada server sebagai tempat menjalankannya. Pengguna mengakses aplikasi HC Portal pada laptop mereka sebagai tempat mengaksesnya.

Lingkungan pengembangan aplikasi memerlukan perangkat lunak penyunting teks (*text editor*). Penulis menggunakan Visual Studio Code (vscode) versi 1.47.3 untuk memenuhi kebutuhan ini. Penulis juga memasang perangkat lunak sesuai dengan identifikasi komponen aplikasi pada poin a di subbab ini. Penulis juga memasang perangkat lunak web server Apache versi 2.4.52 untuk menjalankan aplikasi web HC Portal. Penulis juga melakukan hal yang sama untuk lingkungan server produksi, hanya saja penulis menggunakan IIS sebagai perangkat lunak web servernya. Lingkungan pengguna tidak perlu memasang perangkat lunak yang disebutkan sebelumnya. Pengguna cukup menyiapkan perangkat komputer atau laptop yang mampu menjalankan aplikasi peramban web, seperti Firefox, Chrome, Opera, atau peramban web lainnya.

#### **4.1.8 Pembuatan Lingkungan Pengembangan *Software* dengan membuat repositori digital di platform Github**

Pembuatan Repositori Github adalah cara untuk membuat lingkungan pengembangan *software* di Github. Repositori yang telah dibuat tidak hanya digunakan sebagai tempat arsip kode program, melainkan juga berbagai fitur lainnya dalam hal *platform social coding*. Penulis telah membahas lebih lengkap tentang Github dan fitur-fitur lainnya di [bab sebelumnya](#). Paragraf berikutnya membahas sekilas tentang repositori aplikasi HC Portal saat ini.

Penulis sebenarnya telah menggunakan Github ketika melaksanakan program magang di CTI dengan melakukan pengembangan HC Portal. Sebelumnya, penulis melakukan *forking* repositori dari anak magang lama di tautan berikut:

[https://github.com/julians22/hc\\_system](https://github.com/julians22/hc_system)

Hasil *forking* repositori tersebut kemudian membuat penulis memiliki repositori ini dengan penanda “*forked from julians22/hc\_system*” di tautan berikut:

<https://github.com/ryumada/centratama-hcportal>

Selang beberapa minggu pengembangan aplikasi, penulis perlu membuat agar aplikasi HC Portal dapat sepenuhnya memanfaatkan fitur *responsive design* dari [Bootstrap](#). Templat CSS yang telah dibuat oleh anak magang sebelumnya belum memenuhi fitur ini. Inilah yang membuat penulis merombak ulang *User Interface* aplikasi HC Portal dengan templat CSS baru, yakni AdminLTE. Terdapat banyak perubahan yang penulis lakukan di waktu tersebut. Penulis juga masih memerlukan desain templat lama untuk melihat bagaimana cara penerapannya untuk menampilkan data dan informasi di *User Interface* aplikasi HC Portal sebelumnya. Hal ini membuat penulis untuk membuat repositori baru dengan kode pengembangan HC Portal yang baru. Penulis beri nama repositori ini sebagai “hcportal-new”. Berikut tautan untuk menuju repositori tersebut:

<https://github.com/ryumada/centratama-hcportal-new-public>

Pada penelitian ini, penulis membuat repositori baru untuk menunjukkan bagaimana memulai pengembangan *software* dengan Github. Penulis akan mencontohkan pembuatan repositori baru sampai mirip dengan kondisi repositori “hcportal-new”. Setelah itu, penulis akan memulai pengembangan *e-learning* dengan repositori “hcportal-new”.

Langkah-langkah dalam membuat repositori baru sebenarnya telah ada di laman berikut:

<https://docs.github.com/en/get-started/quickstart/create-a-repo?tool=webui>

Namun, beberapa poin berikut menjelaskan bagaimana penulis membuat repositorinya untuk penelitian ini.

#### A. Membuat Repositori di Github

1. Pertama penulis **mengunjungi halaman web** [github.com](https://github.com), lalu melakukan **login**.
2. Kemudian penulis membuat repositori baru dengan melakukan **klik tombol berlogo tambah** (+) dan **memilih menu new repository**.
3. Lalu akan muncul halaman pembuatan repositori baru. Lalu penulis mengisi formulir pembuatan repositori baru seperti pada tabel di bawah ini:

**Tabel 4.4 Isian formulir pembuatan repositori baru.**

<i>Repository Name</i>	Isi dengan nama repositori sesuai nama aplikasi yang dibuat.
<i>Description</i>	Deskripsi lebih detail yang menjelaskan repositori proyek.
Pilihan Repositori <i>Public</i> atau <i>Private</i>	<ul style="list-style-type: none"> <li>Dengan memilih <i>public</i>, maka sumber kode repositori dapat dilihat oleh publik.</li> <li>Dengan memilih <i>private</i>, maka sumber kode repositori tidak dapat dilihat oleh publik.</li> </ul>
<i>Repository Template</i>	Buat repositori dengan templat yang sudah dibuat sebelumnya. Info lebih lanjut ada di tautan berikut: <a href="https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository">https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository</a> .
<i>Initialize this repository with:</i> 1. Add a <i>README file</i> 2. Add <i>.gitignore</i> 3. Choose a license	<ol style="list-style-type: none"> <li>Ini berfungsi untuk menambahkan penjelasan panjang terkait repositori.</li> <li>Ini berfungsi untuk menambahkan berkas yang tidak ingin dipantau perubahannya oleh Git. Biasanya ini dibuat berdasarkan templat pada bahasa pemrograman tertentu. Opsi ini akan membuat berkas bernama “.gitignore” di repositori.</li> <li>Ini berfungsi untuk memberikan lisensi atas repositori yang dibuat.</li> </ol>

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

**Repository template**  
Start your repository with a template repository's contents.

No template ▾

**Owner \*** ryumada / **Repository name \*** helloda ✓

Great repository names are short and memorable. Need inspiration? How about [verbose-rotary-phone](#)?

**Description (optional)**  
A repository used for introduction.

 **Public**  
Anyone on the internet can see this repository. You choose who can commit.

 **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

**Add a README file**  
This is where you can write a long description for your project. [Learn more](#).

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: None ▾

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more](#).

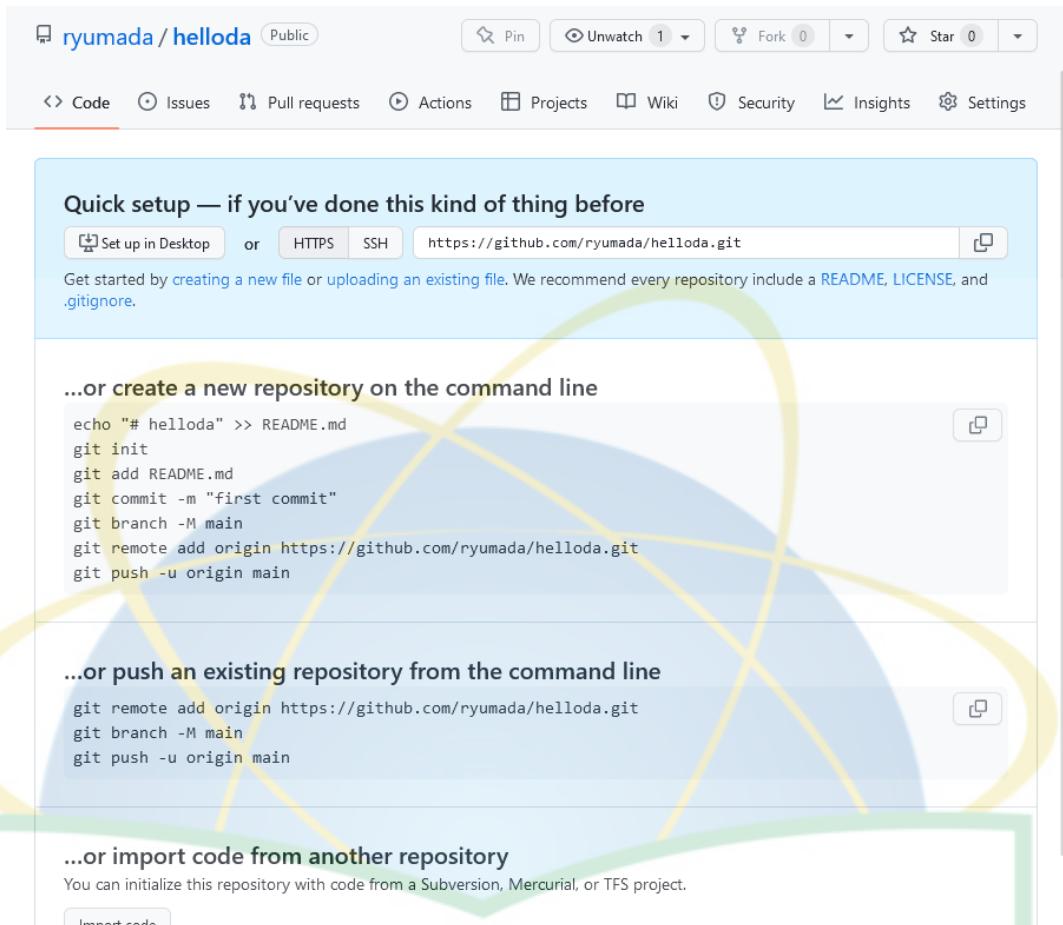
License: None ▾

ⓘ You are creating a public repository in your personal account.

**Create repository**

Gambar 4.12 Pengaturan repositori baru yang penulis buat.

4. Penulis tidak menggunakan opsi-opsi pada *initialize this repository with* dan *repository template*. Ini karena penulis akan membuat repositori Git lokal dari awal, kemudian menghubungkannya ke repositori yang ada di Github. **Klik tombol *Create repository* untuk membuat repositori.**
5. Setelah itu akan muncul halaman yang memuat *command line* untuk menambahkan repositori lokal ke repositori di Github.

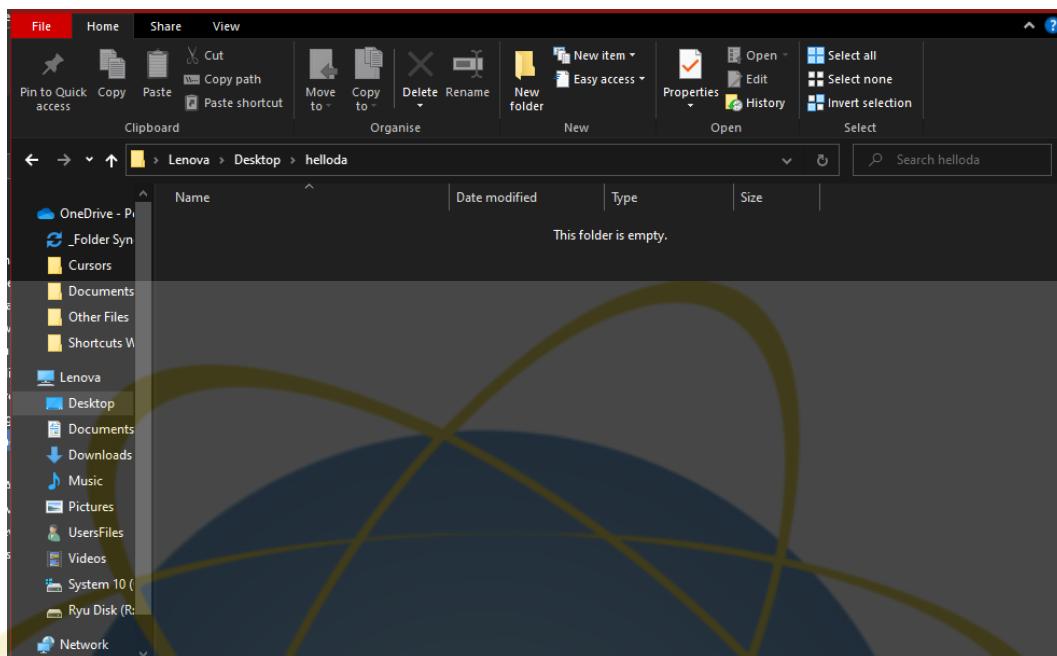


Gambar 4.13 Tampilan awal setelah membuat repositori baru tanpa memilih opsi-opsi pada *initialize this repository* dan *opsi repository template*

## B. Membuat Repositori Lokal dan Membuat Berkas Awal Repositori

**Membuat repositori lokal** dengan membuat folder baru yang kosong lalu membuka *command line* dengan perintah berikut:

```
git init
```



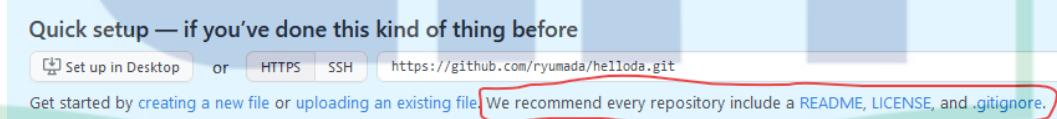
Gambar 4.14 Folder baru kosong untuk membuat repositori baru

```
PS R:\Desktop\helloda> git init
Initialized empty Git repository in R:/Desktop/helloda/.git/

```

Gambar 4.15 Command line dengan perintah `git init` yang telah dimasukkan

Setelah itu kita bisa membuat berkas teks **README.md**, **LICENSE**, dan **.gitignore** sesuai dengan yang disarankan oleh Github saat membuat repositori baru. Ketiga berkas tersebut merupakan berkas teks biasa yang bisa dibuat dengan aplikasi *text editor* seperti notepad, notepad++, vscode, atau lapce.



Gambar 4.16 Github menyarankan untuk membuat berkas teks **README.md**, **LICENSE**, dan **.gitignore**

Penulis telah membahas tentang berkas **README.md** dan menjelaskan bagaimana cara membuatnya pada [bab sebelumnya \(2.11.6\)](#). Tautan berikut merupakan berkas **README.md** yang telah penulis buat untuk penelitian ini.

<https://github.com/ryumada/centratama-hcportal-new-public#readme>

Setelah berkas **README.md** selesai dibuat, langkah selanjutnya adalah membuat berkas **LICENSE**. Sebelumnya di **README.md** telah ada pembahasan

terkait lisensi. Namun, berkas ini memuat pernyataan lisensinya secara lengkap. Pernyataan lisensi ini dapat diambil pada halaman web: <https://choosealicense.com/>. Tautan berikut merupakan berkas LICENSE yang telah penulis buat untuk penelitian ini.

<https://github.com/ryumada/centratama-hcportal-new-public/blob/main/LICENSE>

Berkas terakhir adalah **.gitignore** yang berisi daftar nama berkas yang tidak ingin dimasukkan ke dalam repositori Git di Github. Tautan berikut ini menjelaskan terkait berkas .gitignore ini.

<https://docs.github.com/en/get-started/getting-started-with-git/ignoring-files>

Github telah membuat berbagai templat berkas .gitignore berdasarkan bahasa pemrograman yang dipakai pada satu proyek. Tautan berikut ini merupakan repositori yang berisi berkas-berkas templat .gitignore.

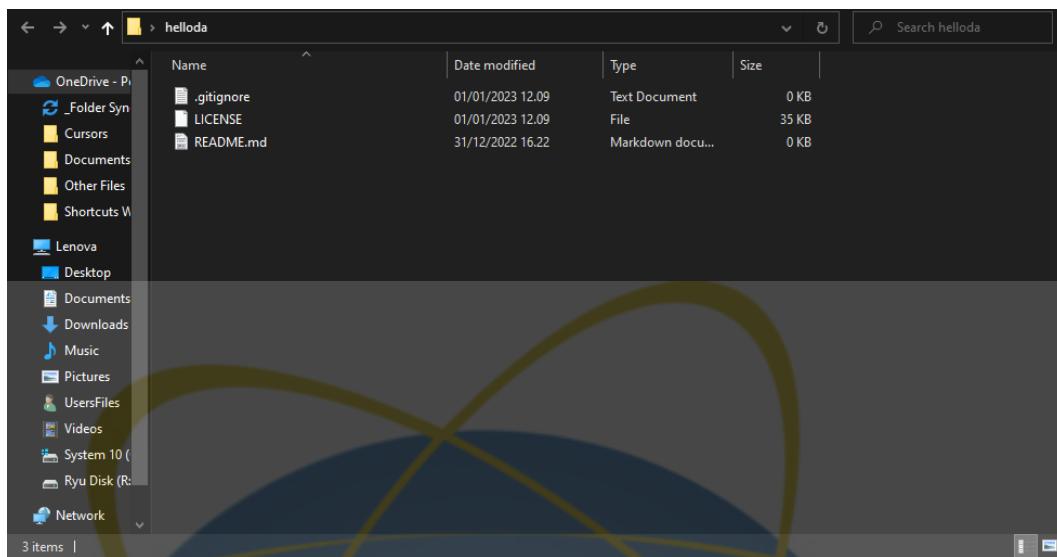
<https://github.com/github/gitignore>

Penulis menggunakan CodeIgniter sebagai *framework* PHP yang penulis gunakan untuk melakukan pengembangan aplikasi HC Portal. Untuk itu, penulis menggunakan templat CodeIgniter dari repositori templat .gitignore tersebut. Tautan berikut ini merupakan berkas .gitignore untuk pengembangan *e-learning* pada penelitian ini.

<https://github.com/ryumada/centratama-hcportal-new-public/blob/main/.gitignore>

### C. Menambahkan berkas-berkas CodeIgniter ke dalam repositori lokal

Pembahasan terkait struktur folder CodeIgniter telah dibahas pada poin 4.1.5 saat membahas tentang analisis struktur aplikasi. Di langkah ini, penulis menambahkan berkas-berkas tersebut ke dalam repositori lokal.



Gambar 4.17 Tampilan folder repositori lokal sebelum ditambahkan berkas *framework* CodeIgniter

Name	Type	Compressed size	Password ...	Size
application	File folder			
system	File folder			
user_guide	File folder			
.editorconfig	Editor Config Source File	1 KB	No	1 KB
.gitignore	Text Document	1 KB	No	1 KB
composer.json	JSON Source File	1 KB	No	1 KB
contributing.md	Markdown Source File	3 KB	No	7 KB
index.php	PHP Source File	4 KB	No	11 KB
LICENSE	File	1 KB	No	2 KB
license.txt	Text Document	1 KB	No	2 KB
README.md	Markdown Source File ...	5 KB	No	12 KB
readme.rst	Restructured Text Source ...	1 KB	No	3 KB

Gambar 4.18 Tampilan folder repositori lokal setelah ditambahkan berkas *framework* CodeIgniter

#### D. Melakukan *Initial Commit*

Setelah menambahkan berkas *framework* CodeIgniter ke dalam repositori lokal, kemudian kita akan melakukan *commit* awal untuk menyimpan kondisi awal repositori. Sebelumnya, kita harus menambahkan alamat repositori daring (*remote repository*) ke dalam repositori lokal kita, baru setelah itu kita dapat melakukan *push* repositori lokal ke *remote* yang ada di Github. Gambar berikut merupakan rangkaian perintah Git yang dapat kita jalankan untuk melakukan *initial commit*:

...or push an existing repository from the command line

```
git remote add origin https://github.com/ryumada/helloda.git  
git branch -M main  
git push -u origin main
```



**Gambar 4.19 Rangkaian perintah Git untuk menambahkan repositori lokal ke dalam repositori *remote***

### E. Membuat templat *Issue* dan PR

Kedua templat tersebut digunakan untuk membuat standardisasi informasi yang diperlukan saat membuat *issue* dan PR di Github. Hal ini menjaga kualitas informasi pada dokumentasi perubahan yang terjadi pada pengembangan *software*. Informasi yang berkualitas dari *issue* dan PR yang dibuat oleh developer dapat membantu developer baru untuk mempelajari pengembangan *software* sebelumnya dan melanjutkan pengembangannya.

McMinn menjelaskan bagaimana membuat PR yang baik yaitu dengan membuat semua komunikasi yang terjadi memiliki tujuan dan berusaha untuk meraihnya. Mereka menjelaskan bagaimana menulis PR, serta memberikan *feedback* dan meresponsnya. Kedua hal ini diterapkan pada lingkungan internal Github. (McMinn, 2015)

Tautan berikut menjelaskan bagaimana cara membuat templat PR:

<https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/creating-a-pull-request-template-for-your-repository>

Sedangkan untuk templat PR yang penulis buat ada di tautan berikut ini:

[https://github.com/ryumada/centratama-hcportal-new-public/blob/main/.github/pull\\_request\\_template.md](https://github.com/ryumada/centratama-hcportal-new-public/blob/main/.github/pull_request_template.md)

*Issue* juga harus dibuat dan dapat memuat informasi yang ingin disampaikan. Templat *issue* membantu developer untuk menjelaskan permasalahan atau ide dan agar tersampaikan dengan baik. Tautan berikut menjelaskan bagaimana cara membuat templat *issue* di Github:

<https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/configuring-issue-templates-for-your-repository#creating-issue-forms>

Sedangkan dua tautan berikut ini adalah templat *issue* yang penulis buat:

- [https://github.com/ryumada/centratama-hcportal-new-public/blob/main/.github/ISSUE\\_TEMPLATE/bug\\_report.md](https://github.com/ryumada/centratama-hcportal-new-public/blob/main/.github/ISSUE_TEMPLATE/bug_report.md)
- [https://github.com/ryumada/centratama-hcportal-new-public/blob/main/.github/ISSUE\\_TEMPLATE/user-story.md](https://github.com/ryumada/centratama-hcportal-new-public/blob/main/.github/ISSUE_TEMPLATE/user-story.md)

Fitur *issue* Github bisa kita gunakan untuk mencatat kebutuhan pengembangan *e-learning* dan dapat kita masukkan langsung ke dalam alur kerja Git (*Git Workflow*). Penulis selanjutnya mencatat kebutuhan tersebut dalam fitur *issues* ini.

## 4.2 Iteration (Iterasi)

### 4.2.1 Analysis & Design (Analisis & Desain)

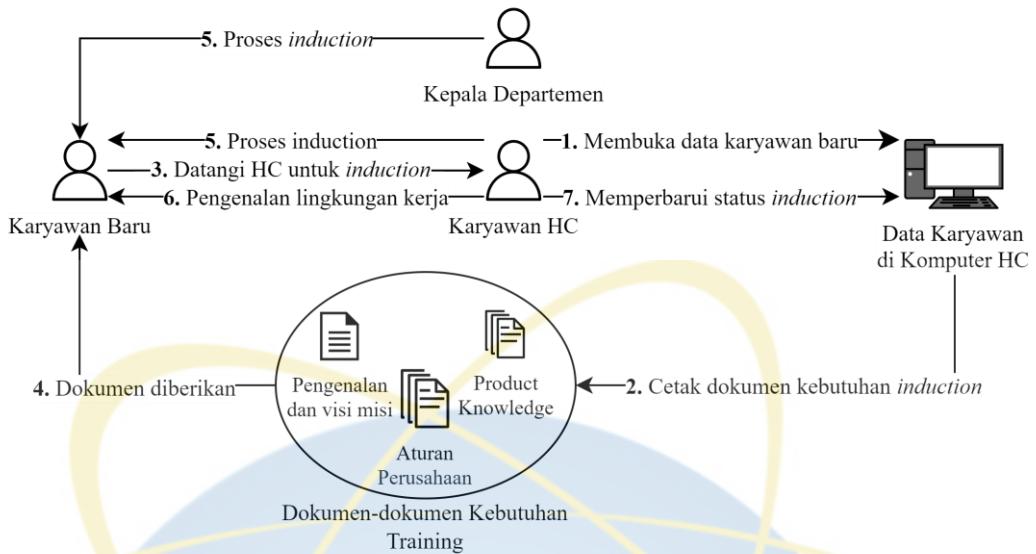
Tahapan ini merupakan tahapan untuk membuat desain aplikasi. Penulis melakukan analisis sistem berjalan dan membuat usulannya apabila menggunakan aplikasi. Penulis kemudian membuat berbagai cetak biru aplikasi dengan UML dan membuat cetak biru lainnya seperti membuat struktur aplikasi, struktur menu, dan *user interface* (UI).

#### A. Analisis sistem berjalan

Pembahasan analisis sistem berjalan diuraikan berdasarkan setiap proses yang terjadi pada *Training* karyawan. Hal ini dimulai dengan membahas *training* awal untuk karyawan baru (*induction*), pengajuan *training* karyawan lama secara luring dan daring, sampai kepada proses diskusi *training* yang terjadi.

##### 1. Training Karyawan Baru (*induction*)

Karyawan yang baru menempati posisi kerja di hari pertamanya akan diberikan semacam pelantikan sebelum ia terjun langsung di lingkungan kerjanya. Hal ini juga merupakan sebuah ajang untuk karyawan baru mengenal proses bisnis perusahaan secara keseluruhan dan meluruskan pemahaman akan tujuan perusahaan dalam membangun bisnis. Tentu saja perusahaan akan mendukung karyawannya untuk meningkatkan kemampuan dirinya. Namun untuk mencapainya, karyawan harus bisa beradaptasi dan menyatukan tujuan dengan lingkungan baru ia bekerja. Berikut ini adalah alur proses *induction* saat ini di CTI:



**Gambar 4.20 Rich Picture proses berjalan *training* karyawan baru**

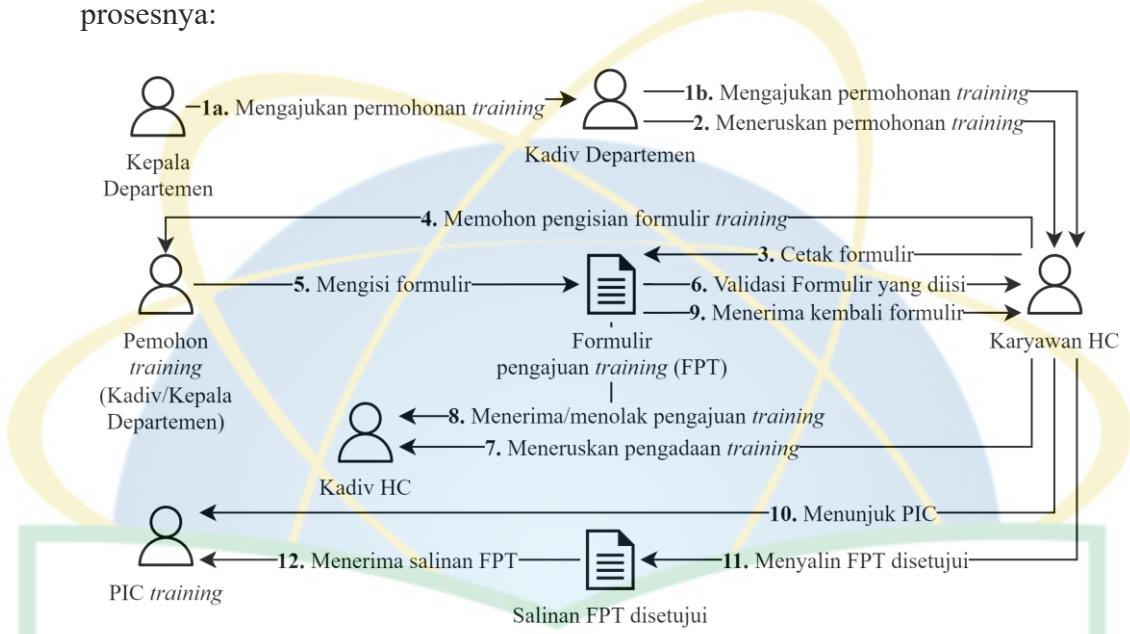
Berikut keterangan alur proses *training* karyawan baru di atas:

1. Karyawan HC membuka data karyawan baru di komputernya untuk melihat status *training* awal (*induction*).
2. Karyawan HC mencetak dokumen kebutuhan *training* awal untuk diberikan.
3. Karyawan baru pada hari pertama bekerja mendatangi HC untuk menerima *training* awal berupa pengenalan visi misi perusahaan *product knowledge* (produk) dan aturan yang berlaku.
4. Karyawan HC memberikan dokumen kebutuhan *training* kepada karyawan baru.
5. Karyawan HC melakukan presentasi terkait pengenalan perusahaan dan berdiskusi dengan karyawan baru serta pengenalan lainnya dari kepala departemen posisi kerjanya.
6. Karyawan HC mengenalkan karyawan baru tersebut ke lingkungan dan rekan kerjanya.
7. Karyawan HC memperbarui status *induction* pada data karyawan baru.

## 2. Pengajuan *Training* Karyawan Lama

Perusahaan mana yang tidak ingin karyawannya bisa memiliki performa yang lebih baik dalam bekerja. CTI merupakan salah satunya yang memfasilitasi hal ini dengan terus berupaya untuk memberikan dukungan agar karyawannya bisa menjadi pribadi yang lebih baik, serta mendapatkan pelatihan *skill* yang menunjang

pekerjaannya. Namun untuk prosesnya diperlukan persetujuan terlebih dahulu dari atasan departemen serta divisi HC. Divisi yang bergerak pada pengelolaan dan pengembangan SDM ini memiliki sarana agar pelaksanaan *training* karyawan bisa lebih mudah untuk dilakukan. Untuk itu, setiap proses *training* yang ingin diadakan pada CTI harus melalui persetujuan HC terlebih dahulu. Berikut ini adalah alur prosesnya:



**Gambar 4.21 Rich Picture proses berjalan pengajuan *training* karyawan lama**

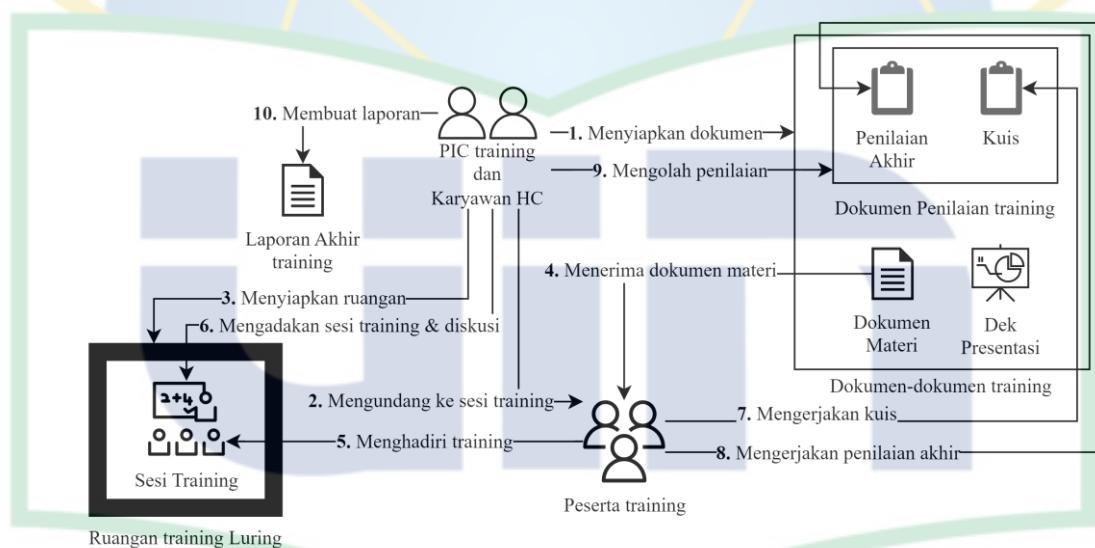
Berikut keterangan alur proses di atas:

1. Terdapat dua kemungkinan di tahapan ini:
  - a. Kepala departemen mengajukan permohonan *training* kepada Kadivnya.
  - b. Kadiv melakukan permohonan *training* kepada karyawan HC.
2. Kadiv departemen melakukan persetujuan permohonan *training* dari kepala departemen **dan** meneruskannya ke karyawan HC.
3. Karyawan HC mencetak formulir pengajuan *training*.
4. Karyawan HC memohon pengisian formulir pengajuan *training* kepada pihak pemohon *training*.
5. Pihak yang mengajukan *training* mengisi formulir pengajuan *training*.
6. Karyawan HC memvalidasi formulir yang telah diisi.
7. Karyawan HC meneruskan permohonan *training* kepada Kadiv HC untuk pelaksanaan *training*.

8. Kadiv HC menerima atau menolak formulir pengajuan *training*.
9. Karyawan HC menerima formulir *training* yang sudah disetujui.
10. Karyawan HC menunjuk PIC (*Person in Charge*/Orang yang bertanggung jawab) *training* biasanya yang ditunjuk adalah yang memohon *training* atau karyawan di departemennya yang dipilih secara langsung.
11. Karyawan HC menyalin formulir permohonan *training* yang telah disetujui.
12. PIC *training* menerima salinan formulir permohonan *training*.

### 3. ***Training Karyawan Lama secara Luring***

Setelah karyawan HC menunjuk PIC dari departemen pemohon *training*, biasanya mereka akan memutuskan untuk mengadakan *training* secara luring atau daring. Jika diadakan secara luring, mereka perlu menyiapkan ruangan *meeting* yang tidak terpakai untuk mengadakan sesi *training*. Mereka juga perlu menyiapkan beberapa dokumen untuk memfasilitasi saat proses *training* berlangsung. Alur berikut menjelaskan prosesnya:



Gambar 4.22 Rich Picture proses berjalan *training* karyawan lama secara luring

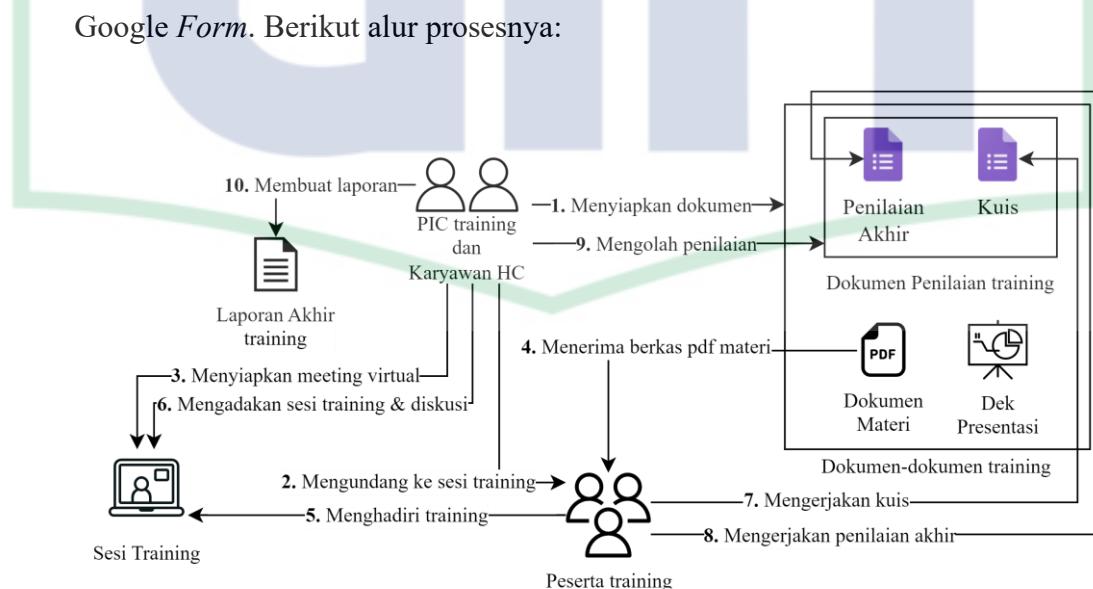
Berikut keterangan alur proses di atas:

1. PIC dan Karyawan HC menyiapkan dokumen materi *training*, dek presentasi (*slide*), serta dokumen kuis dan penilaian akhir.
2. PIC dan Karyawan HC mengundang para peserta *training* yang terdiri dari karyawan di divisi atau departemen yang bersangkutan.

3. PIC dan Karyawan HC menyiapkan ruangan *meeting* untuk mengadakan *training* luring.
4. Peserta *training* menerima dokumen materi *training* sebelum dimulai dari karyawan HC.
5. Peserta *training* menghadiri *training* yang diadakan.
6. PIC dan Karyawan HC mengadakan *training* dan diskusi untuk peserta *training* di sebuah ruangan *meeting* yang sudah disiapkan.
7. Peserta *training* mengerjakan kuis selama *training* berlangsung dan diberikan waktu sejenak untuk mengisinya.
8. Peserta *training* mengerjakan penilaian akhir untuk mengetes pemahaman mereka dalam mengikuti proses *training*.
9. PIC dan karyawan HC mengolah data penilaian *training* yang terdiri dari kuis dan penilaian akhir.
10. Karyawan HC membuat laporan akhir *training*.

#### **4. Training Karyawan Lama secara Daring**

Jika *training* yang diadakan secara daring, maka mereka perlu menyiapkan *meeting* virtual melalui platform konferensi video seperti Zoom *Meeting*. Mereka tetap perlu menyiapkan beberapa dokumen pendukung *training* walaupun tidak harus dicetak. Mereka dapat membagikan dokumen materi dalam bentuk berkas *pdf*, serta membuat kuis dan penilaian akhir pada platform pembuat formulir seperti Google *Form*. Berikut alur prosesnya:



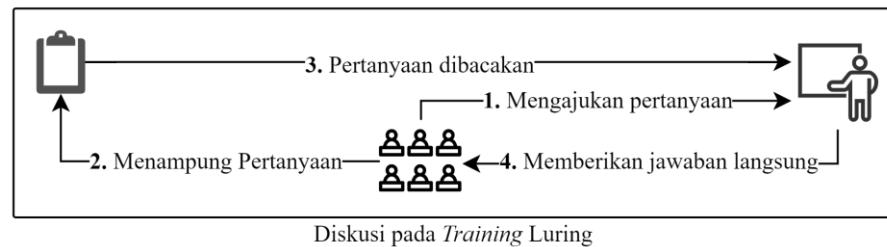
**Gambar 4.23 Rich Picture proses berjalan *training* secara daring.**

Berikut keterangan alur proses di atas:

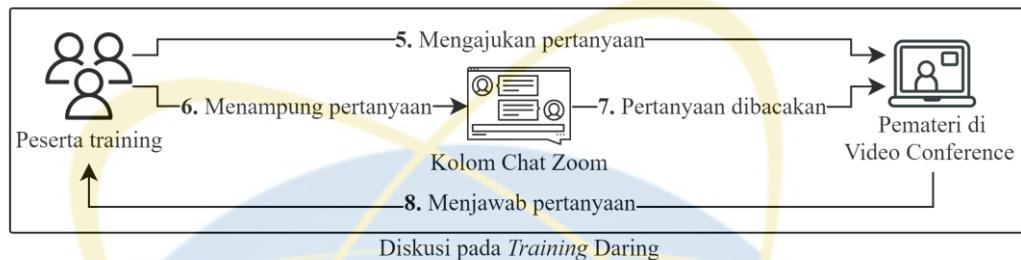
1. PIC dan karyawan HC menyiapkan berkas pdf materi *training*, dek presentasi, serta menyiapkan kuis dan penilaian akhir menggunakan platform Google *Form*.
2. PIC dan karyawan HC mengundang para peserta *training* yang terdiri dari karyawan di divisi atau departemen yang bersangkutan.
3. PIC dan karyawan HC menyiapkan *meeting* virtual pada platform Zoom.
4. Peserta *training* menerima berkas pdf materi *training* sebelum *training* dimulai dari PIC dan karyawan HC.
5. Peserta *training* menghadiri *training* pada *meeting* virtual.
6. PIC dan karyawan HC mengadakan *training* dan diskusi untuk peserta *training* di platform Zoom yang sudah disiapkan.
7. Peserta *training* mengerjakan kuis di Google *Form* selama *training* secara daring berlangsung dan diberikan waktu untuk mengerjakannya.
8. Peserta *training* mengerjakan penilaian akhir di Google *Form* untuk mengetes pemahaman mereka dalam mengikuti proses *training*.
9. Tim pengurus *training* mengolah data kuis dan penilaian akhir.
10. Karyawan HC membuat laporan akhir *training*.

## 5. Diskusi *Training*

Diskusi *Training* pada sistem berjalan dilakukan saat *training* berlangsung. Terdapat sesi khusus diskusi bersama untuk membahas satu topik tertentu, sesi tanya jawab, ataupun pertanyaan langsung di sela pemaparan materi. Cara diskusi seperti ini sangat efektif bila dilakukan secara luring, namun terkadang tidak untuk dilakukan secara daring. Kendala koneksi internet menjadi penyebab hal ini terjadi pada diskusi secara daring. Berikut ini adalah alur proses berjalan diskusi *training* yang terjadi secara luring maupun daring:



Diskusi pada *Training Luring*



Diskusi pada *Training Daring*

**Gambar 4.24 Rich Picture proses berjalan diskusi *training* yang terjadi secara luring (atas) dan daring (tengah), serta melalui platform *chat grup* (bawah)**

Berikut penjelasan alur proses diskusi secara luring, daring di atas:

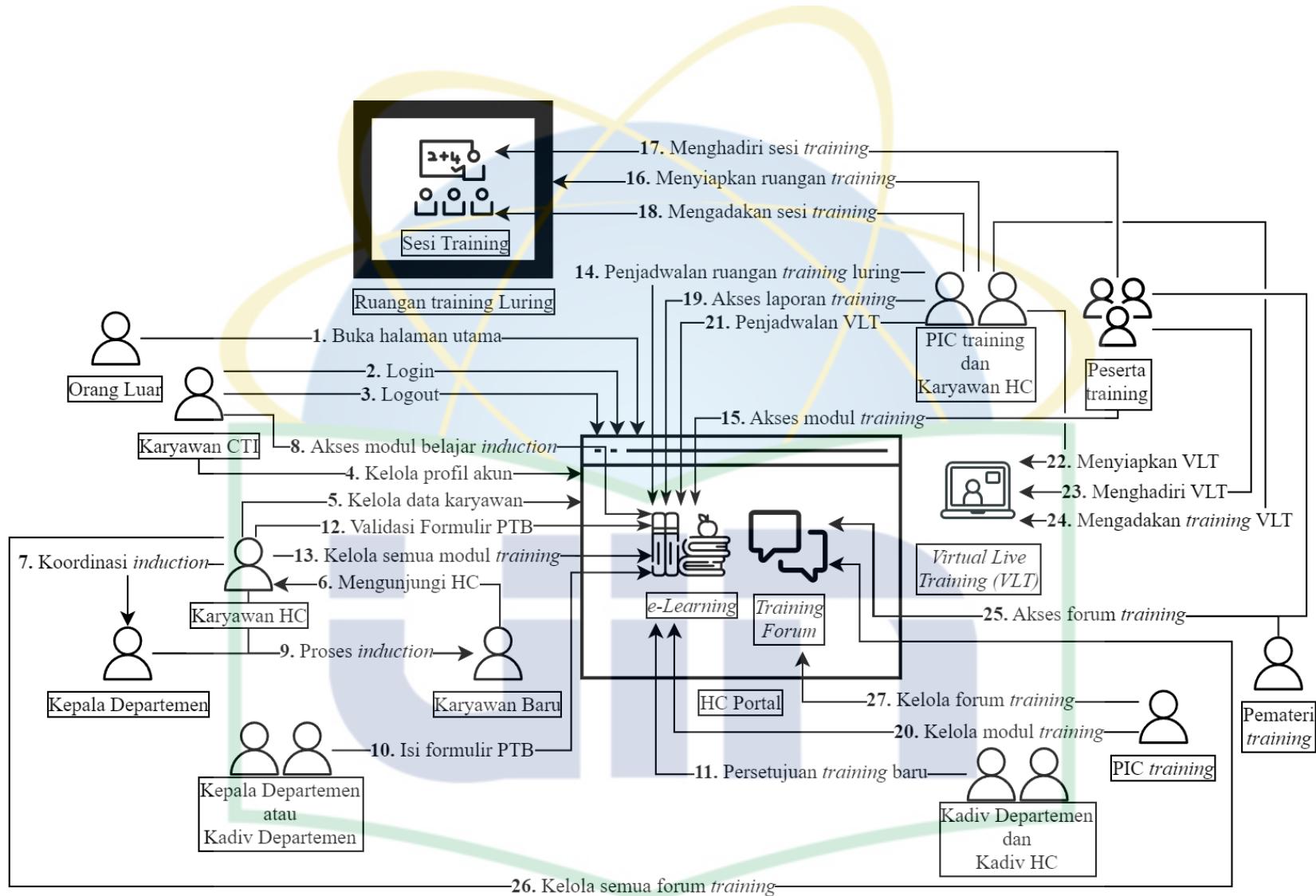
1. *Training* sedang berlangsung, peserta *training* bisa langsung mengajukan pertanyaan jika diizinkan atau diberikan kesempatan oleh pemateri.
2. Peserta *training* dapat menyiapkan pertanyaan dan menuliskannya pada platform yang disediakan.
3. Pemateri membacakan pertanyaan yang telah ditampung.
4. Pemateri bisa langsung memberikan penjelasan dari pertanyaan yang diajukan karyawan *training* atau menjawabnya saat sesi tanya jawab.
5. Peserta *training* menanyakan pertanyaannya secara langsung kepada pemateri.
6. Peserta *training* dapat menuliskan pertanyaannya pada kolom *chat* Zoom *Meeting*.
7. Pemateri dapat membacakan pertanyaan yang telah ditampung.
8. Pemateri menjawab pertanyaan-pertanyaan yang diajukan para peserta *training* secara langsung di sela pemaparan materi atau saat sesi tanya jawab berlangsung.

## B. Analisis Sistem Usulan

Sistem usulan yang penulis buat berikut ini mencakup dari kelima proses analisis sistem berjalan di atas. Proses *induction* dibuat menjadi satu modul *training* khusus yang tersedia untuk semua karyawan CTI. Mereka yang telah mengikuti

*induction* dapat terus melakukan reviu pembaruan dari visi-misi perusahaan, *product knowledge*, dan aturan perusahaan. Pengajuan *training* baru untuk karyawan lama dapat dilakukan dengan mengisi formulir Pengajuan *Training* Baru (PTB), lalu menjalankan proses persetujuannya (*approval*). Aplikasi HC Portal juga harus dapat mengolah penjadwalan *training* langsung (*live training*), baik secara luring maupun daring. Pelaksanaan *live training* berada di luar aplikasi HC Portal. Pelaksanaan *live training* secara luring dilaksanakan di ruangan *meeting* yang telah disiapkan, sedangkan *live training* secara daring dilaksanakan di platform virtual *meeting* Zoom. *E-learning* membantu *live learning* dalam mengelola materi *training*, fitur kuis, dan penilaian akhir. Selain itu, *e-learning* juga dapat menjadi media *training* secara mandiri (*self-paced learning*). Fitur forum dapat membantu pada proses diskusi *training*. Setiap karyawan yang terdaftar pada *training* tertentu akan mendapatkan akses untuk menuju forum *training*-nya. Gambar berikut merupakan alur sistem usulannya:

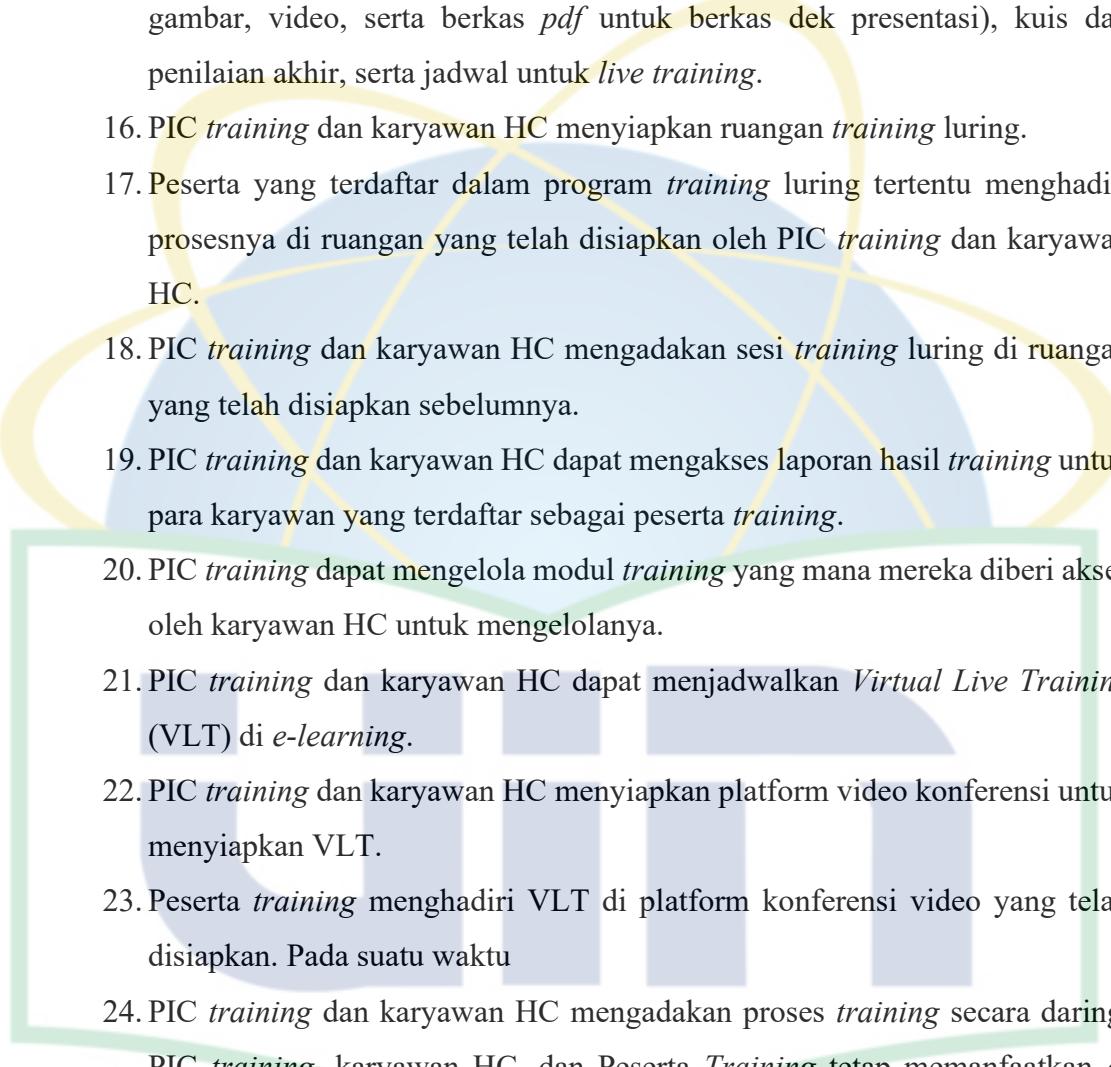




Gambar 4.25 Rich Picture analisis sistem usulan

Gambar berikut menjelaskan alur proses usulan dengan *e-learning* dan forum HC Portal:

1. Orang luar selain karyawan HC dapat mengakses halaman utama aplikasi web HC Portal.
2. Karyawan CTI dapat *login* ke HC Portal untuk mengakses modul-modul aplikasi HC Portal.
3. Karyawan CTI melakukan *logout* untuk mengakhiri sesi penggunanya.
4. Karyawan CTI dapat mengelola profil pengguna berupa pengubahan nama, surel, dan kata sandi.
5. Karyawan HC yang menjadi Admin dapat mengelola data karyawan dengan melakukan penambahan data karyawan baru, serta pengeditan dan penghapusan data karyawan yang telah terdaftar.
6. Karyawan baru yang baru bergabung dengan perusahaan mengunjungi HC untuk melaksanakan proses *induction*.
7. Karyawan HC dan Kepala Departemen melakukan koordinasi untuk melakukan proses *induction* kepada karyawan baru.
8. Semua karyawan CTI baik yang baru maupun yang sudah lama bergabung dapat mengakses modul belajar *induction*.
9. Karyawan HC dan Kepala Departemen melakukan proses *induction* berupa perkenalan awal seperti visi-misi perusahaan, *product knowledge*, dan aturan perusahaan. Lalu, karyawan HC mengajak karyawan baru berkeliling perusahaan untuk perkenalan lingkungan hingga pada akhirnya karyawan HC mengenalkan departemen posisi kerjanya.
10. Pada proses pengajuan *training* baru untuk karyawan lama, Kepala Departemen atau Kadiv Departemen dapat mengisi formulir Pengajuan *Training* Baru (PTB).
11. Apabila Kepala Departemen yang mengajukan proses *training* baru, maka Kadiv Departemen harus menyetujui formulir PTB tersebut. Kadiv HC juga akan melakukan persetujuan akhir karyawan *training*.
12. Sebelum formulir PTB sampai ke proses persetujuan oleh Kadiv HC, karyawan HC akan memvalidasi formulir PTB terlebih dahulu.

- 
13. Karyawan HC yang bertindak sebagai admin dapat mengelola semua modul *training* karyawan.
  14. PIC *training* dan Karyawan HC dapat menjadwalkan ruangan *training* luring.
  15. Peserta yang terdaftar dalam *training* tertentu dapat mengakses modul *training*-nya. Modulnya dapat berupa modul materi (berbentuk teks dan gambar, video, serta berkas *pdf* untuk berkas dek presentasi), kuis dan penilaian akhir, serta jadwal untuk *live training*.
  16. PIC *training* dan karyawan HC menyiapkan ruangan *training* luring.
  17. Peserta yang terdaftar dalam program *training* luring tertentu menghadiri prosesnya di ruangan yang telah disiapkan oleh PIC *training* dan karyawan HC.
  18. PIC *training* dan karyawan HC mengadakan sesi *training* luring di ruangan yang telah disiapkan sebelumnya.
  19. PIC *training* dan karyawan HC dapat mengakses laporan hasil *training* untuk para karyawan yang terdaftar sebagai peserta *training*.
  20. PIC *training* dapat mengelola modul *training* yang mana mereka diberi akses oleh karyawan HC untuk mengelolanya.
  21. PIC *training* dan karyawan HC dapat menjadwalkan *Virtual Live Training* (VLT) di *e-learning*.
  22. PIC *training* dan karyawan HC menyiapkan platform video konferensi untuk menyiapkan VLT.
  23. Peserta *training* menghadiri VLT di platform konferensi video yang telah disiapkan. Pada suatu waktu
  24. PIC *training* dan karyawan HC mengadakan proses *training* secara daring. PIC *training*, karyawan HC, dan Peserta *Training* tetap memanfaatkan *e-learning* HC Portal untuk mengakses berbagai media pendukung *training* seperti berkas materi, kuis, penilaian akhir, dan dek presentasi.
  25. Peserta *training* dan pemateri *training* dapat mengakses forum *training* yang mana mereka telah diberikan akses sebelumnya oleh karyawan HC.
  26. Karyawan HC dapat mengelola semua forum yang ada di aplikasi forum *e-learning*.

27. PIC *training* dapat mengelola forum yang mana mereka telah diberikan akses sebelumnya oleh karyawan HC.

Setelah memahami sistem usulan di atas, penulis memutuskan untuk membagi pengembangan *software* menjadi dua iterasi pengembangan.

### C. Desain Proses, Logis, dan Basis Data

Pada bagian ini, penulis menguraikan desain proses, logis, dan basis data. Desain proses dilakukan dengan menggunakan diagram *Use Case*, *Activity*, dan *Sequence*. Sementara itu, desain logis dilakukan dengan menggunakan diagram *Class*. Terakhir desain basis data dibuat dengan membuat pemodelannya dari diagram *Class* dan membuat spesifikasinya. Diagram *sequence* dibuat terakhir setelah pemodelan skema basis data untuk menyesuaikan dengan nama tabel yang telah dibuat pada pemodelan basis datanya.

#### 1. *Use Case Diagram*

*Use Case diagram* menggambarkan interaksi antara aktor dengan aplikasi *e-learning* HC Portal. Penulis membuat tabel identifikasi aktor, identifikasi *Use Case*, diagram *Use Case*, dan spesifikasi *Use Case*.

**Tabel 4.5 Identifikasi Aktor pada *e-Learning* dan Forum HC Portal**

No.	Aktor	Deskripsi
1.	Karyawan CTI	Individu karyawan yang bergabung dengan CTI dan dapat mengakses aplikasi <i>e-learning</i> dan forum HC Portal.
2.	Karyawan HC	Individu karyawan yang bekerja di divisi HC CTI. Ia juga menjadi admin dari modul aplikasi <i>e-learning</i> dan forum HC Portal.
3.	Kepala Departemen	Individu karyawan CTI yang menjadi kepala atau pemimpin dari suatu departemen kerja.
4.	Kadiv Departemen (Kepala Divisi)	Individu karyawan CTI yang menjadi kepala atau pemimpin dari beberapa departemen yang membentuk suatu divisi kerja. Individu aktor ini diidentifikasi sebagai kepala divisi selain divisi HC.
5.	Kadiv HC	Individu karyawan CTI yang menjadi kepala divisi kerja HC CTI.

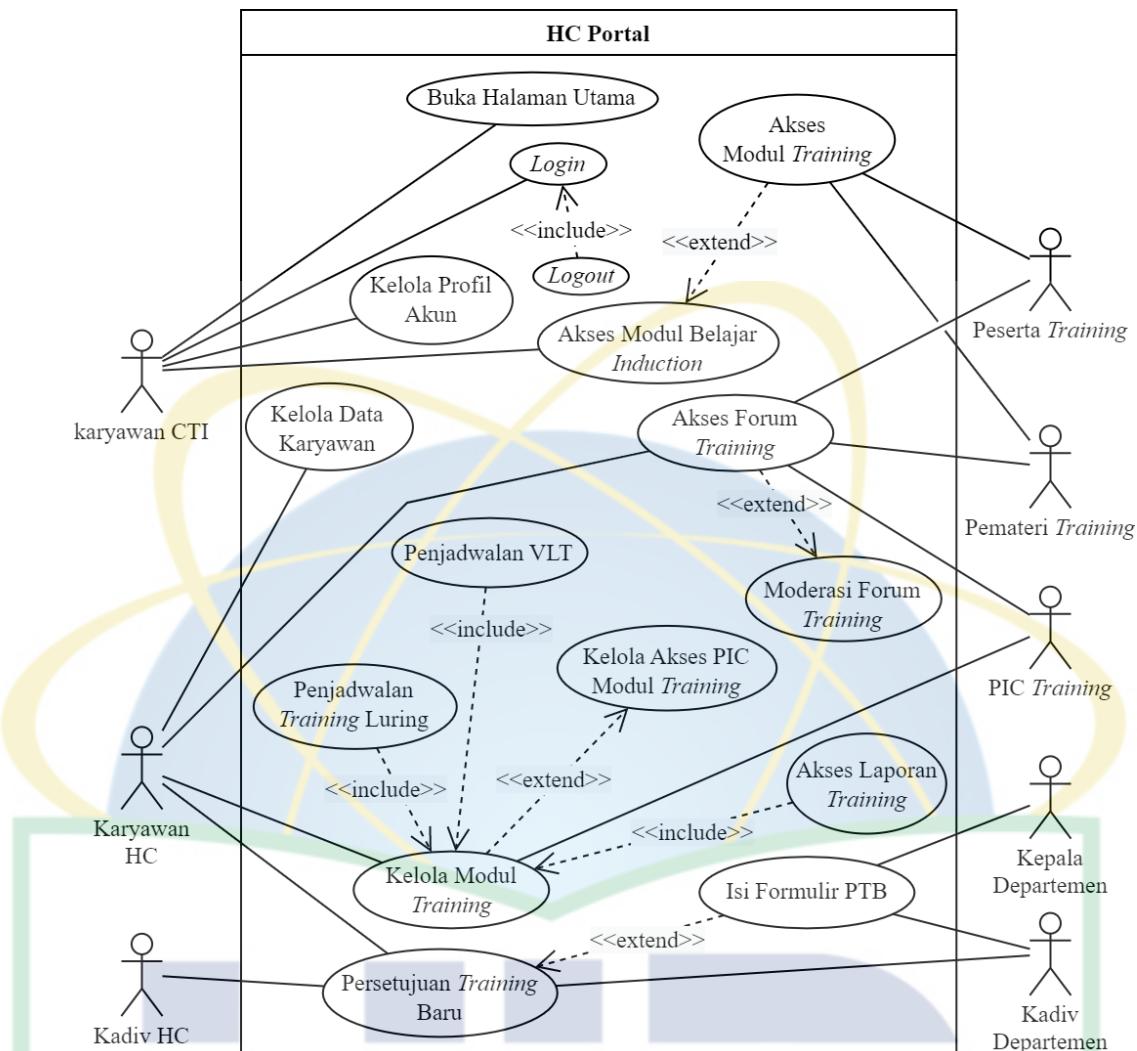
6.	<i>PIC (Person in Charge) Training</i>	Individu karyawan CTI yang bertugas sebagai penanggung jawab dari proses <i>training</i> karyawan. Biasanya karyawan yang dipilih menjadi penanggung jawab <i>training</i> adalah karyawan yang mengajukan proses <i>training</i> baru atau yang diunjuk olehnya.
7.	<i>Peserta Training</i>	Individu karyawan CTI yang terdaftar pada suatu proses <i>training</i> karyawan.
8.	<i>Pemateri Training</i>	Individu luar yang menjadi pembawa materi <i>training</i> karyawan CTI.

Tabel 4.6 Identifikasi *Use Case* pada *e-learning* dan forum HC Portal

No.	<i>Use Case</i>	Deskripsi	Aktor
1.	Buka Halaman Utama	<i>Use Case</i> ini menggambarkan kegiatan untuk melihat halaman awal aplikasi saja tanpa melakukan proses <i>login</i> .	Semua Aktor
2.	<i>Login</i>	<i>Use case</i> ini menggambarkan kegiatan pengguna untuk dapat masuk ke dalam aplikasi dan mengakses modul-modul aplikasi di dalamnya.	Seluruh Karyawan CTI
3.	<i>Logout</i>	<i>Use case</i> ini menggambarkan kegiatan pengguna untuk mengakhiri sesi pengaksesan dan keluar dari aplikasi.	Seluruh Karyawan CTI
4.	Kelola Profil Akun	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola profil pengguna seperti mengubah nama, surel, dan kata sandi.	Seluruh Karyawan CTI
5.	Kelola Data Karyawan	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola seluruh data karyawan seperti melihat data karyawan, mengeditnya, menghapusnya, dan membuat data karyawan baru.	Karyawan HC

6.	Akses Modul Belajar <i>Induction</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengakses modul belajar khusus yaitu <i>induction</i> .	Seluruh Karyawan CTI
7.	Isi Formulir Pengajuan <i>Training</i> Baru (PTB)	<i>Use case</i> ini menggambarkan kegiatan untuk mengisi formulir PTB ketika ada <i>training</i> karyawan baru yang ingin diadakan di CTI.	Kepala Departemen, Kadiv Departemen
8.	Persetujuan <i>Training</i> Baru	<i>Use case</i> ini menggambarkan proses persetujuan PTB dengan memproses formulirnya. Pada karyawan HC, ia bertugas untuk memvalidasi formulir PTB dengan memastikan semua data formulir terisi dengan semestinya sebelum diserahkan kepada Kadiv HC.	Kadiv Departemen, Karyawan HC, dan Kadiv HC
9.	Kelola Modul <i>Training</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola modul <i>training</i> yang diberi akses dengan mengedit konten modul <i>training</i> dan pemberian akses kepada peserta <i>training</i> . Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul <i>training</i> ditambah dengan penambahan dan pengarsipan modul <i>training</i> .	PIC <i>Training</i> dan Karyawan HC
10.	Kelola Akses PIC Modul <i>Training</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola semua modul <i>training</i> . <i>Use case</i> ini ditujukan untuk pengelolaan modul <i>training</i> di level administrator, serta memberikan akses terhadap modul <i>learning</i> .	Karyawan HC
11.	Penjadwalan <i>Training</i> Luring	<i>Use case</i> ini menggambarkan kegiatan untuk menandakan ruangan <i>meeting</i> untuk	PIC <i>Training</i> dan Karyawan HC

		dijadikan ruangan <i>training</i> luring.	
12.	Akses Modul <i>Training</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengakses modul <i>training</i> dengan melihat modul dan membuat progres belajar.	Peserta <i>Training</i> , Pemateri <i>Training</i>
13.	Akses Laporan <i>Training</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengakses laporan belajar Peserta <i>Training</i> seperti progres dan nilai belajarnya.	PIC <i>Training</i> dan Karyawan HC
14.	Penjadwalan VLT ( <i>Virtual Live Training</i> )	<i>Use case</i> ini menggambarkan kegiatan untuk menjadwalkan VLT di platform konferensi video ( <i>video conference</i> ).	PIC <i>Training</i> dan Karyawan HC
15.	Akses Forum <i>Training</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengakses forum <i>training</i> seperti melihat forum, membuat posting di forum, membalas posting,	Peserta <i>Training</i> , Pemateri <i>Training</i> , PIC <i>Training</i> , Karyawan HC
16.	Moderasi Forum <i>Training</i>	<i>Use case</i> ini menggambarkan kegiatan untuk melakukan moderasi forum <i>training</i> di level administrator yang diberikan akses oleh Karyawan HC. Karyawan HC sendiri memiliki akses ini di semua forum <i>Training</i> .	PIC <i>Training</i> , Karyawan HC



Gambar 4.26 Use Case Diagram e-Learning dan Forum pada HC Portal

Tabel 4.7 Spesifikasi Use Case Buka Halaman Utama

<b>Use Case Name</b>	Buka Halaman Awal	
<b>Actor(s)</b>	Semua Aktor	
<b>Description</b>	Use Case ini mendeskripsikan halaman aplikasi yang dapat diakses oleh semua aktor tanpa melakukan proses login.	
<b>Pre-Condition(s)</b>	Aktor membuka aplikasi peramban web ( <i>web browser</i> ) pada komputer mereka.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b> 1. Masuk ke alamat web aplikasi HC Portal. <b>Application Response</b> 2. Menampilkan halaman awal aplikasi HC Portal.	

<b>Alternative Course(s)</b>	-
<b>Conclusion</b>	<i>Use Case</i> berakhir saat pengguna telah masuk ke halaman awal aplikasi HC Portal.
<b>Post-Condition(s)</b>	Pengguna masuk ke halaman utama aplikasi HC Portal.
<b>Implementation Constraint(s) and Specification(s)</b>	Halaman utama aplikasi HC Portal.

**Tabel 4.8 Spesifikasi *Use Case Login***

<b>Use Case Name</b>	<i>Login</i>	
<b>Actor(s)</b>	Seluruh Karyawan CTI	
<b>Description</b>	<i>Use Case</i> ini mendeskripsikan proses autentikasi aktor agar dapat mengakses fungsi aplikasi.	
<b>Pre-Condition(s)</b>	Aktor telah berada di halaman utama aplikasi HC Portal.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor melakukan klik tombol <i>login</i> yang tersedia pada halaman utama.	2. Menampilkan <i>modal</i> (jendela <i>popup</i> ) <i>login</i> .
	3. Memasukkan nomor induk pegawai dan kata sandinya pada kotak formulir yang tersedia.	4. Memvalidasi format input pengguna.
	5. Klik tombol “ <i>Login</i> ”.	6. Aplikasi melakukan autentikasi input pengguna berupa nomor nik dan kata sandinya. 7. Aplikasi menampilkan halaman awal <i>Job Profile</i> .
<b>Alternative Course(s)</b>	Alternatif 1: 4a. Aplikasi memvalidasi format input pengguna dan menemukan eror penulisan.	

	<p>5a. Aplikasi memberikan pesan eror pada kotak formulirnya.</p> <p>6a. Aktor mengulangi langkah ketiga.</p> <p>Alternatif 2:</p> <p>7a. Pengguna akan diarahkan ke halaman awal dengan jendela <i>popup login</i> yang juga menampilkan pesan eror atas kegagalan autentifikasi. Pengguna dapat menjalankan ulang <i>use case</i> ini ke langkah 1 kembali.</p>
<b>Conclusion</b>	<i>Use Case</i> berakhir saat pengguna telah masuk ke halaman awal <i>Job Profile</i> .
<b>Post-Condition(s)</b>	Pengguna masuk ke halaman awal <i>Job Profile</i> .
<b>Implementation Constraint(s) and Specification(s)</b>	Halaman awal <i>Job Profile</i> ..

Tabel 4.9 Spesifikasi *Use Case Logout*

<b>Use Case Name</b>	<i>Logout</i>	
<b>Actor(s)</b>	Seluruh Karyawan CTI	
<b>Description</b>	<i>Use Case</i> ini mendeskripsikan proses keluar dari sesi autentifikasi aktor sebagai aktivitas mengakhiri akses aplikasi HC Portal.	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman portal pengguna dan dalam keadaan telah berhasil melaksanakan <i>Use Case Login</i> .	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b> <ul style="list-style-type: none"> <li>1. Aktor melakukan klik pada area profil di pojok kanan baki navigasi (<i>navigation bar/navbar</i>).</li> <li>3. Aktor melakukan klik tombol “Logout”.</li> </ul>	<b>Application Response</b> <ul style="list-style-type: none"> <li>2. Menampilkan <i>dropdown</i> tombol profil dan “logout”.</li> <li>4. Aplikasi menghapus sesi pengguna pada basis data.</li> </ul>
		<ul style="list-style-type: none"> <li>5. Aplikasi mengarahkan pengguna ke halaman utama aplikasi HC Portal.</li> </ul>

<b>Alternative Course(s)</b>	-
<b>Conclusion</b>	<i>Use Case</i> berakhir saat pengguna telah masuk ke halaman utama aplikasi HC Portal.
<b>Post-Condition(s)</b>	Pengguna masuk ke halaman utama aplikasi HC Portal.
<b>Implementation Constraint(s) and Specification(s)</b>	<i>Dropdown</i> profil pengguna.

Tabel 4.10 Spesifikasi *Use Case* Kelola Profil Akun

<b>Use Case Name</b>	Kelola Profil Akun	
<b>Actor(s)</b>	Seluruh Karyawan CTI	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola profil pengguna seperti mengubah nama, surel, dan kata sandi.	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman awal JP.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b> <ul style="list-style-type: none"> <li>1. Aktor melakukan klik pada area profil di pojok kanan baki navigasi (<i>navigation bar/navbar</i>).</li> <li>3. Aktor melakukan klik tombol profil.</li> <li>5. Aktor mengedit nama dan surel di kotak formulir yang tersedia.</li> <li>6. Aktor melakukan klik tombol “Save” pada akhir formulir.</li> </ul>	<b>Application Response</b> <ul style="list-style-type: none"> <li>2. Aplikasi menampilkan <i>dropdown</i> tombol profil dan “logout”.</li> <li>4. Aplikasi menampilkan halaman profil pengguna.</li> </ul>
<b>Alternative Course(s)</b>	Alternatif 1: 5a. Aktor memperbarui kata sandi pengguna, kemudian memasukkan kata sandi lamanya dengan benar pada kotak formulir yang tersedia.	

	<p>Alternatif 2:</p> <p>5b. Aktor memperbarui kata sandi pengguna, kemudian memasukkan kata sandi lamanya dengan salah pada kotak formulir yang tersedia.</p> <p>7a. Aplikasi menampilkan pesan eror bahwa pengguna salah memasukkan kata sandi lamanya.</p>
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menyimpan perubahan data profil pengguna, menyimpan kata sandi baru pengguna, atau menampilkan pesan eror bahwa pengguna salah memasukkan kata sandi lamanya.
<b>Post-Condition(s)</b>	Aplikasi berhasil menjalankan perintah pengubahan data profil, kata sandi, atau menampilkan pesan eror apabila pengguna salah memasukkan kata sandi lamanya.
<b>Implementation Constraint(s) and Specification(s)</b>	Halaman pengelola profil pengguna.

Tabel 4.11 Spesifikasi *Use Case* Kelola Data Karyawan

<b>Use Case Name</b>	Kelola Data Karyawan	
<b>Actor(s)</b>	Karyawan HC	
<b>Description</b>	<i>Use case</i> ini mendeskripsikan kegiatan untuk mengelola seluruh data karyawan seperti melihat data karyawan, mengeditnya, menghapusnya, dan membuat data karyawan baru.	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman awal JP.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor melakukan klik menu “Settings” lalu ke sub menu “Master Data”.	2. Aplikasi menampilkan halaman menu edit “Master Data”.
	3. Aktor melakukan klik tombol “Manage Employee”.	4. Aplikasi menampilkan halaman yang berisi daftar karyawan.
	5. Aktor melakukan klik tombol tambah karyawan di halaman tersebut.	6. Aplikasi menampilkan jendela <i>popup</i> untuk menambahkan karyawan baru.

	<p>7. Aktor mengisi formulir penambahan data karyawan baru, kemudian melakukan klik tombol “Save” pada akhir formulir.</p> <p>8. Aplikasi menyimpan data karyawan baru pada basis data.</p>
<i>Alternative Course(s)</i>	<p>Alternatif 1:</p> <p>5a. Aktor melakukan klik tombol edit pada daftar karyawan yang tersedia pada tabel.</p> <p>6a. Aplikasi menampilkan jendela <i>popup</i> untuk mengedit data karyawan.</p> <p>7a. Aktor mengisi formulir edit data karyawan, kemudian melakukan klik tombol “Save” pada akhir formulir.</p> <p>8a. Aplikasi menyimpan perubahan data karyawan pada basis data.</p> <p>Alternatif 2:</p> <p>5b. Aktor melakukan klik tombol hapus pada daftar karyawan yang tersedia pada tabel.</p> <p>6a. Aplikasi menampilkan jendela <i>popup</i> untuk meyakinkan admin dalam menghapus karyawan.</p> <p>7a. Aktor melakukan klik tombol “Ya”.</p> <p>8a. Aplikasi menghapus data karyawan pada basis data.</p>
<i>Conclusion</i>	<i>Use Case</i> berakhir saat aplikasi berhasil menyimpan data karyawan baru, menyimpan data karyawan, atau menghapus data karyawan.
<i>Post-Condition(s)</i>	Aplikasi berhasil menjalankan perintah penambahan data karyawan baru, pengubahan data karyawan, atau penghapusan data karyawan.
<i>Implementation Constraint(s) and Specification(s)</i>	Halaman pengelola data karyawan.

Tabel 4.12 Spesifikasi *Use Case* Akses Modul Belajar *Induction*

<b>Use Case Name</b>	Akses Modul Belajar <i>Induction</i>
<b>Actor(s)</b>	Seluruh Karyawan CTI
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengakses modul belajar khusus yaitu <i>induction</i> .

<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman awal JP.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor melakukan klik menu <i>e-learning</i> pada bilah samping ( <i>navbar/navigation bar</i> ) utama aplikasi HC Portal.  3. Aktor melakukan klik modul <i>Induction e-learning</i> .	2. Aplikasi menampilkan halaman awal aplikasi <i>e-learning</i> HC Portal.  4. Aplikasi menampilkan halaman <i>overview</i> modul belajar <i>Induction</i> .
<b>Alternative Course(s)</b>	-	
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menampilkan halaman <i>overview</i> modul belajar <i>induction</i> untuk seluruh pengguna yang merupakan karyawan CTI.	
<b>Post-Condition(s)</b>	Aplikasi berhasil menampilkan halaman <i>overview</i> modul belajar <i>induction</i> untuk seluruh pengguna yang merupakan karyawan CTI.	
<b>Implementation Constraint(s) and Specification(s)</b>	Halaman awal aplikasi <i>e-learning</i> HC Portal, halaman <i>overview</i> modul <i>learning</i> .	

Tabel 4.13 Spesifikasi *Use Case* Isi Formulir Pengajuan Training Baru (PTB)

<b>Use Case Name</b>	Isi Formulir Pengajuan Training Baru (PTB)	
<b>Actor(s)</b>	Kepala Departemen, Kadiv Departemen	
<b>Description</b>	<i>Use case</i> ini mendeskripsikan kegiatan untuk mengisi formulir PTB ketika ada <i>training</i> karyawan baru yang ingin diadakan di CTI.	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning</i> HC Portal.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor melakukan klik menu Pengajuan <i>Training Baru</i> pada	2. Aplikasi menampilkan halaman <i>overview</i> PTB.

	menu sidebar aplikasi e-learning HC Portal.	
	3. Aktor melakukan klik tombol Pengajuan Training Baru.	4. Aplikasi menampilkan halaman pengisian formulir PTB.
	5. Aktor mengisi formulir dengan benar, lalu melakukan klik tombol Submit.	6. Aplikasi mendeteksi pengguna yang melakukan submit adalah Kepala Departemen, menandainya sebagai label status progres, menyimpan formulir PTB ke dalam basis data, dan mengirimkan notifikasi surel kepada Kadiv Departemen.
<b>Alternative Course(s)</b>	<p>Alternatif 1:</p> <p>5a. Aktor mengisi formulir dengan tidak benar, lalu melakukan klik tombol <i>Submit</i> atau <i>Save</i>.</p> <p>6a. Aplikasi menampilkan pesan eror validasi formulir.</p> <p>Alternatif 2:</p> <p>6b. Aplikasi mendeteksi pengguna yang melakukan submit adalah Kadiv Departemen, menandainya sebagai label status progres, menyimpan formulir PTB ke dalam basis data, dan mengirimkan notifikasi surel kepada Karyawan HC.</p> <p>Alternatif 3:</p> <p>5c. Aktor mengisi formulir dengan benar, lalu melakukan klik tombol <i>Save</i>.</p> <p>6c. Aplikasi mendeteksi pengguna yang melakukan submit adalah Kadiv Departemen atau Kepala Departemen, dan menandai formulir sebagai <i>draft</i> dan menyimpannya ke dalam basis data.</p>	
<b>Conclusion</b>	<p><i>Use Case</i> berakhir saat aplikasi menyimpan formulir ke dalam basis data dan mengirimkan surel notifikasi untuk menuju proses persetujuan, atau menampilkan pesan eror validasi formulir apabila pengguna tidak mengisinya dengan benar.</p>	
<b>Post-Condition(s)</b>	Aplikasi berhasil menyimpan formulir ke dalam basis data dan mengirimkan surel notifikasi untuk menuju	

	proses persetujuan, atau menampilkan pesan eror validasi formulir apabila pengguna tidak mengisinya dengan benar. Hal ini membuat aplikasi dapat diandalkan sebagai media penyimpanan formulir PTB dan sebagai penjaga validitas formulir.
<b>Implementation Constraint(s) and Specification(s)</b>	Kerangka dasar modul aplikasi dengan <i>sidebar</i> , halaman <i>overview</i> PTB dan pengisian formulir PTB, serta komponen validasi formulir dengan AJAX.

Tabel 4.14 Spesifikasi *Use Case* Persetujuan *Training Baru*

<b>Use Case Name</b>	Persetujuan <i>Training Baru</i>	
<b>Actor(s)</b>	Kadiv Departemen, Karyawan HC, dan Kadiv HC	
<b>Description</b>	<i>Use case</i> ini mendeskripsikan proses persetujuan PTB dengan memproses formulirnya. Pada karyawan HC, ia bertugas untuk memvalidasi formulir PTB dengan memastikan semua data formulir terisi dengan semestinya sebelum diserahkan kepada Kadiv HC.	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning HC Portal</i> .	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b> <ol style="list-style-type: none"> <li>Aktor melakukan klik menu Pengajuan <i>Training Baru</i> pada menu <i>sidebar</i> aplikasi <i>e-learning HC Portal</i>.</li> <li>Aktor melakukan klik salah satu data pengajuan <i>training baru</i>.</li> <li>Aktor melakukan reviu formulir PTB dan melakukan klik tombol “Terima”.</li> </ol>	<b>Application Response</b> <ol style="list-style-type: none"> <li>Aplikasi menampilkan halaman <i>overview</i> PTB dengan menampilkan data pengajuan <i>training baru</i>.</li> <li>Aplikasi menampilkan halaman persetujuan formulir PTB.</li> <li>Aplikasi mendeteksi pengguna yang melakukan persetujuan adalah Kadiv Departemen, menandai formulir dengan label “Diterima oleh Kadiv Departemen”, memperbarui data formulir PTB, dan mengirimkan notifikasi</li> </ol>

		surel kepada Karyawan HC.
<i>Alternative Course(s)</i>	<p>Alternatif 1:</p> <p>6a. Aplikasi mendeteksi pengguna yang melakukan persetujuan adalah Karyawan HC, menandai formulir dengan label “Diterima oleh Karyawan HC”, memperbarui data formulir PTB, dan mengirimkan notifikasi surel kepada Kadiv HC.</p> <p>Alternatif 2:</p> <p>6b. Aplikasi mendeteksi pengguna yang melakukan persetujuan adalah Kadiv HC, menandai formulir dengan label “Diterima oleh Kadiv HC”, memperbarui data formulir PTB, dan mengirimkan notifikasi surel kepada Karyawan HC untuk selanjutnya diproses ke tahap selanjutnya.</p> <p>Alternatif 3:</p> <p>5c. Aktor melakukan reviu formulir PTB, melakukan klik tombol “Tolak Formulir”.</p> <p>6c. Aplikasi menampilkan pesan dialog untuk menuliskan catatan alasan penolakan formulir PTB.</p> <p>7c. Aktor mengisi catatan penolakan formulir PTB dan melakukan klik tombol “OK” di pesan dialog.</p> <p>8c. Aplikasi memperbarui status formulir dan memperbarunya pada basis data dengan label “Permintaan PTB ditolak oleh [nama aktor yang melakukannya]”, serta mengirimkan pesan surel notifikasi kepada para/satu aktor di level persetujuan sebelumnya.</p> <p>Alternatif 3:</p> <p>5d. Aktor melakukan reviu formulir PTB, melakukan klik tombol “Revisi Formulir”.</p> <p>6d. Aplikasi menampilkan pesan dialog untuk menuliskan catatan alasan revisi formulir PTB.</p> <p>7d. Aktor mengisi catatan revisi formulir PTB dan melakukan klik tombol “OK” di pesan dialog.</p> <p>8d. Aplikasi memperbarui status revisi menjadi “Revisi dibutuhkan dari [nama aktor yang melakukannya]”, memperbarui datanya pada basis data, serta mengirimkan pesan surel notifikasi kepada aktor di level persetujuan sebelumnya.</p> <p>Alternatif 4:</p>	

	<p>7e. Aktor tidak mengisi catatan revisi/penolakan formulir PTB. Kemudian, ia mengeklik tombol “OK” di pesan dialog.</p> <p>8e. Aplikasi menampilkan pesan eror dengan mengatakan bahwa aktor harus mengisi catatan revisi/penolakan.</p> <p>Alternatif 5:</p> <p>7f. Aktor mengisi catatan revisi/penolakan formulir PTB. Kemudian, ia mengeklik tombol “Cancel” di pesan dialog.</p> <p>8f. Aplikasi menutup pesan dialog.</p>
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi memperbarui status persetujuan data formulir PTB dan mengirimkan surel notifikasi sesuai proses persetujuan yang dipilih aktor tertentu.
<b>Post-Condition(s)</b>	Data formulir diperbarui sesuai dengan level persetujuannya dan surel notifikasi terkirim ke karyawan tertentu yang terlibat dalam proses persetujuan formulir PTB.
<b>Implementation Constraint(s) and Specification(s)</b>	Halaman persetujuan dan <i>popup</i> dialog pengisian pesan revisi dan penutupan formulir PTB.

Tabel 4.15 Spesifikasi *Use Case* Kelola Modul Training/Penambahan Modul Training Baru

<b>Use Case Name</b>	Kelola Modul Training/Penambahan Modul Training Baru	
<b>Actor(s)</b>	Karyawan HC	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola modul training yang diberi akses dengan mengedit konten modul training dan pemberian akses kepada peserta training. Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul training ditambah dengan penambahan dan pengarsipan modul training.	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi e-learning HC Portal.	
	<b>Actor Action</b>	<b>Application Response</b>

<b>Typical Course of Event(s)</b>	1. Aktor dapat melihat tombol penambahan modul <i>training</i> baru.	
	2. Aktor melakukan klik tombol penambahan modul <i>training</i> baru.	3. Aplikasi menampilkan halaman pemandu awal penambahan modul <i>training</i> baru.
	4. Aktor mengisi formulir pada halaman pemandu awal penambahan modul <i>training</i> dengan benar dan melakukan klik tombol “Buat modul <i>training</i> baru”.	5. Aplikasi membuat data modul <i>training</i> baru di basis data.
		6. Aplikasi menampilkan halaman kelola modul <i>training</i> .
<b>Alternative Course(s)</b>	Alternatif 1: 4a. Aktor mengisi formulir pada layar pemandu awal penambahan modul <i>training</i> dengan salah dan melakukan klik tombol “Buat modul <i>training</i> baru”. 5a. Aplikasi menampilkan pesan eror validasi formulir, dan aktor harus mengulang ke langkah 4.	
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi membuat data <i>training</i> baru dan menunjukkan halaman kelola modul <i>training</i> tersebut atau menampilkan pesan eror saat pengguna mengisi formulir dengan tidak benar.	
<b>Post-Condition(s)</b>	Data modul <i>training</i> berhasil dibuat di basis data dan aktor diarahkan ke halaman kelola modul <i>training</i> , atau aplikasi berhasil menampilkan pesan eror validasi.	
<b>Implementation Constraint(s) and Specification(s)</b>	Halaman awal pemandu awal penambahan modul <i>training</i> baru, validasi formulirnya, halaman kelola modul <i>training</i> .	

Tabel 4.16 Spesifikasi *Use Case* Kelola Modul *Training*/Pengarsipan Modul *Training*

<b>Use Case Name</b>	Kelola Modul <i>Training</i> /Pengarsipan Modul <i>Training</i>
<b>Actor(s)</b>	Karyawan HC

<b>Description</b>	<p>Use case ini menggambarkan kegiatan untuk mengelola modul <i>training</i> yang diberi akses dengan mengedit konten modul <i>training</i> dan pemberian akses kepada peserta <i>training</i>. Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul <i>training</i> ditambah dengan penambahan dan pengarsipan modul <i>training</i>.</p>		
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning</i> HC Portal.		
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>	
	1. Aktor memilih modul <i>training</i> yang ingin diarsipkan.	2. Aplikasi menampilkan halaman detail modul <i>training</i> .	
	3. Aktor memilih menu arsipkan di menu sidebar <i>e-learning</i> .	4. Aplikasi menampilkan halaman pengarsipan modul <i>training</i> .	
	5. Aktor melakukan klik tombol arsip modul <i>training</i> .	6. Aplikasi menampilkan pesan dialog notifikasi peringatan dengan tombol ya dan tidak.	
	7. Aktor melakukan klik tombol ya.	8. Aplikasi menampilkan pesan dialog konfirmasi teks. Aktor diminta untuk mengetikkan jawabannya.	
	9. Aktor mengisi konfirmasi teks dengan benar dan melakukan klik tombol ok.	10. Aplikasi berhasil memvalidasi konfirmasi teks dan menampilkan pesan pengarsipan modul <i>training</i> sedang diproses.	
		11. Aplikasi mengarsipkan modul <i>training</i> ke basis data hcportal_archives.	
		12. Aplikasi menampilkan halaman awal modul <i>training</i> serta menampilkan pesan dialog sukses	

		mengarsipkan modul <i>training</i> .
<b>Alternative Course(s)</b>	<p>Alternatif 1:</p> <p>7a. Aktor melakukan klik tombol tidak.</p> <p>8a. Aplikasi menutup pesan dialog notifikasi peringatan.</p> <p>Alternatif 2:</p> <p>9b. Aktor mengisi konfirmasi teks dengan salah dan melakukan klik tombol ok.</p> <p>9c. Aplikasi menampilkan eror validasi konfirmasi teks.</p>	
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi mengarsipkan data modul <i>training</i> ke basis data <i>heportal_archives</i> lalu menampilkan kembali halaman pengarsipan modul <i>training</i> , atau menutup pesan dialog konfirmasi peringatan, atau mematikan tombol ok ketika aktor tidak mengisi konfirmasi teks dengan benar.	
<b>Post-Condition(s)</b>	Data modul <i>training</i> berhasil diarsipkan, atau tidak diarsipkan jika proses konfirmasi pengarsipan gagal.	

Tabel 4.17 Spesifikasi *Use Case* Kelola Modul *Training*/Pengeditan Konten Modul *Training* Bertipe Tekstual dan Gambar

<b>Use Case Name</b>	Kelola Modul <i>Training</i> /Pengeditan Konten Modul <i>Training</i> Bertipe Tekstual dan Gambar	
<b>Actor(s)</b>	<i>PIC Training</i> , Karyawan HC	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola modul <i>training</i> yang diberi akses dengan mengedit konten modul <i>training</i> dan pemberian akses kepada peserta <i>training</i> . Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul <i>training</i> ditambah dengan penambahan dan pengarsipan modul <i>training</i> .	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning HC Portal</i> .	
	<b>Actor Action</b>	<b>Application Response</b>

<b>Typical Course of Event(s)</b>	<ol style="list-style-type: none"> <li>1. Apabila kedua aktor ini mengakses modul aplikasi <i>e-learning</i>, mereka dapat melihat modul <i>training</i> yang diberi akses dan melihat tombol kelola di masing-masing modulnya.</li> </ol>	
	<ol style="list-style-type: none"> <li>2. Aktor mengeklik tombol kelola modul di salah satu modul <i>training</i>.</li> </ol>	<ol style="list-style-type: none"> <li>3. Aplikasi menampilkan halaman kelola modul <i>training</i>.</li> </ol>
	<ol style="list-style-type: none"> <li>4. Aktor memilih tombol tambah konten di sidebar <i>e-learning</i>.</li> </ol>	<ol style="list-style-type: none"> <li>5. Aplikasi menampilkan <i>popup dialog</i> pembuatan konten.</li> </ol>
	<ol style="list-style-type: none"> <li>6. Aktor mengetik judul konten, kemudian memilih untuk menambahkan konten bertipe tekstual, gambar dan video.</li> </ol>	<ol style="list-style-type: none"> <li>7. Aplikasi membuat data modul awal di basis data (keperluan simpan otomatis/<i>autosave</i>) serta menampilkan pengedit teks CKEditor di bawah pilihannya.</li> </ol>
	<ol style="list-style-type: none"> <li>8. Aktor mengetik konten teks.</li> </ol>	<ol style="list-style-type: none"> <li>9. Aplikasi menyimpan perubahan aktor ke dalam basis data secara otomatis selama 15 detik sekali. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.</li> </ol>
	<ol style="list-style-type: none"> <li>10. Aktor memilih tombol “simpan dan tutup”.</li> </ol>	<ol style="list-style-type: none"> <li>11. Aplikasi menyimpan perubahan aktor ke basis data, menutup pengedit teks CKEditor, menampilkan pesan perubahan berhasil disimpan, kemudian menampilkan judul konten di sidebar.</li> </ol>

	<p>12. Aktor mengeklik tombol edit di konten tekstual yang baru saja ia buat.</p>	<p>13. Aplikasi menampilkan <i>popup dialog</i> pengeditan teks CKEditor untuk konten dipilih.</p>
	<p>14. Aktor memilih tombol penambahan gambar ke dalam konten.</p>	<p>15. Aplikasi menampilkan <i>popup pemilihan berkas</i> untuk menambahkan gambar.</p>
	<p>16. Aktor memilih gambar yang ingin dimasukkan.</p>	<p>17. Aplikasi mengunggah gambar dan menyimpannya pada penyimpanan aplikasi, kemudian menyimpan perubahan aktor ke dalam basis data, serta menampilkan gambar yang diunggah pada pengedit teks CKEditor.</p>
	<p>18. Aktor mengedit judul konten, kemudian mengeklik tombol “simpan dan tutup”.</p>	<p>19. Aplikasi menyimpan konten ke dalam basis data, menutup pengedit teks CKEditor, menampilkan pesan perubahan berhasil disimpan, serta menampilkan judul konten di <i>sidebar e-learning</i>.</p>
	<p>20. Aktor mengeklik tombol hapus konten.</p>	<p>21. Aplikasi menampilkan <i>popup notifikasi peringatan penghapusan konten</i>.</p>
	<p>22. Aktor mengeklik tombol ya.</p>	<p>23. Aplikasi menampilkan <i>popup dialog verifikasi judul konten</i> untuk memastikan bahwa pengguna benar-benar ingin menghapus konten.</p>

	24. Aktor mengetik judul konten dengan benar.	25. Aplikasi menghapus konten dari basis data, kemudian menampilkan pesan berhasil dihapus.
<b>Alternative Course(s)</b>	Alternatif 1: 18a. Aktor memilih berkas selain gambar atau format gambar yang tidak diizinkan. 19a. Aplikasi menampilkan pesan eror bahwa format berkas tidak diizinkan. Alternatif 2: 24b. Aktor mengeklik tombol tidak. 25b. Aplikasi menutup <i>popup</i> notifikasi peringatan penghapusan konten. Alternatif 3: 26b. Aktor mengetik nama konten dengan salah. 27b. Aplikasi menampilkan eror verifikasi nama konten dan mematikan tombol hapus di <i>popup</i> dialog verifikasi nama konten.	
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan serangkaian proses pembuatan, pengeditan, dan penghapusan konten bertipe teksual dan gambar, serta tiga alternatif skenario yang sudah dijelaskan.	
<b>Post-Condition(s)</b>	Menampilkan pesan eror validasi, atau kembali ke tampilan sebelumnya.	
<b>Implementation Constraint(s) and Specification(s)</b>	Halaman kelola modul <i>training</i> ; <i>popup</i> dialog pembuatan konten bertipe teks dan gambar; komponen pengedit tipografi teks; <i>popup</i> dialog pengeditan konten modul bertipe teks dan gambar; <i>popup</i> notifikasi untuk peringatan, dan pesan berhasil.	

Tabel 4.18 Spesifikasi *Use Case* Kelola Modul *Training*/Pengeditan Konten Modul *Training* Bertipe Video

<b>Use Case Name</b>	Kelola Modul <i>Training</i> / Pengeditan Konten Modul <i>Training</i> Bertipe Video
<b>Actor(s)</b>	PIC <i>Training</i> , Karyawan HC
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola modul <i>training</i> yang diberi akses dengan mengedit konten modul <i>training</i> dan pemberian akses kepada

	peserta <i>training</i> . Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul <i>training</i> ditambah dengan penambahan dan pengarsipan modul <i>training</i> .	
<b>Pre-Condition(s)</b>	Aktor telah membuka platform video YouTube.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor mengunggah video <i>training</i> ke platform YouTube, kemudian menyetel videonya ke mode <i>unlisted</i> .	
	2. Aktor menyalin tautan ke video <i>training</i> . Lalu membuka aplikasi HC Portal dan menuju halaman utama aplikasi <i>e-learning</i> .	3. Aplikasi menampilkan halaman utama <i>e-learning</i> .
	4. Apabila kedua aktor ini mengakses modul aplikasi <i>e-learning</i> , mereka dapat melihat modul <i>training</i> yang diberi akses dan melihat tombol kelola di masing-masing modulnya.	
	5. Aktor mengeklik tombol kelola modul di salah satu modul <i>training</i> .	6. Aplikasi menampilkan halaman kelola modul <i>training</i> .
	7. Aktor menambahkan konten modul <i>training</i> .	8. Aplikasi menampilkan <i>popup dialog</i> pembuatan konten.
	9. Aktor mengetik judul konten, kemudian memilih untuk menambahkan konten bertipe textual, gambar dan video.	10. Aplikasi membuat data modul awal di basis data (keperluan simpan otomatis/ <i>autosave</i> ) serta menampilkan

		pengedit teks CKEditor di bawah pilihannya.
	11. Aktor memilih tombol untuk memasukkan konten video.	12. Aplikasi menampilkan <i>popup</i> untuk menambahkan tautan video.
	13. Aktor menempelkan tautan video yang sebelumnya telah ia salin sebelumnya. Kemudian meneklik tombol centang hijau.	14. Aplikasi menanamkan ( <i>embed</i> ) video YouTube ke dalam konten modul <i>training</i> dan menampilkan pratinjau videonya.
	15. Aktor dapat menambahkan teks atau gambar tambahan di bawah videonya.	16. Aplikasi menyimpan perubahan aktor ke dalam basis data secara otomatis selama 15 detik sekali. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.
	17. Aktor memilih tombol “simpan dan tutup”.	18. Aplikasi menyimpan perubahan pengguna, menutup pengedit teks CKEditor, menampilkan pesan perubahan berhasil disimpan, kemudian menampilkan judul konten di sidebar <i>e-learning</i> .
<i>Alternative Course(s)</i>	-	
<i>Conclusion</i>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan serangkaian proses pembuatan konten bertipe video.	
<i>Post-Condition(s)</i>	Terdapat satu nama konten baru di modul <i>training</i> yang diedit.	
<i>Implementation Constraint(s) and Specification(s)</i>	Komponen pengelola pratinjau media video.	

**Tabel 4.19 Spesifikasi Use Case Kelola Modul Training/Pengeditan Konten Modul training Bertipe PDF (presentasi dek atau berkas .pdf lainnya)**

<b>Use Case Name</b>	Kelola Modul Training/Pengeditan Konten Modul Training Bertipe PDF	
<b>Actor(s)</b>	PIC Training, Karyawan HC	
<b>Description</b>	<p><i>Use case</i> ini menggambarkan kegiatan untuk mengelola modul <i>training</i> yang diberi akses dengan mengedit konten modul <i>training</i> dan pemberian akses kepada peserta <i>training</i>. Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul <i>training</i> ditambah dengan penambahan dan pengarsipan modul <i>training</i>.</p>	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning</i> HC Portal.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	<ol style="list-style-type: none"> <li>Apabila kedua aktor ini mengakses modul aplikasi <i>e-learning</i>, mereka dapat melihat modul <i>training</i> yang diberi akses dan melihat tombol kelola di masing-masing modulnya.</li> </ol>	
	<ol style="list-style-type: none"> <li>Aktor mengeklik tombol kelola modul di salah satu modul <i>training</i>.</li> </ol>	<ol style="list-style-type: none"> <li>Aplikasi menampilkan halaman kelola modul <i>training</i>.</li> </ol>
	<ol style="list-style-type: none"> <li>Aktor menambahkan konten modul <i>training</i>.</li> </ol>	<ol style="list-style-type: none"> <li>Aplikasi menampilkan <i>popup dialog</i> pembuatan konten.</li> </ol>
	<ol style="list-style-type: none"> <li>Aktor mengetik judul konten, kemudian memilih untuk menambahkan konten bertipe pdf.</li> </ol>	<ol style="list-style-type: none"> <li>Aplikasi membuat data modul awal di basis data serta menampilkan pengunggah berkas PDF.</li> </ol>
	<ol style="list-style-type: none"> <li>Aktor memilih berkas PDF yang ingin diunggah lalu</li> </ol>	<ol style="list-style-type: none"> <li>Aplikasi mengunggah berkas PDF, menyimpan perubahan konten modul <i>training</i>,</li> </ol>

	mengeklik tombol unggah.	kemudian menampilkan pratinjau dari berkas PDF yang telah diunggah.
	10. Aktor memilih tombol “simpan dan tutup”.	11. Aplikasi menutup pengedit konten, menampilkan pesan perubahan berhasil disimpan, kemudian menampilkan judul konten di <i>sidebar e-learning</i> .
	12. Aktor memilih tombol edit di konten modul <i>training</i> yang baru saja ia tambahkan.	13. Aplikasi menampilkan konten modul <i>training</i> , beserta formulir pengunggah berkas PDF.
	14. Aktor memilih berkas PDF baru, kemudian mengeklik tombol unggah.	15. Aplikasi mengunggah berkas PDF, menghapus berkas PDF lama, kemudian memperbarui konten modul <i>training</i> pada basis data, serta menampilkan pesan bahwa modul <i>training</i> berhasil diperbarui.
	16. Aktor memilih tombol “simpan dan tutup”.	17. Aplikasi menutup pengedit konten modul <i>training</i> , kemudian menampilkan judul konten di <i>sidebar e-learning</i> .
	18. Aktor mengeklik tombol hapus konten.	19. Aplikasi menampilkan <i>popup</i> notifikasi peringatan penghapusan konten.
	20. Aktor mengeklik tombol ya.	21. Aplikasi menampilkan <i>popup</i> dialog verifikasi judul konten untuk memastikan bahwa pengguna benar-benar

		ingin menghapus konten.
	22. Aktor mengetik verifikasi konten dengan benar, lalu mengeklik tombol hapus.	23. Aplikasi menghapus konten dari basis data, kemudian menampilkan pesan berhasil dihapus.
<i>Alternative Course(s)</i>	Alternatif 1: 20a. Aktor mengeklik tombol tidak. 20b. Aplikasi menutup <i>popup</i> notifikasi peringatan penghapusan konten. Alternatif 2: 22b. Aktor mengetik nama konten dengan salah, lalu mengeklik tombol hapus. 23b. Aplikasi menampilkan eror verifikasi nama konten.	
<i>Conclusion</i>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses pembuatan konten bertipe berkas PDF, mengeditnya dan menghapusnya.	
<i>Post-Condition(s)</i>	Penambahan konten baru bertipe PDF telah dihapus.	
<i>Implementation Constraint(s) and Specification(s)</i>	Komponen pengelola konten modul <i>training</i> bertipe PDF.	

Tabel 4.20 Spesifikasi *Use Case* Kelola Modul *Training*/Pembuatan Konten Kuis

<i>Use Case Name</i>	Kelola Modul <i>Training</i> /Pembuatan Konten Kuis	
<i>Actor(s)</i>	PIC <i>Training</i> , Karyawan HC	
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola modul <i>training</i> yang diberi akses dengan mengedit konten modul <i>training</i> dan pemberian akses kepada peserta <i>training</i> . Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul <i>training</i> ditambah dengan penambahan dan pengarsipan modul <i>training</i> .	
<i>Pre-Condition(s)</i>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning</i> HC Portal.	
	<i>Actor Action</i>	<i>Application Response</i>

<b>Typical Course of Event(s)</b>	<p>1. Apabila kedua aktor ini mengakses modul aplikasi <i>e-learning</i>, mereka dapat melihat modul <i>training</i> yang diberi akses dan melihat tombol kelola di masing-masing modulnya.</p>	
	<p>2. Aktor mengeklik tombol kelola modul di salah satu modul <i>training</i>.</p>	<p>3. Aplikasi menampilkan halaman kelola modul <i>training</i>.</p>
	<p>4. Aktor menambahkan konten modul <i>training</i>.</p>	<p>5. Aplikasi menampilkan <i>popup dialog</i> pembuatan konten.</p>
	<p>6. Aktor mengetik judul konten, kemudian memilih untuk menambahkan konten kuis.</p>	<p>7. Aplikasi membuat data modul awal di basis data (keperluan simpan otomatis/<i>autosave</i>) serta menampilkan komponen pengelola kuis.</p>
	<p>8. Aktor mengisi formulir kuis.</p>	<p>9. Aplikasi menyimpan perubahan aktor ke dalam basis data secara otomatis selama 15 detik sekali. Aplikasi juga menampilkan status penyimpanan perubahan aktor.</p>
	<p>10. Aktor memilih tombol “simpan dan tutup”.</p>	<p>11. Aplikasi menyimpan perubahan aktor, menutup pengedit konten, menampilkan pesan perubahan berhasil disimpan, kemudian menampilkan judul konten di <i>sidebar e-learning</i>.</p>

	<p>12. Aktor mengeklik tombol edit di konten modul <i>training</i> yang baru saja ia tambahkan.</p>	<p>13. Aplikasi menampilkan konten modul <i>training</i> bertipe kuis.</p>
	<p>14. Aktor menghapus satu pertanyaan di konten kuis yang telah dibuat dan mengubah beberapa pertanyaan pada konten kuis.</p>	<p>15. Aplikasi menyimpan perubahan aktor ke dalam basis data secara otomatis selama 15 detik sekali. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.</p>
	<p>16. Aktor memilih tombol “simpan dan tutup”.</p>	<p>17. Aplikasi menyimpan perubahan konten pada basis data, menutup pengedit konten modul <i>training</i>, kemudian menampilkan judul konten di sidebar <i>e-learning</i>.</p>
	<p>18. Aktor mengeklik tombol hapus konten.</p>	<p>19. Aplikasi menampilkan <i>popup</i> notifikasi peringatan penghapusan konten.</p>
	<p>20. Aktor mengeklik tombol ya.</p>	<p>21. Aplikasi menampilkan <i>popup</i> dialog verifikasi judul konten untuk memastikan bahwa pengguna benar-benar ingin menghapus konten.</p>
	<p>22. Aktor mengetik verifikasi judul konten dengan benar.</p>	<p>23. Aplikasi menghapus konten dari basis data, kemudian menampilkan pesan berhasil dihapus.</p>
<p><i>Alternative Course(s)</i></p>	<p>Alternatif 1:</p> <p>20a. Aktor mengeklik tombol tidak.</p> <p>20b. Aplikasi menutup <i>popup</i> notifikasi peringatan penghapusan konten.</p> <p>Alternatif 2:</p>	

	<p>22b. Aktor mengetik verifikasi judul konten dengan salah.</p> <p>23b. Aplikasi menampilkan eror verifikasi judul konten.</p> <p>Alternatif 3:</p> <p>14c. Aktor menghapus semua data kuis.</p> <p>15c. Aplikasi menampilkan pesan eror untuk meminta aktor agar setidaknya kuis memiliki satu pertanyaan.</p>
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses pembuatan konten kuis, pengeditannya, dan penghapusannya, serta aplikasi dapat menjalankan tiga alternatif skenario yang sudah dijelaskan.
<b>Post-Condition(s)</b>	Penambahan konten baru bertipe kuis telah dihapus.
<b>Implementation Constraint(s) and Specification(s)</b>	Komponen pengelola konten kuis modul <i>training</i> .

Tabel 4.21 Spesifikasi *Use Case* Penjadwalan *Training Luring*

<b>Use Case Name</b>	Penjadwalan <i>Training Luring</i>	
<b>Actor(s)</b>	<i>PIC Training</i> , Karyawan HC	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk membuat penjadwalan <i>training luring</i> pada modul <i>training</i> yang diberi akses. Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul <i>training</i> .	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning</i> HC Portal.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	<ol style="list-style-type: none"> <li>Apabila kedua aktor ini mengakses modul aplikasi <i>e-learning</i>, mereka dapat melihat modul <i>training</i> yang diberi akses dan melihat tombol kelola di masing-masing modulnya.</li> </ol>	

	<p>2. Aktor mengeklik tombol kelola modul di salah satu modul <i>training</i>.</p>	<p>3. Aplikasi menampilkan halaman kelola modul <i>training</i>.</p>
	<p>4. Aktor menambahkan konten modul <i>training</i>.</p>	<p>5. Aplikasi menampilkan <i>popup dialog</i> pembuatan konten.</p>
	<p>6. Aktor mengetik judul konten, kemudian memilih untuk membuat konten menjadi jadwal <i>training</i> luring.</p>	<p>7. Aplikasi membuat data awal konten (untuk keperluan penyimpanan otomatis/<i>autosave</i>) dan menampilkan formulir pembuat jadwal <i>training</i>.</p>
	<p>8. Aktor mengisi formulir jadwal <i>training</i> dan memilih ruangannya.</p>	<p>9. Aplikasi menyimpan perubahan aktor ke dalam basis data secara otomatis selama 15 detik sekali. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.</p>
	<p>10. Aktor memilih tombol “simpan dan tutup”.</p>	<p>11. Aplikasi menyimpan perubahan data pada basis data, menutup pengedit konten, menampilkan pesan perubahan berhasil disimpan, kemudian menampilkan judul konten di sidebar <i>e-learning</i>.</p>
	<p>12. Aktor memilih tombol edit di konten jadwal <i>training</i> luring yang baru saja ia tambahkan.</p>	<p>13. Aplikasi menampilkan konten jadwal <i>training</i> luring.</p>
	<p>14. Aktor mengedit data jadwal <i>training</i> luring.</p>	<p>15. Aplikasi menyimpan perubahan aktor ke dalam basis data secara otomatis selama 15 detik sekali. Aplikasi juga menampilkan</p>

		status penyimpanan perubahan pengguna.
16. Aktor memilih tombol “simpan dan tutup”.	17. Aplikasi menyimpan perubahan konten pada basis data, menutup pengedit konten modul <i>training</i> , kemudian menampilkan judul konten di sidebar <i>e-learning</i> .	
18. Aktor membuka konten jadwal <i>training</i> luring yang baru saja ditambahkan.	19. Aplikasi menampilkan <i>popup</i> konten jadwal <i>training</i> luring.	
20. Aktor mengeklik tombol “tambahkan ke Google Calendar”.	21. Aplikasi mengarahkan pengguna ke halaman penambahan <i>event</i> di Google Calendar.	
22. Pengguna kembali ke halaman aplikasi <i>e-learning</i> HC Portal dan mengeklik tombol silang untuk menutup <i>popup</i> konten jadwal <i>training</i> luring.	23. Aplikasi menutup <i>popup</i> konten jadwal <i>training</i> luring.	
24. Aktor mengeklik tombol hapus konten.	25. Aplikasi menampilkan <i>popup</i> notifikasi peringatan penghapusan konten.	
26. Aktor mengeklik tombol ya.	27. Aplikasi menampilkan <i>popup</i> dialog verifikasi judul konten untuk memastikan bahwa pengguna benar-benar ingin menghapus konten.	
28. Aktor mengetik judul konten dengan benar.	29. Aplikasi menghapus konten dari basis data, kemudian menampilkan pesan berhasil dihapus.	

<b>Alternative Course(s)</b>	<p>Alternatif 1:</p> <p>26a. Aktor mengeklik tombol tidak.</p> <p>27a. Aplikasi menutup <i>popup</i> notifikasi peringatan penghapusan konten.</p> <p>Alternatif 2:</p> <p>28b. Aktor mengetik nama konten dengan salah.</p> <p>29b. Aplikasi menampilkan eror verifikasi nama konten.</p> <p>Alternatif 3:</p> <p>14c. Aktor menghapus semua data jadwal.</p> <p>15c. Aplikasi menampilkan pesan eror untuk meminta aktor agar mengisi semua data yang diperlukan.</p>
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses pembuatan jadwal <i>training</i> luring, pengeditannya, dan penghapusannya, serta aplikasi dapat menjalankan tiga alternatif skenario yang sudah dijelaskan.
<b>Post-Condition(s)</b>	Penambahan konten jadwal <i>training</i> luring telah dihapus.
<b>Implementation Constraint(s) and Specification(s)</b>	Komponen pengelola konten jadwal <i>training</i> luring; komponen pembuat tautan <i>add-to-Google-Calendar</i> .

Tabel 4.22 Spesifikasi *Use Case* Penjadwalan VLT (*Virtual Live Training*)

<b>Use Case Name</b>	Penjadwalan VLT ( <i>Virtual Live Training</i> )	
<b>Actor(s)</b>	<i>PIC Training</i> , Karyawan HC	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk membuat penjadwalan VLT pada modul <i>training</i> yang diberi akses. Karyawan HC sebagai super administrator juga memiliki akses ini di semua modul <i>training</i> .	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning</i> HC Portal.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Apabila kedua aktor ini mengakses modul aplikasi <i>e-learning</i> , mereka dapat melihat modul <i>training</i> yang diberi akses dan	

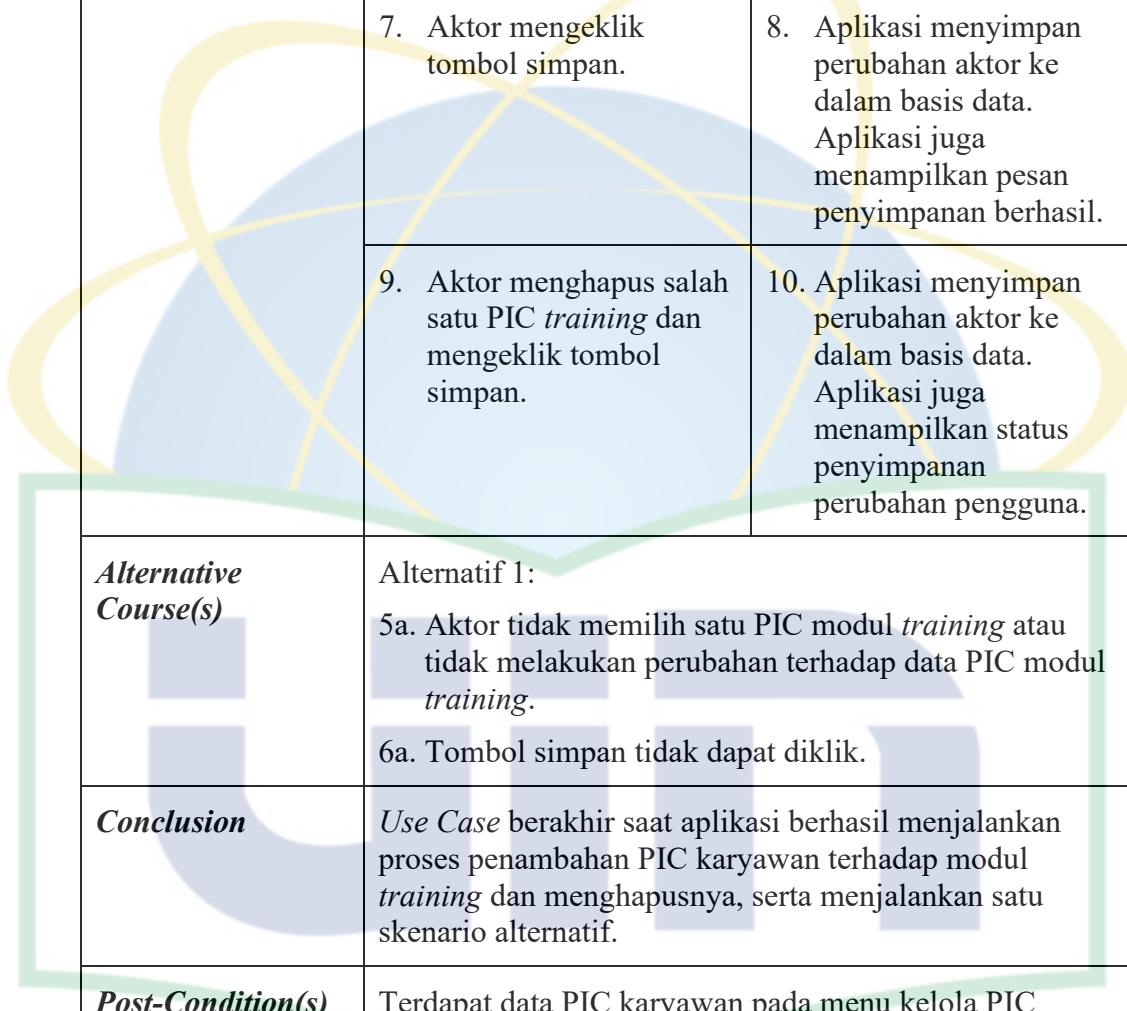
	<p>melihat tombol kelola di masing-masing modulnya.</p>	
	<p>2. Aktor mengeklik tombol kelola modul di salah satu modul <i>training</i>.</p>	<p>3. Aplikasi menampilkan halaman kelola modul <i>training</i>.</p>
	<p>4. Aktor menambahkan konten modul <i>training</i>.</p>	<p>5. Aplikasi menampilkan <i>popup dialog</i> pembuatan konten.</p>
	<p>6. Aktor mengetik judul konten, kemudian memilih untuk membuat konten menjadi jadwal VLT.</p>	<p>7. Aplikasi membuat data awal konten (untuk keperluan penyimpanan otomatis/<i>autosave</i>) dan menampilkan formulir membuat jadwal VLT.</p>
	<p>8. Aktor mengisi formulir jadwal VLT dan mengisi tautan ke konferensi video.</p>	<p>9. Aplikasi menyimpan perubahan aktor ke dalam basis data secara otomatis selama 15 detik sekali. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.</p>
	<p>10. Aktor memilih tombol “simpan dan tutup”.</p>	<p>11. Aplikasi menutup pengedit konten, menyimpan perubahan data pada basis data, menampilkan pesan perubahan berhasil disimpan, kemudian menampilkan judul konten di <i>sidebar e-learning</i>.</p>
	<p>12. Aktor memilih tombol edit di konten jadwal VLT yang baru saja ia tambahkan.</p>	<p>13. Aplikasi menampilkan konten jadwal VLT.</p>
	<p>14. Aktor mengedit data jadwal VLT.</p>	<p>15. Aplikasi menyimpan perubahan aktor ke dalam basis data secara otomatis selama 15</p>

		detik sekali. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.
	16. Aktor memilih tombol “simpan dan tutup”.	17. Aplikasi menyimpan perubahan konten pada basis data, menutup pengedit konten modul <i>training</i> , kemudian menampilkan judul konten di <i>sidebar e-learning</i> .
	18. Aktor membuka konten jadwal VLT yang baru saja ditambahkan.	19. Aplikasi menampilkan <i>popup</i> konten jadwal VLT.
	20. Aktor mengeklik tombol “tambahkan ke Google Calendar”.	21. Aplikasi mengarahkan pengguna ke halaman penambahan <i>event</i> di Google Calendar.
	22. Pengguna kembali ke halaman aplikasi <i>e-learning HC Portal</i> dan mengeklik tombol silang untuk menutup <i>popup</i> konten jadwal VLT.	23. Aplikasi menutup <i>popup</i> konten jadwal VLT.
	24. Aktor mengeklik tombol hapus konten.	25. Aplikasi menampilkan <i>popup</i> notifikasi peringatan penghapusan konten.
	26. Aktor mengeklik tombol ya.	27. Aplikasi menampilkan <i>popup dialog</i> verifikasi judul konten untuk memastikan bahwa pengguna benar-benar ingin menghapus konten.
	28. Aktor mengetik judul konten dengan benar.	29. Aplikasi menghapus konten dari basis data, kemudian menampilkan pesan berhasil dihapus.

<b>Alternative Course(s)</b>	<p>Alternatif 1:</p> <p>26a. Aktor mengeklik tombol tidak.</p> <p>27a. Aplikasi menutup <i>popup</i> notifikasi peringatan penghapusan konten.</p> <p>Alternatif 2:</p> <p>28b. Aktor mengetik nama konten dengan salah.</p> <p>29b. Aplikasi menampilkan eror verifikasi nama konten dan mematikan tombol hapus di <i>popup</i> dialog verifikasi nama konten.</p> <p>Alternatif 3:</p> <p>14c. Aktor menghapus semua data jadwal.</p> <p>15c. Aplikasi menampilkan pesan eror untuk meminta aktor agar mengisi semua data yang diperlukan.</p>
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses pembuatan jadwal VLT, pengeditannya, dan penghapusannya, serta aplikasi dapat menjalankan tiga alternatif skenario yang sudah dijelaskan.
<b>Post-Condition(s)</b>	Penambahan konten jadwal VLT telah dihapus.
<b>Implementation Constraint(s) and Specification(s)</b>	Komponen pengelola konten jadwal VLT.

Tabel 4.23 Spesifikasi *Use Case* Kelola Akses PIC Modul Training

<b>Use Case Name</b>	Kelola Akses PIC Modul Training	
<b>Actor(s)</b>	Karyawan HC	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola akses PIC terhadap modul <i>training</i> yang telah dibuat.	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam halaman utama aplikasi <i>e-learning</i> HC Portal.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor mengeklik tombol kelola modul di salah satu modul <i>training</i> .	2. Aplikasi menampilkan halaman kelola modul <i>training</i> .



	<p>3. Aktor melihat menu kelola PIC <i>training</i>.</p> <p>5. Aktor memilih satu atau beberapa karyawan untuk dijadikan PIC <i>training</i>.</p> <p>7. Aktor mengeklik tombol simpan.</p> <p>9. Aktor menghapus salah satu PIC <i>training</i> dan mengeklik tombol simpan.</p>	<p>4. Aplikasi menampilkan pengaturan akses PIC <i>training</i>.</p> <p>6. Aplikasi menampilkan nama karyawan pada tampilan daftar PIC.</p> <p>8. Aplikasi menyimpan perubahan aktor ke dalam basis data. Aplikasi juga menampilkan pesan penyimpanan berhasil.</p> <p>10. Aplikasi menyimpan perubahan aktor ke dalam basis data. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.</p>
<i>Alternative Course(s)</i>	Alternatif 1: 5a. Aktor tidak memilih satu PIC modul <i>training</i> atau tidak melakukan perubahan terhadap data PIC modul <i>training</i> . 6a. Tombol simpan tidak dapat diklik.	
<i>Conclusion</i>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses penambahan PIC karyawan terhadap modul <i>training</i> dan menghapusnya, serta menjalankan satu skenario alternatif.	
<i>Post-Condition(s)</i>	Terdapat data PIC karyawan pada menu kelola PIC <i>training</i> .	
<i>Implementation Constraint(s) and Specification(s)</i>	Komponen pengelola PIC <i>training</i> .	

Tabel 4.24 Spesifikasi *Use Case* Kelola Modul *Training*/Kelola Akses Peserta Modul *Training*

<b><i>Use Case Name</i></b>	Kelola Akses Peserta Modul <i>Training</i>
<b><i>Actor(s)</i></b>	PIC <i>Training</i> , Karyawan HC

<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengelola akses peserta <i>training</i> . PIC <i>training</i> dapat menjalankan <i>Use Case</i> ini hanya pada modul <i>training</i> yang diberi akses. Karyawan HC sebagai super administrator memiliki akses ini di semua modul <i>training</i> .		
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam salah satu modul <i>training</i> .		
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>	
	1. Aktor mengeklik menu kelola peserta <i>training</i> .	2. Aplikasi menampilkan pengelola peserta <i>training</i> .	4. Aplikasi menampilkan formulir untuk menambahkan jenis peserta <i>training</i> .
	3. Aktor mengeklik tombol tambah jenis peserta <i>training</i> baru.		
	5. Aktor dapat memilih peserta <i>training</i> berdasarkan divisi, departemen, atau langsung ke karyawan yang spesifik.	6. Aplikasi menyimpan perubahan aktor ke dalam basis data. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.	8. Aplikasi menyimpan perubahan aktor ke dalam basis data. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.
	7. Aktor mengeklik tombol simpan.		
	9. Aktor menghapus salah satu jenis peserta <i>training</i> dan mengeklik tombol simpan.	10. Aplikasi menyimpan perubahan aktor ke dalam basis data. Aplikasi juga menampilkan status penyimpanan perubahan pengguna.	
<b>Alternative Course(s)</b>	Alternatif 1: 5a. Aktor tidak memilih satu jenis peserta <i>training</i> atau tidak melakukan perubahan terhadap data peserta <i>training</i> .		

	6a. Tombol simpan tidak dapat diklik.
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses penambahan peserta terhadap modul <i>training</i> dan menghapusnya, serta menjalankan satu skenario alternatif.
<b>Post-Condition(s)</b>	Terdapat data peserta <i>training</i> pada menu kelola peserta <i>training</i> .
<b>Implementation Constraint(s) and Specification(s)</b>	Komponen pengelola peserta <i>training</i> .

Tabel 4.25 Spesifikasi *Use Case* Akses Laporan *Training*

<b>Use Case Name</b>	Akses Laporan <i>Training</i>	
<b>Actor(s)</b>	PIC <i>Training</i> , Karyawan HC	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengakses laporan <i>training</i> . PIC <i>training</i> dapat menjalankan <i>Use Case</i> ini hanya pada modul <i>training</i> yang diberi akses. Karyawan HC sebagai super administrator memiliki akses ini di semua modul <i>training</i> .	
<b>Pre-Condition(s)</b>	Aktor telah berada di dalam salah satu modul <i>training</i> .	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b> <ul style="list-style-type: none"> <li>1. Aktor mengeklik menu laporan <i>Training</i>.</li> <li>3. Aktor memfilter penampilan laporan <i>training</i>.</li> <li>5. Aktor memilih salah satu data peserta <i>training</i>.</li> </ul>	<b>Application Response</b> <ul style="list-style-type: none"> <li>2. Aplikasi menampilkan laporan <i>training</i>.</li> <li>4. Aplikasi memperbarui tampilan laporan <i>training</i> berdasarkan filter yang dipilih.</li> <li>6. Aplikasi menampilkan detail data progres peserta <i>training</i>.</li> </ul>
<b>Alternative Course(s)</b>	-	
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menampilkan data laporan <i>training</i> , filter datanya, dan menampilkan detail data progres peserta <i>training</i> .	

<b>Post-Condition(s)</b>	Aktor dapat melihat data laporan <i>training</i> sesuai dengan filter yang ia pilih dan detail progres peserta <i>training</i> .
<b>Implementation Constraint(s) and Specification(s)</b>	Pengelola laporan <i>training</i> dan tampilannya.

Tabel 4.26 Spesifikasi Use Case Akses Modul Training

<b>Use Case Name</b>	Akses Modul Training	
<b>Actor(s)</b>	Peserta <i>Training</i> , Pemateri <i>Training</i>	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengakses modul <i>training</i> dengan berbagai tipe konten, seperti teksual dan gambar, video, presentasi dek atau berkas PDF lainnya, kuis dan penilaian akhir, serta jadwal <i>training</i> luring dan daring.	
<b>Pre-Condition(s)</b>	Aktor telah berada di halaman awal <i>e-learning HC Portal</i> .	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b> <ul style="list-style-type: none"> <li>1. Aktor melihat modul <i>training</i> yang dapat ia akses, lalu mengeklik modul tersebut untuk mengaksesnya.</li> <li>3. Aktor mengeklik konten <i>training</i> pertama.</li> <li>5. Aktor membaca konten modul <i>training</i> sampai ke bagian akhir, lalu mengeklik tombol selanjutnya.</li> <li>7. Aktor menonton video materi <i>training</i>, lalu mengeklik tombol selanjutnya.</li> </ul> Application Response	<b>Application Response</b> <ul style="list-style-type: none"> <li>2. Aplikasi menampilkan halaman awal modul <i>training</i> dengan statistik progres aktor.</li> <li>4. Aplikasi menampilkan konten <i>training</i> bertipe teksual dan gambar.</li> <li>6. Aplikasi menandai bahwa aktor telah menyelesaikan konten tersebut. Kemudian, aplikasi mengarahkan aktor ke konten selanjutnya yang bertipe video.</li> <li>8. Aplikasi menandai bahwa aktor telah menyelesaikan konten tersebut. Kemudian, aplikasi mengarahkan aktor ke konten selanjutnya yang</li> </ul>

		bertipe presentasi dek atau berkas PDF lainnya.
	9. Aktor membaca berkas PDF sampai selesai, lalu mengeklik tombol selanjutnya.	10. Aplikasi menandai bahwa aktor telah menyelesaikan konten tersebut. Kemudian, aplikasi mengarahkan aktor ke konten selanjutnya yang bertipe kuis.
	11. Aktor mengeklik tombol mulai.	12. Aplikasi menampilkan soal kuis dan memulai penghitung waktu mundur ( <i>timer</i> ).
	13. Aktor memilih jawaban kuis.	14. Aplikasi menyimpan jawaban aktor dan menyinkronkan waktu penghitung mundur dengan server.
	15. Aktor menyelesaikan kuis sampai sebelum waktu yang telah ditentukan, melakukan reviu jawaban, lalu mengeklik tombol <i>submit</i> .	16. Aplikasi menyimpan jawaban aktor, memeriksa skor yang didapat, menandai aktor sebagai <b>Lulus</b> dalam kuis, menandai bahwa aktor telah menyelesaikan konten, kemudian menampilkan hasilnya kepada aktor.
	17. Aktor melihat hasil skor kuis. Kemudian, aktor mengeklik tombol selanjutnya.	18. Aplikasi menampilkan konten selanjutnya yaitu jadwal <i>training</i> luring.
	19. Aktor melihat jadwal <i>training</i> luring dan mengeklik tombol tambahkan ke Google Calendar.	20. Aplikasi mengarahkan aktor ke halaman web penambahan <i>event</i> kalender baru di Google Calendar.
	21. Aktor menambahkan <i>event</i> ke kalender Google miliknya. Lalu,	22. Aplikasi menampilkan konten selanjutnya

	aktor kembali ke halaman <i>e-learning</i> HC Portal. Kemudian, aktor mengeklik tombol selanjutnya.	yaitu jadwal <i>training</i> daring.
	23. Aktor melihat jadwal <i>training</i> daring dan mengeklik tombol tambahkan ke Google Calendar.	24. Aplikasi mengarahkan aktor ke halaman web penambahan <i>event</i> kalender baru di Google Calendar.
<i>Alternative Course(s)</i>	<p>Alternatif 1:</p> <p>15a. Aktor menyelesaikan kuis sampai waktu berakhir dan tidak sempat mengeklik tombol <i>submit</i>.</p> <p>16a. Aplikasi menyimpan jawaban aktor seadanya. memproses skor yang didapat, menandai aktor sebagai <b>Lulus</b> dalam kuis, menandai bahwa aktor telah menyelesaikan konten, kemudian menampilkan hasilnya kepada aktor.</p> <p>Alternatif 2:</p> <p>16b. Aplikasi menyimpan jawaban aktor, memeriksa skor yang didapat, menandai aktor sebagai <b>Tidak Lulus</b> dalam kuis, kemudian menampilkan hasilnya kepada aktor. Aplikasi juga menampilkan waktu mundur aktor dapat melakukan pengerojan ulang kuis.</p>	
<i>Conclusion</i>	<p><i>Use Case</i> berakhir saat aplikasi berhasil menjalankan seluruh proses penampilan konten modul <i>training</i> bertipe tekstual dan gambar, video, presentasi dek atau berkas PDF lainnya, kuis dan penilaian akhir, serta jadwal <i>training</i> luring dan daring. Aplikasi dapat menandai progres aktor di beberapa konten <i>training</i> kecuali jadwal <i>live training</i> secara luring dan daring. Aplikasi juga berhasil menjalankan alternatif skenario seperti yang telah dijelaskan sebelumnya.</p>	
<i>Post-Condition(s)</i>	Aktor telah menyelesaikan beberapa konten <i>training</i> dengan tipe konten yang berbeda.	
<i>Implementation Constraint(s) and Specification(s)</i>	Komponen pengaksesan <i>training</i> bertipe tekstual dan gambar, video, presentasi dek atau berkas PDF lainnya, kuis dan penilaian akhir, serta jadwal <i>training</i> luring dan daring; Komponen <i>dashboard</i> progres <i>training</i> aktor.	

**Tabel 4.27 Spesifikasi Use Case Akses Forum Training/Menampilkan Posting Forum**

<b>Use Case Name</b>	Akses Forum Training/Menampilkan Posting Forum	
<b>Actor(s)</b>	Peserta <i>Training</i> , Pemateri <i>Training</i> , PIC <i>Training</i> , Karyawan HC	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk mengakses forum <i>training</i> dengan melihat posting forum.	
<b>Pre-Condition(s)</b>	Aktor telah berada di halaman awal <i>e-learning</i> HC Portal.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor melihat modul <i>training</i> yang dapat ia akses, lalu mengeklik modul tersebut untuk mengaksesnya.	2. Aplikasi menampilkan halaman awal modul <i>training</i> dengan statistik progres aktor.
	3. Aktor mengeklik salah satu konten modul <i>training</i> .	4. Aplikasi menampilkan konten modul <i>training</i> .
	5. Aktor mengeklik tombol forum pada tampilan konten modul <i>training</i> .	6. Aplikasi menampilkan forum modul <i>training</i> untuk konten tersebut.
<b>Alternative Course(s)</b>	7. Aktor membuka salah satu posting forum.	8. Aplikasi menampilkan posting forum.
	-	
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses penampilan posting forum modul <i>training</i> .	
<b>Post-Condition(s)</b>	Aktor dapat melihat posting forum untuk konten modul <i>training</i> yang dipilih.	
<b>Implementation Constraint(s) and Specification(s)</b>	Halaman penampil posting forum.	

**Tabel 4.28 Spesifikasi Use Case Akses Forum Training/Balasan Posting Forum**

<b>Use Case Name</b>	Akses Forum Training/Balasan Posting Forum
----------------------	--

<b>Actor(s)</b>	Peserta Training, Pemateri Training, PIC Training, Karyawan HC.	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk membuat, mengedit, dan menghapus balasan posting forum modul <i>training</i> .	
<b>Pre-Condition(s)</b>	Aktor telah menjalankan <i>Use Case</i> Akses Forum <i>Training</i> /Menampilkan Posting Forum.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor mengetik balasan pada kotak formulir yang telah tersedia di bawah posting forum <i>training</i> . Lalu, aktor mengeklik tombol <i>submit</i> .	2. Aplikasi menyimpan balasan aktor dan menampilkannya di bawah posting forum <i>training</i> .
	3. Aktor mengeklik tombol kebab menu (⋮), lalu memilih menu edit balasan.	4. Aplikasi mengubah tampilan balasan menjadi pengedit teks.
	5. Aktor mengedit teks balasan forum, lalu ia mengeklik tombol simpan.	6. Aplikasi menyimpan perubahan balasan forum, menutup pengedit teks, dan menampilkan perubahannya.
	7. Aktor mengeklik tombol kebab menu (⋮), lalu memilih menu hapus balasan.	8. Aplikasi menampilkan jendela <i>popup</i> peringatan terkait penghapusan balasan forum.
	9. Aktor mengeklik tombol ya.	10. Aplikasi menampilkan jendela <i>popup</i> untuk verifikasi penghapusan balasan forum.
	11. Aktor mengetik verifikasi penghapusan balasan forum dengan benar.	12. Aplikasi mengaktifkan tombol hapus pada jendela <i>popup</i> verifikasi.

	<p>13. Aktor mengeklik tombol hapus.</p> <p>14. Aplikasi mengganti teks balasan sebagai “balasan telah dihapus oleh penulisnya” pada basis data dan tampilan balasan.</p>	
<i>Alternative Course(s)</i>	<p>Alternatif 1:</p> <p>1a. Aktor tidak mengetik balasan pada kotak formulir yang telah tersedia di bawah posting forum <i>training</i>. Lalu, aktor mengeklik tombol <i>submit</i>.</p> <p>2a. Aplikasi menampilkan pesan eror bahwa aktor harus mengisi formulir terlebih dahulu.</p> <p>Alternatif 2:</p> <p>5b. Aktor menghapus teks balasan forum, lalu ia mengeklik tombol simpan.</p> <p>6b. Aplikasi menampilkan pesan eror bahwa aktor harus mengisi formulir terlebih dahulu.</p> <p>Alternatif 3:</p> <p>9c. Aktor mengeklik tombol tidak.</p> <p>10c. Aplikasi menutup jendela <i>popup</i> peringatan terkait penghapusan balasan forum.</p> <p>Alternatif 4:</p> <p>13d. Aktor mengetik verifikasi penghapusan balasan forum dengan salah.</p> <p>14d. Aplikasi tidak mengaktifkan tombol hapus pada jendela <i>popup</i> verifikasi.</p>	
<i>Conclusion</i>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses pembuatan, pengeditan, dan penghapusan balasan forum modul <i>training</i> .	
<i>Post-Condition(s)</i>	Aktor dapat melihat balasan posting forum <i>training</i> -nya dengan teks “balasan telah dihapus oleh pemiliknya” sebagai tanda bahwa balasannya telah dihapus.	
<i>Implementation Constraint(s) and Specification(s)</i>	Pengelola balasan forum <i>training</i> .	

**Tabel 4.29 Spesifikasi Use Case Akses Forum Training/Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul Training**

<b>Use Case Name</b>	Akses Forum <i>Training</i> /Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul <i>Training</i>	
<b>Actor(s)</b>	Peserta <i>Training</i> , Pemateri <i>Training</i> , PIC <i>Training</i> , Karyawan HC.	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk membuat, mengedit, dan menghapus posting forum modul <i>training</i> .	
<b>Pre-Condition(s)</b>	Aktor telah menjalankan <i>Use Case</i> Akses Forum <i>Training</i> /Menampilkan Posting Forum.	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b> <ol style="list-style-type: none"> <li>1. Aktor mengeklik tombol kembali di sebelah judul forum.</li> <li>3. Aktor mengeklik tombol buat posting forum modul <i>training</i> baru yang ditandai ikon tambah (+).</li> <li>5. Aktor mengetik formulir pembuatan posting forum dengan benar, kemudian mengeklik tombol buat posting forum baru.</li> <li>7. Aktor mengeklik posting forum <i>training</i> yang telah ia buat sebelumnya.</li> <li>9. Aktor mengeklik tombol kebab menu (:), lalu memilih menu edit posting.</li> <li>11. Aktor mengedit posting forum <i>training</i>, lalu mengeklik tombol simpan.</li> </ol>	<b>Application Response</b> <ol style="list-style-type: none"> <li>2. Aplikasi menampilkan daftar posting forum <i>training</i>.</li> <li>4. Aplikasi menampilkan jendela <i>popup</i> untuk membuat posting forum baru.</li> <li>6. Aplikasi menyimpan data ke dalam basis data, kemudian menampilkannya di halaman daftar posting forum <i>training</i>.</li> <li>8. Aplikasi menampilkan posting forum <i>training</i> yang dipilih.</li> <li>10. Aplikasi menampilkan pengedit teks posting forum.</li> <li>12. Aplikasi menyimpan perubahan posting forum <i>training</i>, menutup pengedit teks posting forum <i>training</i>, serta menampilkan</li> </ol>

		bahwa perubahan aktor telah disimpan.
	13. Aktor mengeklik tombol kebab menu (:), lalu memilih menu hapus posting.	14. Aplikasi menampilkan jendela <i>popup</i> peringatan terkait penghapusan posting forum.
	15. Aktor mengeklik tombol ya.	16. Aplikasi menampilkan jendela <i>popup</i> untuk verifikasi penghapusan posting forum.
	17. Aktor mengetik verifikasi penghapusan posting forum dengan benar.	18. Aplikasi mengaktifkan tombol hapus pada jendela <i>popup</i> verifikasi.
	19. Aktor mengeklik tombol hapus.	20. Aplikasi mengganti teks posting forum sebagai “posting forum telah dihapus oleh penulisnya” pada basis data dan tampilan balasan.
<i>Alternative Course(s)</i>	<p>Alternatif 1:</p> <p>5a. Aktor tidak mengetik formulir pembuatan posting forum dengan benar, kemudian mengeklik tombol buat posting forum baru.</p> <p>6a. Aplikasi menampilkan pesan eror bahwa aktor harus mengisi formulir pembuatan posting forum dengan benar.</p> <p>Alternatif 2:</p> <p>11b. Aktor menghapus seluruh teks posting forum <i>training</i>, lalu mengeklik tombol simpan.</p> <p>12b. Aplikasi menampilkan pesan eror bahwa posting forum <i>training</i> tidak boleh kosong.</p> <p>Alternatif 3:</p> <p>15c. Aktor mengeklik tombol tidak.</p> <p>16c. Aplikasi menutup jendela <i>popup</i> peringatan terkait penghapusan posting forum.</p> <p>Alternatif 4:</p>	

	<p>17d. Aktor mengetik verifikasi penghapusan posting forum dengan salah.</p> <p>18d. Aplikasi tidak mengaktifkan tombol ya pada jendela <i>popup</i> verifikasi.</p>
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses pembuatan, pengeditan, dan penghapusan posting forum modul <i>training</i> .
<b>Post-Condition(s)</b>	Aktor dapat melihat posting forum <i>training</i> -nya dengan teks “posting forum telah dihapus oleh pemiliknya” sebagai tanda bahwa balasannya telah dihapus.
<b>Implementation Constraint(s) and Specification(s)</b>	Pengelola posting forum <i>training</i> .

**Tabel 4.30 Spesifikasi *Use Case* Akses Forum *Training*/Pemberian *Voting* dan Penandaan Balasan Forum *Training* sebagai Solusi**

<b>Use Case Name</b>	Akses Forum <i>Training</i> /Pemberian <i>Voting</i> dan Penandaan Balasan Forum <i>Training</i> sebagai Solusi.	
<b>Actor(s)</b>	Peserta <i>Training</i> , Pemateri <i>Training</i> , PIC <i>Training</i> , Karyawan HC.	
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk memberikan <i>voting</i> , serta penandaan balasan posting forum sebagai solusi pada posting forum miliknya.	
<b>Pre-Condition(s)</b>	Aktor telah berada di halaman awal forum modul <i>training</i> .	
<b>Typical Course of Event(s)</b>	<b>Actor Action</b>	<b>Application Response</b>
	1. Aktor memilih salah satu posting forum <i>training</i> .	2. Aplikasi menampilkan posting forum <i>training</i> yang dipilih.
	3. Aktor memilih tombol <i>voting</i> ke atas pada salah satu balasan forum <i>training</i> .	4. Aplikasi memperbarui <i>voting</i> balasan <i>training</i> dan menampilkannya bahwa balasan <i>forum</i> mendapatkan satu suara.
	5. Aktor memilih tombol <i>voting</i> ke bawah pada	6. Aplikasi memperbarui <i>voting</i> balasan <i>training</i> dan menampilkannya

	balasan forum <i>training</i> lainnya.	bahwa balasan <i>forum</i> mendapatkan minus satu suara.
	7. Aktor mengeklik tombol kembali untuk ke halaman daftar posting forum <i>training</i> .	8. Aplikasi menampilkan daftar posting forum <i>training</i> .
	9. Aktor memilih posting forum <i>training</i> miliknya.	10. Aplikasi menampilkan posting forum <i>training</i> yang dipilih.
	11. Aktor memilih tombol tandai sebagai solusi dengan ikon centang ( <input checked="" type="checkbox"/> ).	12. Aplikasi menandai balasan sebagai solusi, memperbaruiya pada basis data, lalu menyorot balasan forum <i>training</i> dengan warna latar hijau.
<i>Alternative Course(s)</i>	-	
<i>Conclusion</i>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses pemberian voting dan penandaan balasan forum <i>training</i> sebagai solusi.	
<i>Post-Condition(s)</i>	Aktor dapat melihat posting forum <i>training</i> -nya memiliki satu balasan yang ditandai sebagai solusi. Pada posting forum yang dipilih sebelumnya, posting tersebut memiliki balasan yang mendapat voting +1 dan yang lainnya -1.	
<i>Implementation Constraint(s) and Specification(s)</i>	Pengelola voting dan solusi balasan <i>training</i> .	

Tabel 4.31 Spesifikasi *Use Case* Moderasi Forum *Training*

<b>Use Case Name</b>	Moderasi Forum <i>Training</i>
<b>Actor(s)</b>	PIC <i>Training</i> , Karyawan HC
<b>Description</b>	<i>Use case</i> ini menggambarkan kegiatan untuk melakukan moderasi forum <i>training</i> berupa pengeditan posting forum dan balasan posting forum, serta penghapusannya secara level administrator. PIC <i>training</i> dapat menjalankan <i>Use Case</i> ini hanya pada modul <i>training</i>

	yang diberi akses. Karyawan HC sebagai super administrator memiliki akses ini di semua modul <i>training</i> .	
<i>Pre-Condition(s)</i>	Aktor berada di salah satu posting forum modul <i>training</i> .	
<i>Typical Course of Event(s)</i>	<i>Actor Action</i>	<i>Application Response</i>
	<p>1. Aktor mengeklik tombol kebab menu (:) pada posting forum. Aktor akan melihat beberapa menu tambahan untuk melakukan moderasi posting seperti edit sebagai administrator dan hapus sebagai administrator. Aktor memilih edit sebagai administrator.</p>	<p>2. Aplikasi menampilkan pengedit teks untuk mengedit posting forum dengan pesan peringatan sedang dalam mode administrator.</p>
	<p>3. Aktor mengedit teks posting forum. Kemudian, aktor mengeklik tombol simpan perubahan.</p>	<p>4. Aplikasi menampilkan <i>popup</i> peringatan dan verifikasi perubahan teks posting forum sebagai administrator.</p>
	<p>5. Aktor mengetik alasan perubahan dan mengetik verifikasi dengan benar, lalu aktor mengeklik tombol edit.</p>	<p>6. Aplikasi menyimpan perubahan aktor dan menandai bahwa posting forum diedit oleh administrator, mencadangkan posting orisinal ke basis data, menutup pengedit teks, dan menampilkan perubahannya.</p>
	<p>7. Aktor mengeklik tombol kebab menu (:) pada posting forum. Kemudian, aktor mengeklik menu hapus sebagai administrator.</p>	<p>8. Aplikasi menampilkan <i>popup</i> peringatan dan verifikasi penghapusan posting forum sebagai administrator.</p>
	<p>9. Aktor mengetik alasan penghapusan posting forum dan mengetik verifikasi dengan benar,</p>	<p>10. Aplikasi mencadangkan posting orisinal ke dalam basis data dan mengubah isi teks</p>

	<p>lalu aktor mengeklik tombol hapus.</p>	<p>posting forum menjadi “Posting Dihapus oleh Administrator dengan alasan [alasan dari aktor]”.</p>
	<p>11. Aktor mengeklik tombol kebab menu (:) pada balasan posting forum di bawahnya. Kemudian, aktor mengeklik menu edit sebagai administrator.</p>	<p>12. Aplikasi menampilkan pengedit teks untuk mengedit balasan posting forum dengan pesan peringatan sedang dalam mode administrator.</p>
	<p>13. Aktor mengedit teks balasan posting forum. Kemudian, aktor mengeklik tombol simpan perubahan.</p>	<p>14. Aplikasi menampilkan <i>popup</i> peringatan dan verifikasi perubahan teks balasan posting forum sebagai administrator.</p>
	<p>15. Aktor mengisi alasan perubahan dan mengetik verifikasi dengan benar, lalu aktor mengeklik tombol edit.</p>	<p>16. Aplikasi menyimpan perubahan aktor, menandai bahwa balasan posting forum diedit oleh administrator, mencadangkan balasan posting orisinal ke basis data, menutup pengedit teks, dan menampilkan perubahannya.</p>
	<p>17. Aktor mengeklik tombol kebab menu (:) pada balasan posting forum sebelumnya. Kemudian, aktor mengeklik menu hapus sebagai administrator.</p>	<p>18. Aplikasi menampilkan <i>popup</i> peringatan dan verifikasi perubahan balasan posting forum sebagai administrator.</p>
	<p>19. Aktor mengetik alasan penghapusan balasan posting forum dan mengetik verifikasi dengan benar, lalu aktor mengeklik tombol hapus.</p>	<p>20. Aplikasi mencadangkan balasan posting orisinal ke dalam basis data dan mengubah isi teks posting forum menjadi “Balasan Posting Dihapus oleh Administrator dengan</p>

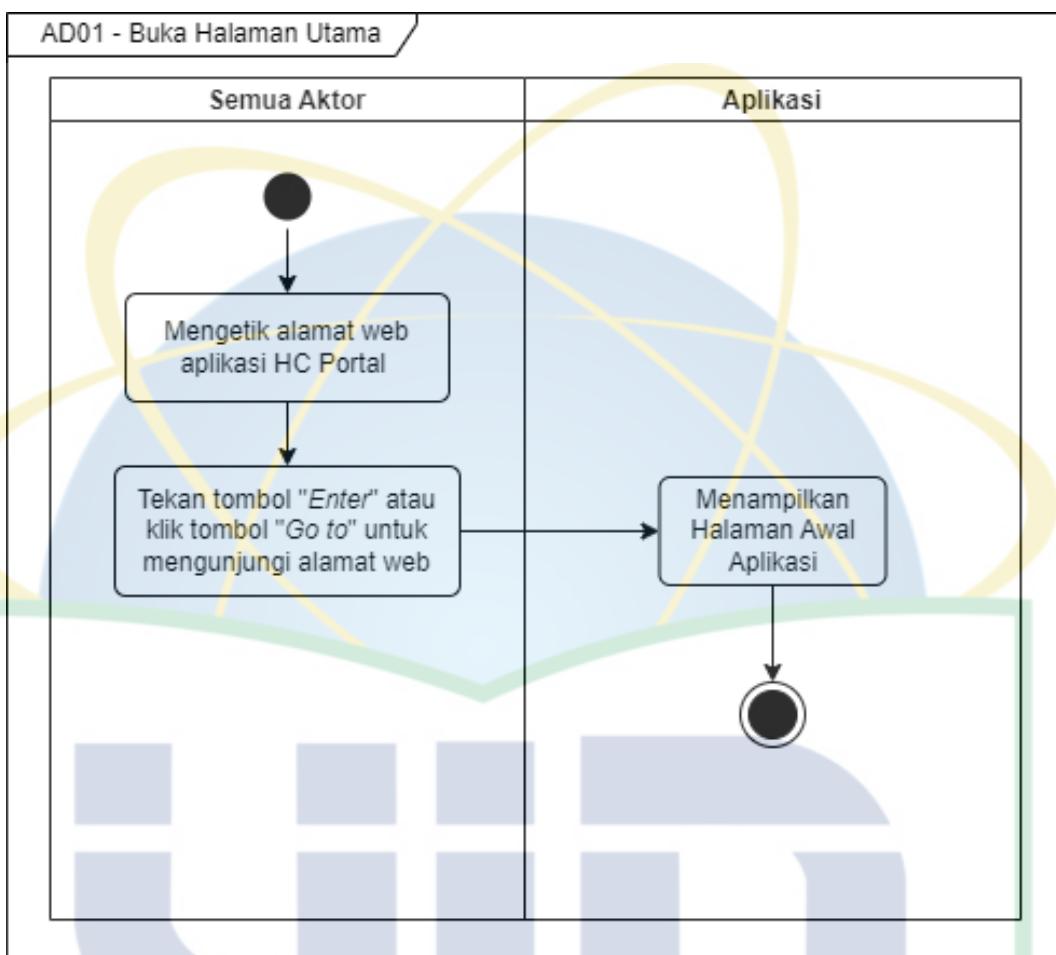
		alasan [alasan dari aktor]".
<i>Alternative Course(s)</i>		
	<p>Alternatif 1:</p> <p>3a. Aktor menghapus teks posting forum. Kemudian, aktor mengeklik tombol simpan perubahan.</p> <p>4a. Aplikasi menampilkan pesan eror bahwa aktor posting forum harus memiliki teks.</p> <p>Alternatif 2:</p> <p>5b. Aktor mengetik alasan perubahan dan mengetik verifikasi dengan salah, lalu aktor mengeklik tombol edit.</p> <p>6b. Aplikasi menampilkan pesan eror bahwa aktor harus mengetik verifikasi dengan benar.</p> <p>Alternatif 3:</p> <p>5c. Aktor mengetik verifikasi dengan benar, lalu aktor mencoba klik tombol edit.</p> <p>6c. Aplikasi menampilkan pesan eror bahwa aktor diwajibkan untuk mengisi alasan perubahan.</p> <p>Alternatif 4:</p> <p>9d. Aktor mengetik alasan penghapusan posting forum dan mengetik verifikasi dengan salah, lalu aktor mengeklik tombol hapus.</p> <p>10d. Aplikasi menampilkan pesan eror bahwa aktor diwajibkan mengetik verifikasi dengan benar.</p> <p>Alternatif 5:</p> <p>9e. Aktor mengetik verifikasi dengan benar, lalu aktor mengeklik tombol hapus.</p> <p>10e. Aplikasi menampilkan pesan eror bahwa aktor diwajibkan mengisi alasan penghapusan posting forum.</p> <p>Alternatif 6:</p> <p>13f. Aktor menghapus teks balasan posting forum. Kemudian, aktor mengeklik tombol simpan perubahan.</p> <p>14f. Aplikasi menampilkan pesan eror bahwa aktor posting forum harus memiliki teks.</p> <p>Alternatif 7:</p>	

	<p>15g. Aktor mengetik alasan perubahan dan mengetik verifikasi dengan salah, lalu aktor mengeklik tombol edit.</p> <p>16g. Aplikasi menampilkan pesan eror bahwa aktor harus mengetik verifikasi dengan benar.</p> <p>Alternatif 8:</p> <p>15h.. Aktor mengetik verifikasi dengan benar, lalu aktor mencoba klik tombol edit.</p> <p>16h. Aplikasi menampilkan pesan eror bahwa aktor diwajibkan untuk mengisi alasan perubahan.</p> <p>Alternatif 9:</p> <p>19i. Aktor mengetik alasan penghapusan posting forum dan mengetik verifikasi dengan salah, lalu aktor mengeklik tombol hapus.</p> <p>20i. Aplikasi menampilkan pesan eror bahwa aktor diwajibkan mengetik verifikasi dengan benar.</p> <p>Alternatif 10:</p> <p>19j. Aktor mengetik verifikasi dengan benar, lalu aktor mengeklik tombol hapus.</p> <p>20j. Aplikasi menampilkan pesan eror bahwa aktor diwajibkan mengisi alasan penghapusan posting forum.</p>
<b>Conclusion</b>	<i>Use Case</i> berakhir saat aplikasi berhasil menjalankan proses moderasi forum seperti pengeditan dan penghapusan posting forum <i>training</i> dan balasannya.
<b>Post-Condition(s)</b>	Aktor dapat melihat balasan posting forum pengguna memiliki teks bahwa telah dihapus oleh administrator, dan salah satu balasan di posting forum tersebut memiliki teks yang serupa.
<b>Implementation Constraint(s) and Specification(s)</b>	Pengelola moderasi posting forum <i>training</i> dan balasannya.

## 2. Activity Diagram

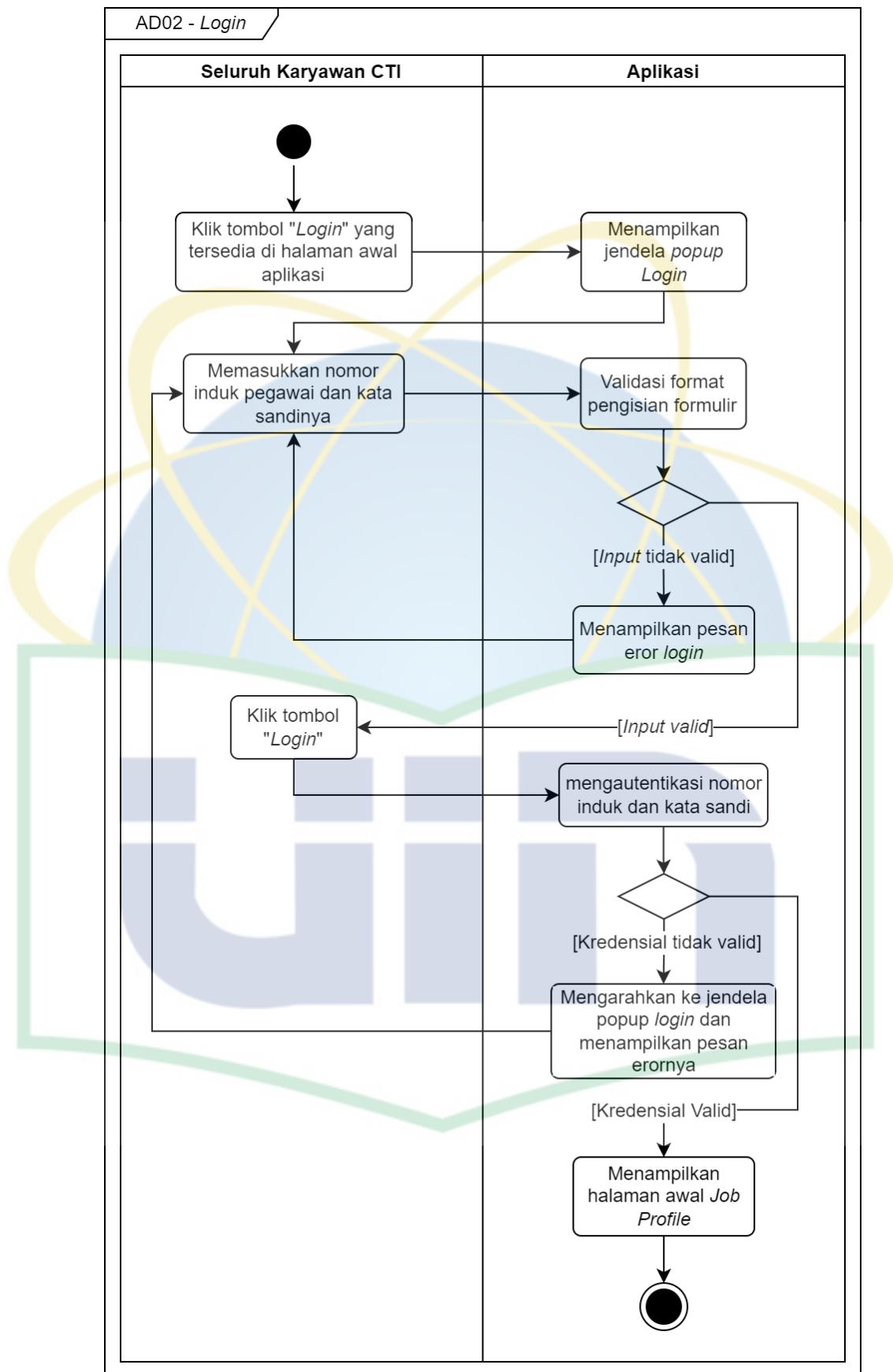
*Activity Diagram* berikut ini dibuat berdasarkan spesifikasi *Use Case* sebelumnya. Diagram-diagram ini menggambarkan alur proses dari aktivitas modul *e-learning* ini.

Aktivitas pertama berikut dapat dilakukan oleh siapa saja yang dapat mengakses alamat web HC Portal. Aktivitas ini menggambarkan bagaimana cara pengguna dapat masuk ke halaman awal aplikasi dan berinteraksi dengan aplikasi HC Portal.



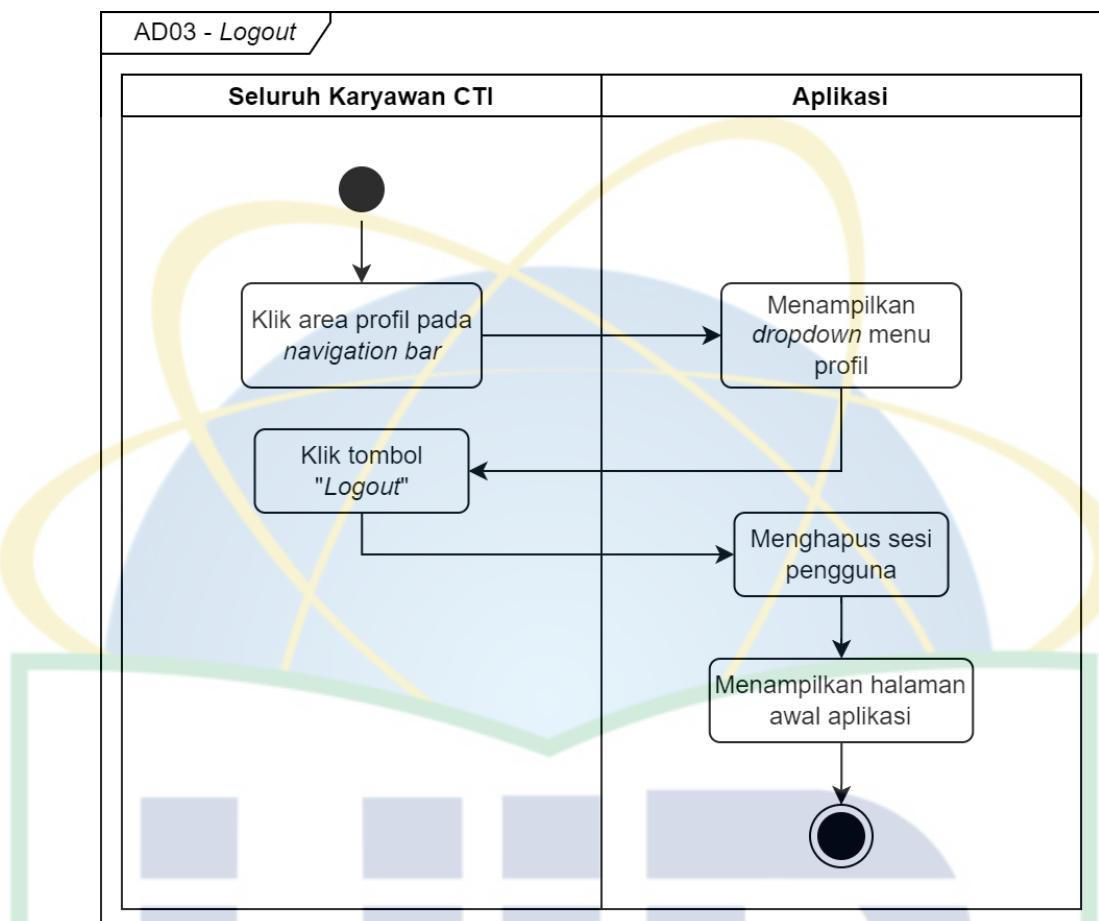
**Gambar 4.27 Activity Diagram Buka Halaman Utama**

Aktivitas berikutnya yaitu *login* dapat diakses setelah aktor dapat masuk ke halaman awal aplikasi. Aktivitas berikut hanya bisa berhasil dijalankan apabila aktor memiliki kredensial untuk mengakses fungsi aplikasi yang sebenarnya.



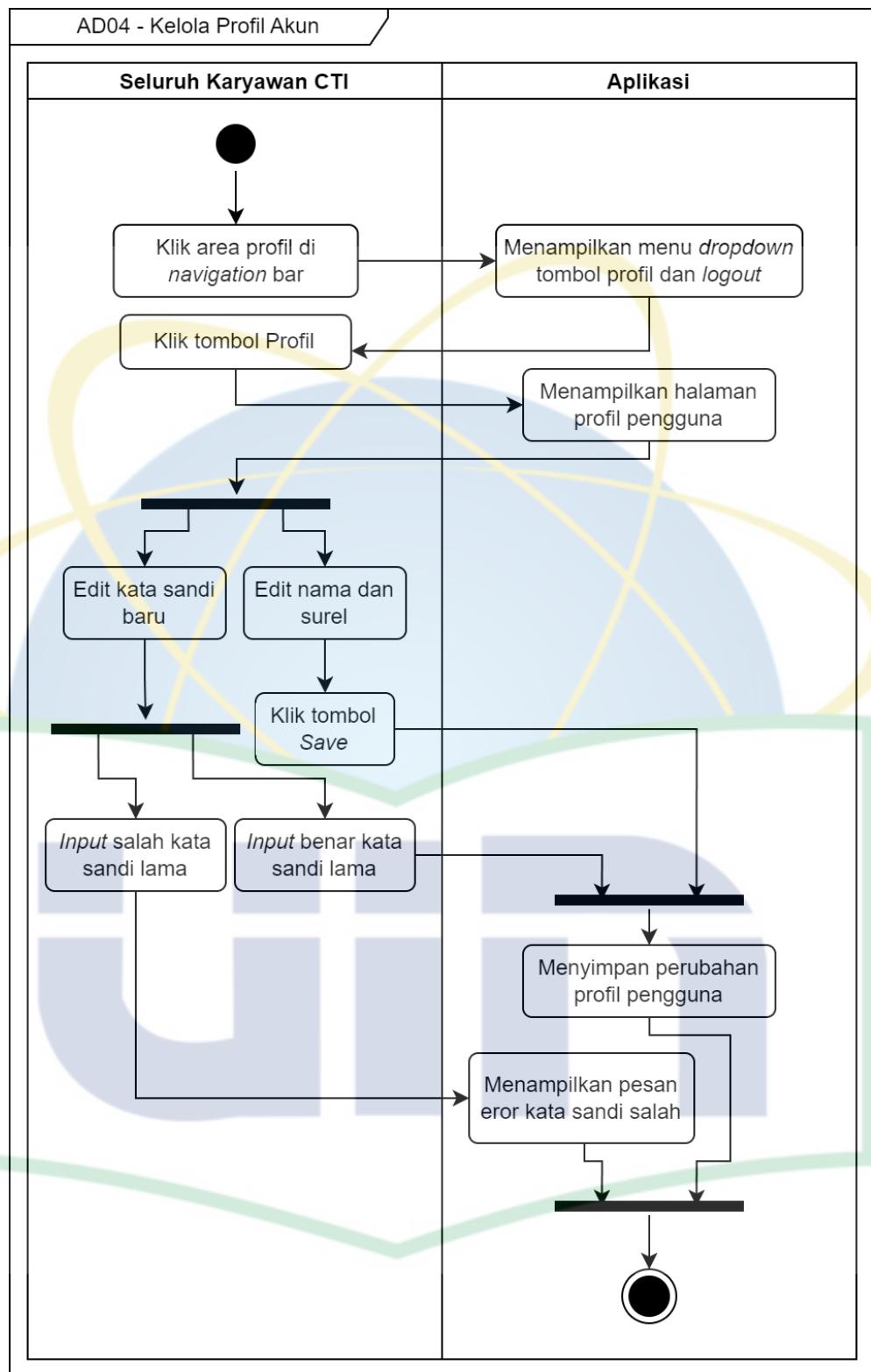
Gambar 4.28 Activity Diagram Login

Aktor yang telah berhasil masuk melalui proses *login* dapat mengakhiri sesinya dengan melakukan proses *logout*. Aktivitas berikutnya menggambarkan proses tersebut.



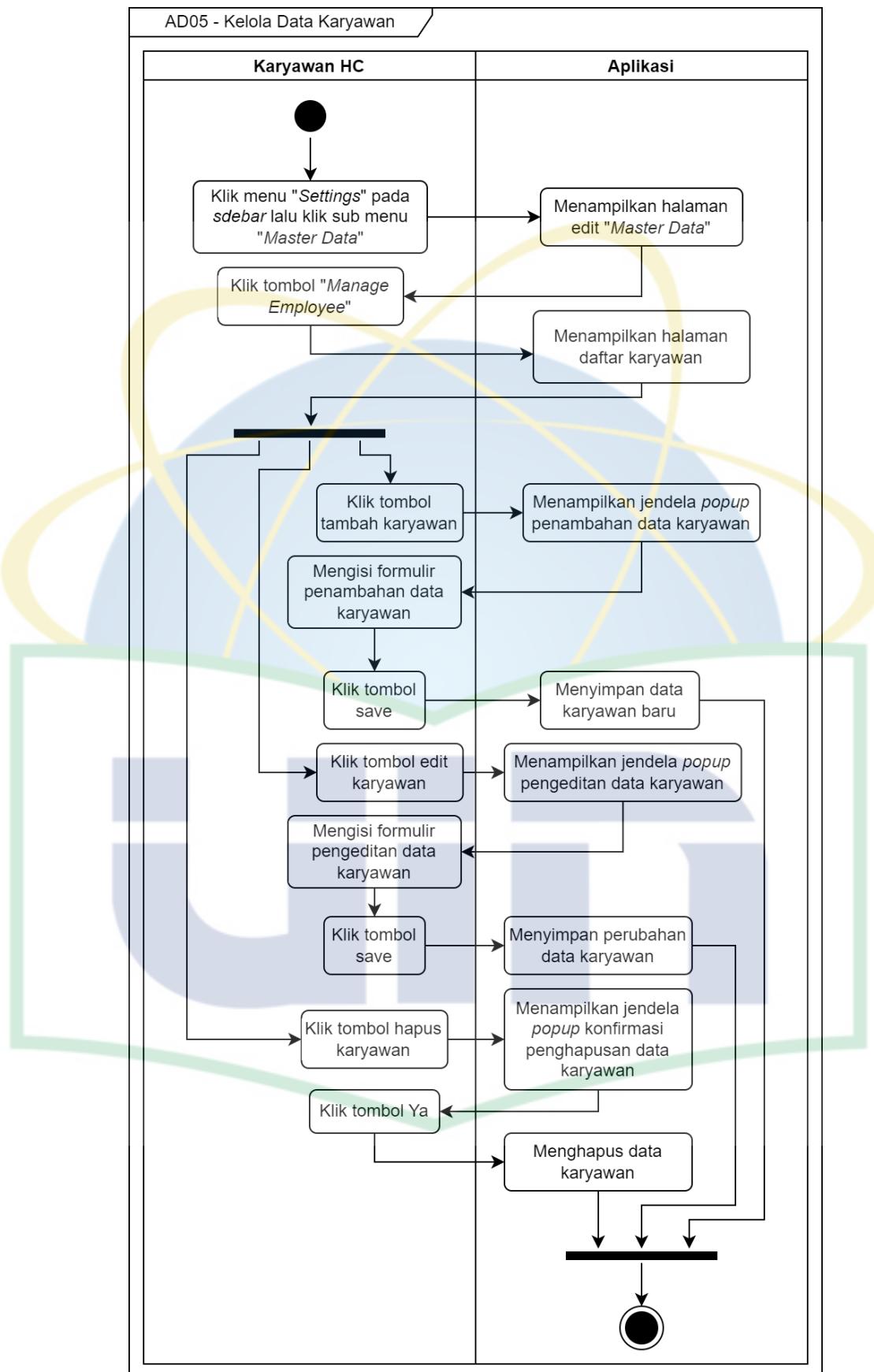
Gambar 4.29 *Activity Diagram Logout*

Aktivitas berikut ini dapat diakses oleh seluruh aktor yang berhasil melaksanakan *login*. Aktivitas ini berfungsi untuk mengelola profil pengguna seperti nama, surel, dan *password*.



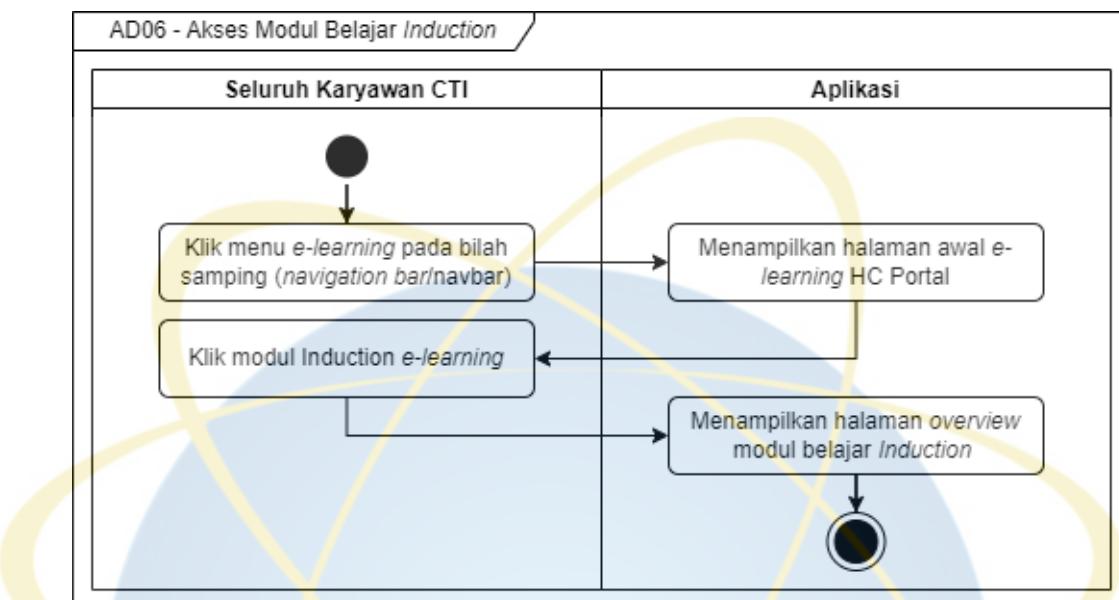
Gambar 4.30 Activity Diagram Kelola Profil Akun

Aktivitas berikutnya berfungsi untuk mengelola data karyawan pada aplikasi. Aktivitas ini hanya dapat diakses oleh *super-admin*. Ia dapat menambahkan data karyawan baru, mengedit data karyawan, maupun menghapus data karyawan.



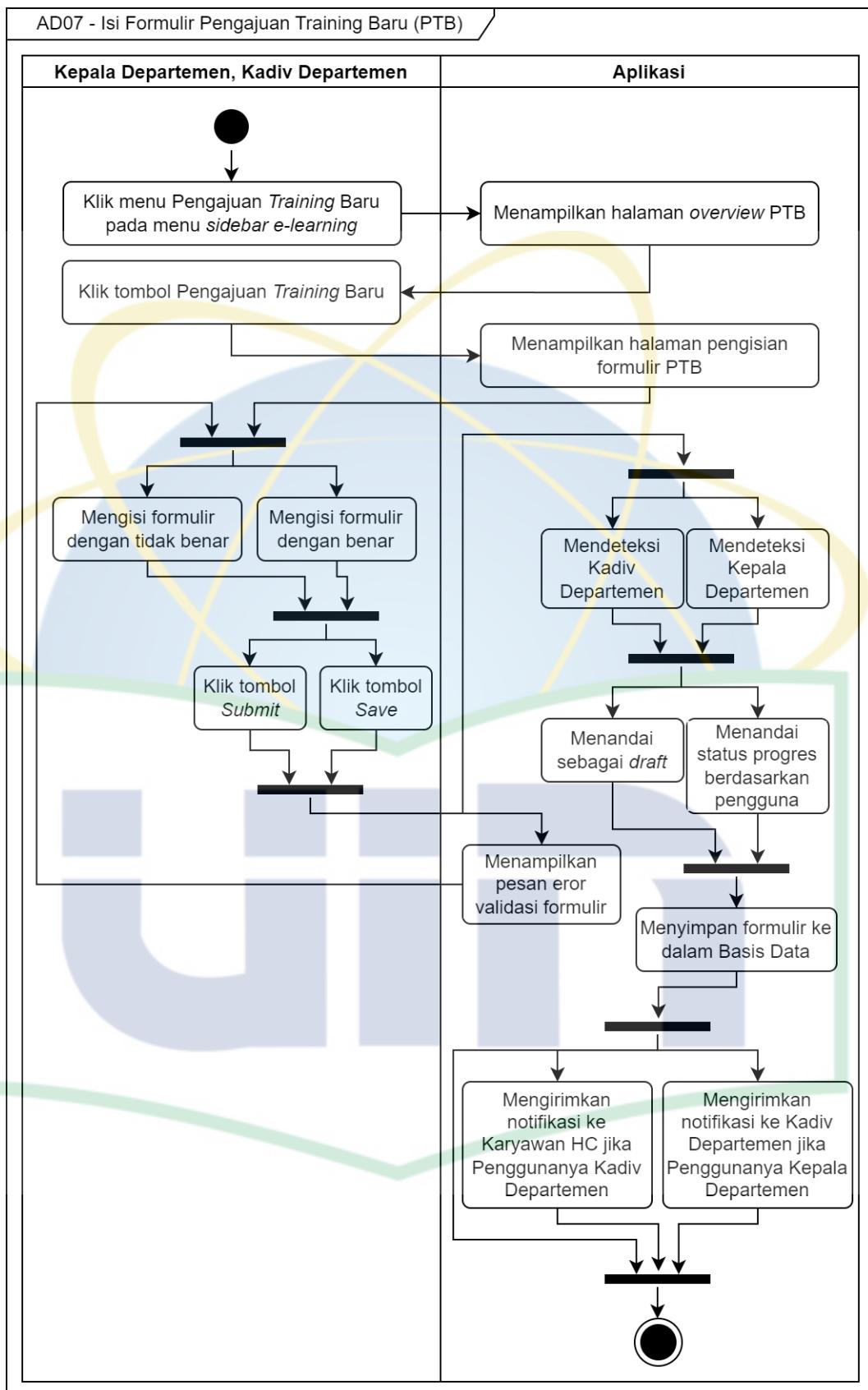
Gambar 4.31 *Activity Diagram Kelola Data Karyawan*

*Activity Diagram* berikutnya berfungsi untuk mengakses modul *Induction* yang dapat diakses oleh seluruh karyawan CTI. Mereka dapat mengaksesnya melalui bilah samping HC Portal.



**Gambar 4.32 Activity Diagram Akses Modul Belajar Induction**

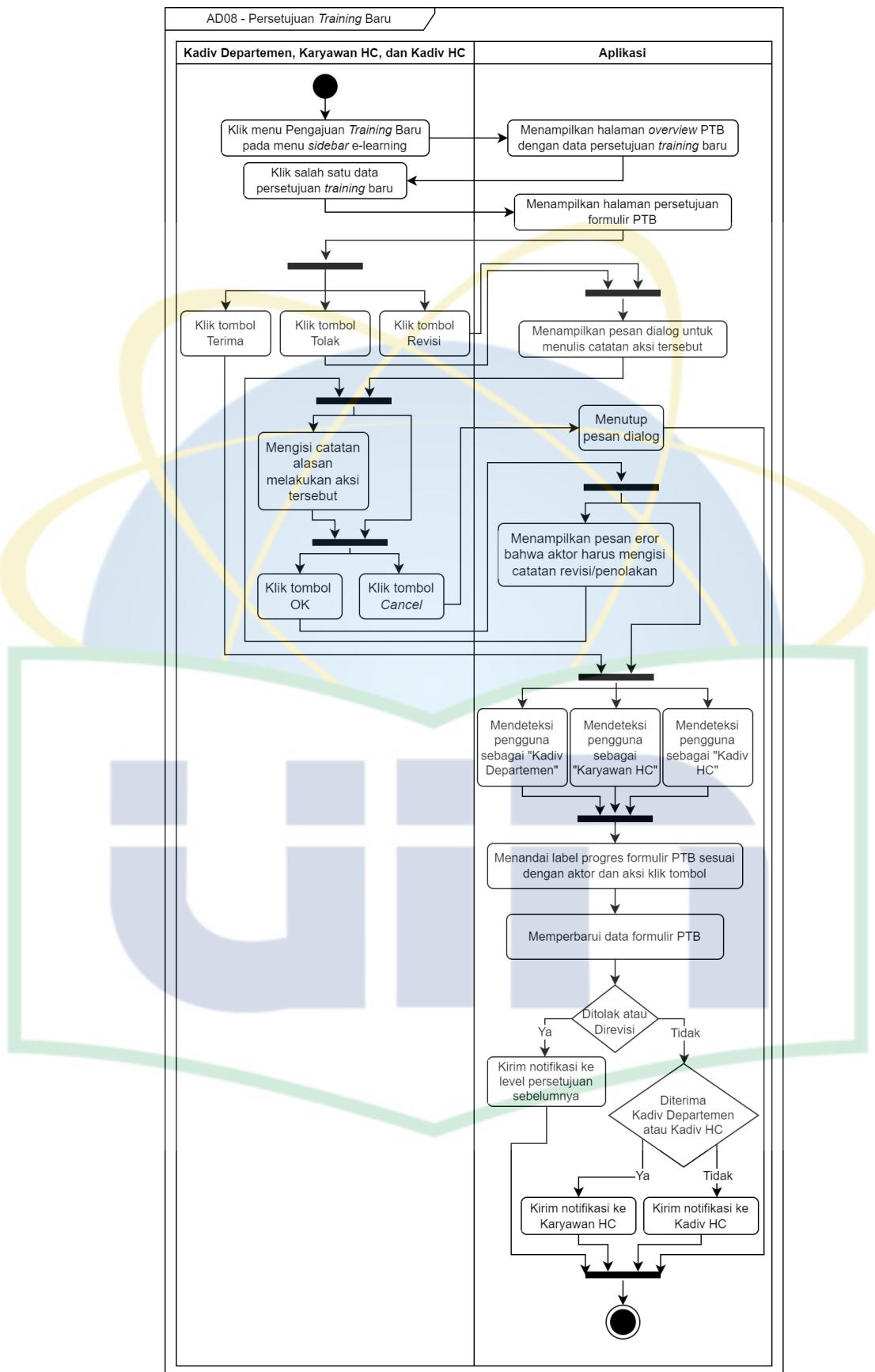
Aktivitas berikut menggambarkan alur pengisian formulir PTB dengan berbagai kondisi, apakah pengguna ingin menyimpannya atau men-submit-nya, mendeteksi aktor apakah Kadiv atau Kepala Departemen yang melakukannya, hingga kondisi validasi formulir. Alur aktivitas akan kembali bila validasi gagal.



**Gambar 4.33 Activity Diagram Isi Formulir Pengajuan Training Baru (PTB)**

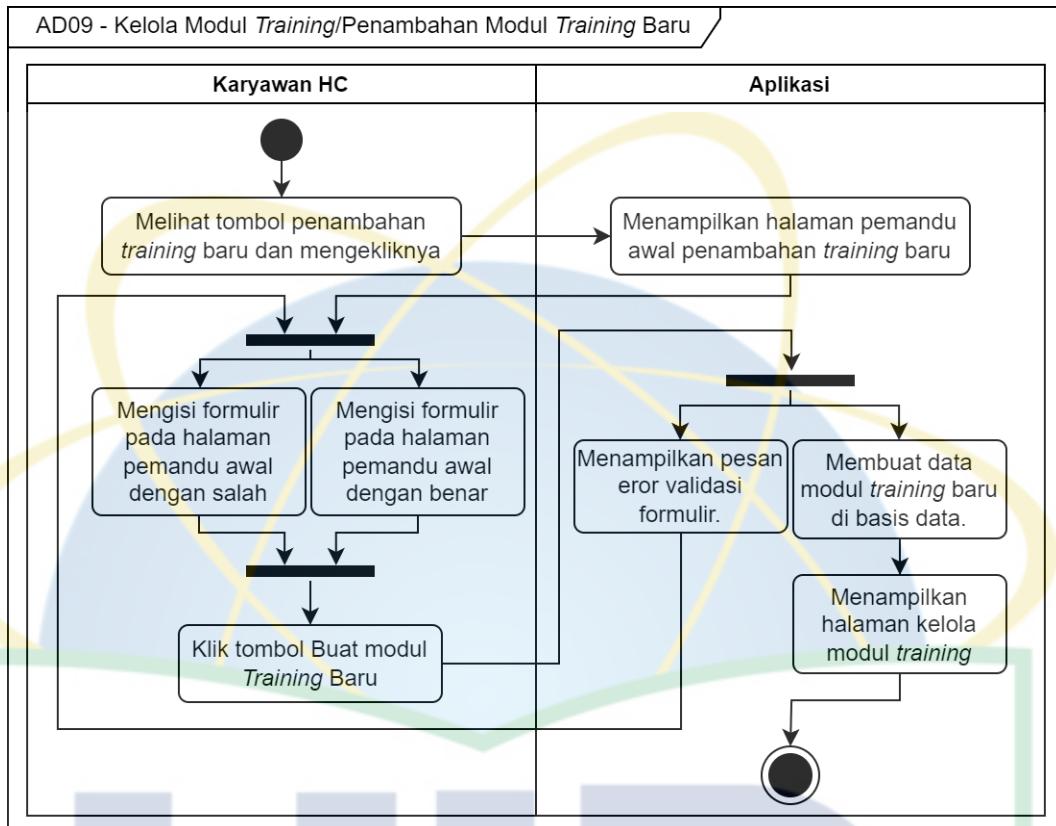
Aktivitas berikutnya menggambarkan alur persetujuan *training* baru. Aktor yang dapat menjalankan aktivitas ini adalah Kadiv Departemen, Karyawan HC, dan Kadiv HC. Berikut alur urutan persetujuannya:

1. Kadiv Departemen mengisi formulir Persetujuan *Training* Baru, kemudian melakukan *submit*. Kemudian, karyawan HC akan mendapatkan notifikasi untuk melakukan validasi formulir.
2. Karyawan HC melakukan validasi formulir. Apabila formulir berhasil melalui proses validasi maka karyawan HC akan menerima formulir tersebut yang mana aplikasi akan mengirimkan notifikasi kepada Kadiv HC. Apabila formulir tidak memenuhi persyaratan validasi karyawan HC maka karyawan HC akan menolaknya atau merevisinya yang mana aplikasi akan mengirimkan notifikasi kepada Kadiv Departemen terkait status persetujuan tersebut. Kemudian, Kadiv Departemen dapat merevisinya bila formulir mendapatkan status revisi dari karyawan HC.
3. Kadiv HC melakukan validasi formulir. Karyawan HC akan menerima pesan notifikasi apabila Kadiv HC menerima formulir persetujuan tersebut untuk memproses proses *training* baru atau merevisi formulir persetujuan tersebut.



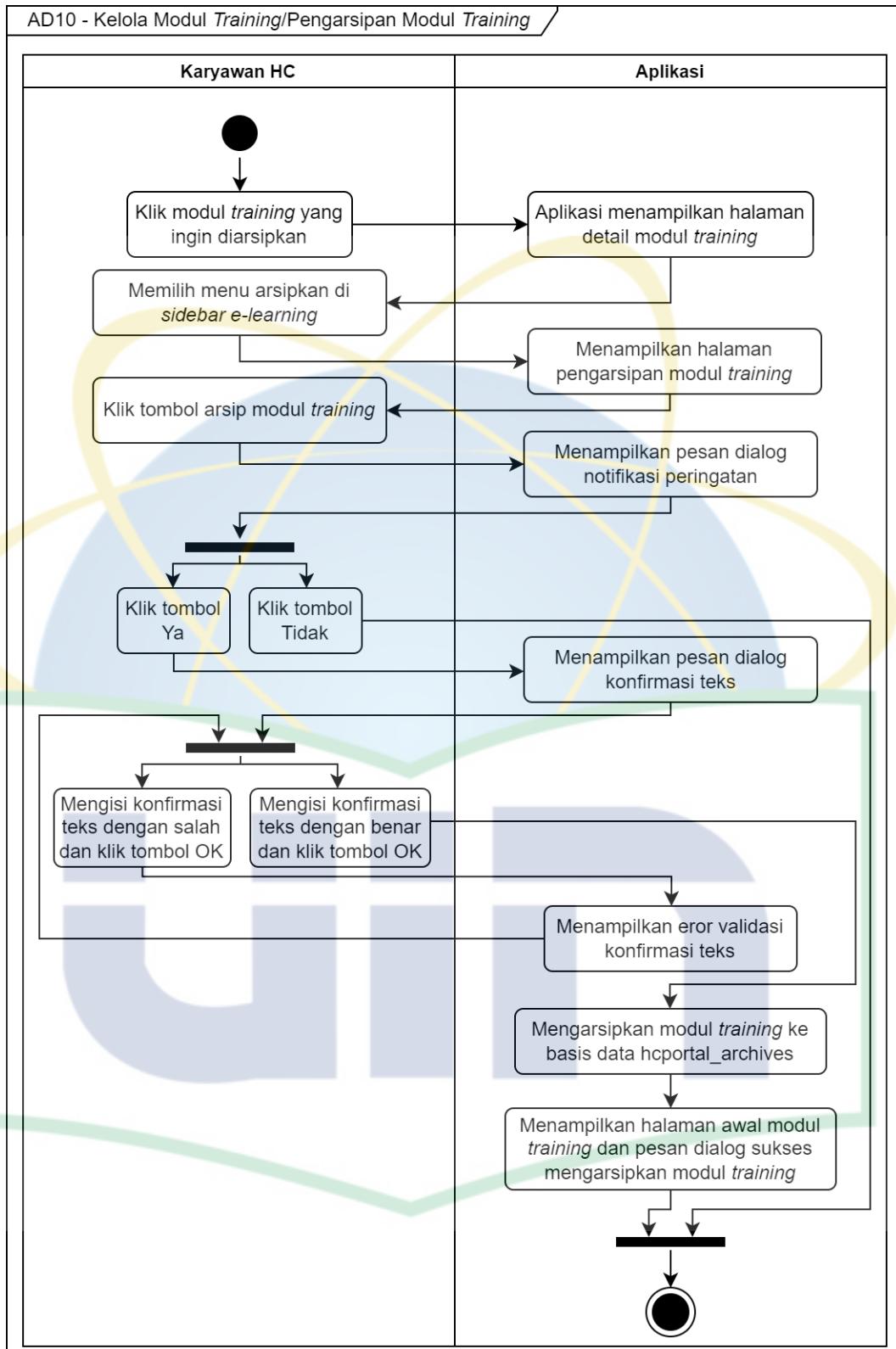
**Gambar 4.34 Activity Diagram Persetujuan Training Baru**

Aktivitas berikutnya berfungsi untuk melakukan penambahan modul *training* baru. Aktivitas ini hanya bisa diakses oleh Karyawan HC yang memegang peran super administrator untuk *e-learning* ini.



Gambar 4.35 Activity Diagram Kelola Modul Training/Penambahan Modul Training Baru

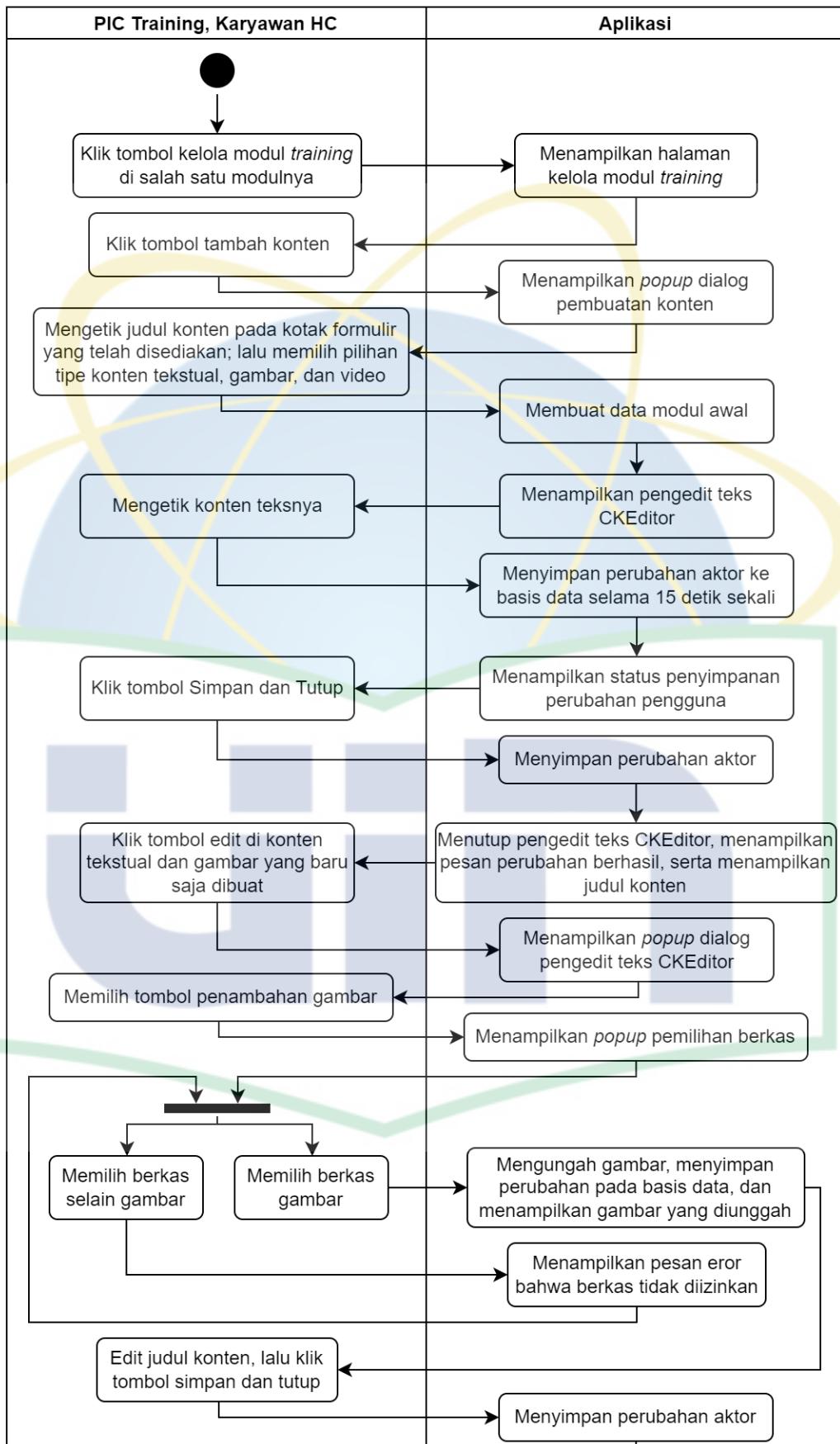
Aktivitas berikutnya merupakan alur untuk melakukan pengarsipan modul *training*. Aktivitas ini juga hanya bisa diakses oleh karyawan HC sebagai pemegang peran super administrator.

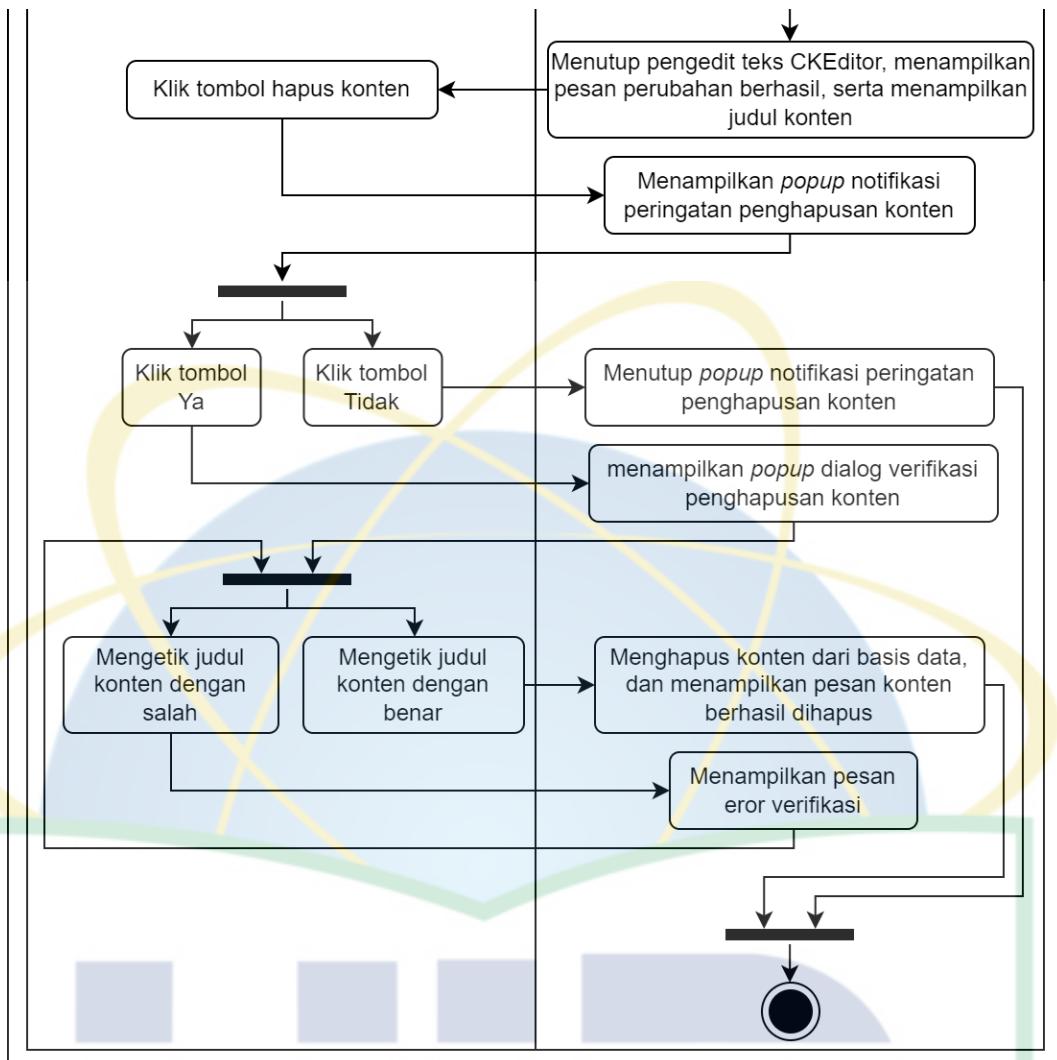


Gambar 4.36 *Activity Diagram Kelola Modul Training/Pengarsipan Modul Training*

Aktivitas berikutnya dibuat untuk mendukung pengelolaan *training* dengan tipe konten *learning*, berbasis teks dan gambar.

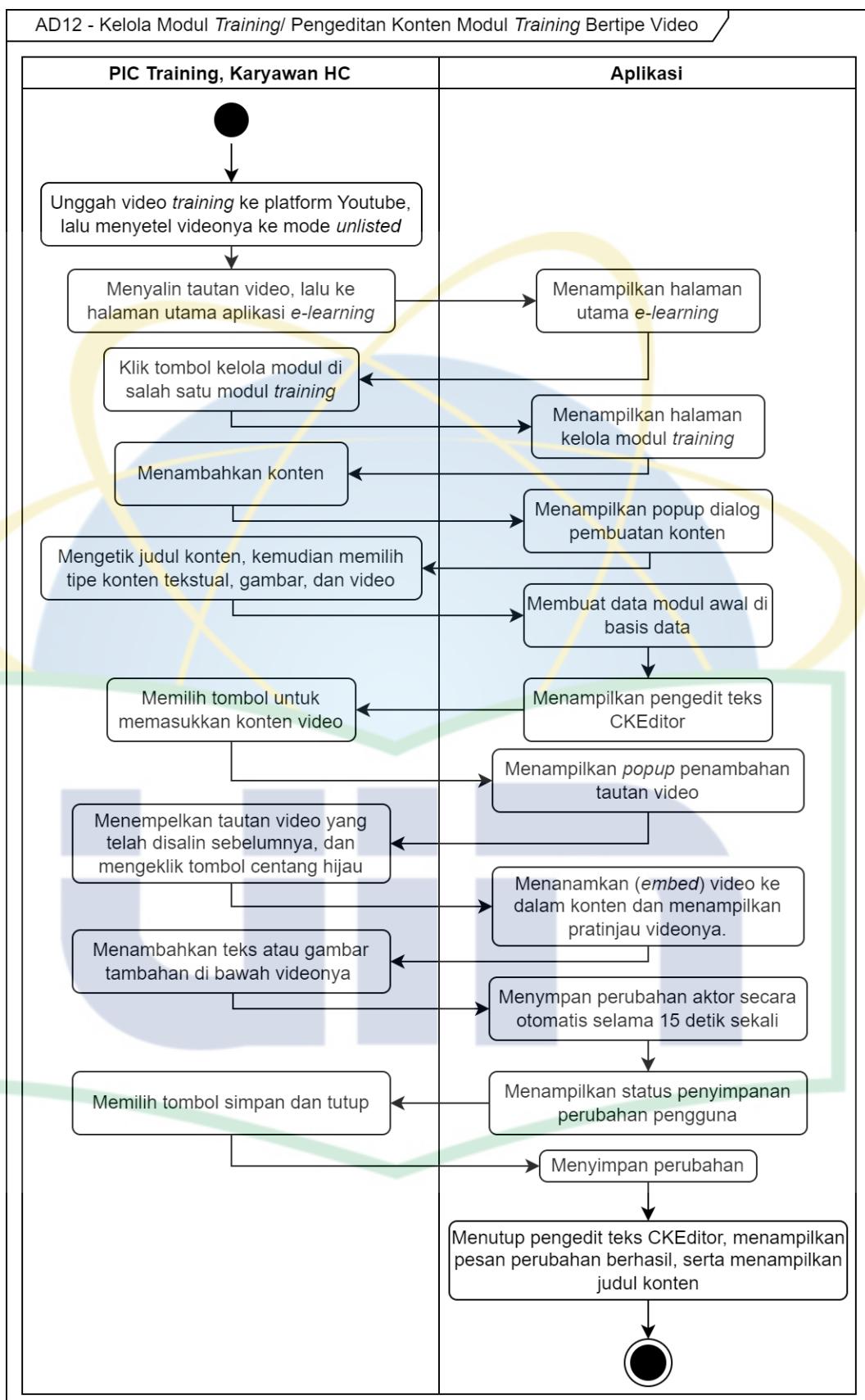
AD11 - Kelola Modul Training/Pengeditan Konten Modul Training Bertipe Tekstual dan Gambar





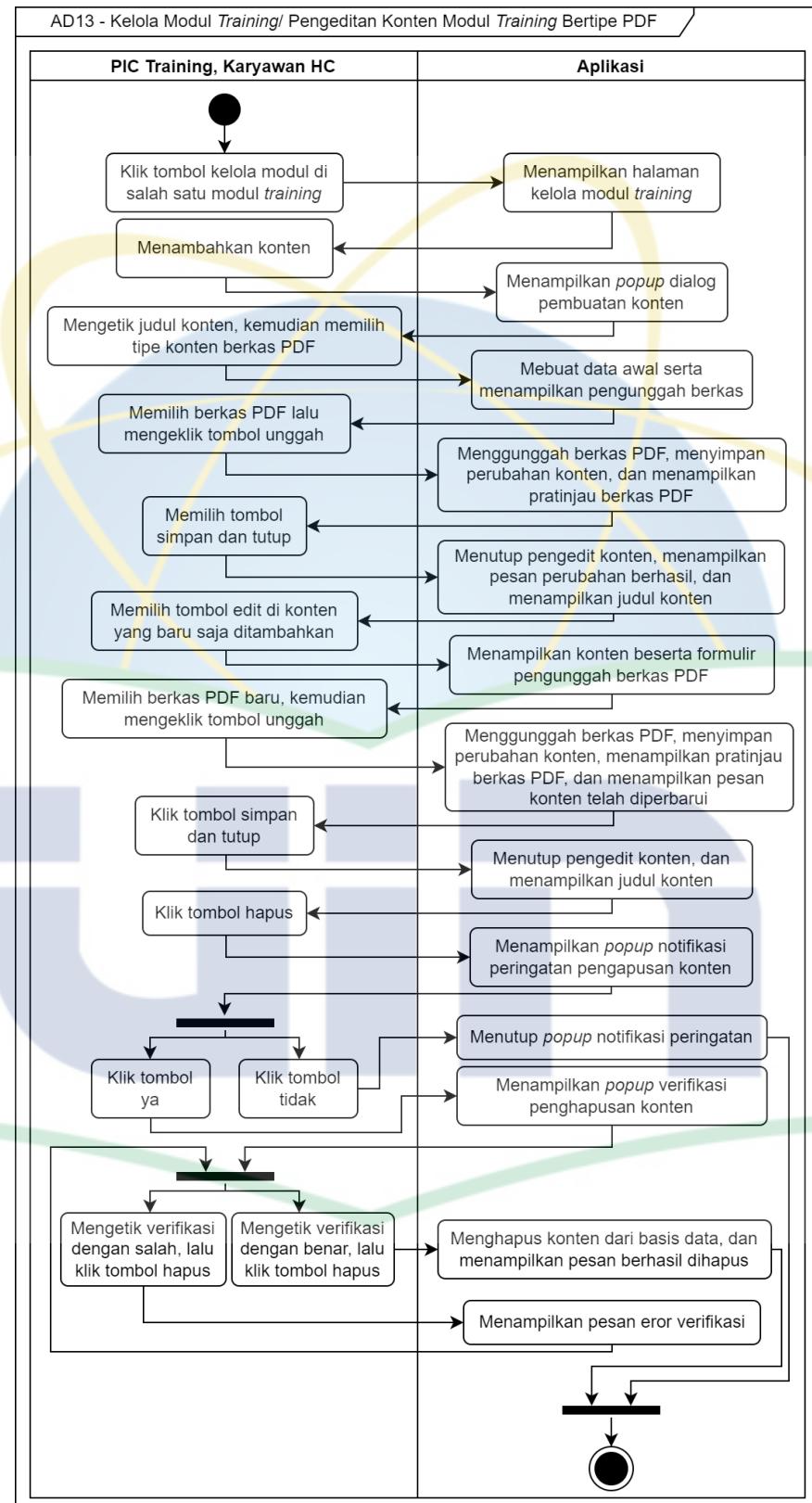
**Gambar 4.37 Activity Diagram Kelola Modul Training/Pengeditan Konten Modul Training Bertipe Tekstual dan Gambar**

Aktivitas berikutnya masih berada di kelola modul *training* yang mana aktivitasnya merupakan pengeditan konten modul *training* bertipe video. Aktivitas ini dan aktivitas sebelumnya dapat diakses oleh PIC *Training* dan Karyawan HC.



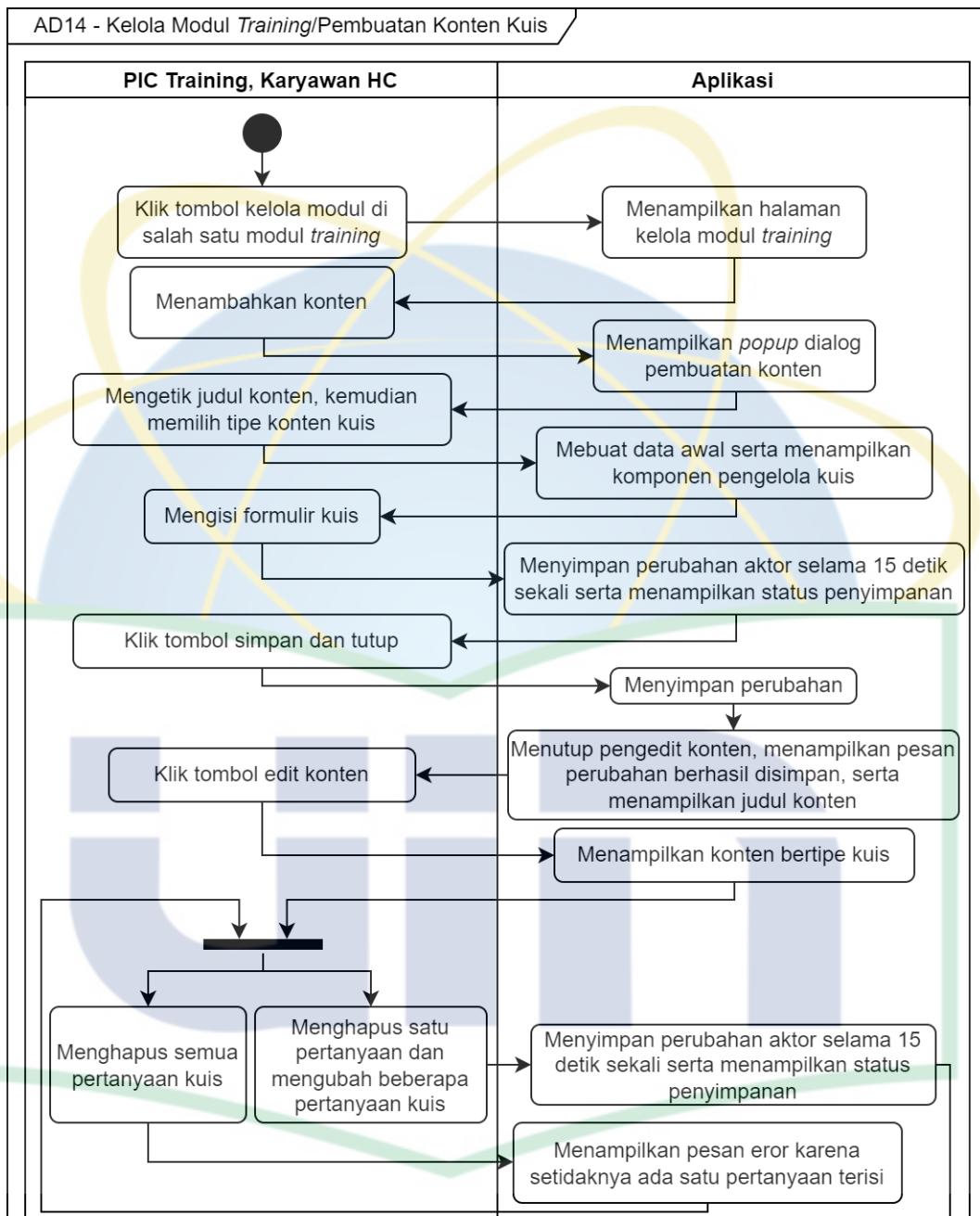
Gambar 4.38 *Activity Diagram Kelola Modul Training/Pengeditan Konten Bertipe Video*

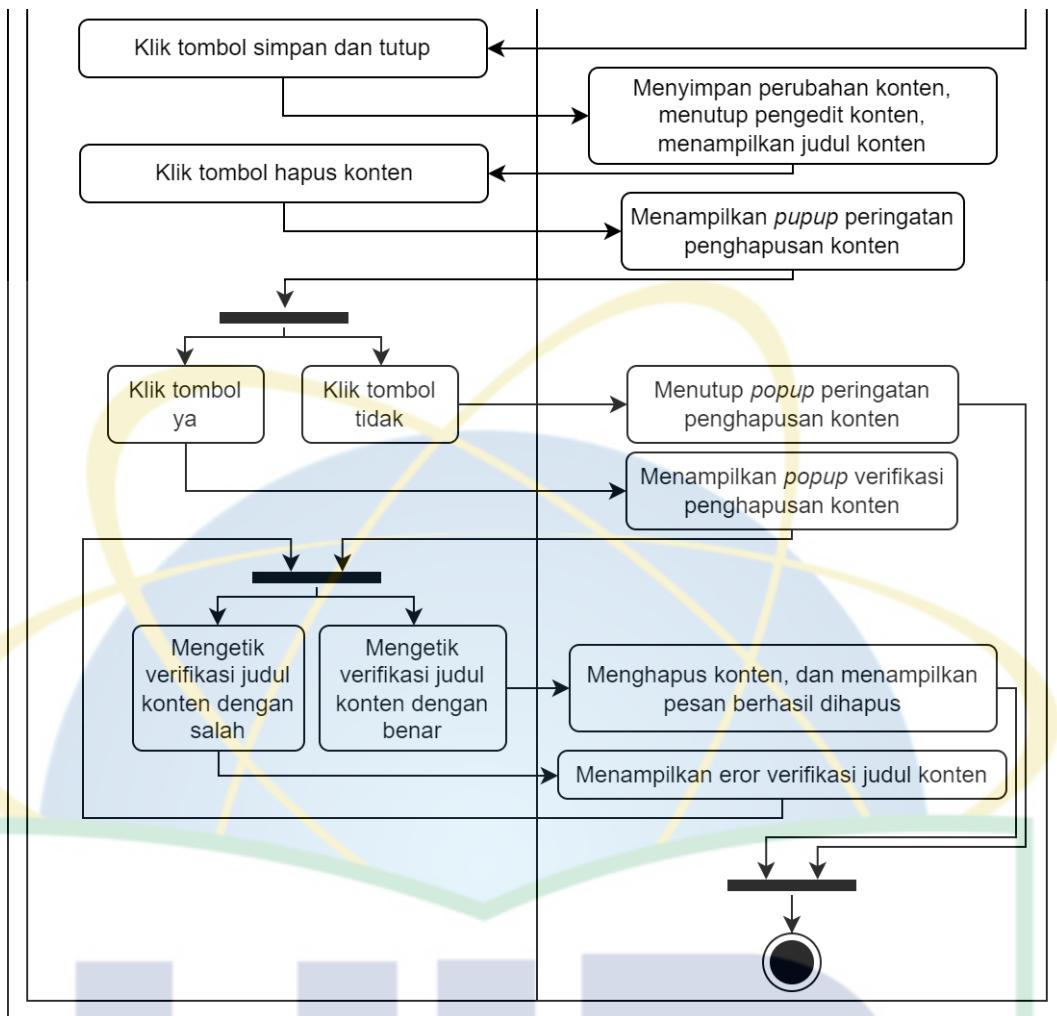
Aktivitas berikutnya juga masih di kelola modul *training* yang mana aktivitasnya merupakan pengeditan konten modul *training* bertipe berkas PDF.



**Gambar 4.39 Activity Diagram Kelola Modul Training/Pengeditan konten bertipe PDF**

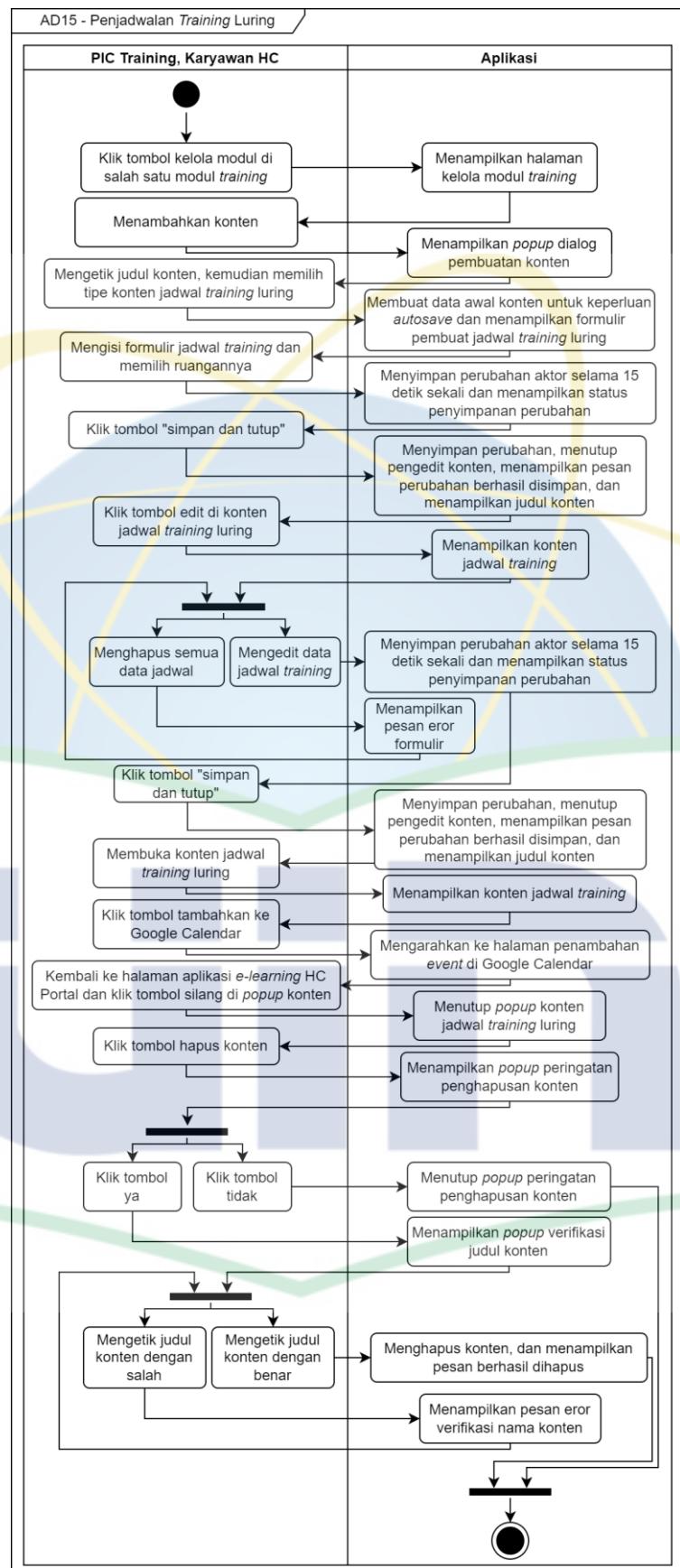
Aktivitas berikutnya masih berada dalam pengelolaan modul *training*. Aktivitas kali ini membahas alur pembuatan konten kuis di *e-learning*. Aktivitas ini dapat diakses oleh PIC *Training* dan Karyawan HC.



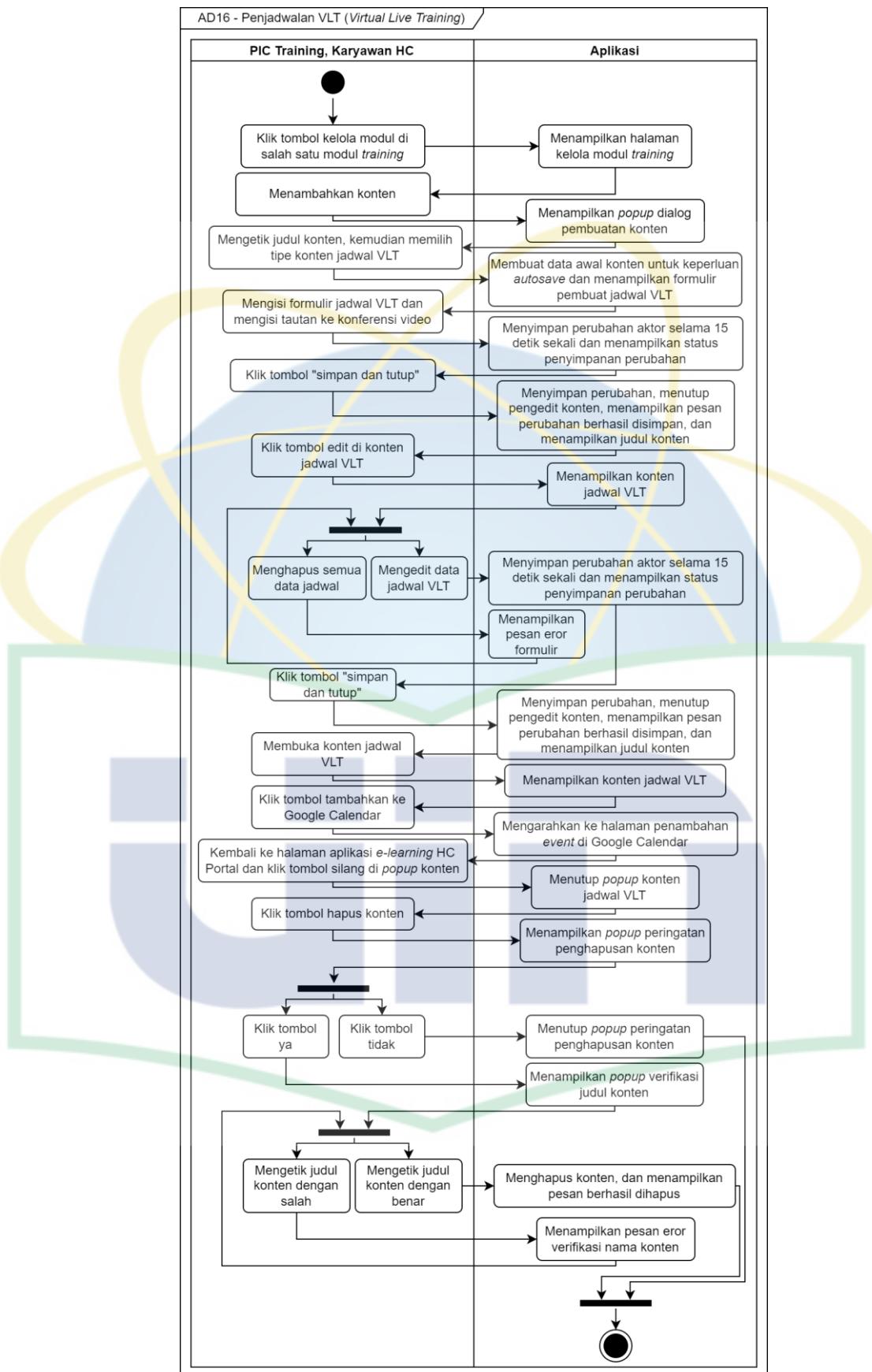


Gambar 4.40 Activity Diagram Kelola Modul Training/Pembuatan Konten Kuis

Aktivitas berikutnya menggambarkan alur pembuatan jadwal *training* luring. Aktivitas ini dapat diakses oleh PIC *Training* dan Karyawan HC. Aktivitasnya meliputi pembuatan jadwal *training*, pengeditannya dan penghapusannya, dan bagaimana pengguna dapat menambahkan jadwal ke Google Calendar miliknya. Kemudian setelahnya, terdapat aktivitas yang memiliki alur yang mirip hanya saja aktivitas tersebut menggambarkan alur pembuatan jadwal VLT. Aktivitas ini juga memiliki akses aktor yang sama.

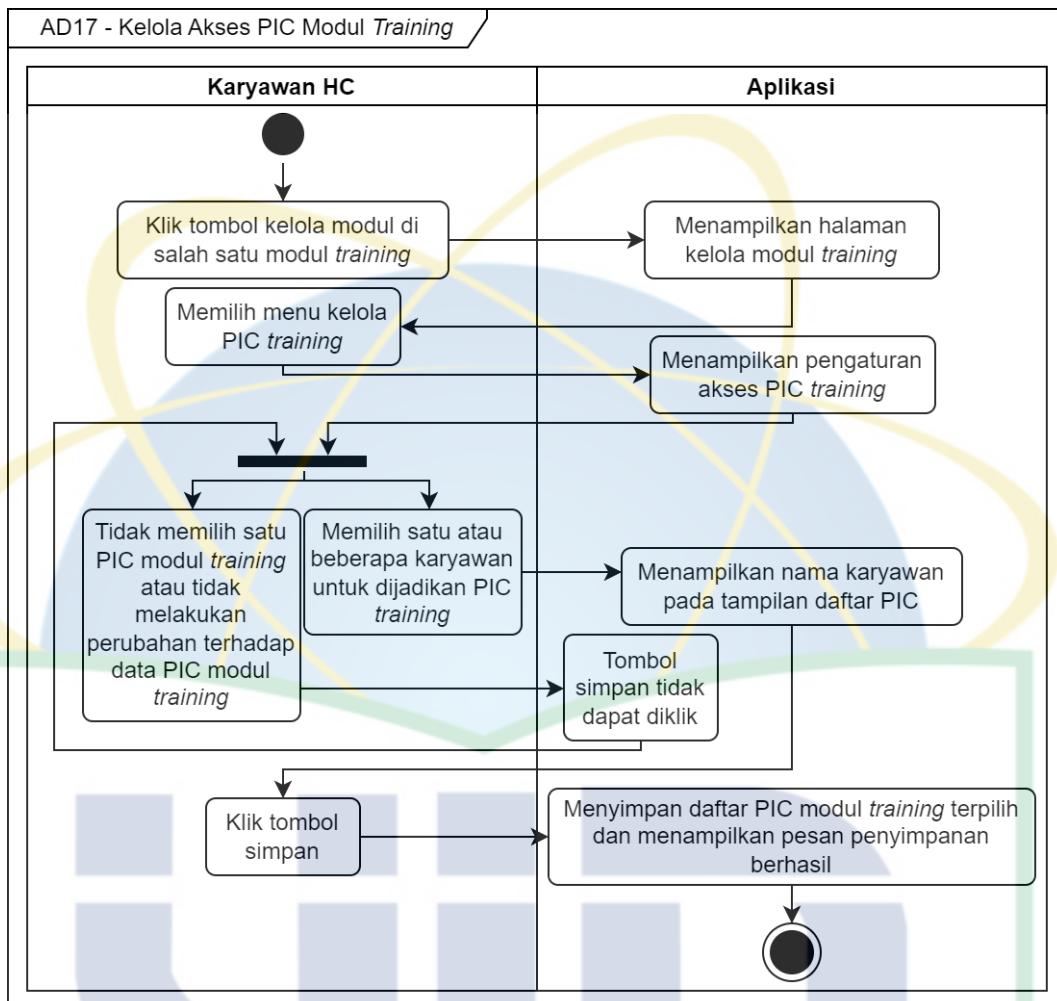


**Gambar 4.41 Activity Diagram Penjadwalan Training Luring**



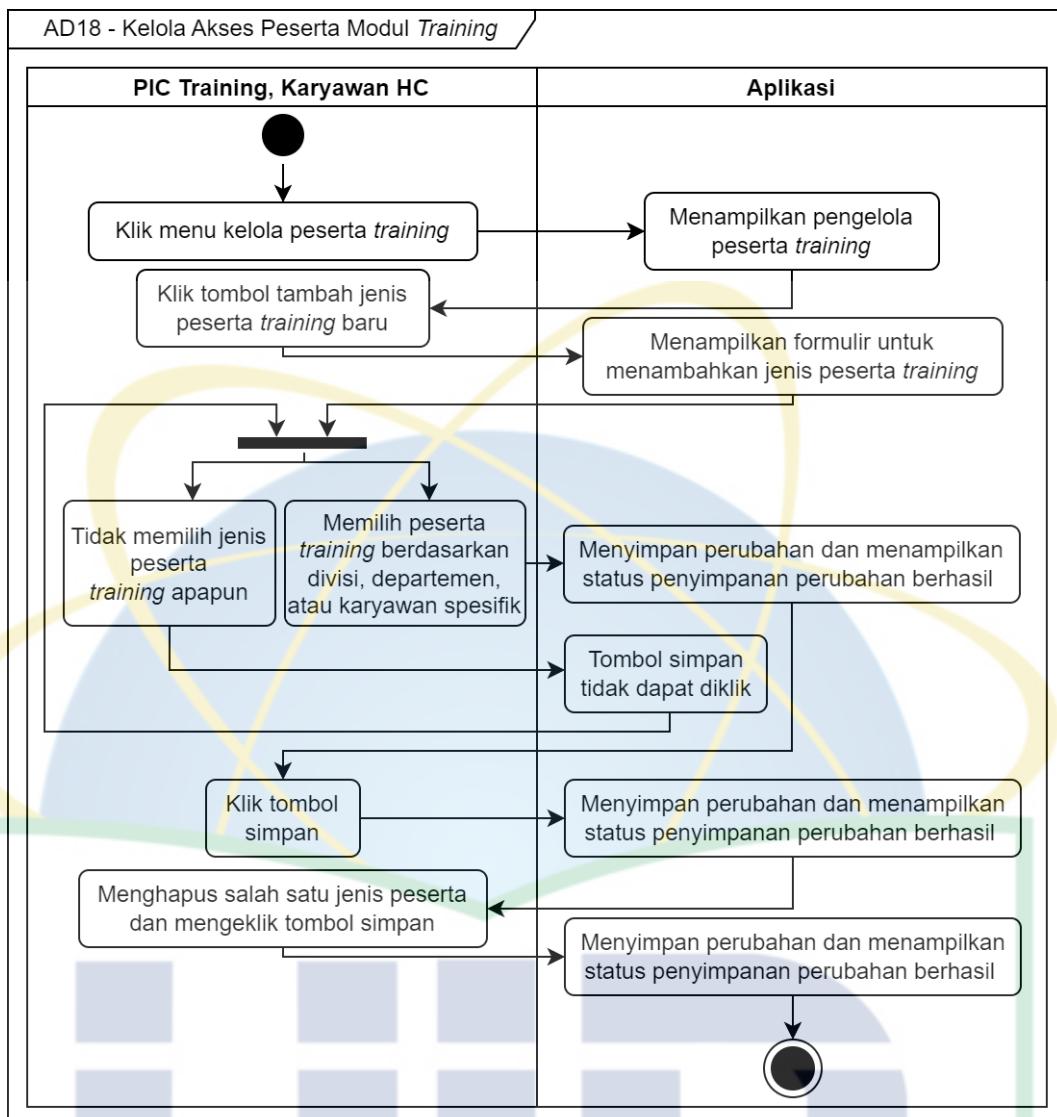
**Gambar 4.42 Activity Diagram Penjadwalan VLT (Virtual Live Training)**

Aktivitas berikutnya menggambarkan alur pengelolaan PIC untuk Modul *Training*. Aktivitas ini hanya bisa diakses oleh karyawan HC yang bertindak sebagai super administrator.



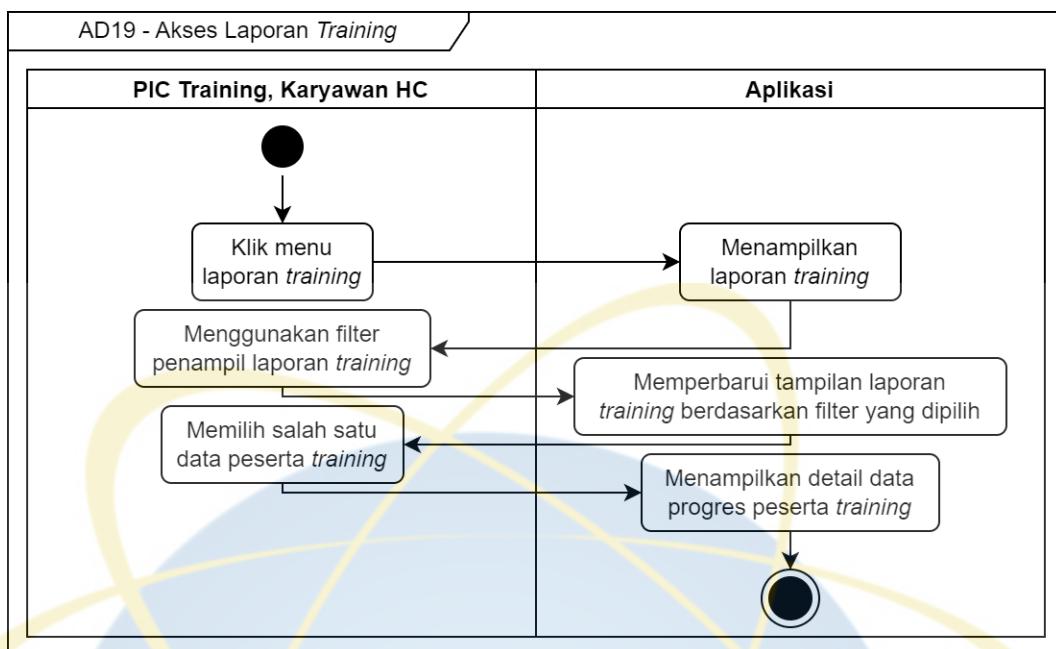
Gambar 4.43 Activity Diagram Kelola Akses PIC Modul Training

Aktivitas berikutnya menggambarkan alur pengelolaan peserta modul *training*. Aktivitas ini dapat diakses oleh PIC *Training* dan Karyawan HC.



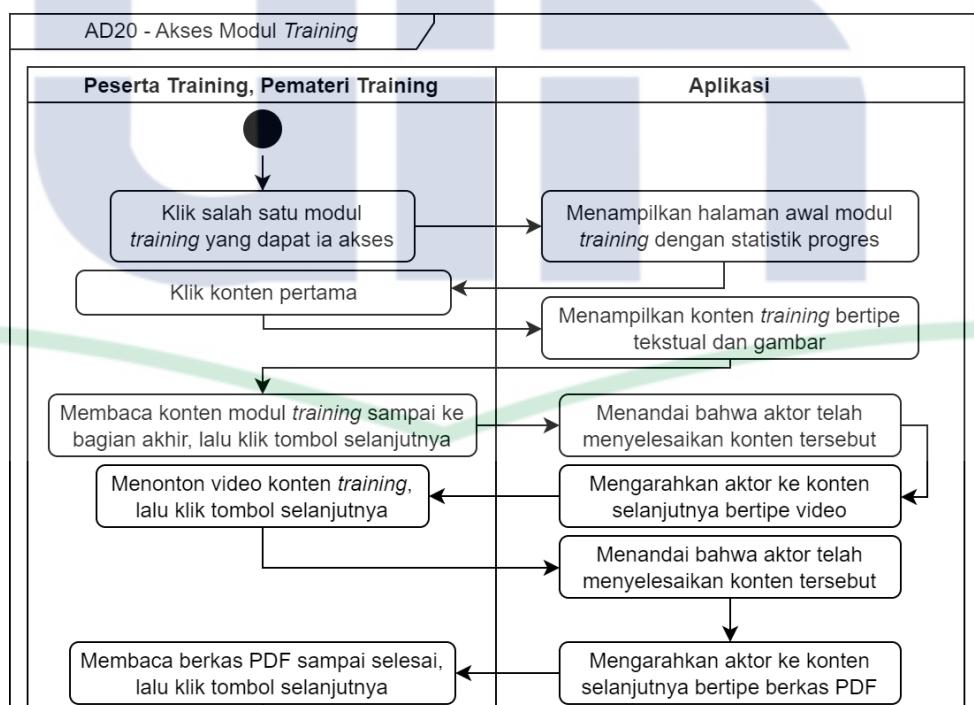
Gambar 4.44 *Activity Diagram Kelola Akses Peserta Modul Training*

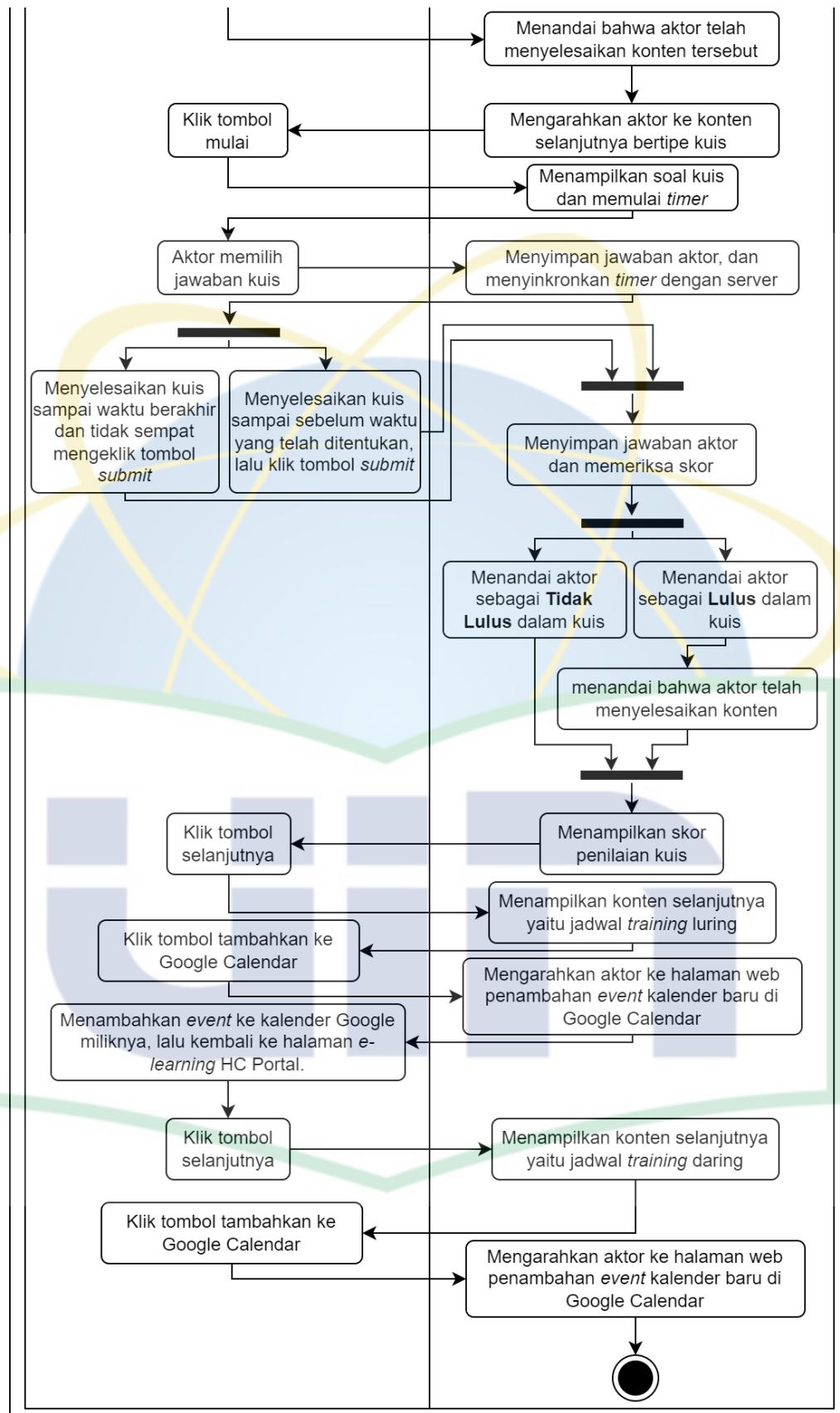
Aktivitas berikutnya menggambarkan alur pengaksesan laporan *training*. Aktivitas ini dapat dijalankan oleh PIC *Training* dan Karyawan HC. Alurnya sederhana untuk menampilkan data laporan *training* dan menggunakan fitur filter data. Alur ini tidak memiliki skenario alternatif.



Gambar 4.45 Activity Diagram Akses Laporan Training

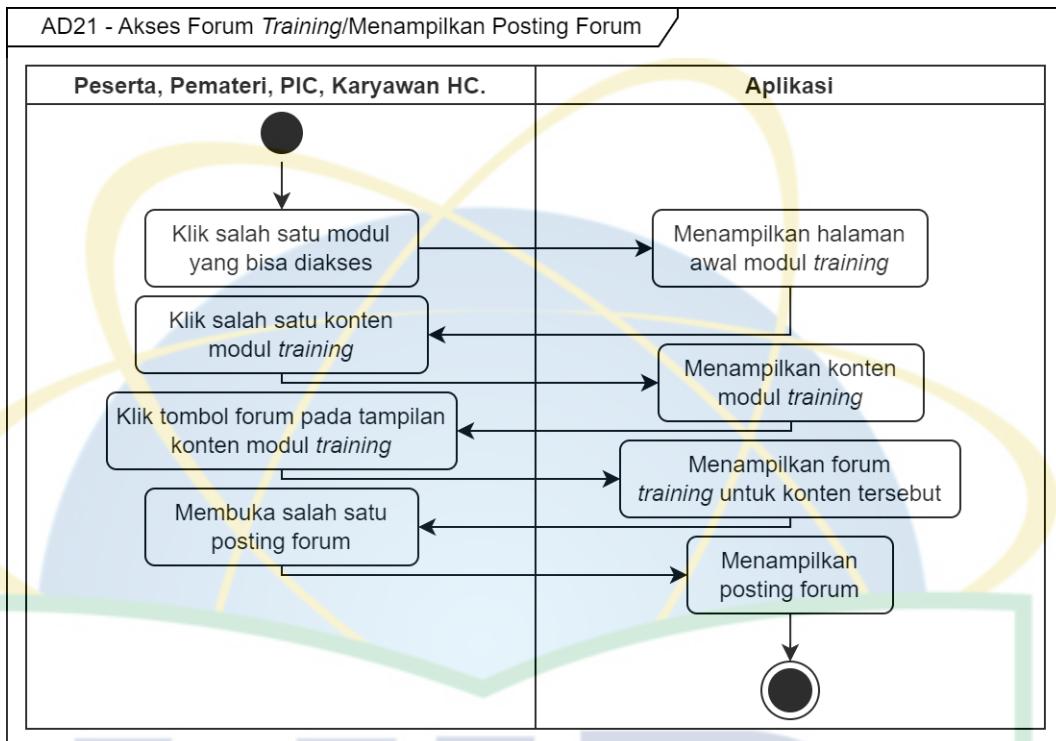
Aktivitas berikutnya menggambarkan alur pengaksesan atau penggerjaan modul *training*. Alur ini mencakup semua jenis konten *training* yang diakses secara berurut mulai dari tipe textual dan gambar, video, berkas PDF, kuis, sampai ke jadwal *training* luring dan daring. Aktivitas ini dapat diakses oleh Peserta *Training* dan Pemateri *Training*.





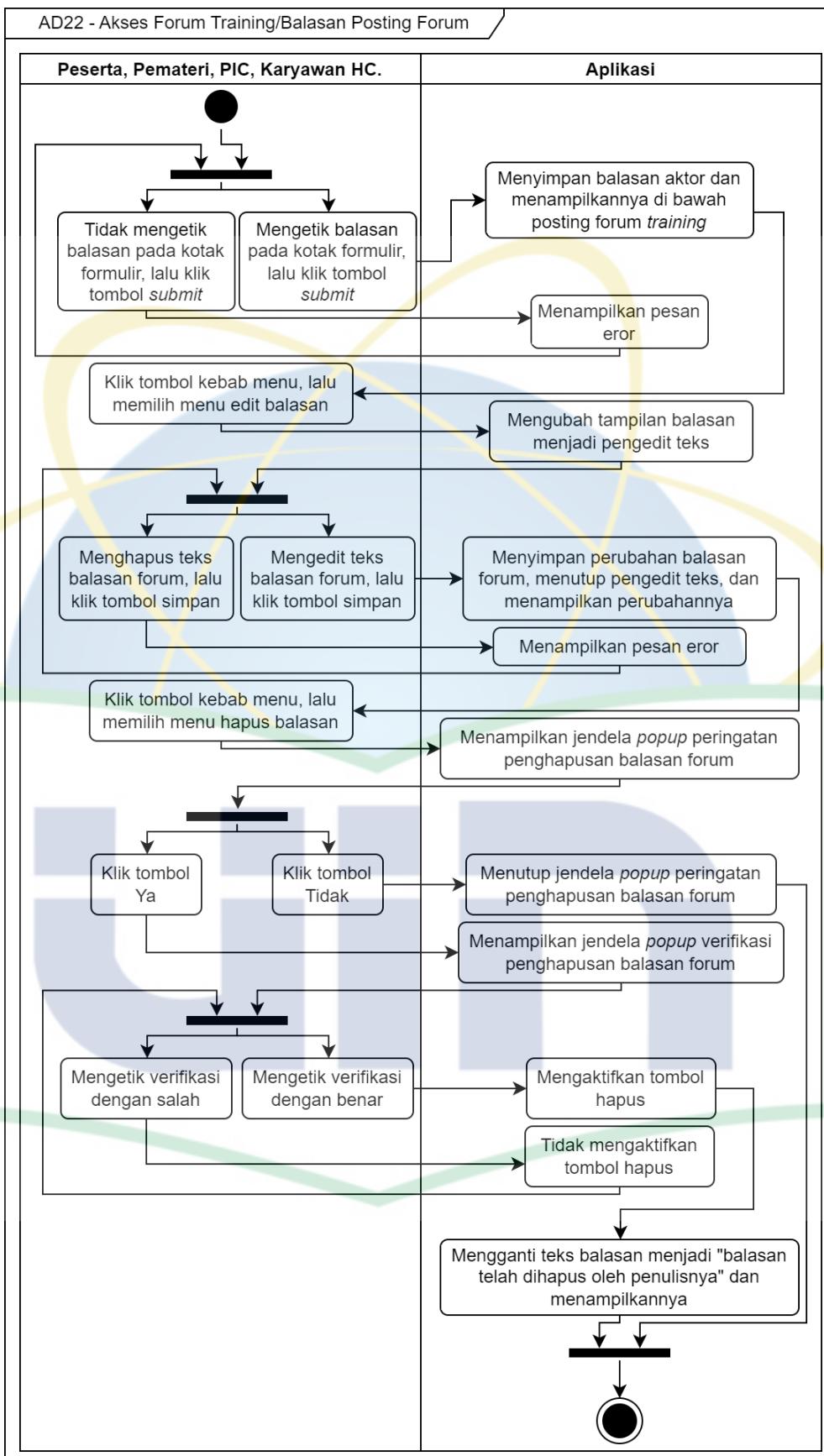
Gambar 4.46 Activity Diagram Akses Modul Training

Aktivitas berikutnya merupakan alur pengaksesan forum modul *training*. Aktivitas ini dapat diakses oleh Peserta *Training*, Pemateri *Training*, PIC *Training* dan Karyawan HC. Mereka dapat melihat forum untuk masing-masing konten modul *training*.



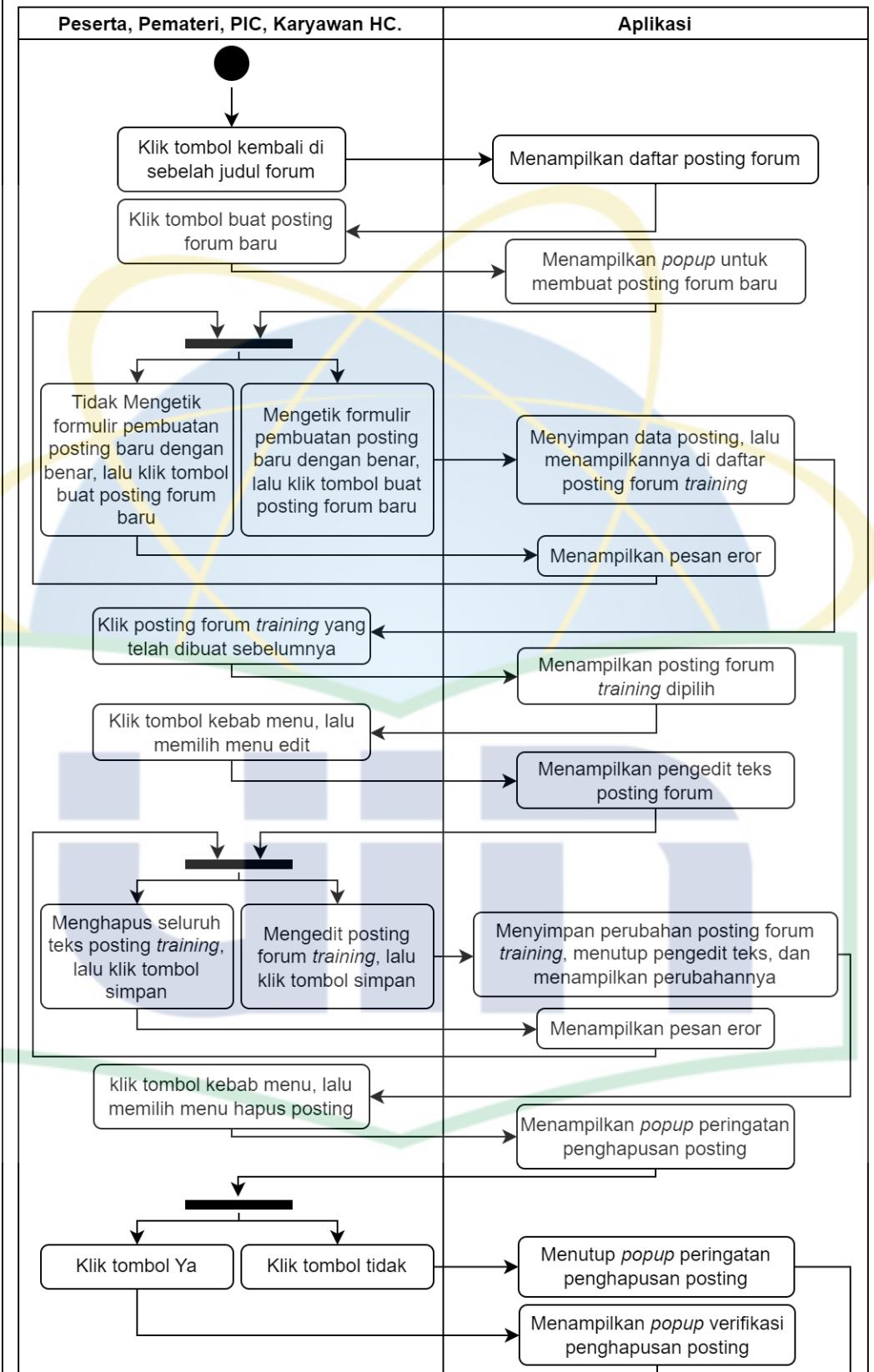
**Gambar 4.47 Activity Diagram Akses Forum Training/Menampilkan Posting Forum**

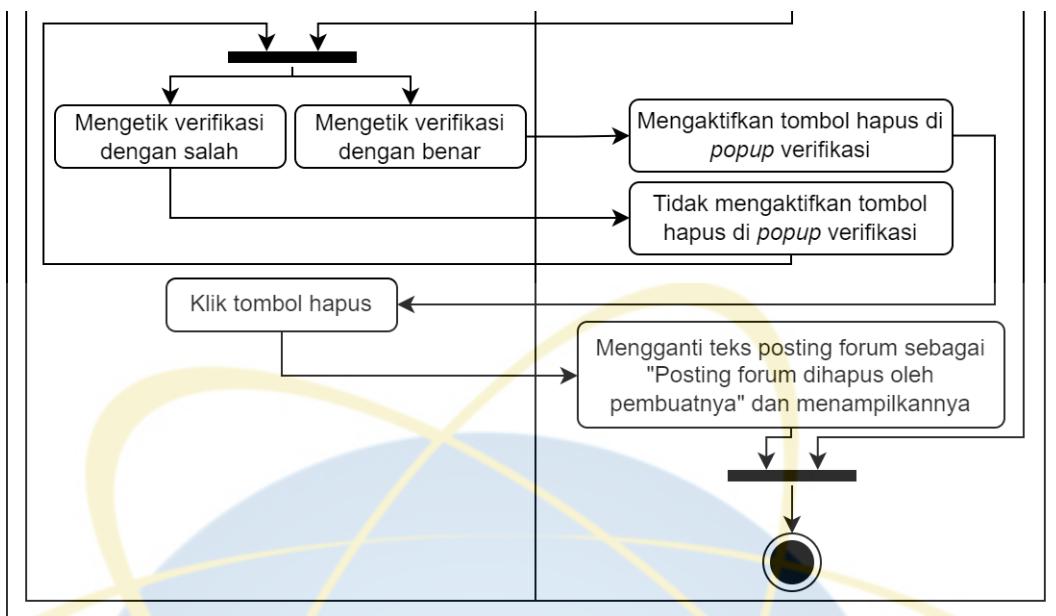
Aktivitas berikutnya merupakan alur pengelolaan balasan forum. Aktivitasnya meliputi pembuatan balasan posting forum, serta pengeditan dan penghapusannya. Aktivitas ini dapat diakses oleh Peserta *Training*, Pemateri *Training*, PIC *Training* dan Karyawan HC.



Gambar 4.48 Activity Diagram Akses Forum Training/Balasan Posting Forum

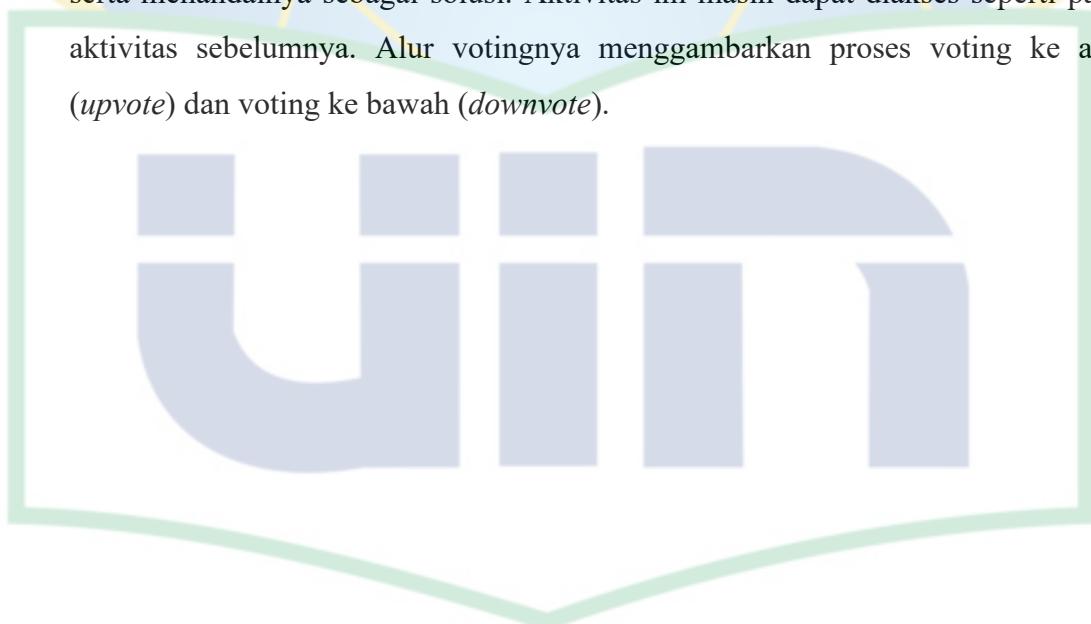
AD23 - Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul Training

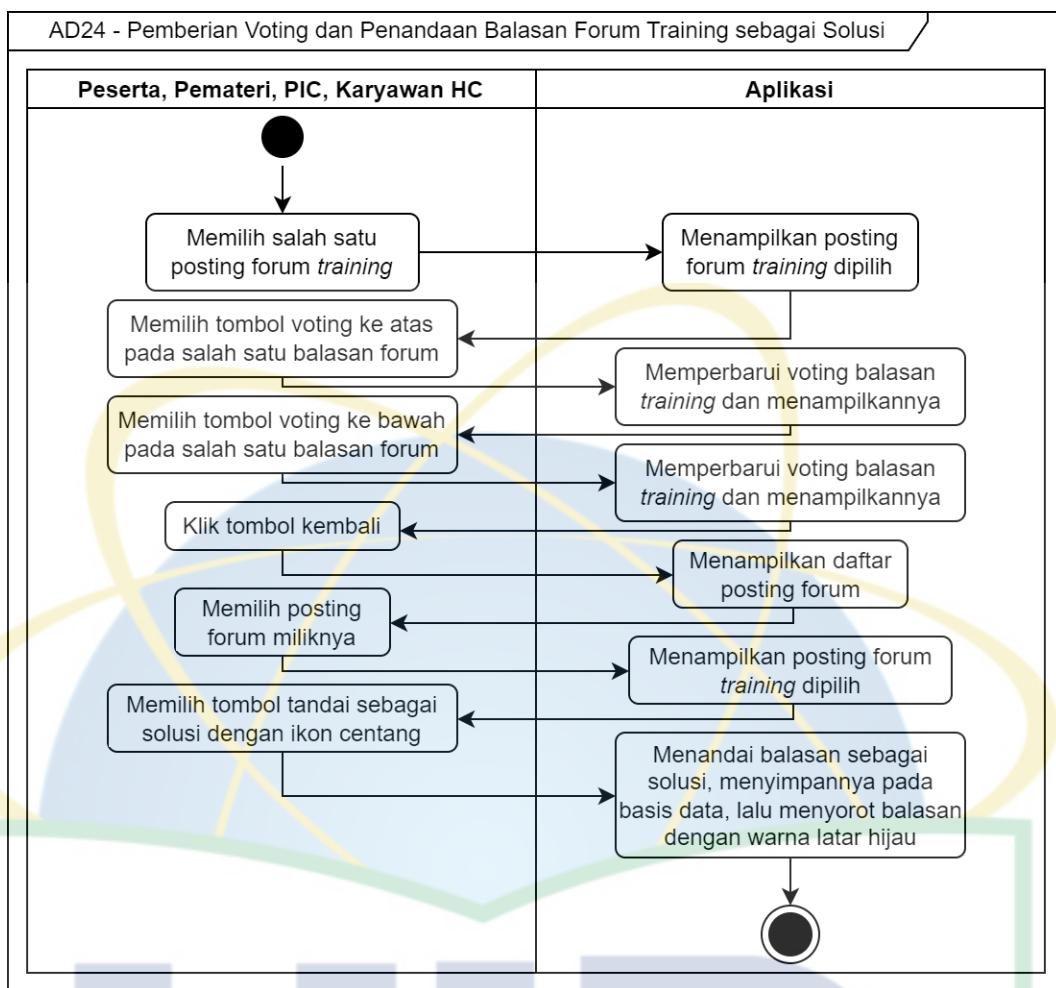




**Gambar 4.49 Activity Diagram Akses Forum Training/Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul Training**

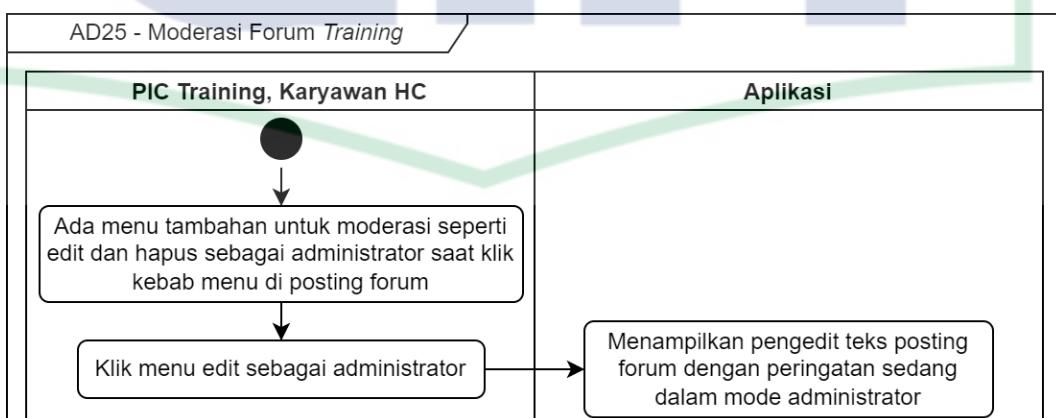
Aktivitas berikutnya menggambarkan alur voting balasan forum *training* serta menandainya sebagai solusi. Aktivitas ini masih dapat diakses seperti pada aktivitas sebelumnya. Alur votingnya menggambarkan proses voting ke atas (*upvote*) dan voting ke bawah (*downvote*).

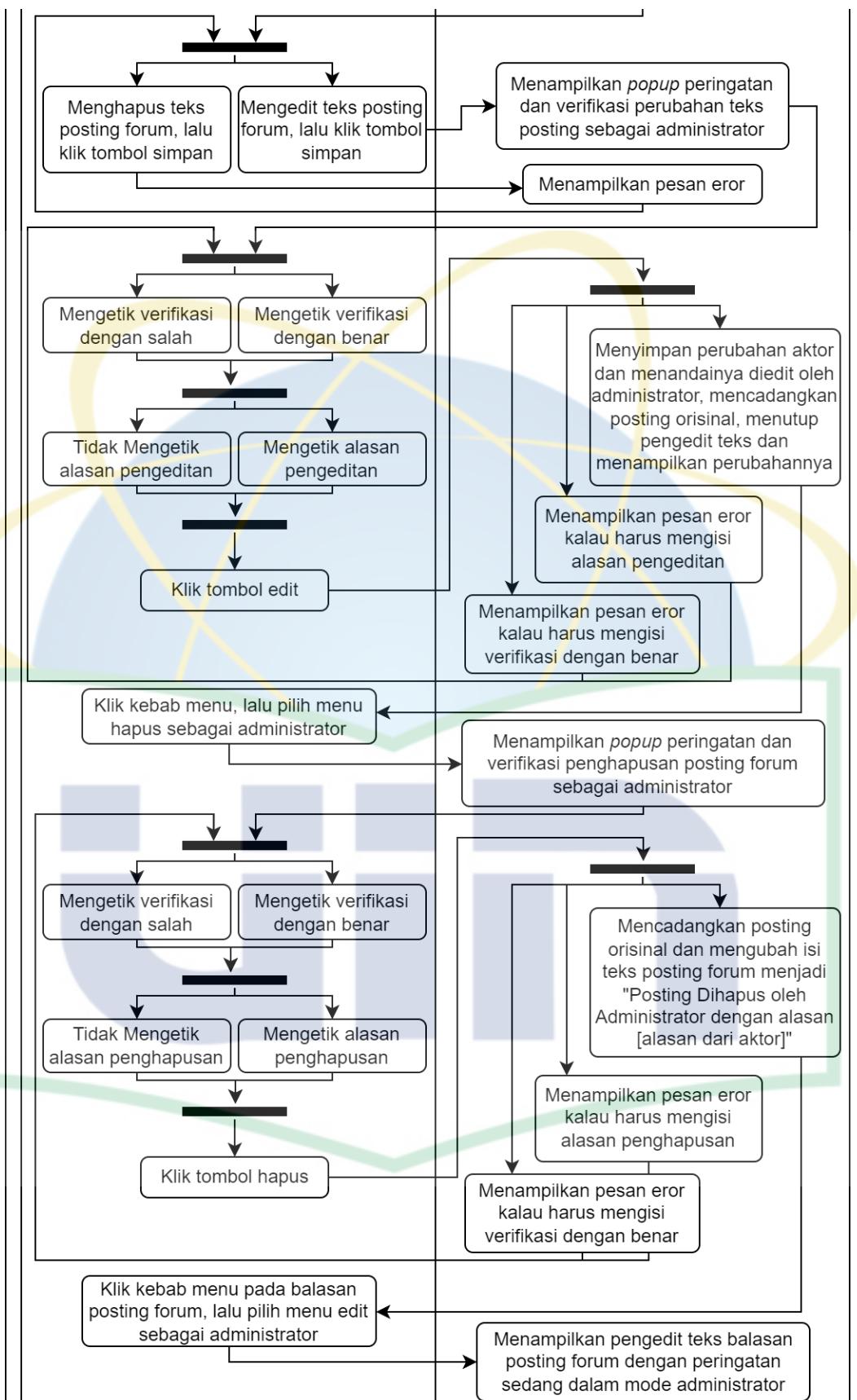


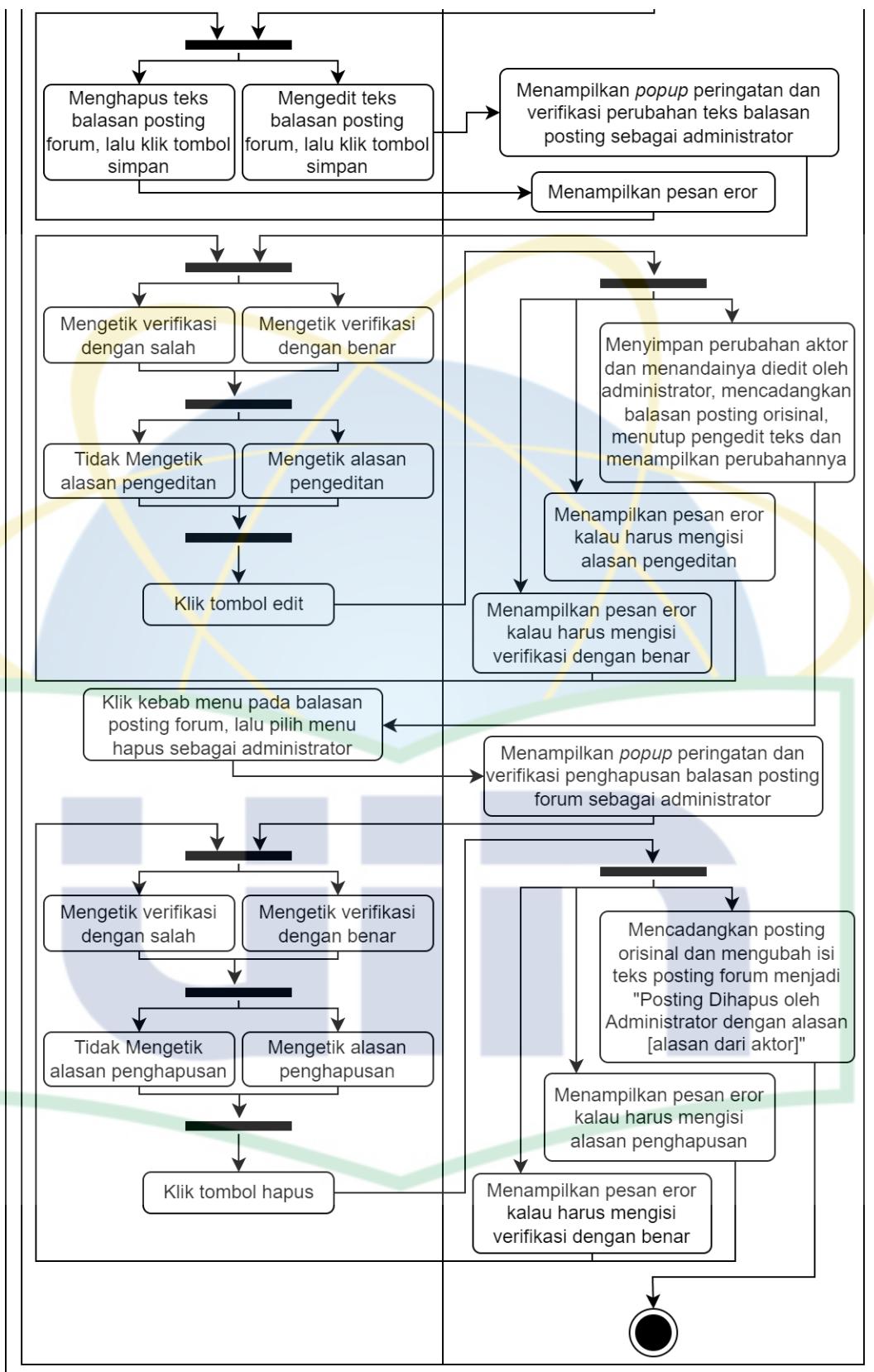


Gambar 4.50 Activity Diagram Akses Forum Training/Pemberian Voting dan Penandaan Balasan Forum Training sebagai Solusi

Aktivitas terakhir menggambarkan alur moderasi forum berupa tindakan pengeditan dan penghapusan sebagai administrator untuk posting forum dan balasannya. Aktivitas ini dapat diakses oleh PIC Training dan Karyawan HC.



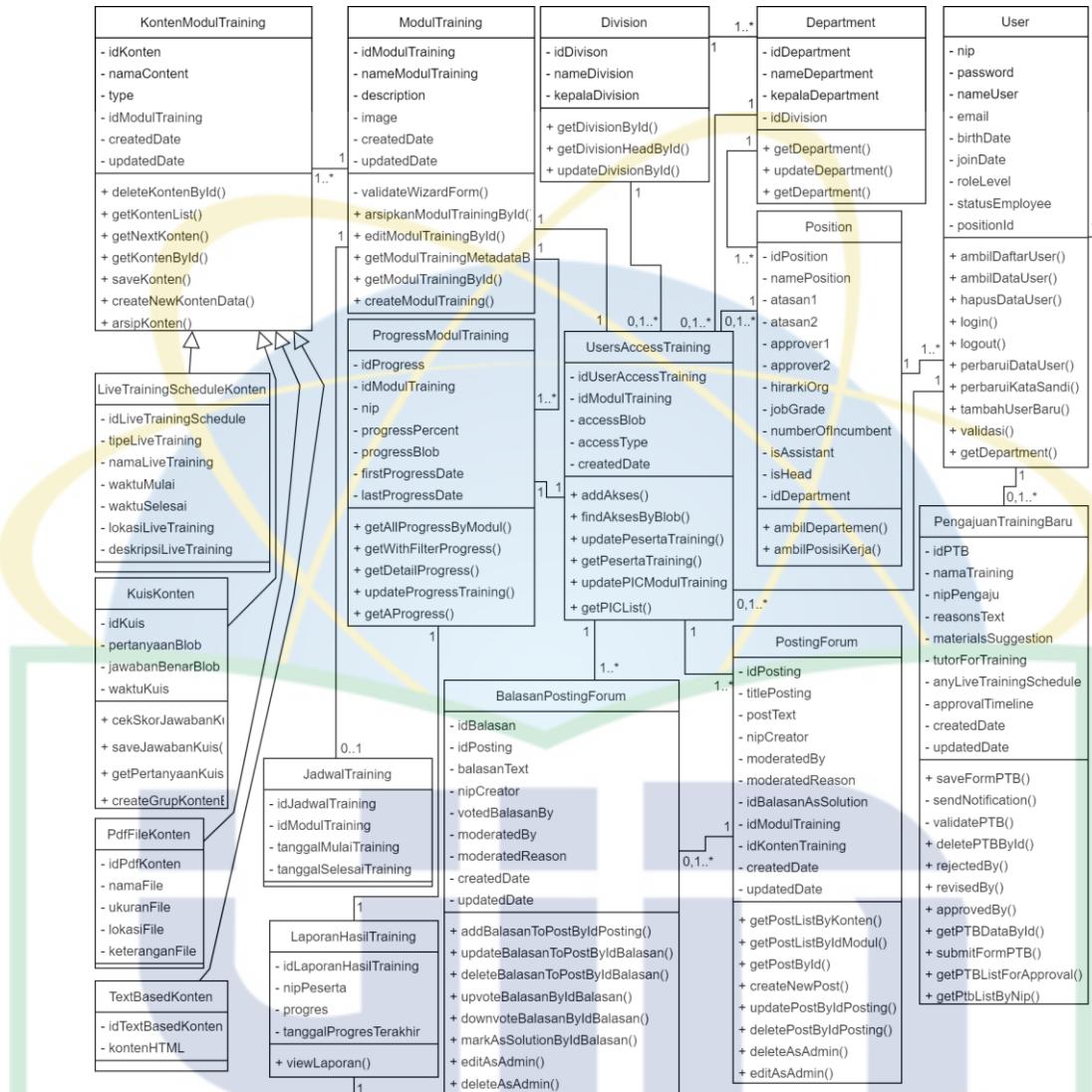




Gambar 4.51 *Activity Diagram* Moderasi Forum Training

### 3. Class Diagram

*Class Diagram* berikut menggambarkan struktur objek aplikasi *e-learning* HC Portal.



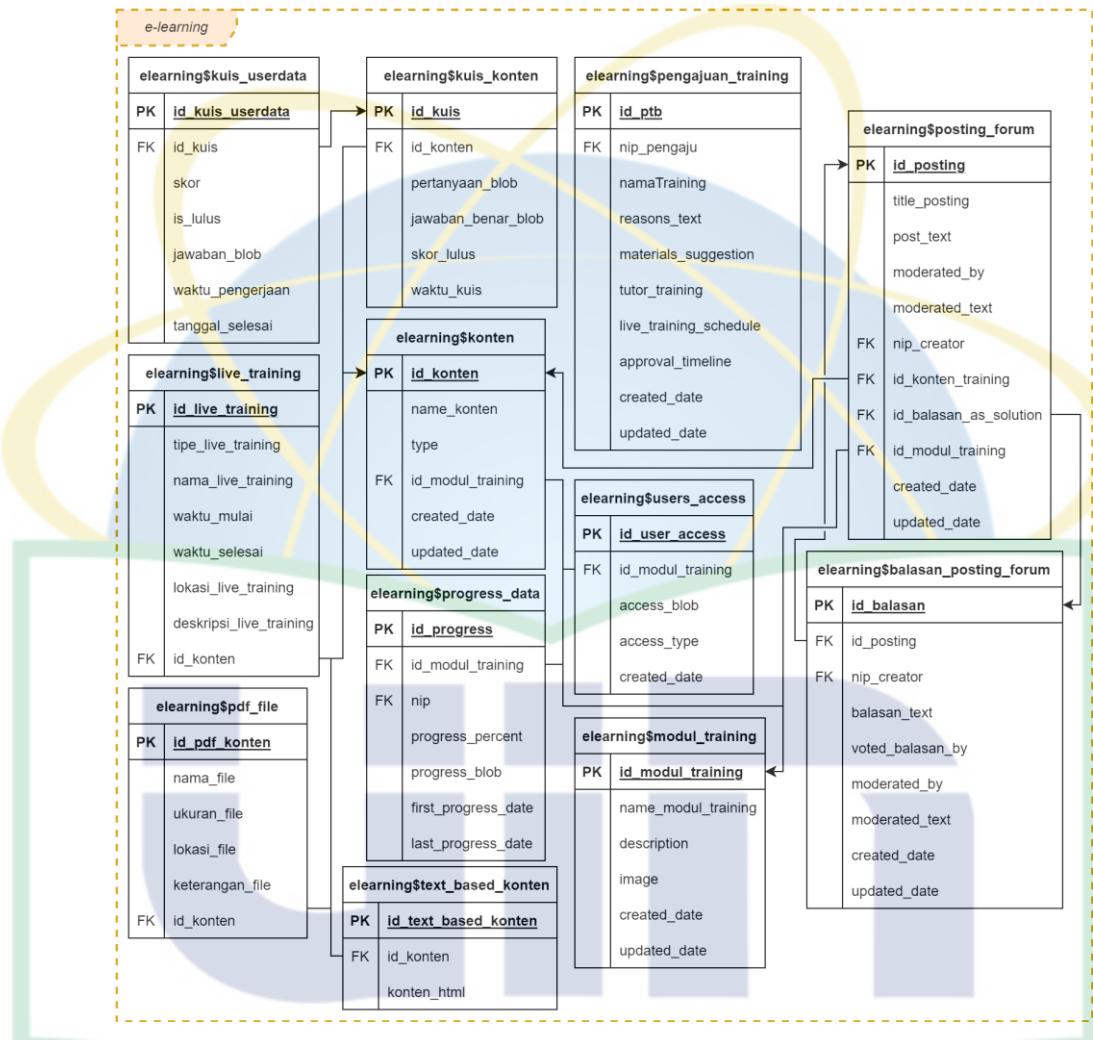
Gambar 4.52 *Class Diagram* Modul Aplikasi *e-learning* dan Forum Training

### 4. Pemodelan Skema Basis Data

Pemodelannya pada penelitian ini dilakukan sesuai dengan aturan yang telah dijelaskan oleh Munawar dan Wagner, seperti yang telah dijelaskan pada subbab 2.10.2 tentang “Skema Basis Data dan Pemodelannya dari *Class Diagram*”.

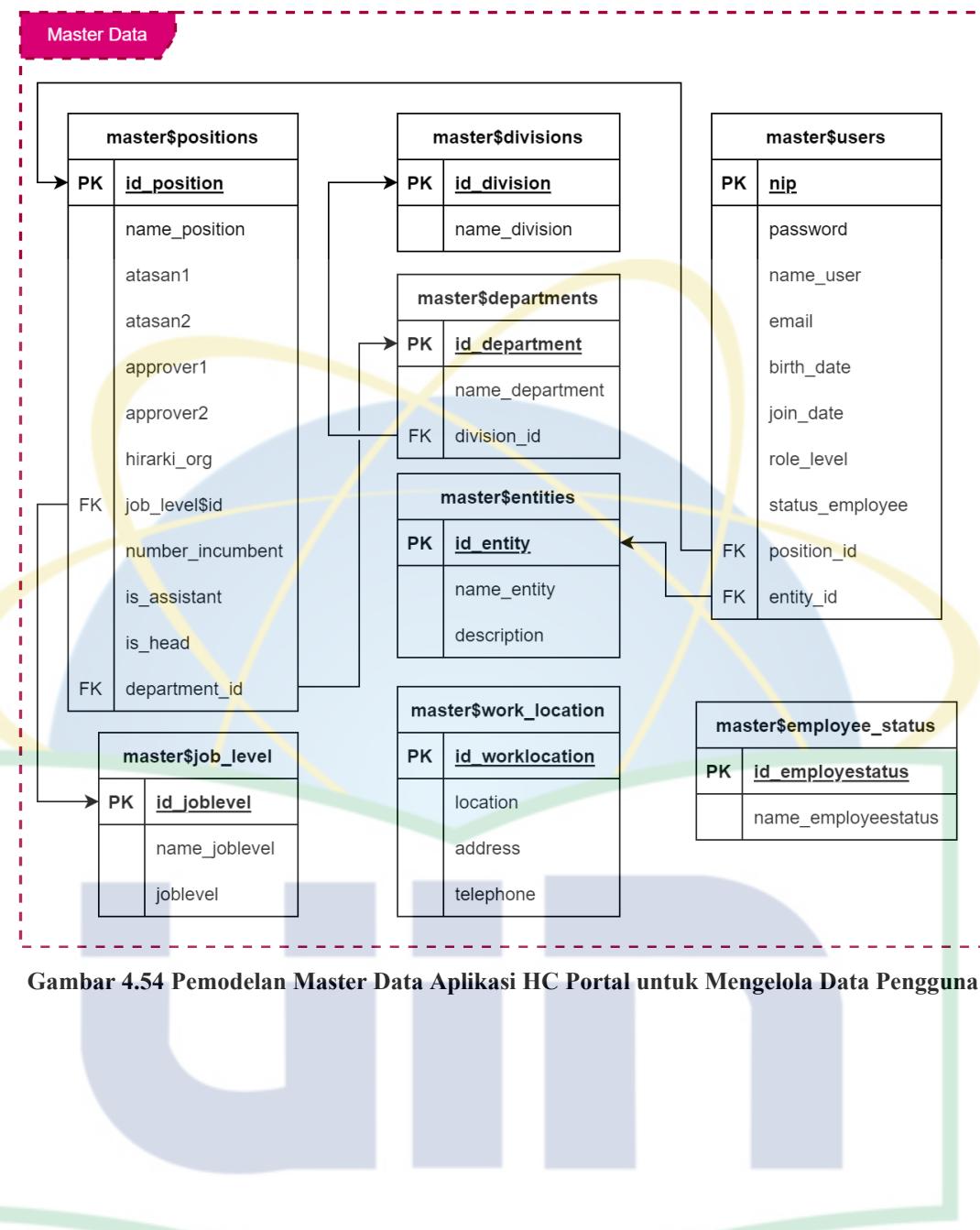
Ada beberapa hal yang penulis lakukan untuk membuat pemodelan skema basis data. Pertama, penulis merumuskan nama tabel dengan membuat nama prefiks agar dapat memisahkannya dengan modul aplikasi HC Portal lain yang sudah ada. Penulis mengambil variabel sederhana pada *Class Diagram* yang dapat langsung

dijadikan atribut pada tabel di basis data. Penulis juga membuat atribut-atribut tambahan, serta tabel tambahan untuk memenuhi penyimpanan seluruh data pada suatu *Class*. Dua desain basis data juga ditambahkan ke dalam pemodelan ini untuk memperlihatkan hubungannya dengan tabel-tabel yang dibuat untuk modul *e-learning* ini.

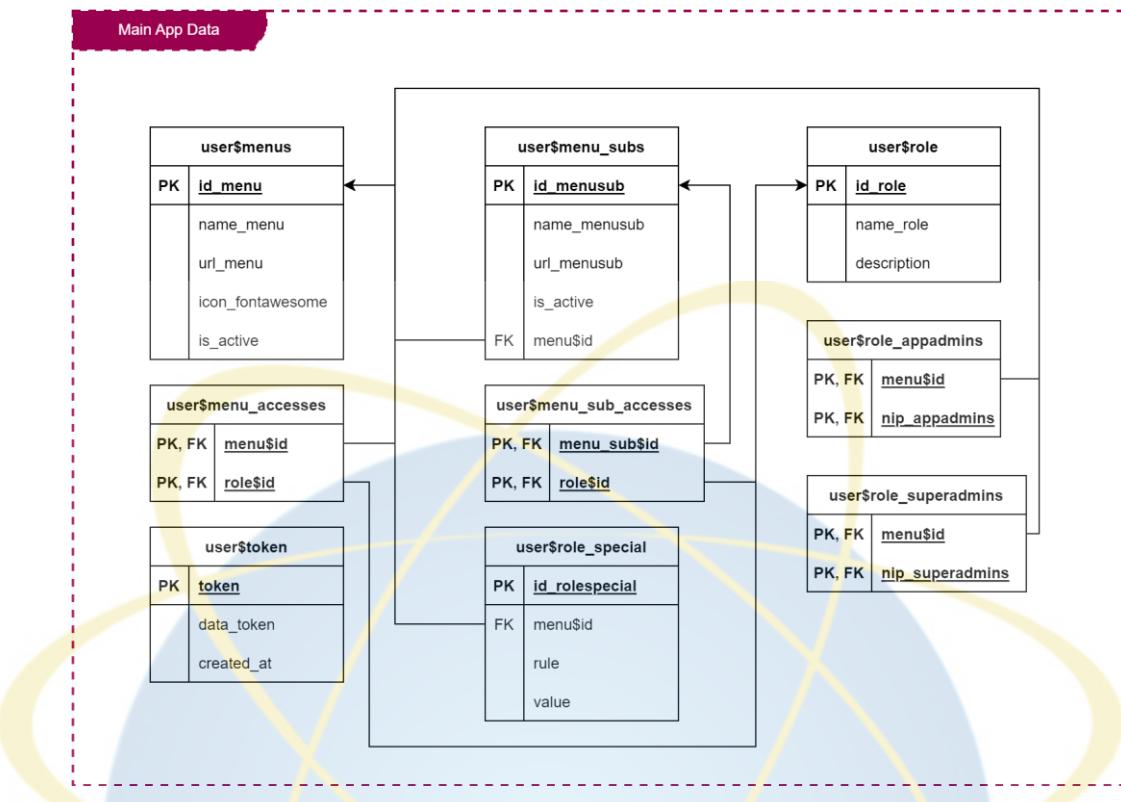


Gambar 4.53 Pemodelan Basis Data untuk modul *e-learning*

Gambar di atas menunjukkan pemodelan basis data yang diperlukan untuk mewadahi penyimpanan data modul aplikasi *e-learning* HC Portal. Dua pemodelan berikut merupakan pemodelan untuk keperluan dasar aplikasi dan pengelolaan data pengguna.

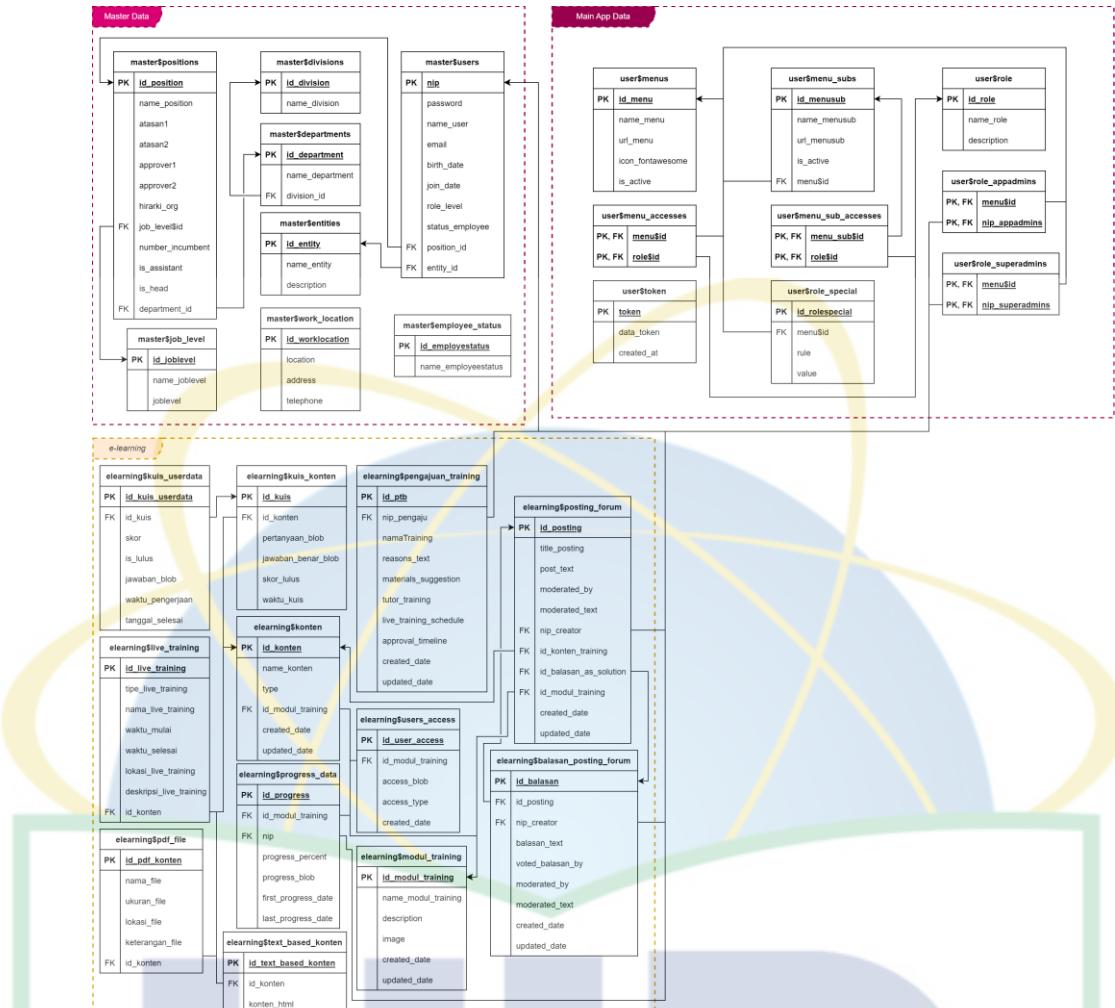


Gambar 4.54 Pemodelan Master Data Aplikasi HC Portal untuk Mengelola Data Pengguna



**Gambar 4.55 Pemodelan Main App Data untuk Keperluan Dasar Aplikasi**

Ketiga pemodelan basis data di atas mencakup permasalahan yang diangkat pada penelitian ini. Sebenarnya desain basis data untuk aplikasi HC Portal telah memiliki berbagai pemodelan untuk modul aplikasi lain. Penulis telah membahasnya pada subbab 4.1.3 yang membahas tentang identifikasi aplikasi web HC Portal. Gambar berikut ini menunjukkan pemodelan basis data keseluruhan untuk studi kasus pada penelitian ini.



Gambar 4.56 Pemodelan Basis Data Keseluruhan untuk Studi Kasus Penelitian Ini

## 5. Spesifikasi Basis Data

Penjelasannya penulis kelompokkan berdasarkan ketiga pemodelan yang telah dibuat sebelumnya. Penjelasan terkait spesifikasi basis data tetap dijelaskan dalam satu tabel.

### a. E-learning

- *E-learning Modul Training*

Nama Tabel : elearning\$modul\_training

*Primary Key* : id\_modul\_training

*Foreign Key* : -

Jenis Tabel : Master

**Tabel 4.32 Spesifikasi Tabel *E-learning* Modul *Training***

No.	Field	Type	Size	Keterangan
1.	id_modul_training	varchar	11	Kode: MT yyyy mm 000  Id Modul <i>Training</i> dengan prefiks MT dilanjutkan dengan 4 karakter tahun, 2 karakter bulan, dan 3 karakter angka inkremental.
2.	name_modul_training	varchar	32	Nama atau judul modul <i>training</i> .
3.	description	varchar	256	Deskripsi singkat modul <i>training</i> yang akan muncul di halaman <i>overview</i> ketika pengguna baru saja mengunjungi sebuah modul <i>training</i> .
4.	image	text	-	Alamat gambar <i>thumbnail</i> modul <i>training</i> .
5.	created_date	date	-	Tanggal dibuatnya modul <i>training</i> .
6.	updated_date	date	-	Tanggal modul <i>training</i> diperbarui.

o *E-learning* Konten

Nama Tabel : elearning\$konten

Primary Key : id\_konten

Foreign Key : id\_modul\_training

Jenis Tabel : Master

**Tabel 4.33 Spesifikasi Tabel *E-learning* Konten**

No.	Field	Type	Size	Keterangan
1.	id_konten	varchar	16	Kode: id_modul_training + KT 000

					Diawali dengan <code>id_modul_training</code> kemudian diikuti dengan Id Konten Modul <i>Training</i> dengan prefiks KT dilanjutkan dengan 3 karakter angka inkremental.
2.	<code>name_konten</code>	<code>varchar</code>	24		Nama atau judul konten modul <i>training</i> .
3.	<code>type</code>	<code>enum</code>	-		<p>Penanda tipe konten modul <i>training</i> dengan tipe data pilihan <i>enum</i>. Ada empat kemungkinan tipe modul <i>training</i> pada penelitian ini:</p> <ul style="list-style-type: none"> <li>• <i>text-based</i> (untuk tipe konten tekstual, gambar, dan video)</li> <li>• <i>quiz</i></li> <li>• <i>pdf_file</i></li> <li>• <i>live_training</i></li> </ul>
4.	<code>id_modul_training</code>	<code>varchar</code>	11		<p>Kode: <u>MT yyyy mm 000</u></p> <p>Field ini merujuk ke id pada tabel modul <i>training</i>.</p>
5.	<code>created_date</code>	<code>date</code>	-		Tanggal dibuatnya konten modul <i>training</i> .
6.	<code>updated_date</code>	<code>date</code>	-		Tanggal konten modul <i>training</i> diperbarui.

○ *E-learning Kuis Konten*

Nama Tabel : elearning\$kuis\_konten

*Primary Key* : `id_kuis`

*Foreign Key* : `id_konten`

Jenis Tabel : Master

Tabel 4.34 Spesifikasi Tabel *E-learning Kuis Konten*

No.	Field	Type	Size	Keterangan
1.	id_kuis	varchar	23	<p>Kode: id_konten + QUIZ 000</p> <p>Id ini diawali dengan id_konten. Kemudian dilanjutkan id Konten Kuis dengan prefiks QUIZ dilanjutkan dengan 3 karakter angka inkremental.</p>
2.	id_konten	varchar	16	<p>Kode: id_modul_training + KT 000</p> <p>Field ini merujuk pada id di tabel <i>E-Learning Konten</i>.</p>
3.	pertanyaan_blob	JSON	-	Konten kuis yang berupa pertanyaan dan pilihan jawabannya. Data ini disimpan dalam struktur JSON.
4.	jawaban_benar_blob	JSON	-	Data jawaban benar yang disimpan dalam struktur JSON.
5.	skor_lulus	tinyint	-	Skor untuk ditandai sebagai lulus dalam mengerjakan kuis.
6.	waktu_kuis	tinyint	-	Penanda waktu untuk mengerjakan kuis dalam menit. Maksimum waktu yang dapat diberikan untuk mengerjakan kuis adalah 120 menit atau 2 jam.

- *E-learning Kuis User-data*

Nama Tabel : elearning\$kuis\_userdata

Primary Key : id\_kuis\_userdata

Foreign Key : id\_kuis

Jenis Tabel : Transaksi

**Tabel 4.35 Spesifikasi Tabel *E-learning Kuis User Data***

No.	Field	Type	Size	Keterangan
1.	id_kuis_userdata	varchar	23	<p>Kode: id_kuis + nip + 000</p> <p>Id ini diawali dengan id_kuis. Kemudian dilanjutkan nip yang mengerjakan kuis. Terakhir dilanjutkan dengan inkremental berjumlah 3 karakter untuk pengguna yang ingin melakukan beberapa percobaan pengerjaan kuis.</p>
2.	id_kuis	varchar	23	<p>Kode: id_konten + QUIZ 000</p> <p>Field ini merujuk pada id di tabel <i>E-Learning Kuis</i>.</p>
3.	skor	tinyint	-	Skor pengguna setelah melakukan pengerjaan kuis.
4.	is_lulus	boolean	-	Penanda apakah pengguna lulus atau tidak setelah mengerjakan kuis.
5.	jawaban_blob	JSON	-	Data jawaban kuis pengguna.
6.	waktu_pengerjaan	tinyint	-	Waktu pengerjaan kuis pengguna dicatat dalam menit.
7.	tanggal_selesai	timestamp	-	Tanggal dan waktu pengguna selesai mengerjakan kuis.

- *E-learning Text-based* Konten

Nama Tabel : elearning\$text\_based\_konten

*Primary Key* : id\_text\_based\_konten

*Foreign Key* : id\_konten

Jenis Tabel : Master

Tabel 4.36 Spesifikasi Tabel *E-learning Text-based* Konten

No.	Field	Type	Size	Keterangan
1.	id_text_based_konten	varchar	24	Kode: id_konten + TXTKT 000  Id ini diawali dengan Konten Kuis dengan prefiks TXTKT dilanjutkan dengan 4 karakter tahun, 2 karakter bulan, dan 3 karakter angka inkremental.
2.	id_konten	varchar	16	Kode: id_modul_training + KT 000  <i>Field</i> ini merujuk pada id di tabel <i>E-Learning</i> Konten.
3.	konten_html	text	-	Kode HTML yang membentuk konten

- *E-learning PDF File* Konten

Nama Tabel : elearning\$pdf\_file

*Primary Key* : id\_pdf\_konten

*Foreign Key* : id\_konten

Jenis Tabel : Master

Tabel 4.37 Spesifikasi Tabel *E-learning PDF File* Konten

No.	Field	Type	Size	Keterangan
1.	id_pdf_konten	varchar	24	Kode: id_konten + PDFKT 000

				Diawali dengan id_konten kemudian diikuti dengan Id PDF Konten dengan prefiks PDFKT dilanjutkan dengan 3 karakter angka inkremental.
2.	id_konten	varchar	16	Kode: id_modul_training + KT 000  Field ini merujuk pada id di tabel E-Learning Konten.
3.	nama_file	varchar	64	Nama berkas PDF.
4.	ukuran_file	int	-	Ukuran berkas PDF yang dinyatakan dalam Bytes.
5.	lokasi_file	text	-	Teks <i>string</i> berisi lokasi direktori berkas di server.
6.	keterangan_file	text	-	Penjelasan singkat terkait berkas PDF.

- *E-learning Live Training* Konten

Nama Tabel : elearning\$live\_training

*Primary Key* : id\_live\_training

*Foreign Key* : id\_konten

Jenis Tabel : Master

**Tabel 4.38 Spesifikasi Tabel *E-learning Live Training* Konten**

No.	Field	Type	Size	Keterangan
1.	id_live_training	varchar	23	Kode: id_konten + LTKT 000  Diawali dengan id_konten kemudian diikuti dengan Id <i>Live Training</i> dengan prefiks LTKT dilanjutkan

				dengan 3 karakter angka inkremental.
2.	id_konten	varchar	16	Kode: <u>id_modul_training + KT 000</u>  <i>Field</i> ini merujuk pada id di tabel <i>E-Learning Konten</i> .
3.	tipe_live_training	enum	-	Penanda tipe <i>Live Training</i> yang terdiri dari: <ul style="list-style-type: none"><li>• offline_learning</li><li>• online_learning</li></ul>
4.	nama_live_training	varchar	64	Nama acara <i>Live Training</i> .
5.	waktu_mulai	timestamp	-	Tanggal dan Waktu mulai acara <i>Live Training</i> yang ditulis dalam <i>timestamp</i> .
6.	waktu_selesai	timestamp	-	Tanggal dan Waktu selesai acara <i>Live Training</i> yang ditulis dalam <i>timestamp</i> .
7.	lokasi_live_training	text	-	Lokasi <i>Live Training</i> diadakan. <i>Field</i> ini akan berisi lokasi ruangan apabila tipenya <i>Offline Learning</i> atau berisi alamat URL konferensi video apabila tipenya <i>Online Learning</i> .
8.	deskripsi_live_training	text	-	Deskripsi singkat terkait acara <i>Live Training</i> .

o *E-learning Progress Data*

Nama Tabel : elearning\$progress\_data

*Primary Key* : id\_progress

*Foreign Key* : id\_modul\_training, nip

Jenis Tabel : Master

Tabel 4.39 Spesifikasi Tabel *E-learning Progress*

No.	Field	Type	Size	Keterangan
1.	id_progress	varchar	19	Kode: id_modul_training + nip  Diawali dengan id_modul_training, kemudian diikuti dengan nip karyawan.
2.	id_modul_training	varchar	11	Kode: MT yyyy mm 000  Field ini merujuk pada id di tabel <i>E-learning Modul Training</i> .
3.	nip	varchar	8	Kode: CG 000000  Field ini merujuk pada tabel Master Users.
4.	progress_percent	tinyint	-	Progres karyawan yang dinyatakan dalam persen, angkanya dari 0 - 100.
5.	progress_blob	JSON	-	Data progres <i>training</i> yang disimpan dalam struktur JSON.
6.	first_progress_date	timestamp	-	Waktu pertama kali karyawan mengakses modul <i>training</i> .
7.	last_progress_date	timestamp	-	Waktu terakhir kali karyawan membuat progres di modul <i>training</i> .

- o *E-learning Pengajuan Training*

Nama Tabel : elearning\$pengajuan\_training

Primary Key : id\_ptb  
 Foreign Key : nip\_pengaju  
 Jenis Tabel : Transaksi

**Tabel 4.40 Spesifikasi Tabel *E-learning Pengajuan Training***

No.	Field	Type	Size	Keterangan
1.	id_ptb	varchar	19	<p>Kode: <u>PTB YYYY MM 000</u></p> <p>Diawali dengan prefix “PTB”, kemudian diikuti dengan 4 karakter tahun, 2 karakter bulan, terakhir 3 karakter angka inkremental.</p>
2.	nip_pengaju	varchar	8	<p>Kode: <u>CG 000000</u></p> <p>Field ini merujuk pada tabel Master Users.</p>
3.	reasons_text	text	-	Isian formulir PTB terkait alasan diperlukannya pengadaan <i>training</i> .
4.	materials_suggestion	text	-	Isian formulir PTB terkait saran materi apa yang ingin diangkat dalam pengadaan <i>training</i> .
5.	tutor_training	text	-	Isian formulir PTB terkait saran tutor untuk <i>training</i> yang diajukan.
6.	live_training_schedule	text	-	Jika <i>training</i> telah direncanakan untuk mengadakan <i>Live Training</i> , Field ini menyimpan data untuk isian formulir tersebut.

7.	approval_timeline	JSON	-	Data <i>timeline approval</i> yang nantinya diolah untuk tampilannya.
8.	created_date	timestamp	-	Tanggal dan waktu dibuatnya formulir PTB.
9.	updated_date	Timestamp	-	Tanggal dan waktu diperbaruiinya formulir PTB.

o *E-learning User Access*

Nama Tabel : elearning\$users\_access

Primary Key : id\_user\_access

Foreign Key : id\_modul\_training

Jenis Tabel : Master

Tabel 4.41 Spesifikasi Tabel *E-learning User Access*

No.	Field	Type	Size	Keterangan
1.	id_user_access	varchar	22	<p>Kode: id_modul_training + access_blob + 000</p> <p>Diawali dengan id_modul_training, kemudian diikuti dengan access_blob, terakhir ditambahkan angka inkremental.</p>
2.	id_modul_training	varchar	11	<p>Kode: MT yyyy mm 000</p> <p>Field ini merujuk pada id di tabel <i>E-learning Modul Training</i>.</p>

3.	access_blob	varchar	8	<p>Field ini bisa berisi di antara 3 nilai berikut:</p> <ul style="list-style-type: none"> <li>• nip → 8 karakter Kode: <u>CG 000000</u></li> <li>• id_division → 6 karakter Kode: <u>div-00</u></li> <li>• id_departemen → 6 karakter Kode: <u>dep-00</u></li> </ul> <p>Tiga kemungkinan ini karena akses terhadap modul <i>training</i>.</p>
4.	access_type	enum	-	<p>Pilihan tipe akses modul <i>training</i> yang terdiri dari:</p> <ul style="list-style-type: none"> <li>• peserta → menandakan bahwa akses ini adalah peserta <i>training</i>.</li> <li>• pic → menandakan bahwa akses ini untuk PIC <i>Training</i>.</li> <li>• pemateri → menandakan bahwa akses ini untuk pemateri <i>training</i>.</li> <li>• admin → menandakan bahwa akses ini untuk Administrator.</li> </ul>
5.	created_date	timestamp	-	Tanggal dibuatnya akses terhadap modul <i>training</i> .

o *E-learning Posting Forum*

Nama Tabel : elearning\$posting\_forum

Primary Key : id\_posting  
 Foreign Key : nip\_creator, id\_konten\_training, id\_balasan\_as\_solution,  
               id\_modul\_training  
 Jenis Tabel : Transaksi

**Tabel 4.42 Spesifikasi Tabel E-learning Posting Forum**

No.	Field	Type	Size	Keterangan
1.	id_posting	varchar	33	<p>Kode: id_modul_training + nip_creator + YYYY MM DD hh mm ss</p> <p>Diawali dengan id_modul_training, kemudian diikuti dengan nip pembuat posting, terakhir ditambahkan <i>timestamp</i> dengan format 4 karakter tahun, 2 karakter bulan, 2 karakter hari, 2 karakter jam, 2 karakter menit, dan 2 karakter detik.</p>
2.	title_posting	text	-	Judul posting yang dibuat.
3.	post_text	mediumtext	-	Teks konten posting.
4.	moderated_by	text	-	Penanda posting apabila di-moderasi oleh PIC <i>Training</i> atau Karyawan HC.
5.	moderated_reason	text	-	Alasan PIC <i>Training</i> atau karyawan HC melakukan moderasi terhadap posting forum.
6	nip_creator	varchar	8	<p>Kode: CG 000000</p> <p>Field ini merujuk pada tabel Master <i>Users</i>.</p>

7.	id_konten_training	varchar	16	Kode: id_modul_training + KT 000  Field ini merujuk pada tabel E-learning Konten.
8.	id_balasan_as_solution	varchar	48	Kode: id_posting + R + YYYY MM DD hh mm ss  Penanda id balasan sebagai solusi yang merujuk dari tabel E-learning Balasan Posting Forum.
9.	id_modul_training	varchar	11	Kode: MT yyyy mm 000  Field ini merujuk pada id di tabel E-learning Modul Training.
10.	created_date	timestamp	-	Tanggal dan waktu posting dibuat.
11.	updated_date	timestamp	-	Tanggal dan waktu posting diperbarui.

o *E-learning* Balasan Posting Forum

Nama Tabel : master\$users

Primary Key : nip

Foreign Key : position\_id, entity\_id

Jenis Tabel : Master

**Tabel 4.43 Spesifikasi Tabel *E-learning* Balasan Posting Forum**

No.	Field	Type	Size	Keterangan
1.	id_balasan	varchar	48	Kode:

				<p><b>id_posting + R + YYYY MM DD hh mm ss</b></p> <p>Diawali dengan <b>id_posting</b>, kemudian diikuti dengan penanda <b>R</b> yang menandakan balasan (<i>reply</i>), terakhir ditambahkan <i>timestamp</i> dengan format 4 karakter tahun, 2 karakter bulan, 2 karakter hari, 2 karakter jam, 2 karakter menit, dan 2 karakter detik.</p>
2.	<b>id_posting</b>	varchar	33	<p>Kode: <b>id_modul_training + nip_creator + YYYY MM DD hh mm ss</b></p> <p><i>Field</i> ini merujuk pada tabel <i>E-learning Posting Forum</i>.</p>
3.	<b>nip_creator</b>	varchar	8	<p>Kode: <b>CG 000000</b></p> <p><i>Field</i> ini merujuk pada tabel <i>Master Users</i>.</p>
4.	<b>balasan_text</b>	text	-	Teks balasan posting.
5.	<b>voted_balasan_by</b>	JSON	-	Daftar voting oleh aktor yang dapat mengakses <b>modul training</b> yang disimpan dalam data berstruktur JSON.
6.	<b>moderated_by</b>	text	-	Penanda posting apabila di-moderasi oleh <i>PIC Training</i> atau Karyawan HC.
7.	<b>moderated_reason</b>	text	-	Alasan <i>PIC Training</i> atau karyawan HC melakukan moderasi terhadap posting forum.

8.	created_date	timestamp	-	Tanggal dan waktu posting dibuat.
9.	updated_date	timestamp	-	Tanggal dan waktu posting diperbarui.

### b. Master Data

- o Master *Users*

Nama Tabel : master\$users

*Primary Key* : nip

*Foreign Key* : position\_id, entity\_id

Jenis Tabel : Master

Tabel 4.44 Spesifikasi Tabel Master *Users*

No.	Field	Type	Size	Keterangan
1.	nip	varchar	8	Kode: CG 000000  Nomor Induk Pegawai
2.	password	varchar	60	Kata sandi pengguna dengan <i>hash</i> Bcrypt yang mana akan selalu menghasilkan 60 karakter mau sepanjang atau sependek apapun kata sandi pengguna
3.	name_user	varchar	32	Nama karyawan
4.	email	varchar	64	Email karyawan
5.	birth_date	date	-	Tanggal lahir karyawan
6.	join_date	date	-	Tanggal awal masuk karyawan ke perusahaan
7.	role_level	varchar	6	Kode: role-0  Kode peran ( <i>role</i> ) karyawan

8.	status_employee	varchar	1	Kode: P = Permanen C = <i>Contract</i> /Kontrak  Status kerja karyawan
9.	position_id	varchar	7	Kode: pos-000  Kode posisi kerja karyawan
10.	entity_id	varchar	10	Kode: entity-000  Kode entitas karyawan

o Master *Positions*

Nama Tabel : master\$positions

Primary Key : id\_position

Foreign Key : department\_id, job\_level\$id

Jenis Tabel : Master

Tabel 4.45 Spesifikasi Tabel Master Positions

No.	Field	Type	Size	Keterangan
1.	id_position	varchar	7	Kode: pos-000  Kode posisi kerja karyawan
2.	name_position	varchar	24	Nama posisi kerja
3.	atasan1	varchar	7	Kode posisi kerja atasan 1 karyawan (kodenya sama seperti field id)

4.	atasan2	varchar	7	Kode posisi kerja atasan 2 karyawan (kodenya sama seperti <i>field id</i> )
5.	approver1	varchar	7	Kode posisi kerja approver 1 karyawan (kodenya sama seperti <i>field id</i> )
6.	approver2	varchar	7	Kode posisi kerja approver 2 karyawan (kodenya sama seperti <i>field id</i> )
7.	hirarki_org	varchar	3	<p>Kode:</p> <p>N = Direksi dan Pimpinan</p> <p>N-1 = Supervisor</p> <p>N-2 = Staf Spesialis</p> <p>N-3/N-4 = Staf</p> <p>Kode hierarki karyawan</p>
8.	job_level\$id	varchar	2	Id kelas posisi pekerjaan yang diambil dari tabel master\$job_level
9.	number_incumbent	int	-	Jumlah karyawan yang menempati posisi kerja
10.	is_assistant	boolean	-	Tanda jika posisi merupakan asisten posisi atasannya
11.	is_head	boolean	-	Tanda jika posisi merupakan kepala dari posisi lain
12.	department_id	varchar	7	<p>Kode:</p> <p>dept-00</p> <p>Kode departemen</p>

- o *Master Divisions*

Nama Tabel : master\$divisions

*Primary Key* : id\_division

*Foreign Key* : -

Jenis Tabel : Master

Tabel 4.46 Spesifikasi Tabel Master Divisions

No.	Field	Type	Size	Keterangan
1.	id_division	varchar	5	Kode: div-0  Kode divisi perusahaan
2.	name_division	varchar	24	Nama divisi

o Master *Departments*

Nama Tabel : master\$departments

*Primary Key* : id\_department

*Foreign Key* : division\_id

Jenis Tabel : Master

Tabel 4.47 Spesifikasi Tabel Master Departments

No.	Field	Type	Size	Keterangan
1.	id_department	varchar	7	Kode: dept-00  Kode departemen perusahaan
2.	name_department	varchar	24	Nama departemen
3.	division_id	varchar	5	Kode: div-0  Kode divisi perusahaan

o Master *Entities*

Nama Tabel : master\$entities

*Primary Key* : id\_entity

*Foreign Key* : -

Jenis Tabel : Master

Tabel 4.48 Spesifikasi Tabel Master Entities

No.	Field	Type	Size	Keterangan
1.	id_entity	varchar	9	Kode: entity-00  Kode entitas perusahaan
2.	name_entity	varchar	32	Nama entitas perusahaan
3.	description	varchar	64	Deskripsi entitas

o Master *Work Location*

Nama Tabel : master\$work\_location

*Primary Key* : id\_worklocation

*Foreign Key* : -

Jenis Tabel : Master

Tabel 4.49 Spesifikasi Tabel Master Work Location

No.	Field	Type	Size	Keterangan
1.	id_worklocation	varchar	15	Kode: worklocation-00  Kode lokasi tempat kerja
2.	location	text	-	Lokasi kota tempat kerja
3.	address	text	-	Alamat tempat kerja
4.	telephone	varchar	13	Telepon tempat kerja

o Master *Employee Status*

Nama Tabel : master\$employee\_status

*Primary Key* : id\_employeestatus

*Foreign Key* : -

Jenis Tabel : Master

Tabel 4.50 Spesifikasi Tabel Master Employee Status

No.	Field	Type	Size	Keterangan
1.	id_employeestatus	varchar	1	Kode: P = Permanen C = <i>Contract/Kontrak</i>  Kode Status kerja karyawan
2.	name_employeestatus	varchar	24	Nama status employee

o Master *Job Level*

Nama Tabel : master\$job\_level

*Primary Key* : id\_joblevel

*Foreign Key* : -

Jenis Tabel : Master

Tabel 4.51 Spesifikasi Tabel Master Job Level

No.	Field	Type	Size	Keterangan
1.	id_joblevel	varchar	11	Kode: joblevel-00  Kode Job Level
2.	name_joblevel	varchar	24	Nama Job Level
3.	joblevel	varchar	2	Level <i>Job</i> berupa angka dari 4 sampai 30

c. Main App Data

o User Menu

Nama Tabel : user\$menu

*Primary Key* : id\_menu

*Foreign Key* : -

Jenis Tabel : Master

Tabel 4.52 Spesifikasi Tabel User Menu

No.	Field	Type	Size	Keterangan
1.	id_menu	varchar	7	Kode: menu-00  Kode menu aplikasi
2.	name_menu	varchar	16	Nama menu
3.	url_menu	varchar	24	Alamat menu
4.	icon_fontawesome	varchar	24	Class CSS ikon Fontawesome
5.	is_active	boolean	-	Tanda jikalau menu aktif atau tidak

o User Menu Access

Nama Tabel : user\$menu\_acceses

*Primary Key* : menu\$id, role\$id

*Foreign Key* : menu\$id, role\$id

Jenis Tabel : Master

Tabel 4.53 Spesifikasi Tabel Menu Accesses

No.	Field	Type	Size	Keterangan
1.	menu\$id	varchar	7	Kode: menu-00  Kode menu aplikasi
2.	role\$id	varchar	7	Kode: role-00  Kode role user

- User Sub Menu

Nama Tabel : user\$menu\_subs

*Primary Key* : id\_menusub

*Foreign Key* : menu\$id

Jenis Tabel : Master

Tabel 4.54 Spesifikasi Tabel Sub Menu

No.	Field	Type	Size	Keterangan
1.	id_menusub	varchar	10	Kode: menu-00 + -00  Kode sub menu aplikasi yang diawali dengan 7 karakter atribut menu\$id
2.	name_menusub	varchar	16	Nama sub menu
3.	url_menusub	varchar	24	Alamat sub menu
4.	is_active	boolean	-	Tanda jika sub menu aktif atau tidak
5.	menu\$id	varchar	7	Kode: menu-00  Kode menu aplikasi yang didapat dari tabel user\$menus

- User Sub Menu Access

Nama Tabel : user\$menu\_sub\_accesses

*Primary Key* : menu\_sub\$id, role\$id

*Foreign Key* : menu\_sub\$id, role\$id

Jenis Tabel : Master

Tabel 4.55 Spesifikasi Tabel Sub Menu Accesses

No.	Field	Type	Size	Keterangan
1.	menu_sub\$id	varchar	10	Kode:

				menu-00 + -00
2.	role\$id	varchar	7	Kode: role-00  Kode <i>role</i> user

- o User Role
 

Nama Tabel	: user\$role
Primary Key	: id_role
Foreign Key	: -
Jenis Tabel	: Master

Tabel 4.56 Spesifikasi Tabel User Role

No.	Field	Type	Size	Keterangan
1.	id_role	varchar	7	Kode: role-00  Kode <i>role</i> user
2.	name_role	varchar	12	Nama peran/ <i>role</i> user
3.	description	varchar	32	Deskripsi <i>role</i> user

- o User Role Appadmins
 

Nama Tabel	: user\$role_appadmins
Primary Key	: menu\$id, nip_appadmins
Foreign Key	: menu\$id, nip_appadmins
Jenis Tabel	: Master

**Tabel 4.57 Spesifikasi Tabel User Role Appadmins**

No.	Field	Type	Size	Keterangan
1.	menu\$id	varchar	7	Kode: menu-00  Kode menu aplikasi
2.	nip_appadmins	varchar	8	Kode CG 000000  Nomor Induk Pegawai untuk admin modul aplikasi

- *User Role Superadmins*

Nama Tabel : user\$role\_superuser

Primary Key : menu\$id, nip\_superuser

Foreign Key : menu\$id, nip\_superuser

Jenis Tabel : Master

**Tabel 4.58 Spesifikasi Tabel User Role Superadmins**

No.	Field	Type	Size	Keterangan
1.	menu\$id	varchar	7	Kode: menu-00  Kode menu aplikasi
2.	nip_superuser	varchar	8	Kode CG 000000  Nomor Induk Pegawai untuk superadmin aplikasi HC Portal

- *User Role Special*

Nama Tabel : user\$role\_special

Primary Key : id\_rolespecial

Foreign Key : menu\$id

Jenis Tabel : Master

Tabel 4.59 Spesifikasi Tabel User Role Special

No.	Field	Type	Size	Keterangan
1.	id_rolespecial	varchar	10	Kode: <u>role-s-000</u>  Kode <i>role special</i> aplikasi
2.	menu\$id	varchar	7	Kode: <u>menu-00</u>  Kode menu aplikasi
3.	rule	text	-	Aturan khusus keperluan teknis pengecekan <i>role special</i> untuk menu aplikasi dengan bentuk <i>string</i> . Aturan bisa berupa divisi, departemen, ataupun kode posisi tertentu. Nilai ini didapat dari tabel posisi maupun nilai <i>foreign key</i> -nya  Contoh: <ul style="list-style-type: none"><li>• “id_division” → menunjukkan divisi</li><li>• “id_department” → menunjukkan departemen</li><li>• “id_position” → menunjukkan id posisi karyawan</li><li>• “hirarki_org” → menunjukkan nilai hierarki posisi karyawan</li></ul>
4.	value	text	-	Nilai aturan spesial yang benar. Jika aturan tidak sama dengan nilai ini, maka pengguna tidak memiliki akses <i>role special</i> .

- User Token
 

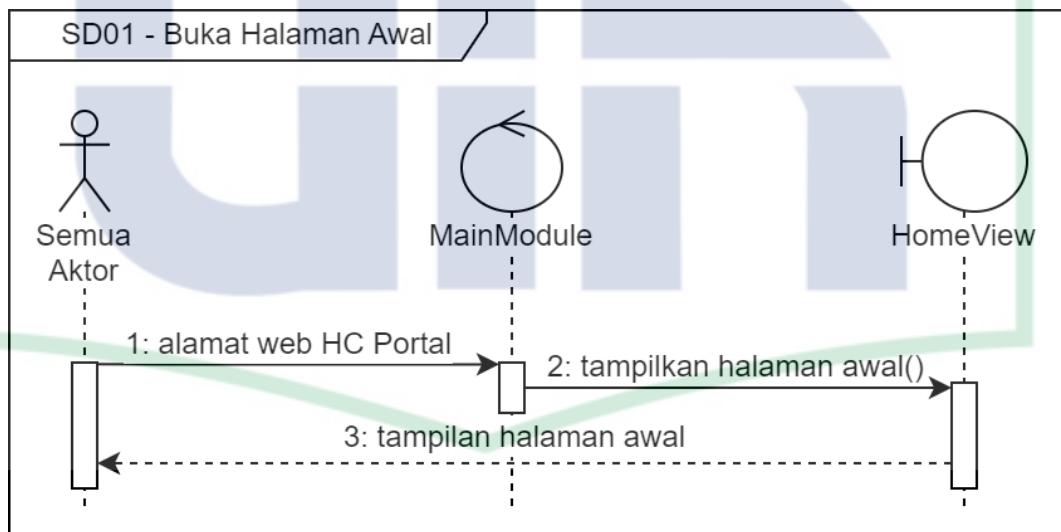
Nama Tabel	:	user\$token
Primary Key	:	token
Foreign Key	:	-
Jenis Tabel	:	Master

**Tabel 4.60 Spesifikasi Tabel User Token**

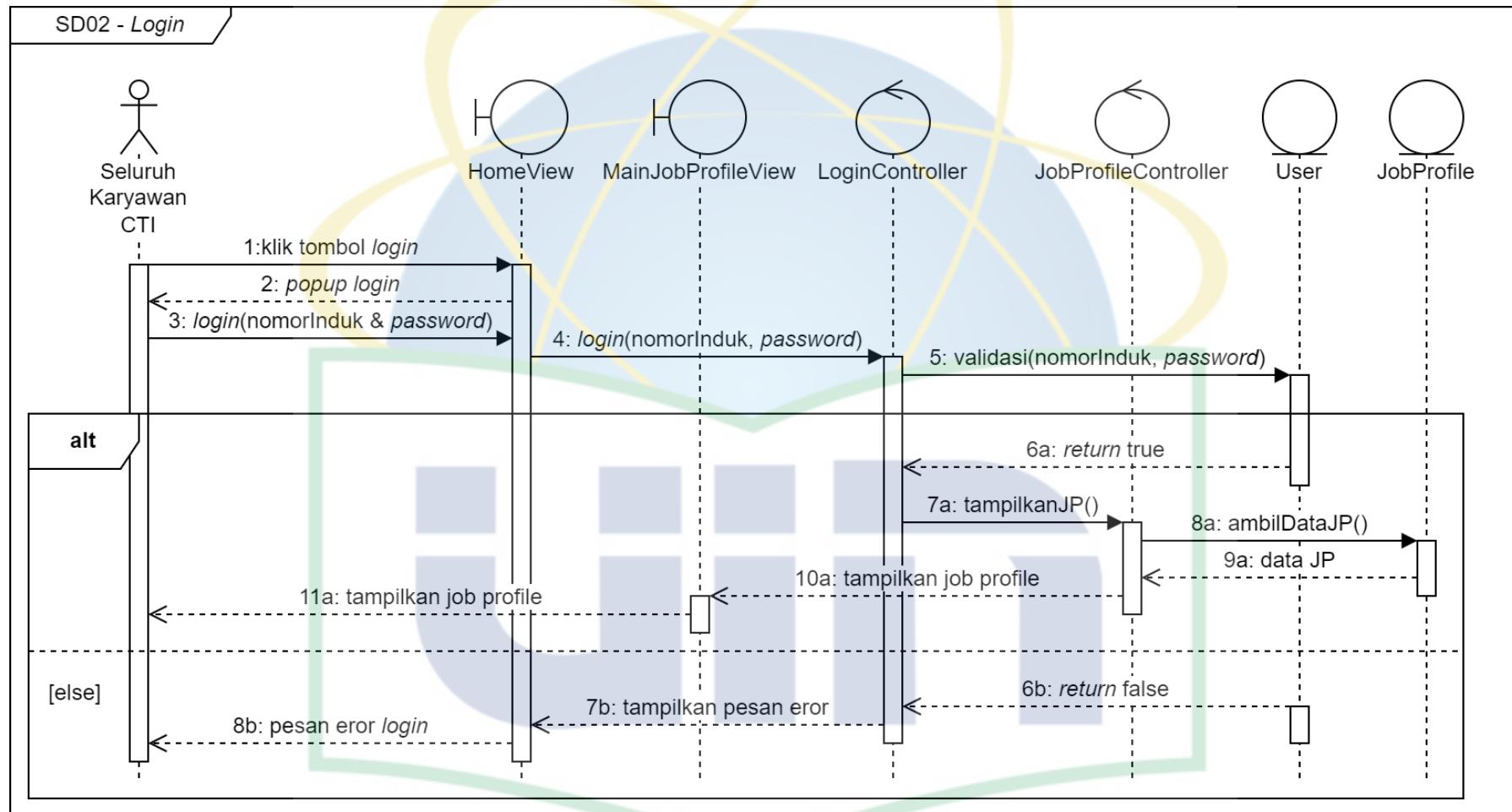
No.	Field	Type	Size	Keterangan
1.	token	varchar	32	Kode Hash MD5 yang memiliki 32 karakter
2.	data_token	mediumtext	-	Data yang tersimpan pada token berupa <i>string JSON</i> .
3.	created_at	date	-	Tanggal token dibuat

## 6. Sequence Diagram

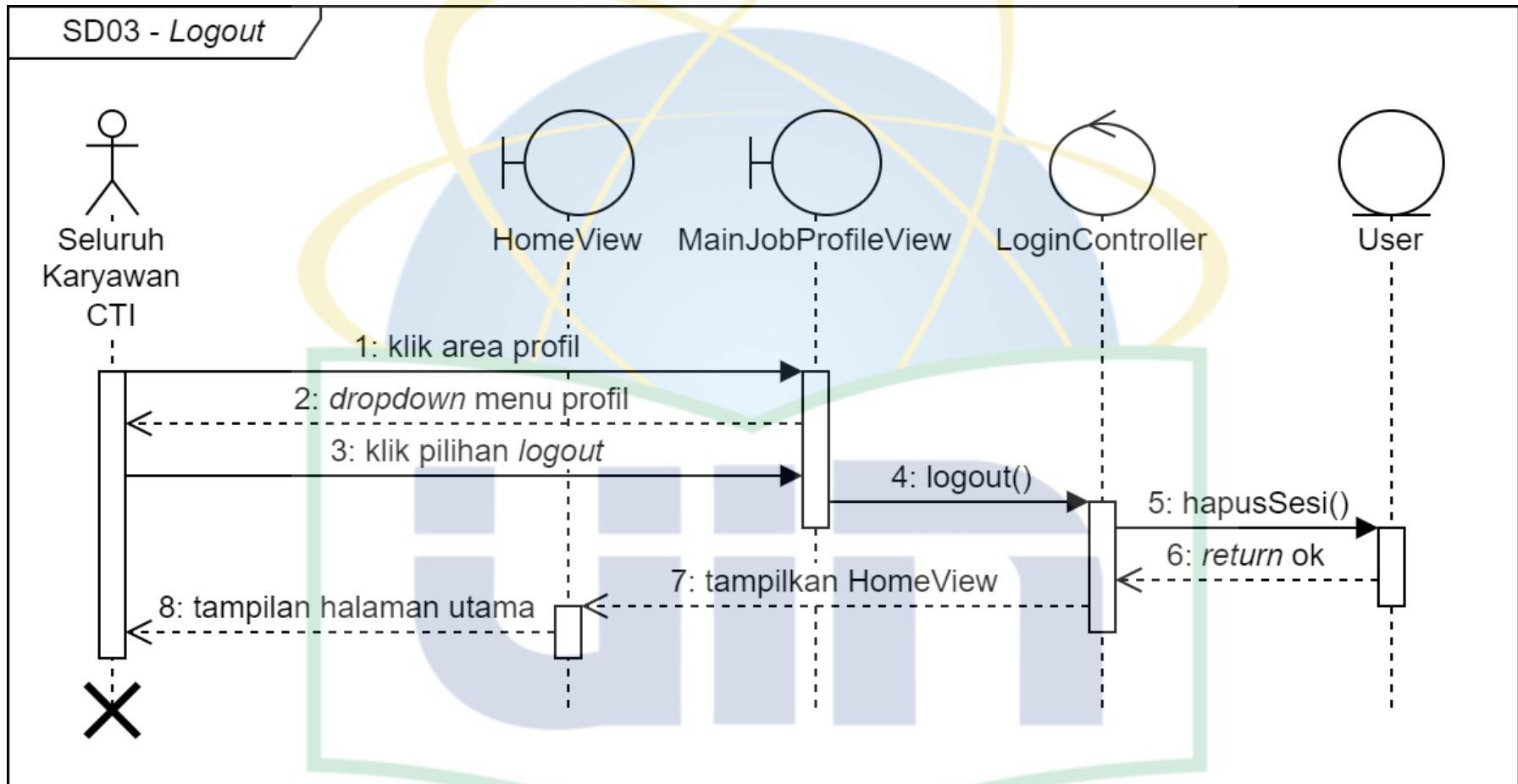
*Sequence Diagram* menggambarkan komunikasi pesan antara komponen aplikasi dan juga dengan pengguna/aktor. Penjelasan *Sequence Diagram* dibahas di setiap Spesifikasi *Use Case* yang telah dibuat sebelumnya.



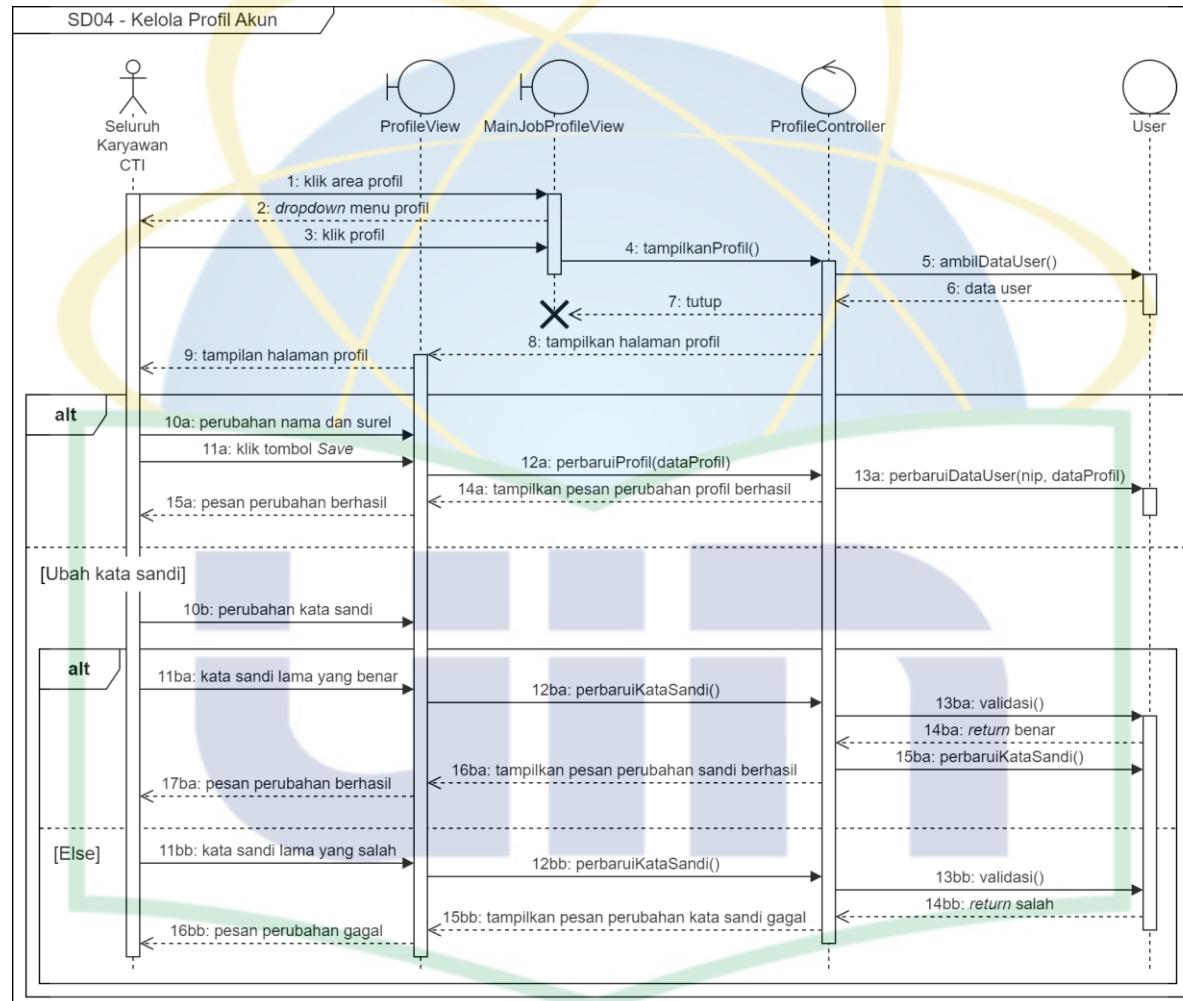
**Gambar 4.57 Sequence Diagram Buka Halaman Awal**



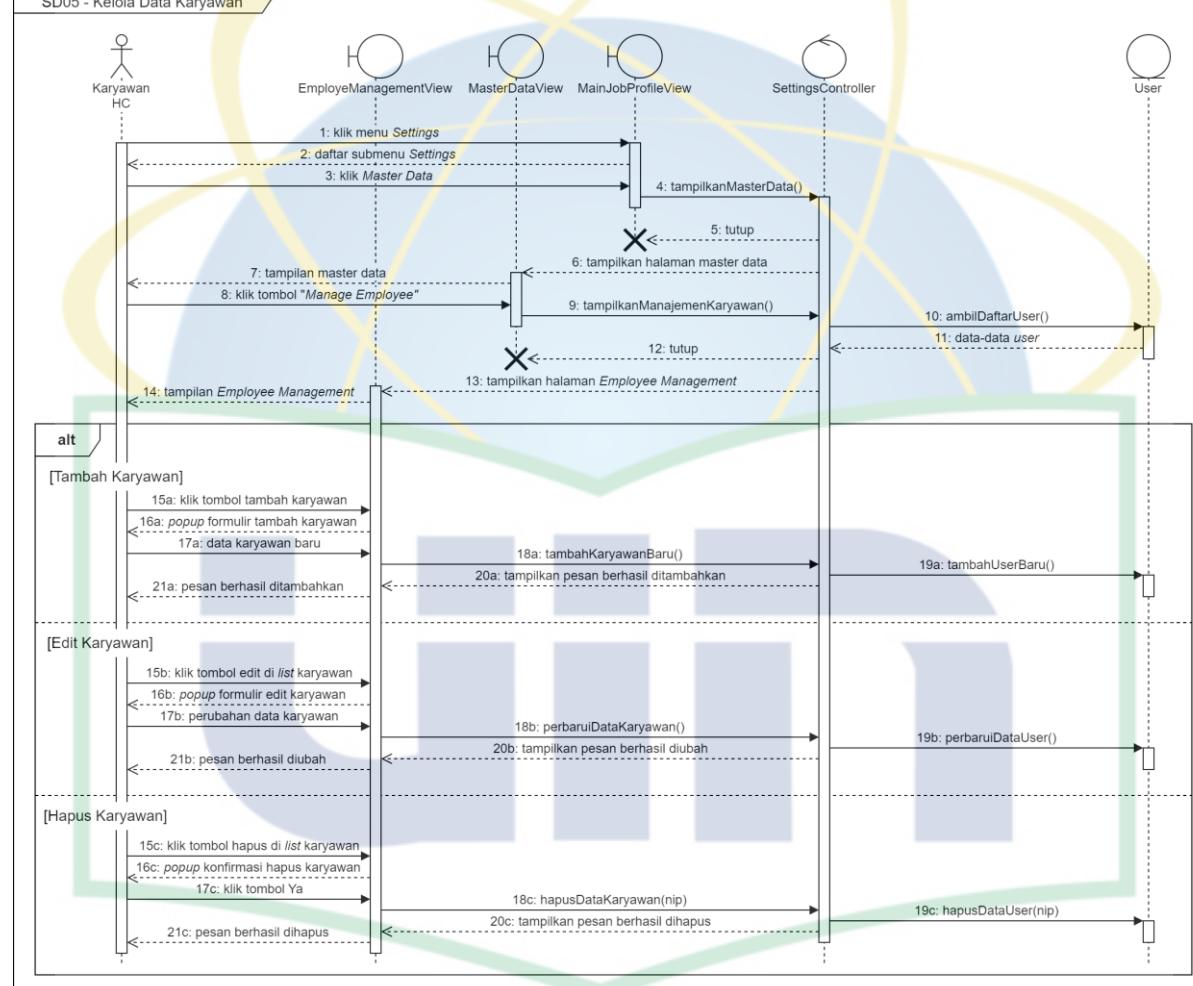
Gambar 4.58 Sequence Diagram Login



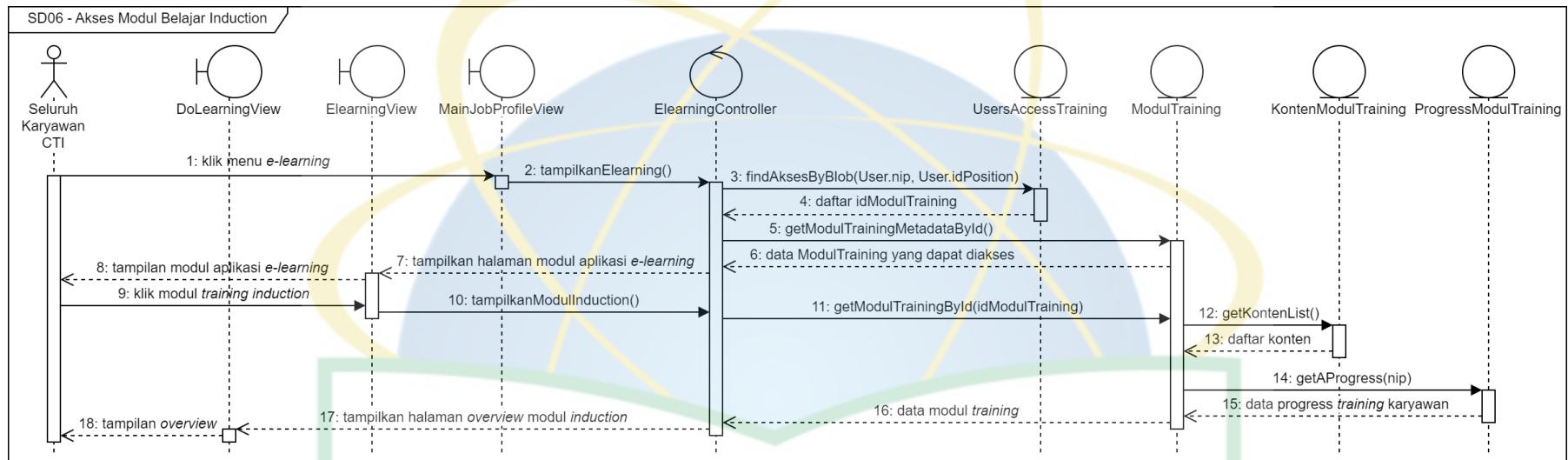
Gambar 4.59 Sequence Diagram Logout



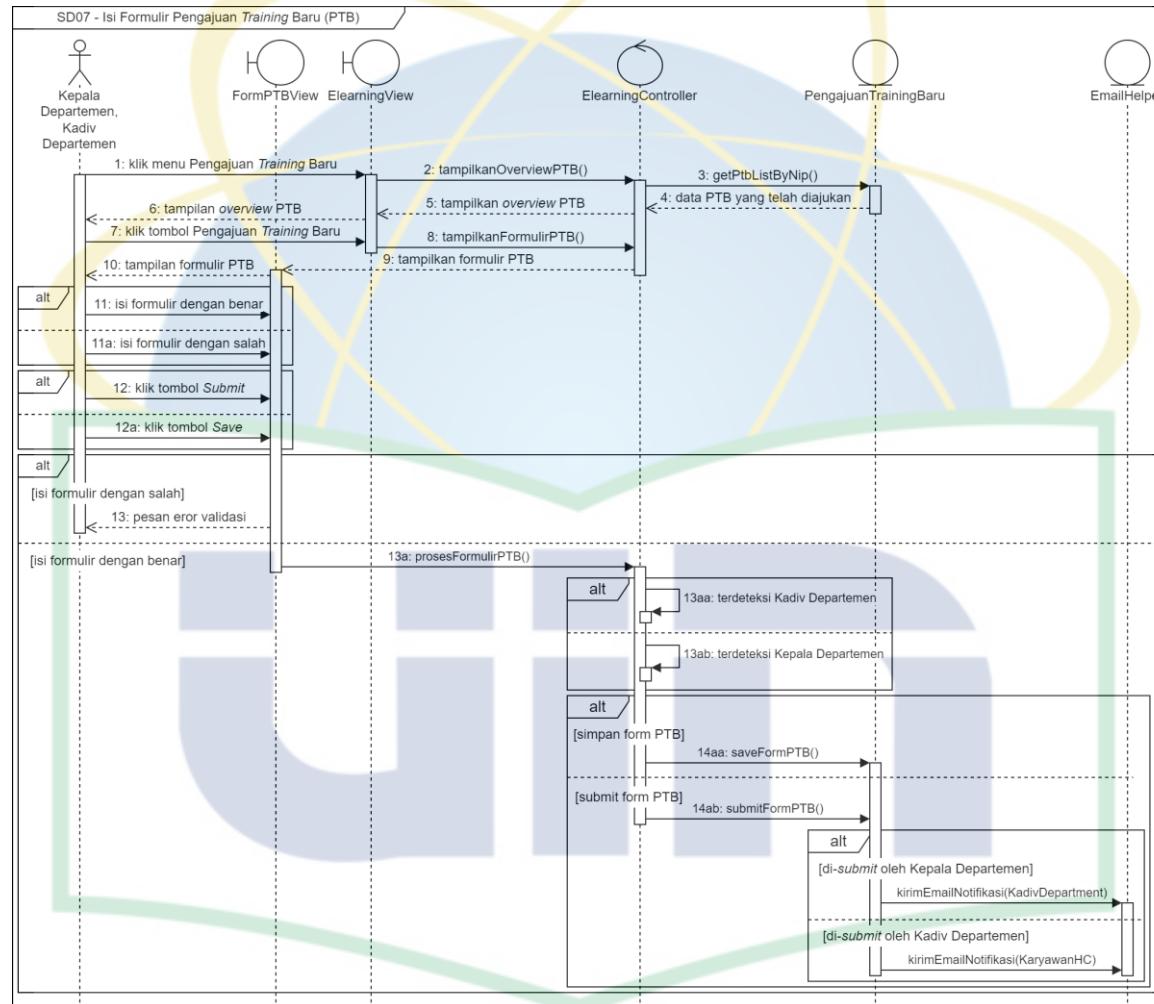
Gambar 4.60 Sequence Diagram Kelola Profil Akun



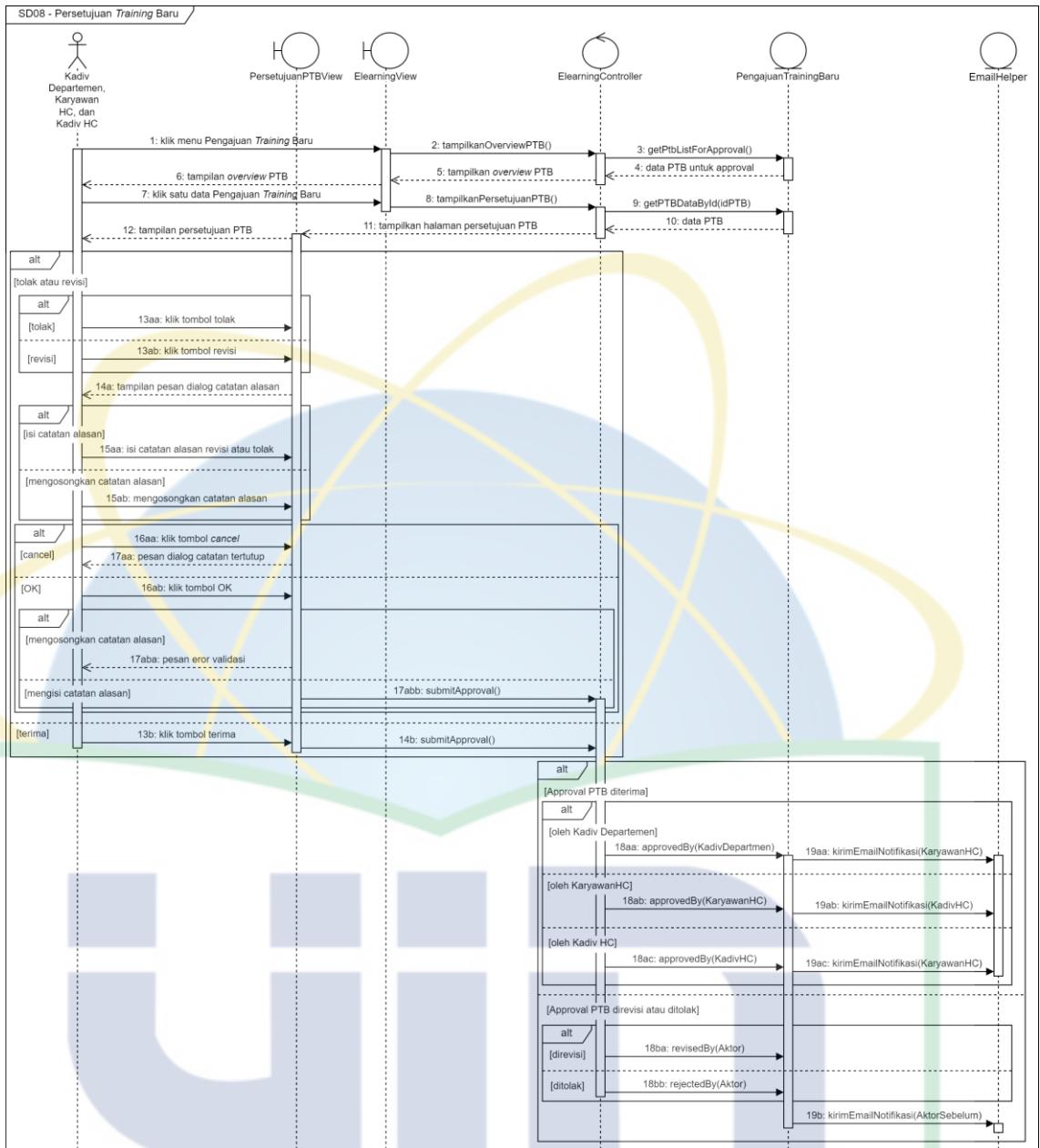
Gambar 4.61 Sequence Diagram Kelola Data Karyawan



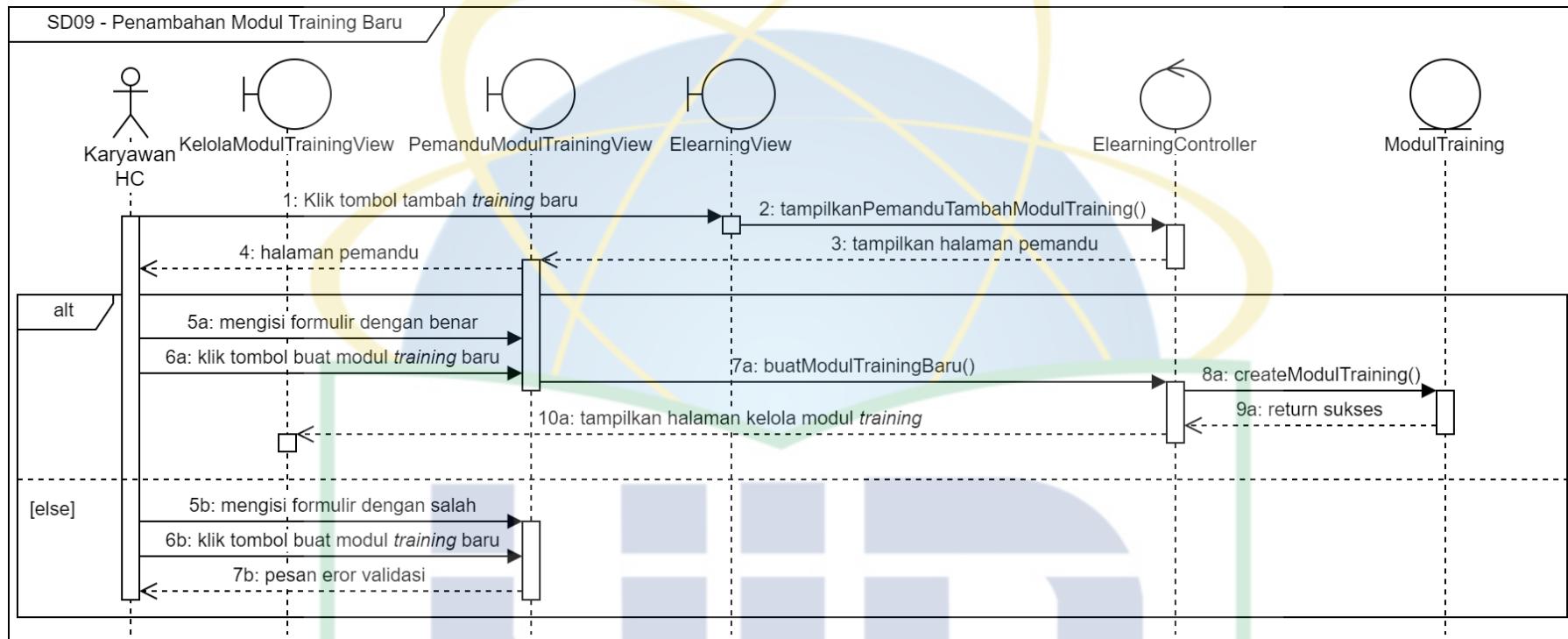
Gambar 4.62 Sequence Diagram Akses Modul Belajar Induction



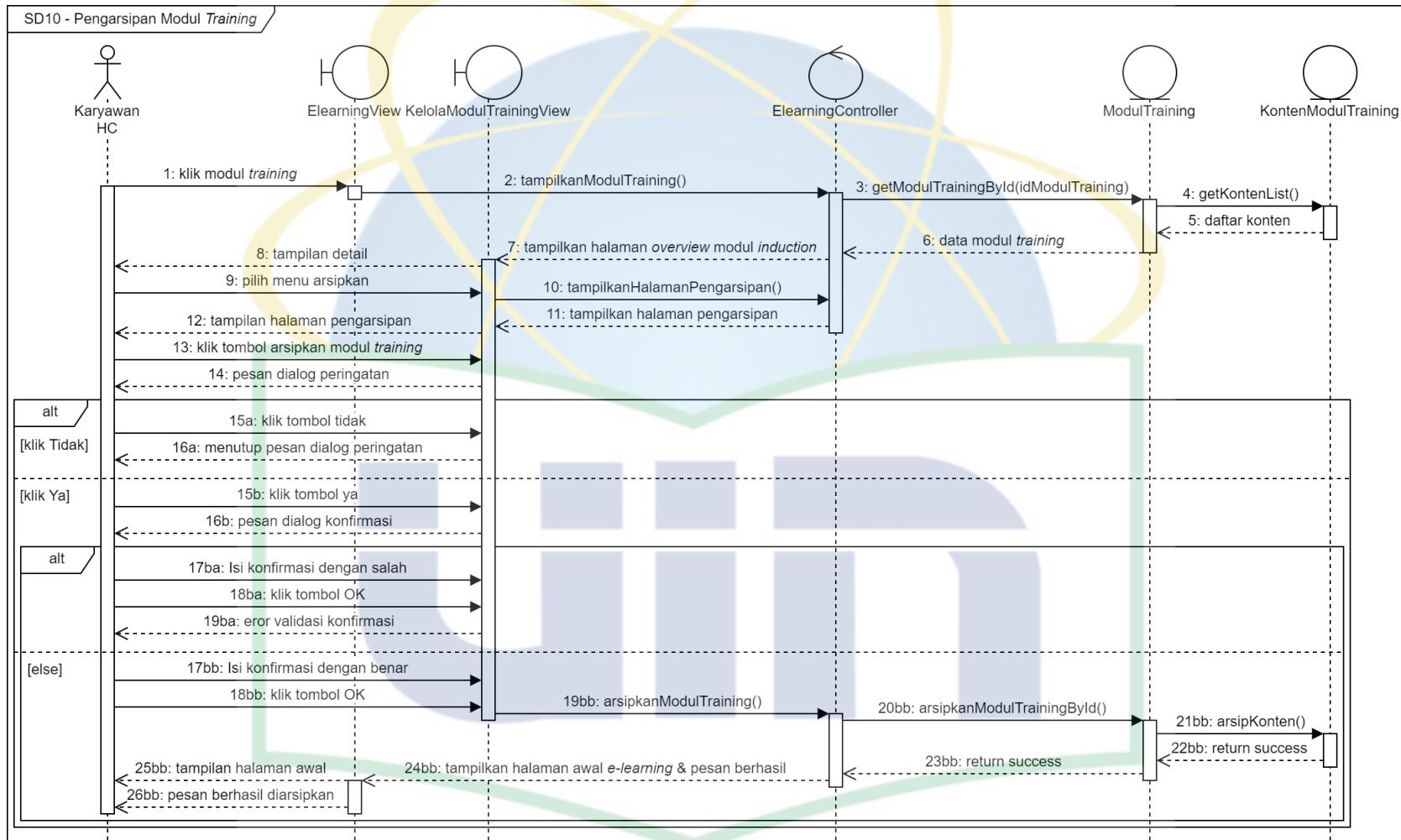
Gambar 4.63 Sequence Diagram Isi Formulir Pengajuan Training Baru (PTB)



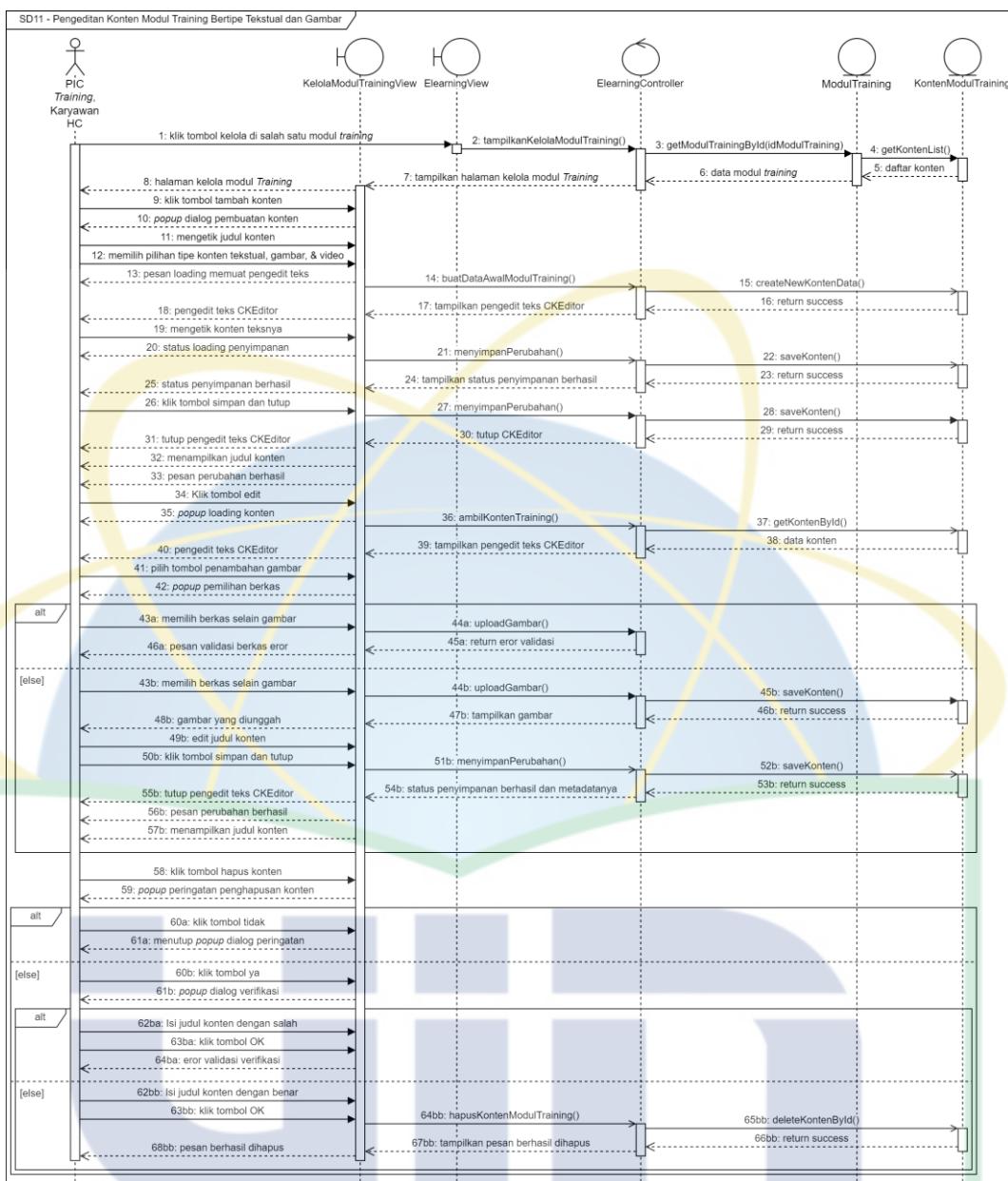
Gambar 4.64 Sequence Diagram Persetujuan Training Baru



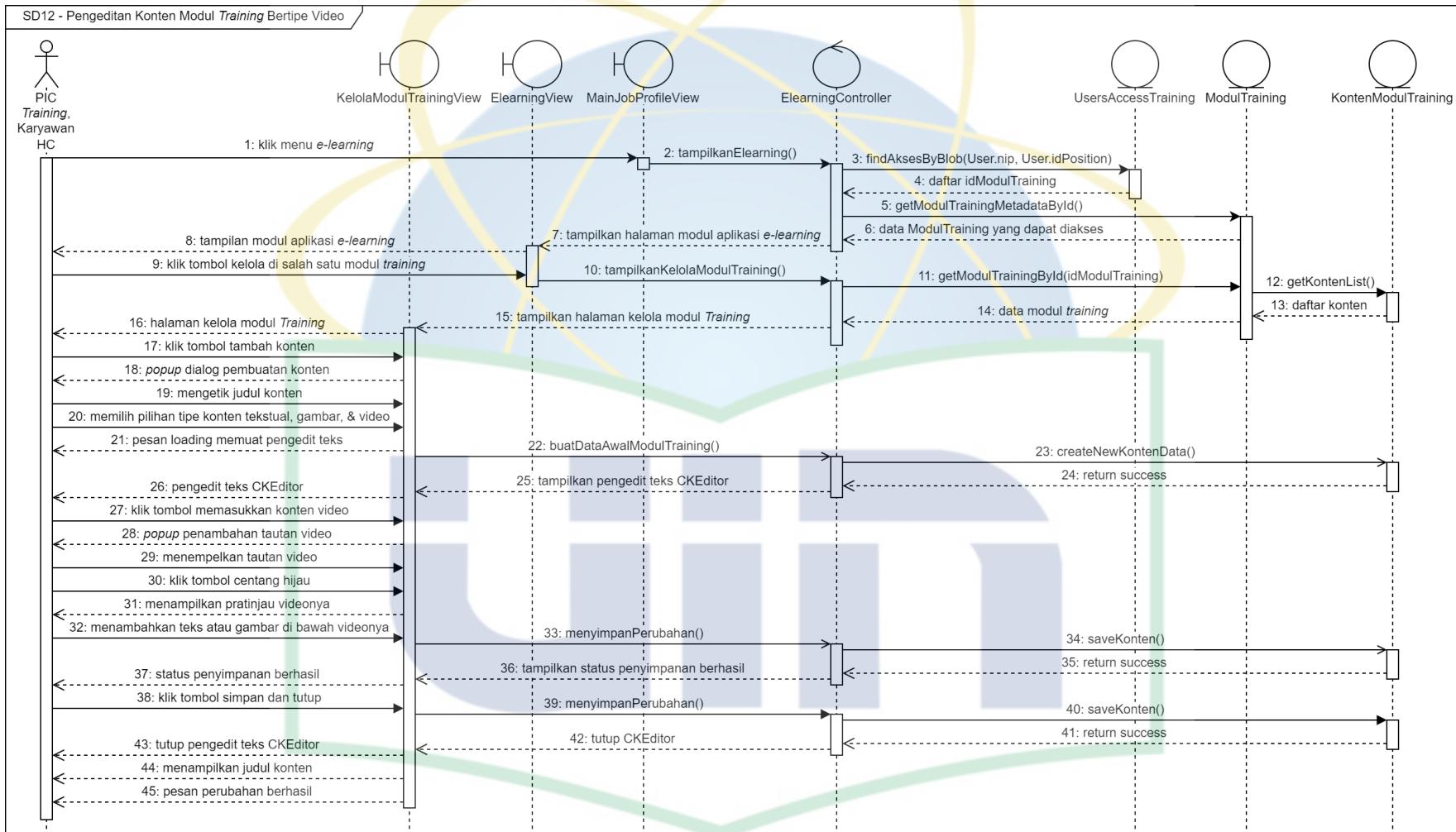
Gambar 4.65 Sequence Diagram Penambahan Modul Training Baru



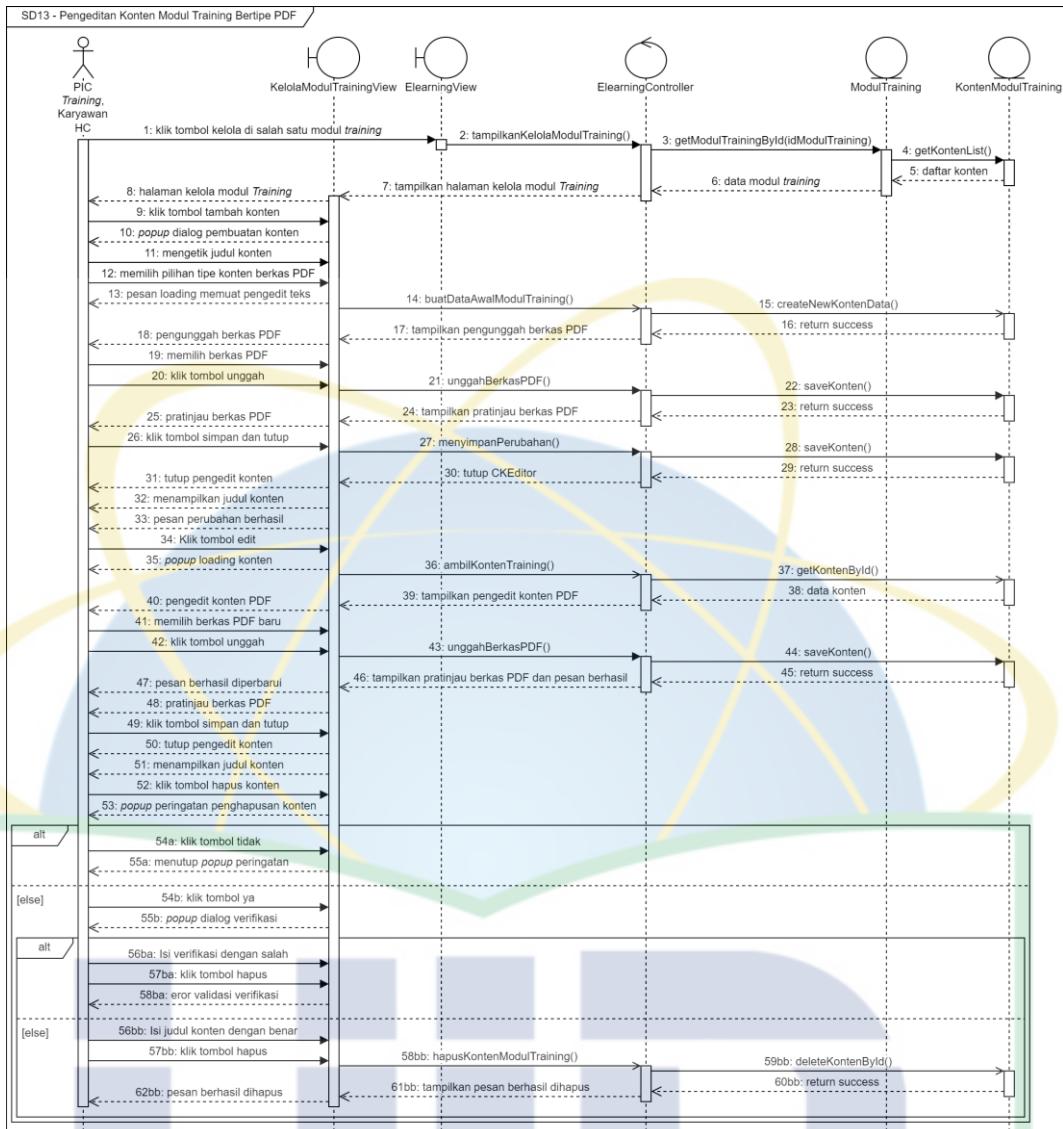
Gambar 4.66 Sequence Diagram Pengarsipan Modul Training



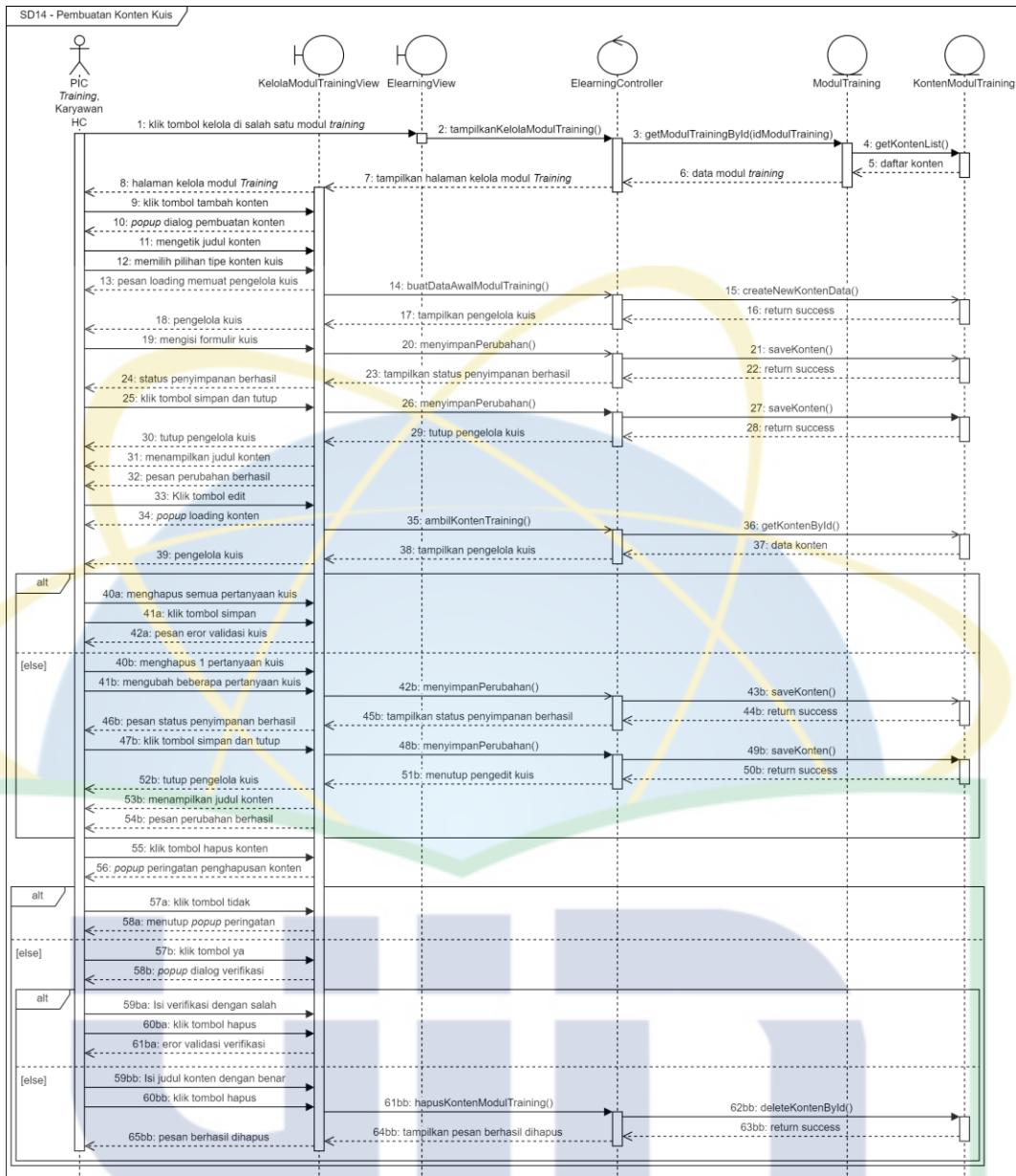
Gambar 4.67 Sequence Diagram Pengeditan Konten Modul Training Bertipe Tekstual dan Gambar



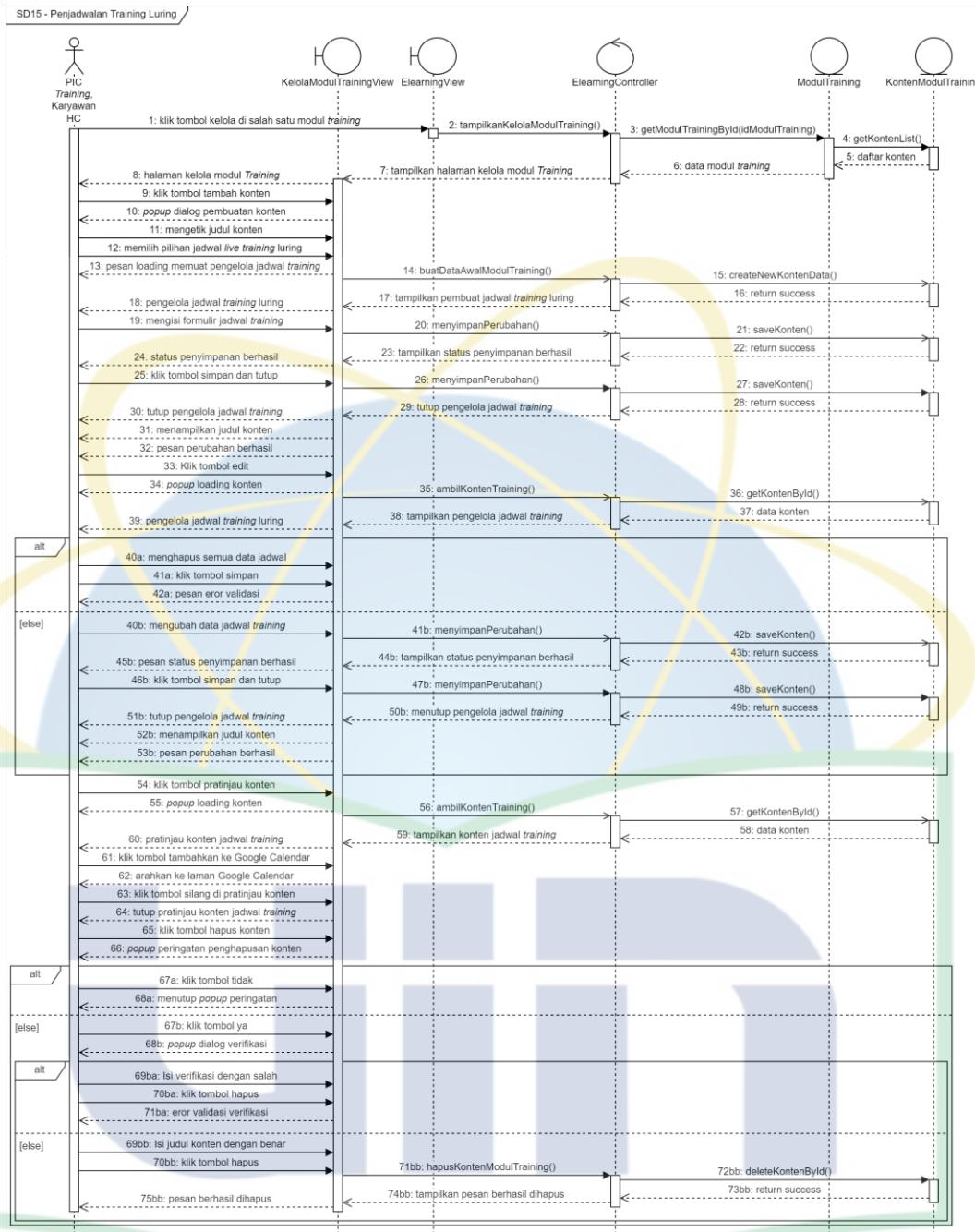
Gambar 4.68 Sequence Diagram Pengeditan Konten Modul Training Bertipe Video



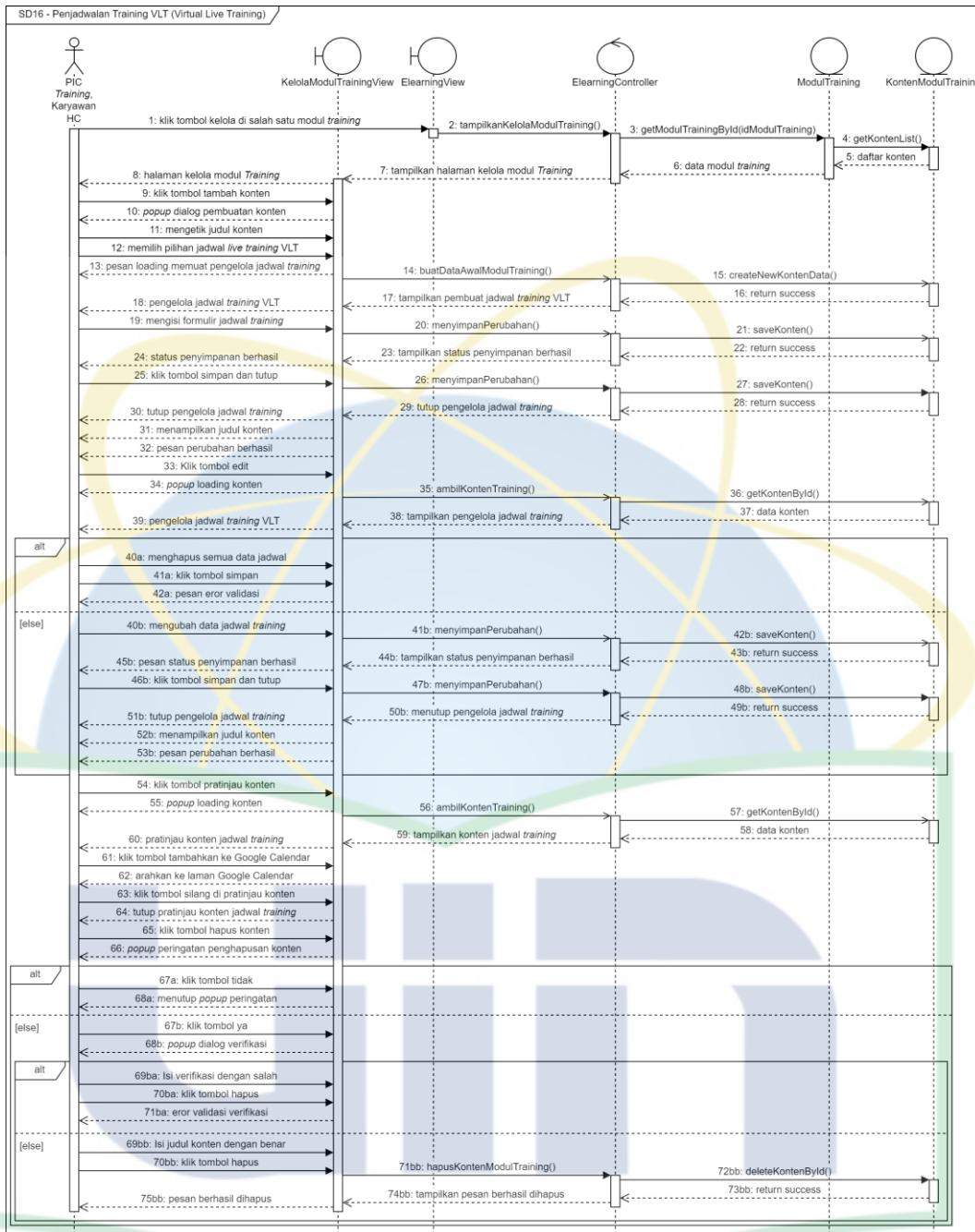
Gambar 4.69 Sequence Diagram Pengeditan Konten Modul Training Bertipe PDF



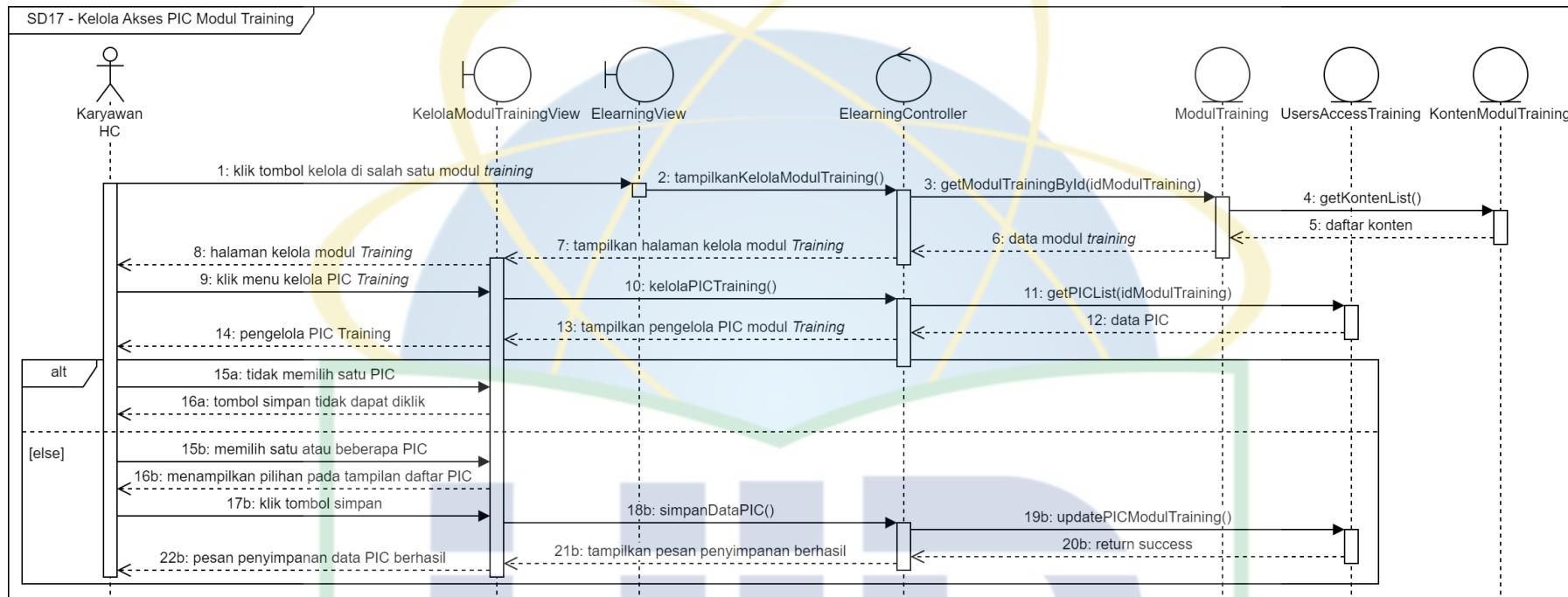
Gambar 4.70 Sequence Diagram Pembuatan Konten Kuis



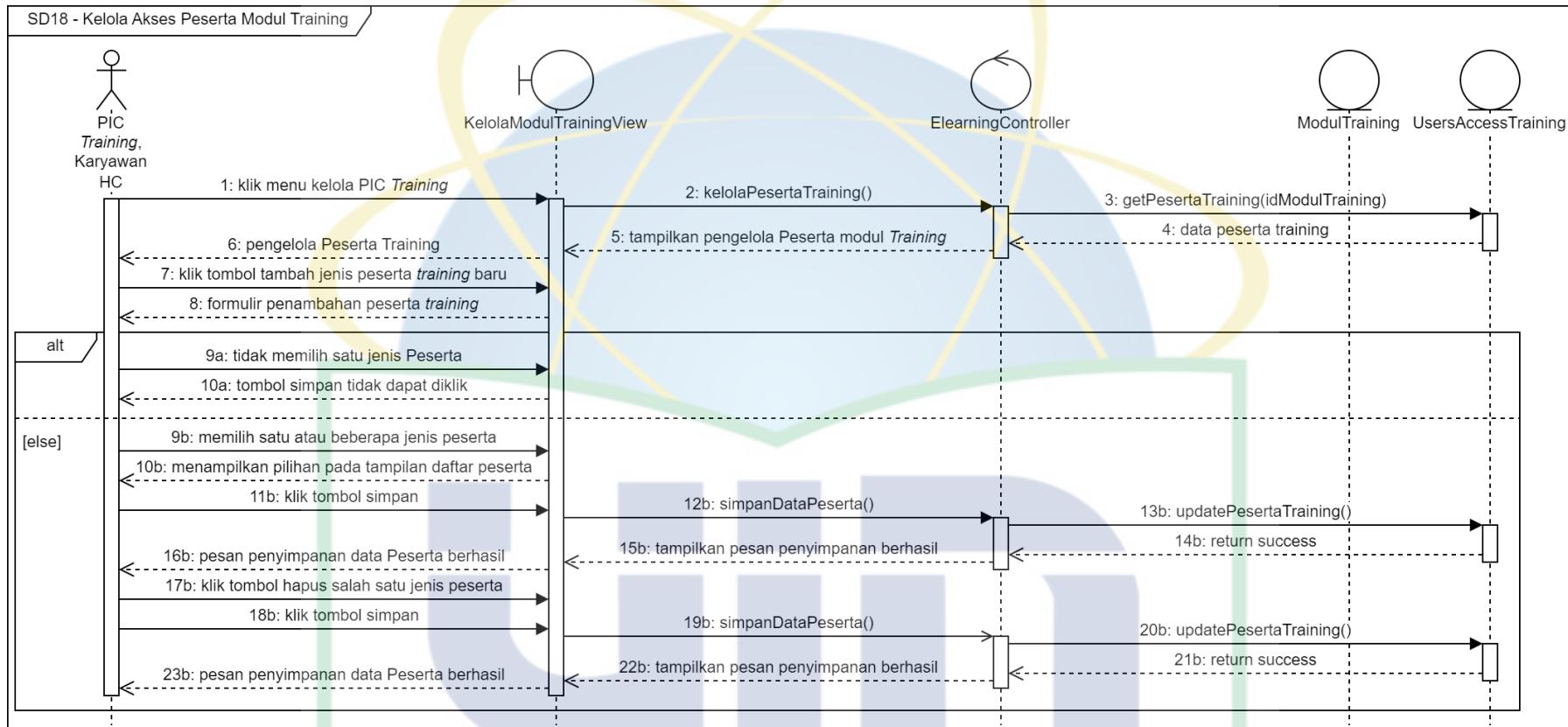
Gambar 4.71 Sequence Diagram Penjadwalan Training Luring



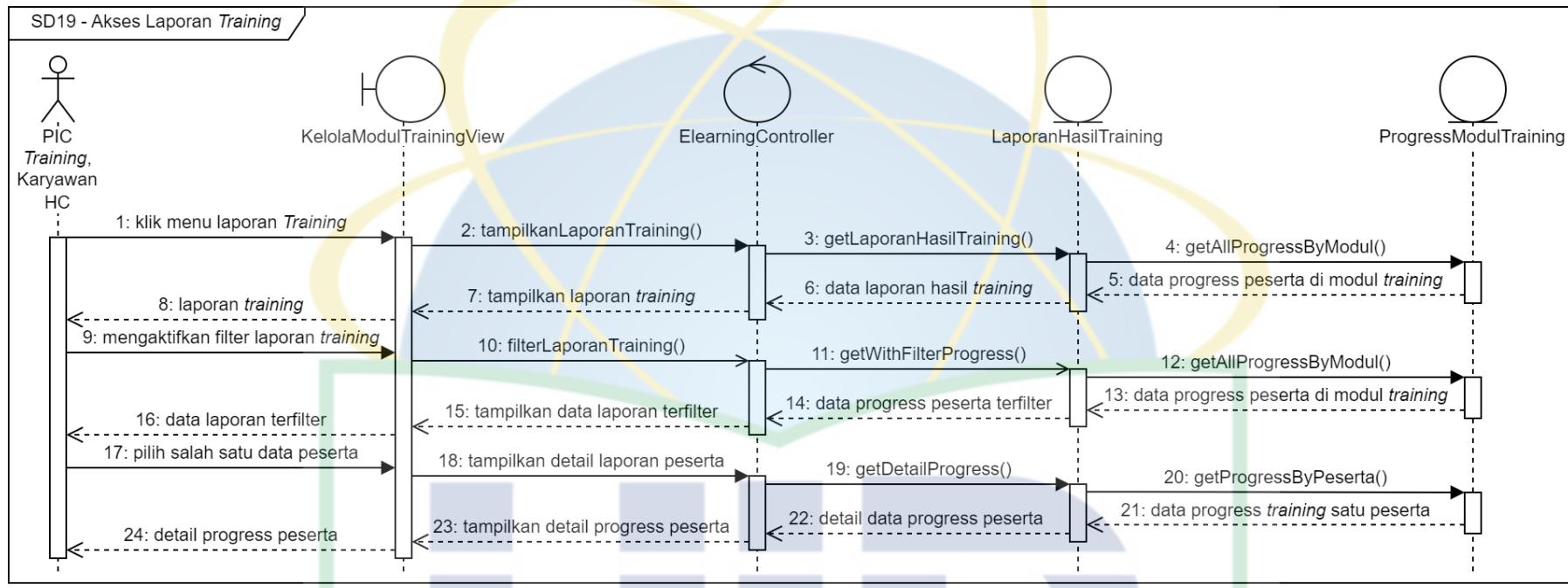
Gambar 4.72 Sequence Diagram Penjadwalan VLT (Virtual Live Training)



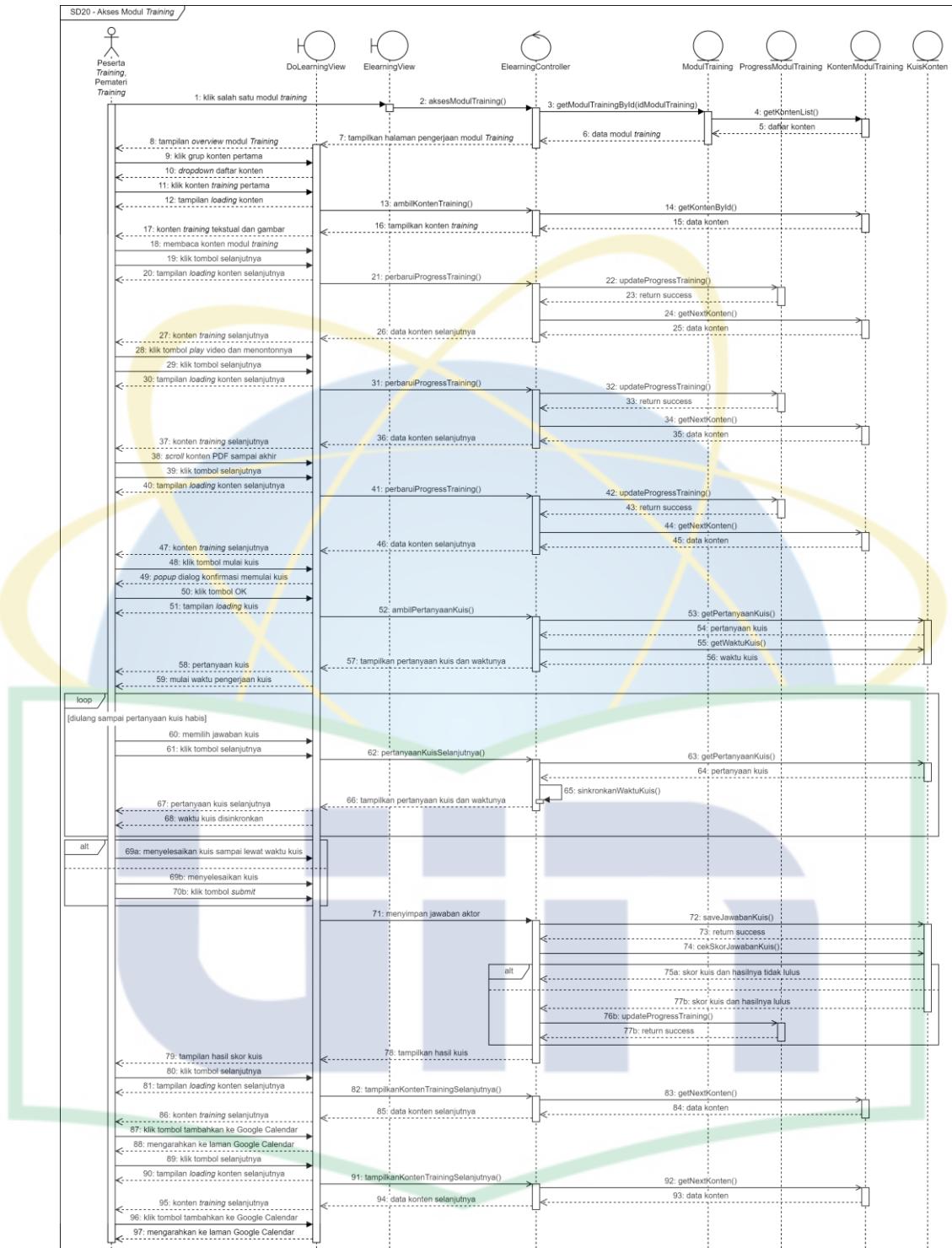
Gambar 4.73 Sequence Diagram Kelola Akses PIC Modul Training



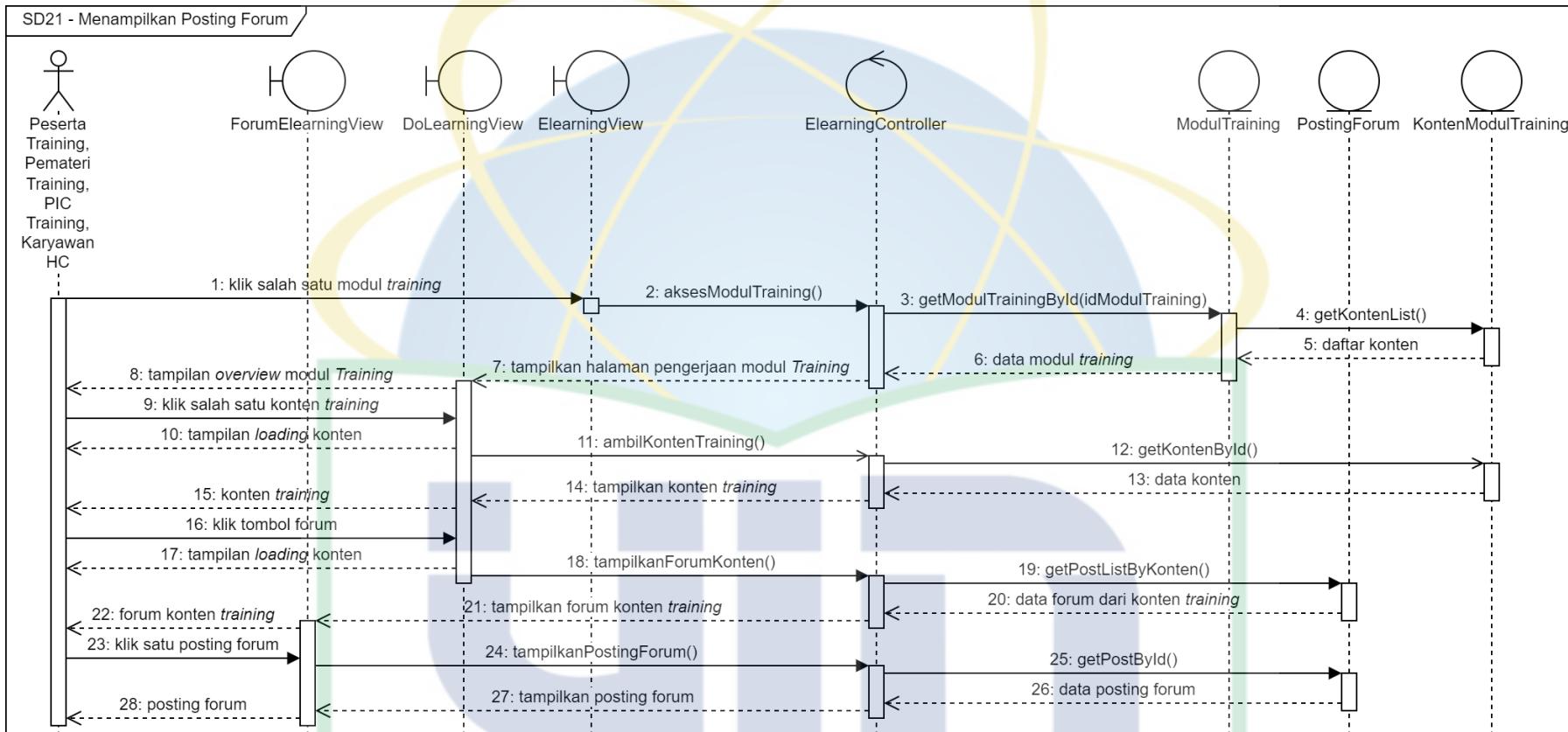
Gambar 4.74 Sequence Diagram Kelola Akses Peserta Modul Training



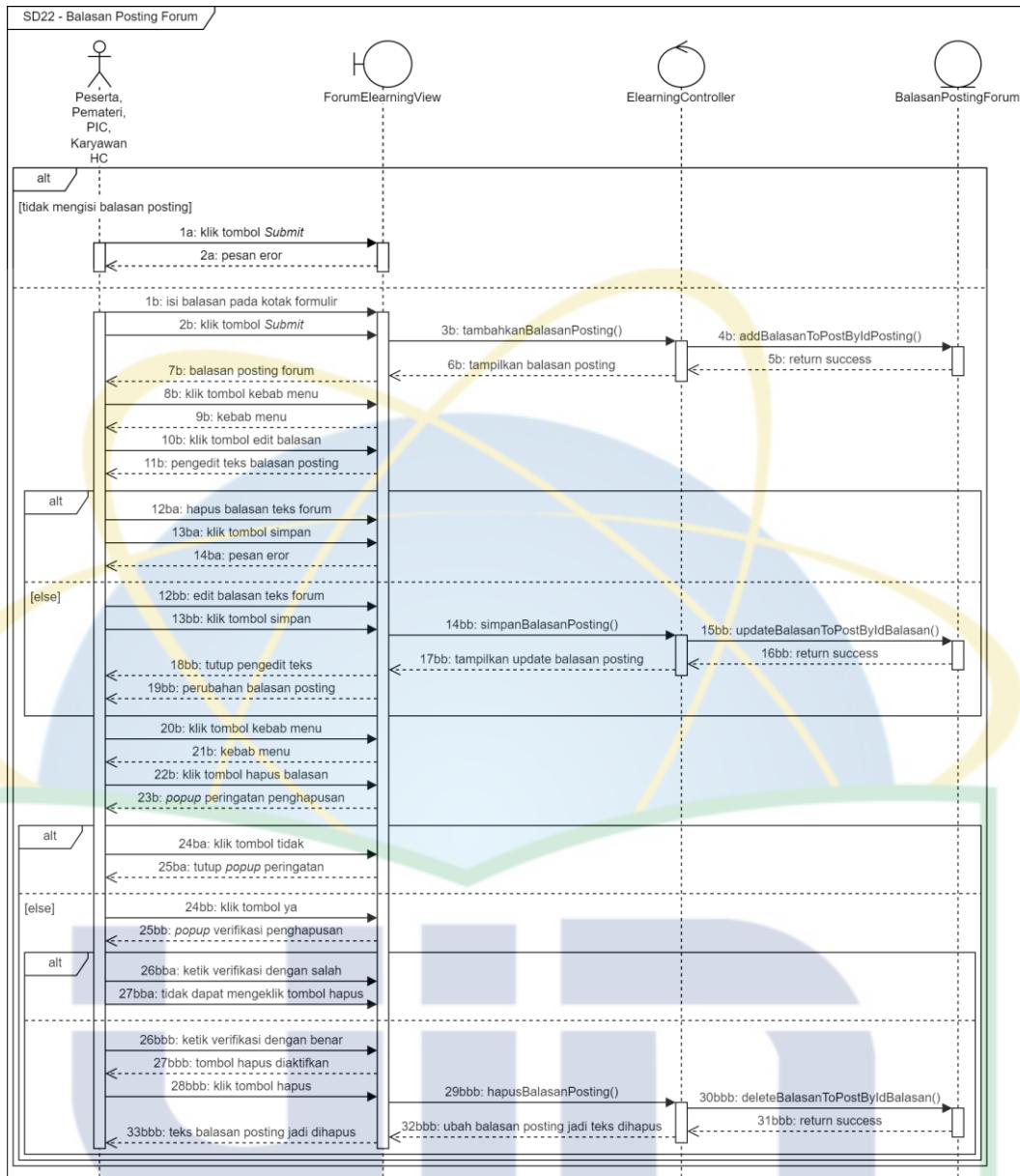
Gambar 4.75 Sequence Diagram Akses Laporan Training



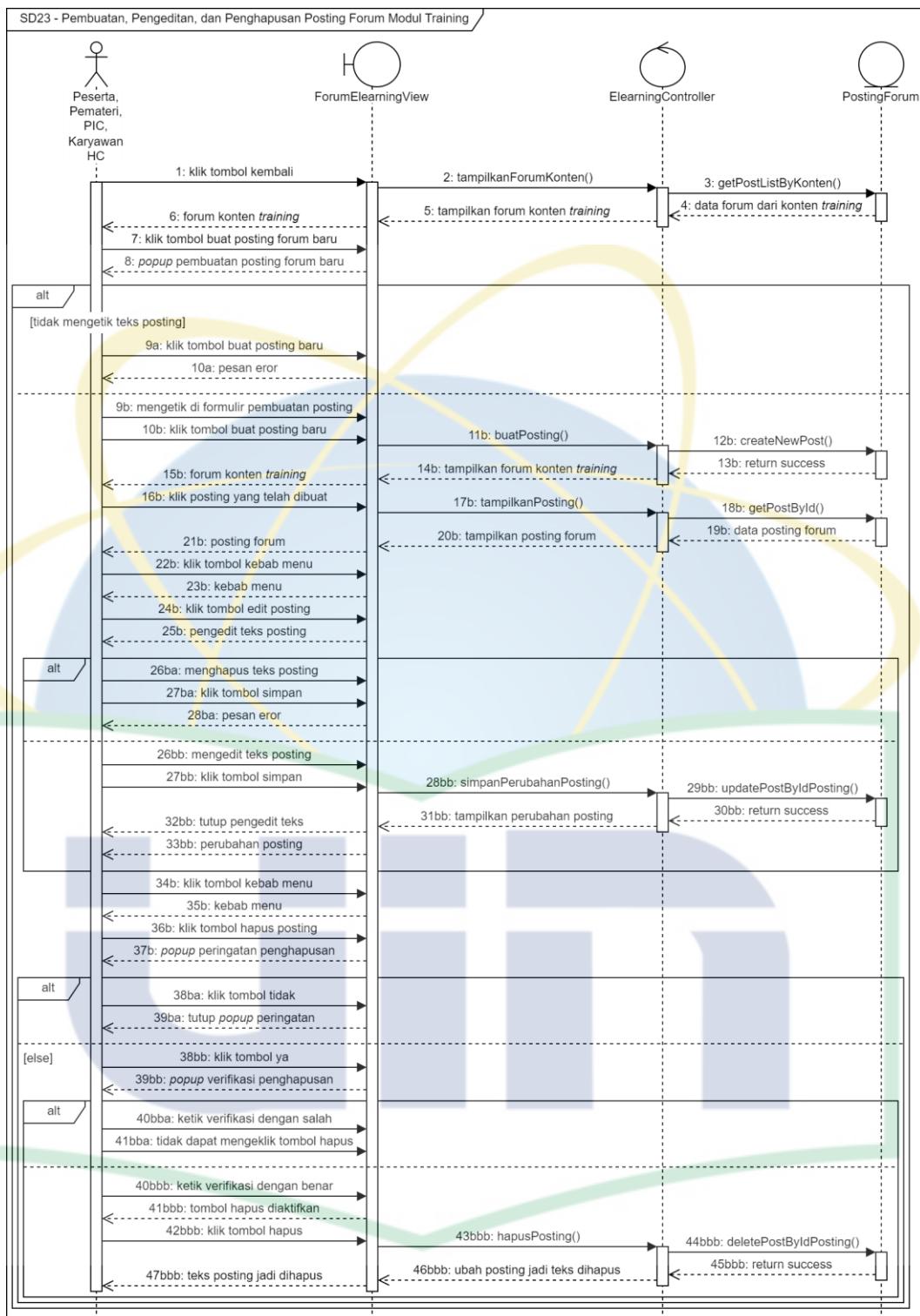
Gambar 4.76 Sequence Diagram Akses Modul Training



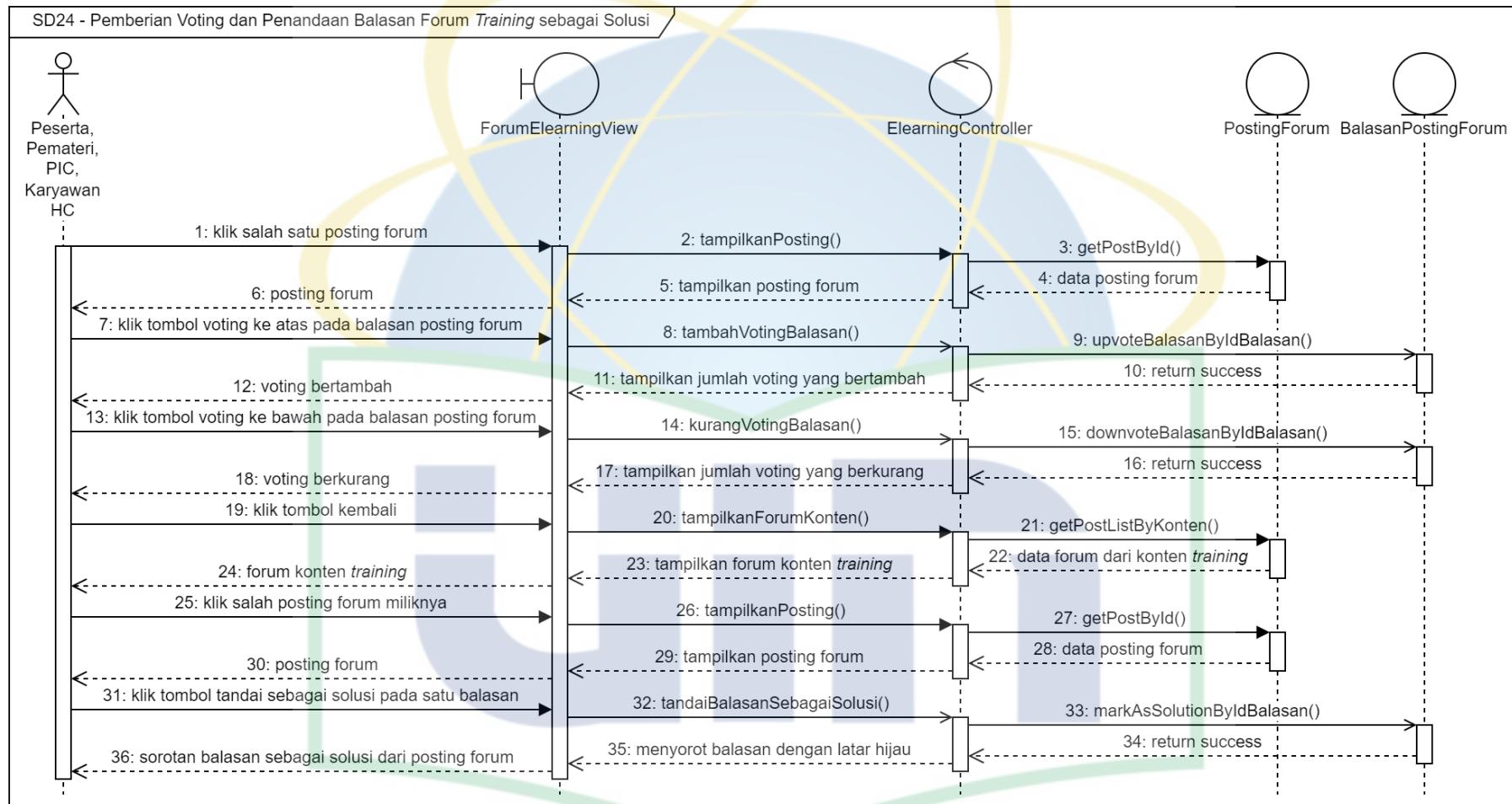
Gambar 4.77 Sequence Diagram Menampilkan Posting Forum



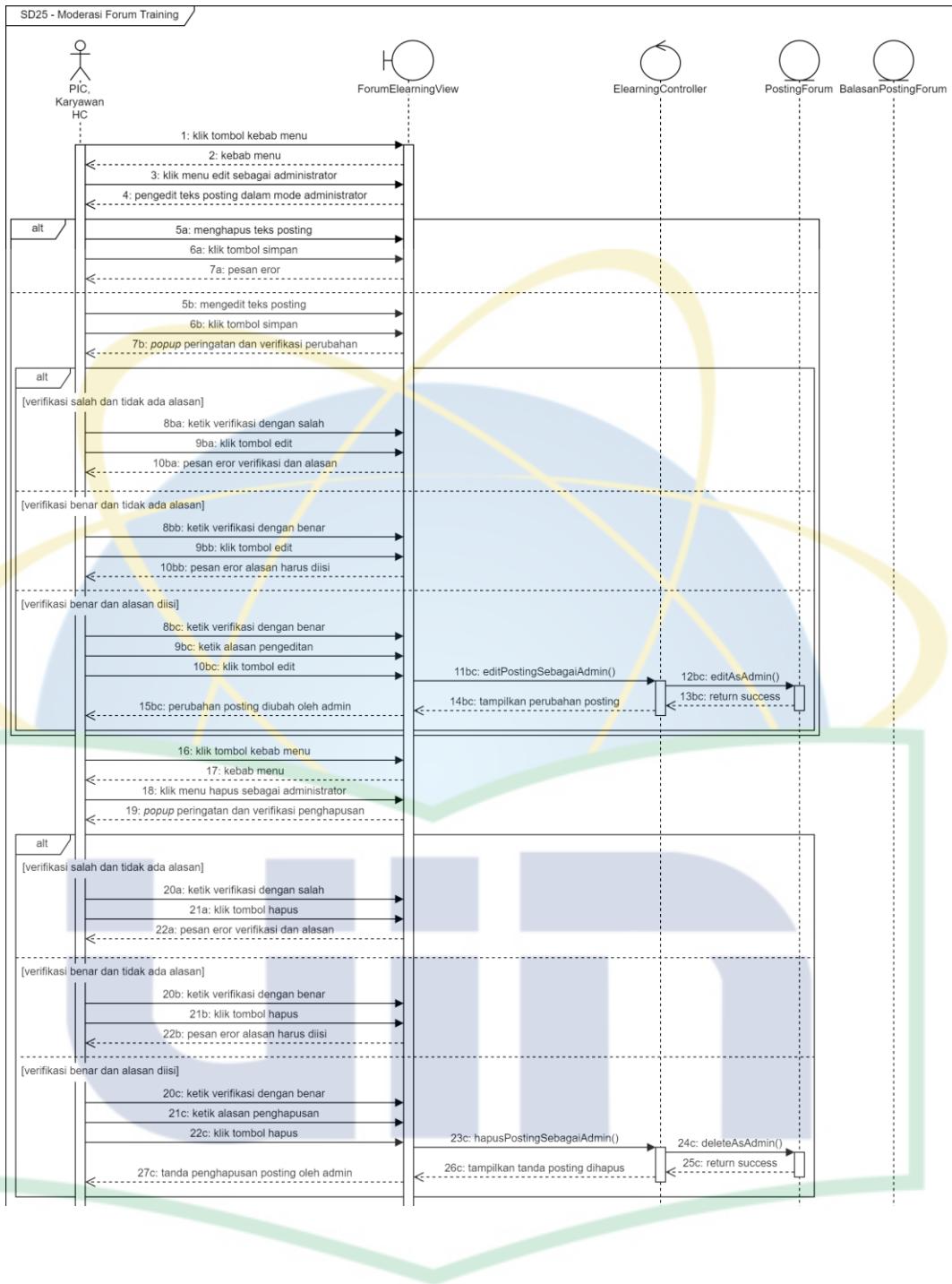
Gambar 4.78 Sequence Diagram Balasan Posting Forum

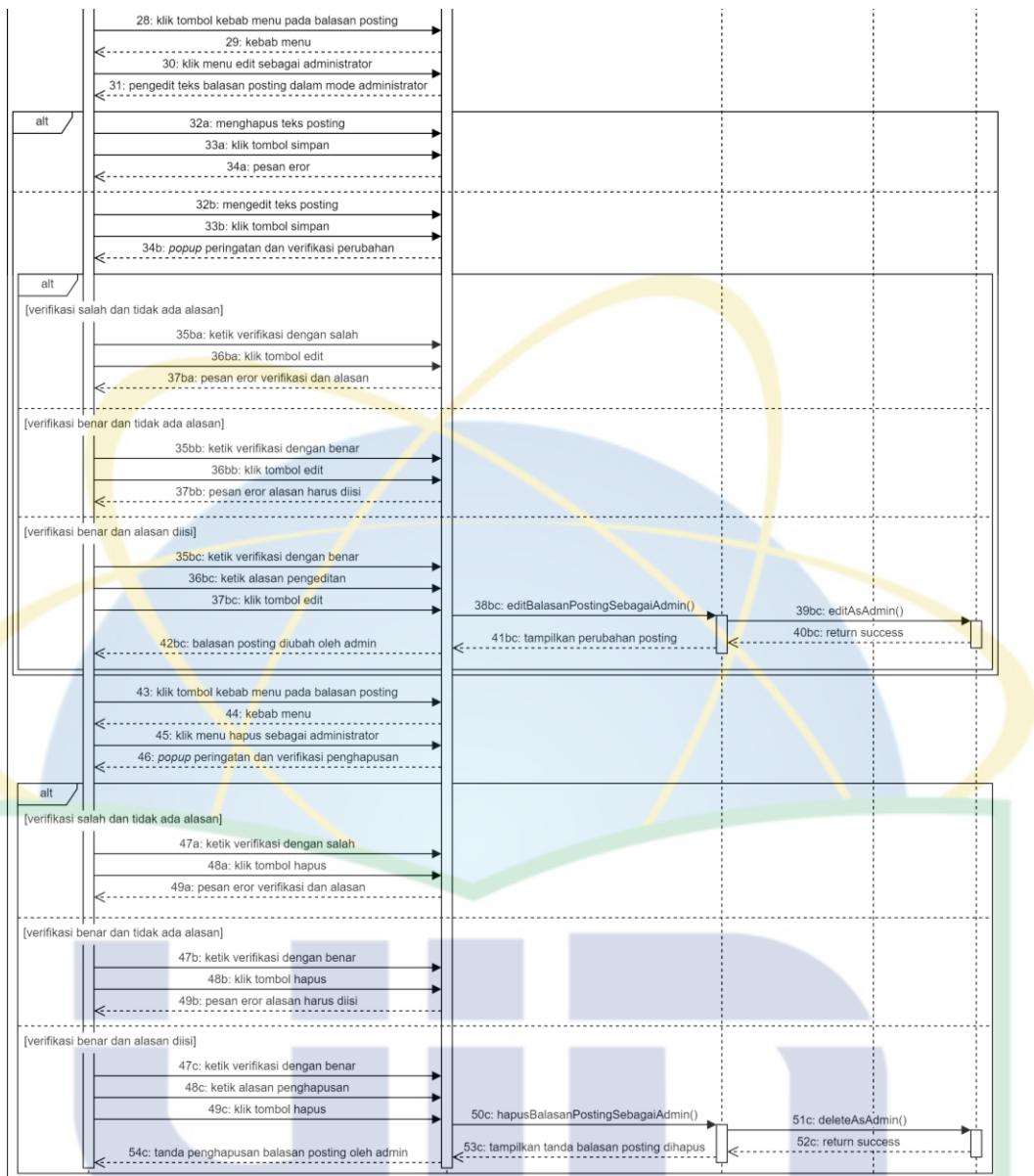


**Gambar 4.79 Sequence Diagram Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul Training**



Gambar 4.80 Sequence Diagram Pemberian Voting dan Penandaan Balasan Forum *Training* sebagai Solusi



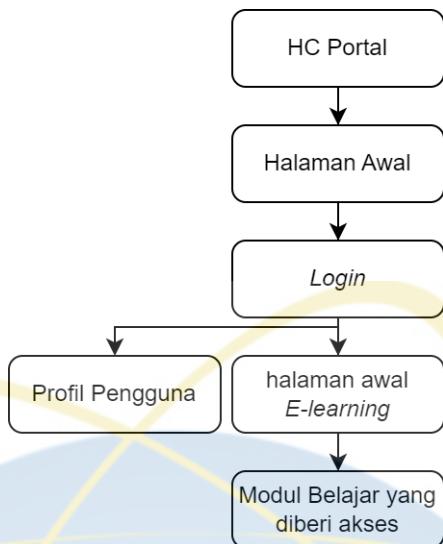


Gambar 4.81 Sequence Diagram Moderasi Forum Training

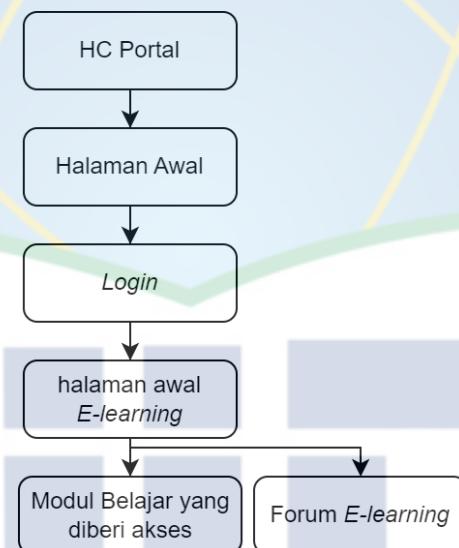
## D. Desain Tampilan

### 1. Struktur Menu

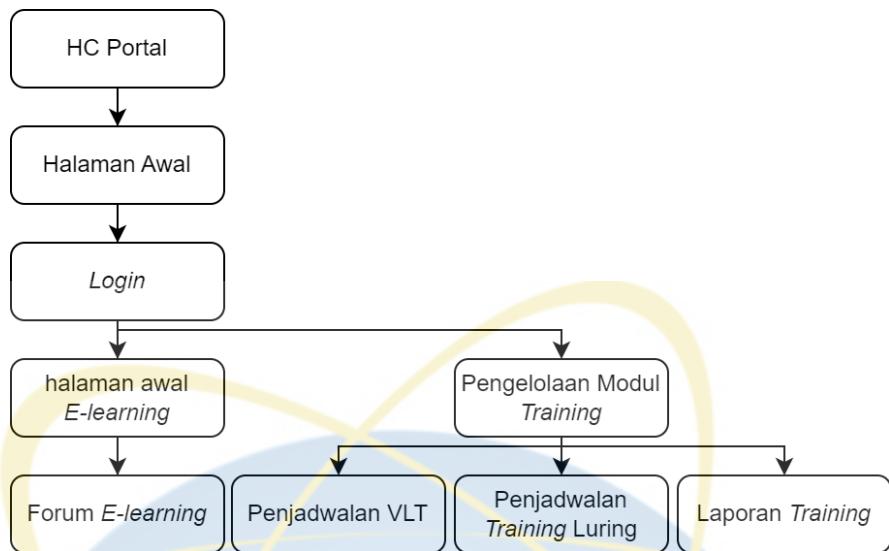
Penulis membuat struktur menu untuk mengidentifikasi tampilan aplikasi (*User Interface/UI*) yang dapat dibuat. Pembahasannya dibagi dalam per aktor untuk memperlihatkan halaman aplikasi yang dapat diakses. Jumlah aktor yang terdapat pada pengembangan modul *e-learning* ini adalah 8 aktor dengan 2 aktor yang memiliki jenis pengaksesan yang sama, yakni Pemateri dan Peserta *Training*.



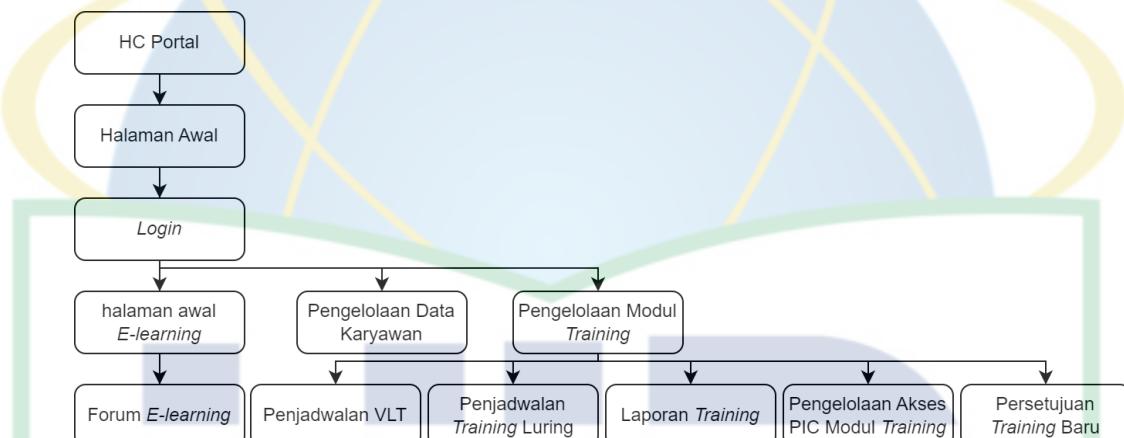
**Gambar 4.82 Struktur Menu untuk Karyawan CTI**



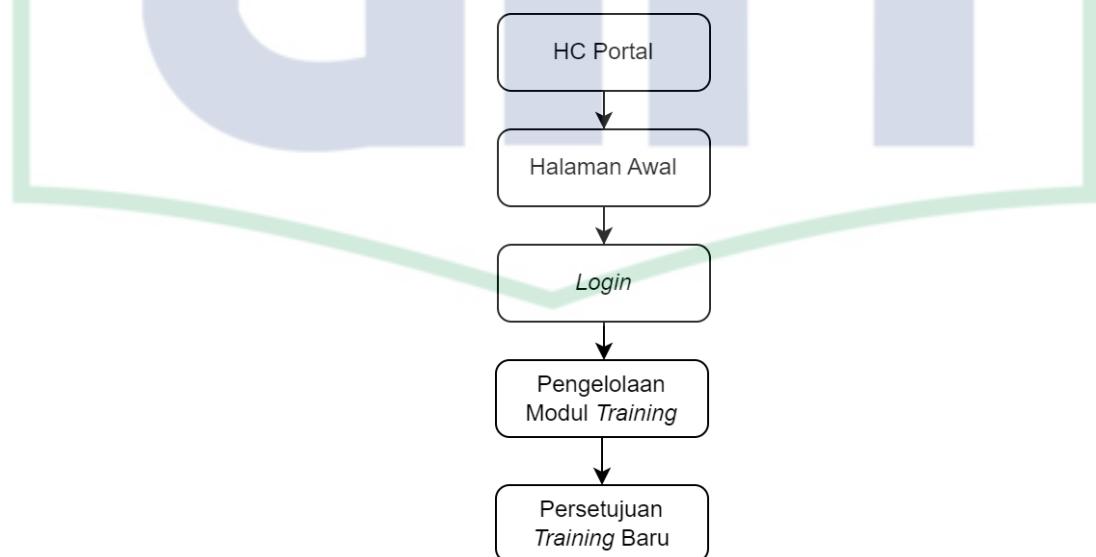
**Gambar 4.83 Struktur Menu untuk Peserta dan Pemateri Training**



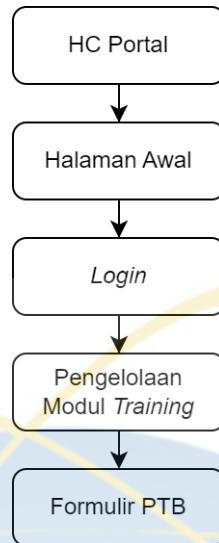
**Gambar 4.84 Struktur Menu untuk PIC Training**



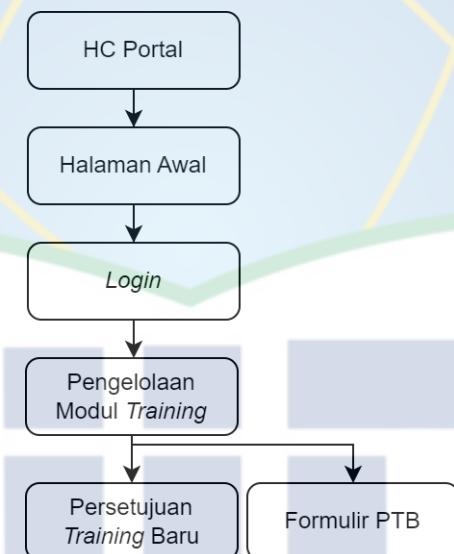
**Gambar 4.85 Struktur Menu untuk Karyawan HC**



**Gambar 4.86 Struktur Menu untuk Kadiv HC**



Gambar 4.87 Struktur Menu untuk Kepala Departemen



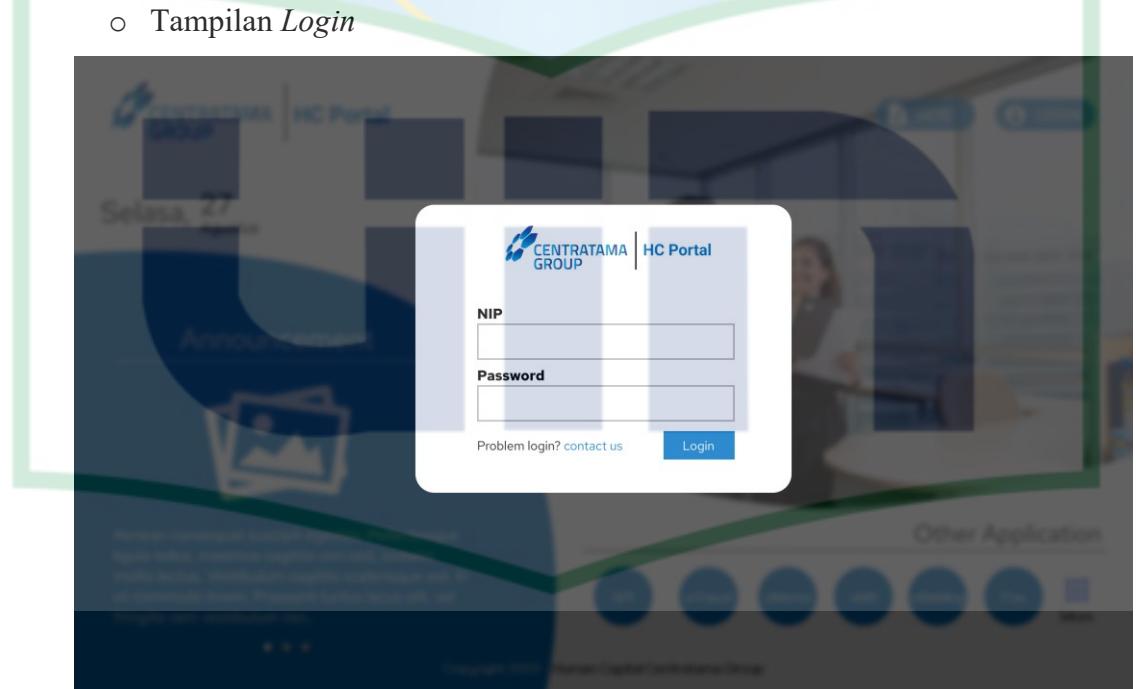
Gambar 4.88 Struktur Menu untuk Kadiv Departemen

## 2. User Interface (UI)

- Tampilan Halaman Awal Aplikasi HC Portal

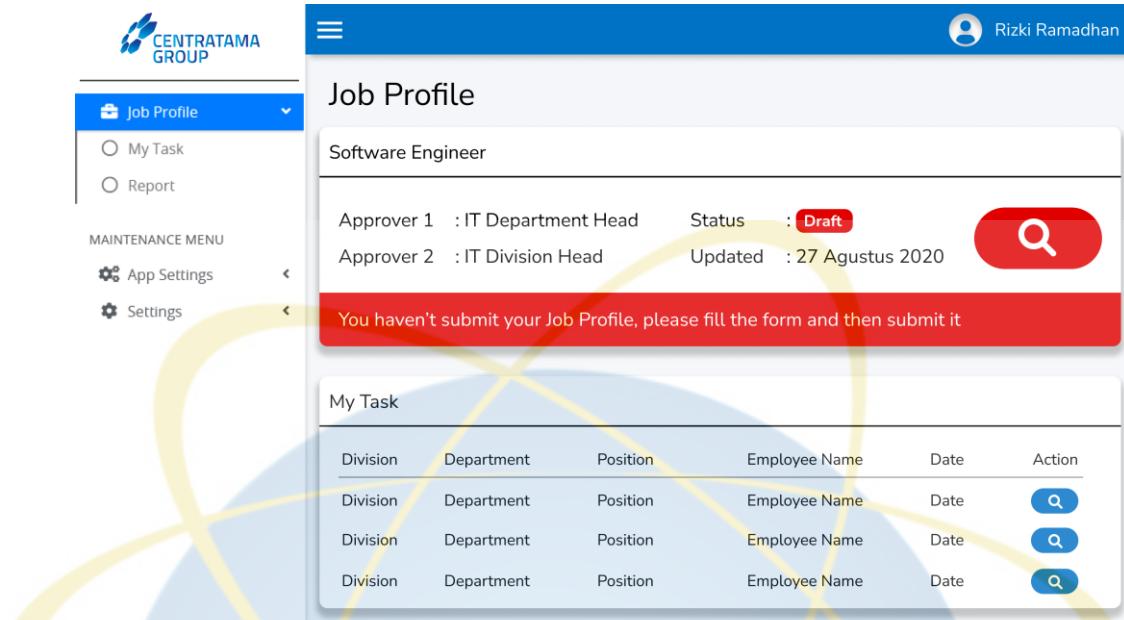


- Tampilan Login



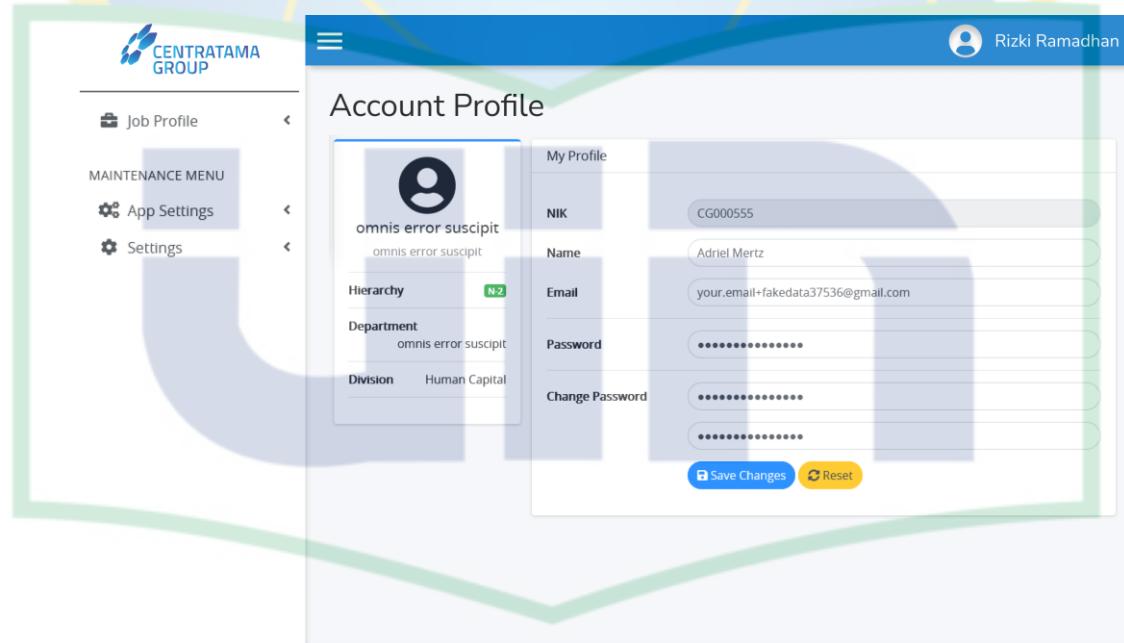
**Gambar 4.90 Tampilan Login Aplikasi berupa Popup (Login Popup)**

- Tampilan awal *Job Profile*



Gambar 4.91 Halaman Awal JP Setelah Karyawan Melakukan *Login* (MainJobProfileView)

- Tampilan Profil Pengguna



Gambar 4.92 Halaman Profil Pengguna (ProfileView)

- Tampilan *Master Data Management*

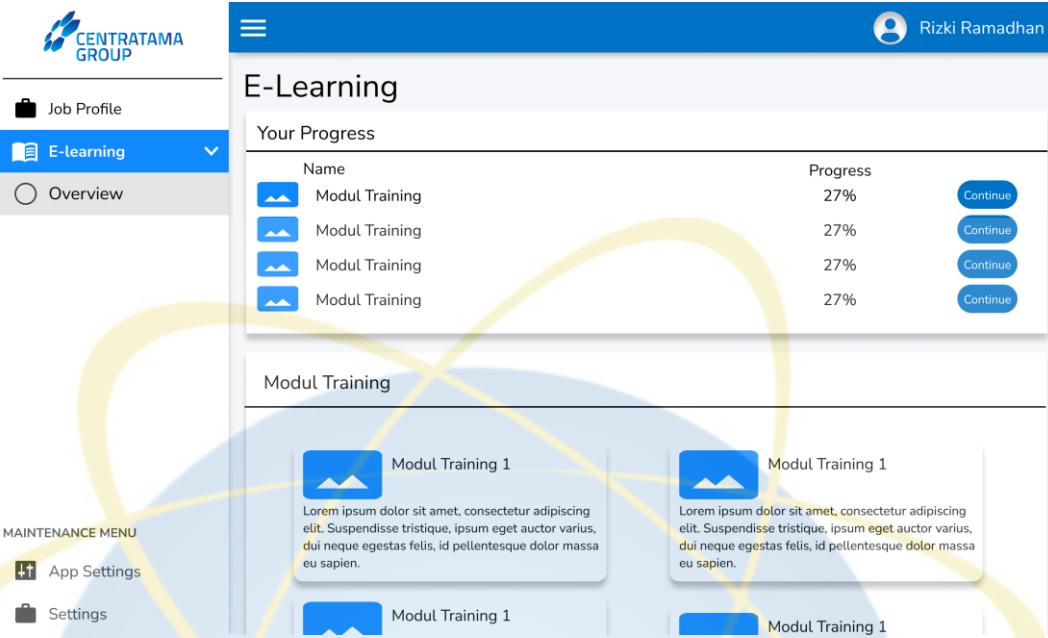


- Tampilan *Employee Data Management*

Nik	Employee Name's	Level Org.	Divisi	Departemen	Aksi
CG000001	Justo quisque conubia!	Taciti?	Orci.	Libero aenean!	
CG000002	Mattis fusce?	Sapien!	Velit.	Lacinia pellentesque.	
CG000003	Tortor dapibus.	Amet.	Est.	Bibendum aenean?	
CG000006	Etiam mollis maecenas.	Sapien!	Dictum?	Lacus lectus velit.	
CG000007	Aliquam mi.	Curabitur!	Vitae.	Dapibus vivamus quam?	
CG000008	Nunc integer!	Mi!	Mauris!	Nibh orci.	
CG000009	Habitant dictumst.	Est!	Libero!	Quisque consectetur eros.	
CG000010	Mi donec.	Ut!	Diam.	Ultrices gravida?	
CG000011	Nullam suscipit.	Condimentum.	Auctor!	Libero laoreet?	

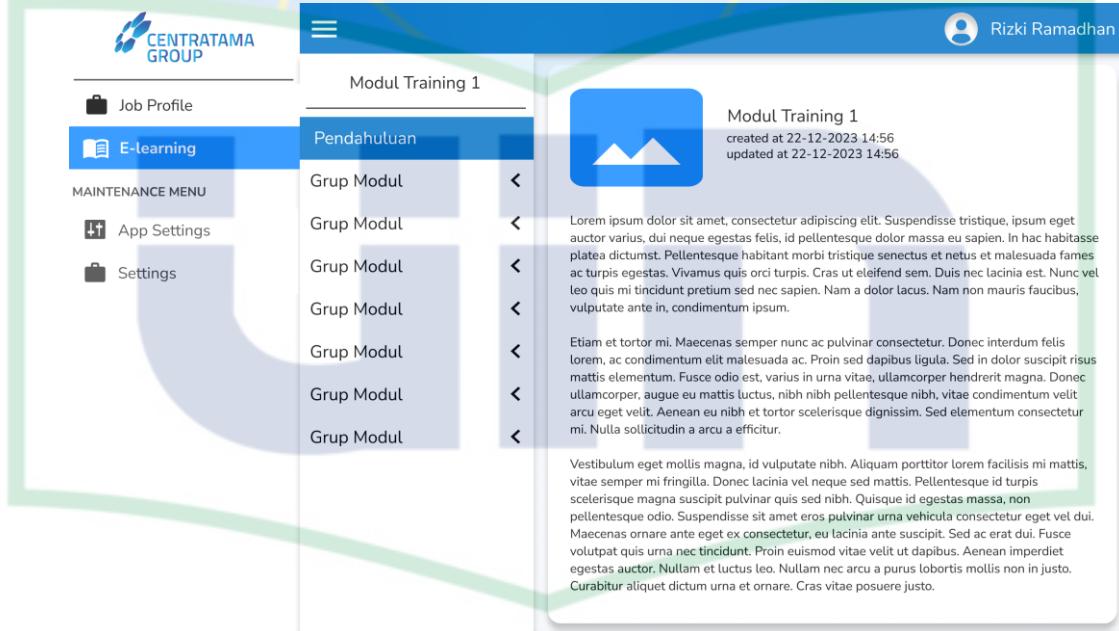
**Gambar 4.94 Halaman Manajemen Data Karyawan (EmployeeManagementView)**

- Halaman Awal *E-Learning*



Gambar 4.95 Halaman awal *E-Learning* (*ElearningView*)

- Tampilan Pengerjaan Modul *Training* dengan menu *sidebar* utama terbuka



Gambar 4.96 Tampilan Pengerjaan Modul *Training* dengan menu *sidebar* utama terbuka (*DoLearning*)

- Tampilan Penggeraan Modul *Training* dengan menu *sidebar* utama tertutup

Modul Training 1

Pendahuluan

Grup Modul

Modul Training 1  
created at 22-12-2023 14:56  
updated at 22-12-2023 14:56

Modul Training 1

created at 22-12-2023 14:56  
updated at 22-12-2023 14:56

Etiam et tortor mi. Maecenas semper nunc ac pulvinar consectetur. Donec interdum felis lorem, ac condimentum elit malesuada ac. Proin sed dapibus ligula. Sed in dolor suscipit risus mattis elementum. Fusce odio est, varius in urna vitae, ullamcorper hendrerit magna. Donec ullamcorper, augue eu mattis luctus, nibh nibh pellentesque nibh, vitae condimentum velit arcu eget velit. Aenean eu nibh et tortor scelerisque dignissim. Sed elementum consectetur mi. Nulla sollicitudin a arcu a efficitur.

Vestibulum eget mollis magna, id vulputate nibh. Aliquam porttitor lorem facilisis mi mattis, vitae semper mi fringilla. Donec lacinia vel neque sed mattis. Pellentesque id turpis scelerisque magna suscipit pulvinar quis sed nibh. Quisque id egestas massa, non pellentesque odio. Suspendisse sit amet eros pulvinar urna vehicula consectetur eget vel dui. Maecenas ornare ante eget ex consectetur, eu lacinia ante suscipit. Sed ac erat dui. Fusce volutpat quis urna nec tincidunt. Proin euismod vitae velit ut dapibus. Aenean imperdiet egestas auctor. Nullam et luctus leo. Nullam nec arcu a purus lobortis mollis non in justo. Curabitur aliquet dictum urna et ornare. Cras vitae posuere justo.

Nulla gravida felis nisl, sed placerat arcu finibus ut. Aliquam et pulvinar est. Etiam tempus scelerisque neque. Nulla viverra enim quis arcu volutpat, vitae volutpat ex fermentum. Donec feugiat leo risus, in maximus dolor gravida sed. Fusce dignissim leo id nunc tempus malesuada. Donec nec metus ex. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Phasellus faucibus non tellus non eleifend. Nulla facilisi. Suspendisse elementum, felis vitae volutpat interdum, purus urna dictum tortor, in varius magna diam ac ex. Pellentesque porta sodales maximus. Morbi velit urna, mollis et ligula et, porttitor fringilla magna. Maecenas vel mauris leo.

**Gambar 4.97 Tampilan Penggeraan Modul *Training* dengan menu *sidebar* utama tertutup (DoLearning)**

- Halaman Awal Pengajuan *Training* Baru

CENTRATAMA GROUP

Job Profile

E-learning

Overview

Pengajuan Training

Tanggal

Divisi

Departemen

Nama

Status

29-08-2020 14:30	Finance	Information Technology	Rizki	<button>Submit</button>
29-08-2020 14:30	Finance	Information Technology	Rizki	<button>Submit</button>
29-08-2020 14:30	Finance	Information Technology	Rizki	<button>Submit</button>
29-08-2020 14:30	Finance	Information Technology	Rizki	<button>Submit</button>
29-08-2020 14:30	Finance	Information Technology	Rizki	<button>Submit</button>
29-08-2020 14:30	Finance	Information Technology	Rizki	<button>Submit</button>
29-08-2020 14:30	Finance	Information Technology	Rizki	<button>Submit</button>
29-08-2020 14:30	Finance	Information Technology	Rizki	<button>Submit</button>

Buat Pengajuan Training Baru

**Gambar 4.98 Halaman Awal Pengajuan *Training* Baru**

- Formulir Pengajuan *Training Baru*

Pengajuan Training Baru (PTB)

Nama Anda (Nama Pengaju)	Rizki Ramadhan
NIP anda (NIP Pengaju)	CG270819

Nama Training

Alasan Pengadaan Training

Saran Materi Training

Tutor Training  
Ex: Nama Tutor - Asal Lembaga Training

Saran Pengadaan Live Training

**Submit**

Gambar 4.99 Formulir Pengajuan *Training Baru* (FormPTBView)

- Tampilan Pembuat Modul *Training*

Buat Modul Training

Gambar Sampul

**Upload**

Nama Training

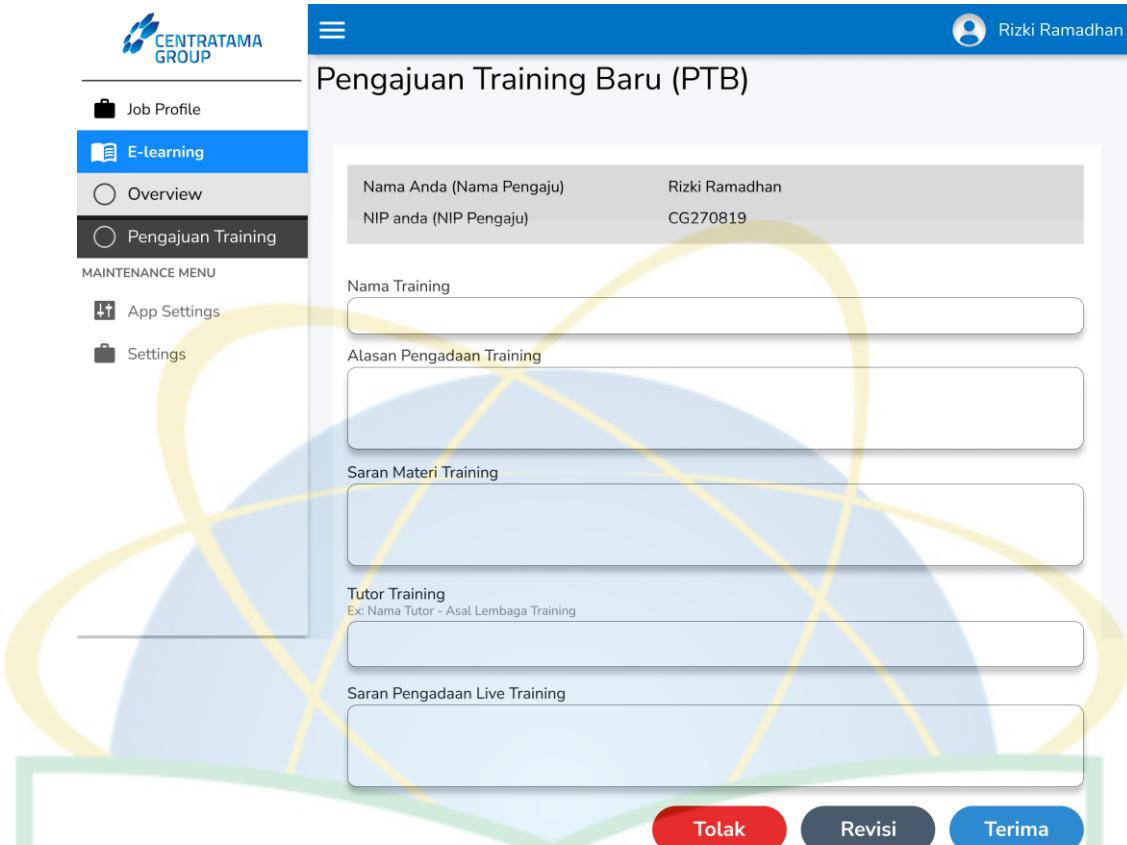
Tanggal Mulai Training      Tanggal Selesai Training

Deskripsi

**Buat Modul Training Baru**

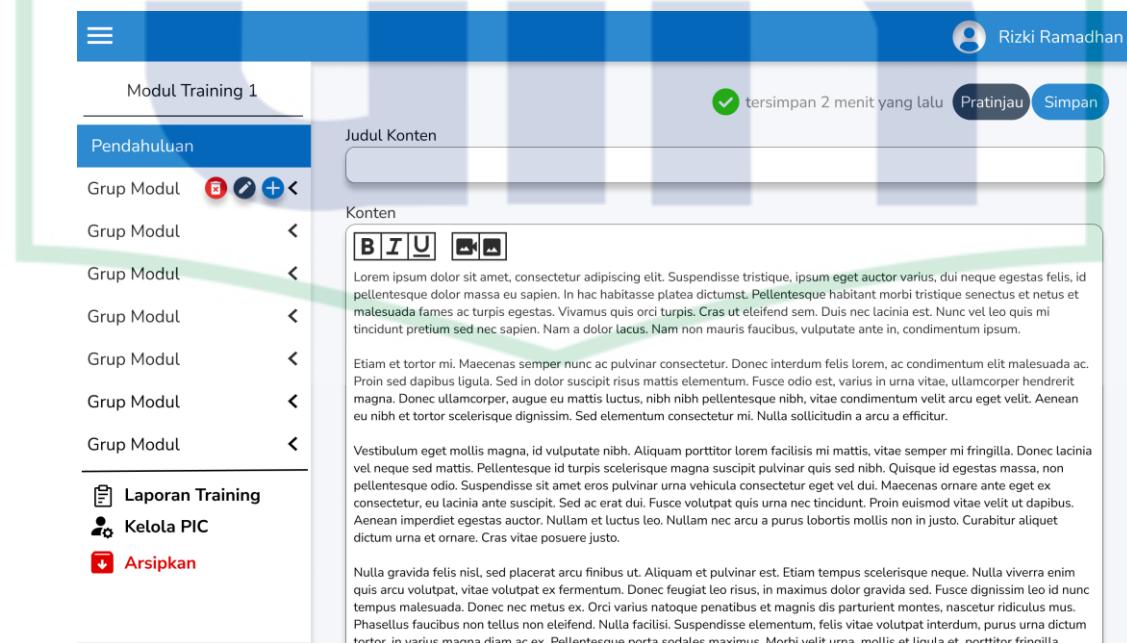
Gambar 4.100 Tampilan Pembuat Modul *Training* (PemanduModulTrainingView)

- Tampilan Persetujuan PTB



Gambar 4.101 Tampilan Persetujuan PTB (PersetujuanPTBView)

- Tampilan Pengelolaan Modul Training



Gambar 4.102 Tampilan Pengelolaan Modul Training (KelolaModulTrainingView)

- Tampilan Pratinjau Konten *Training*

Modul Training 1

Pendahuluan

Grup Modul Delete Edit + <

Grup Modul <

Laporan Training

Kelola PIC

Arsipkan

Judul Konten

tersimpan 2 menit yang lalu Pratinjau Simpan

Etiam ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse tristique, ipsum eget auctor varius, dui neque egestas felis, id pellentesque dolor massa eu sapien. In hac habitasse platea dictumst. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vivamus quis orci turpis. Cras ut eleifend sem. Duis nec lacinia est. Nunc vel leo quis mi tincidunt pretium sed nec sapien. Nam a dolor lacus. Nam non mauris faucibus, vulputate ante in, condimentum ipsum.

Etiam et tortor mi. Maecenas semper nunc ac pulvinar consectetur. Donec interdum felis lorem, ac condimentum elit malesuada ac. Proin sed dapibus ligula. Sed in dolor suscipit risus mattis elementum. Fusce odio est, varius in urna vitae, ullamcorper hendrerit magna. Donec ullamcorper, augue eu mattis luctus, nibh nibh pellentesque nibh, vitae condimentum velit arcu eget velit. Aenean eu nibh et tortor scelerisque dignissim. Sed elementum consectetur mi. Nulla sollicitudin a arcu a efficitur.

Vestibulum eget mollis magna, id vulputate nibh. Aliquam porttitor lorem facilisis mi mattis, vitae semper mi fringilla. Donec lacinia vel neque sed mattis. Pellentesque id turpis scelerisque magna suscipit pulvinar quis sed nibh. Quisque id egestas massa, non pellentesque odio. Suspendisse sit amet eros pulvinar urna vehicula consectetur eget vel dui. Maecenas ornare ante eget ex consectetur, eu lacinia ante suscipit. Sed ac erat dui. Fusce volutpat quis urna nec tincidunt. Proin euismod vitae velit ut dapibus. Aenean imperdiet egestas auctor. Nullam et luctus leo. Nullam nec arcu a purus lobortis mollis non in justo. Curabitur aliquet dictum urna et ornare. Cras vitae posuere justo.

Nulla gravida felis nisl, sed placerat arcu finibus ut. Aliquam et pulvinar est. Etiam tempus scelerisque neque. Nulla viverra enim quis arcu volutpat, vitae volutpat ex fermentum. Donec feugiat leo risus, in maximus dolor gravida sed. Fusce dignissim leo id nunc tempus malesuada. Donec nec metus ex. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Phasellus faucibus non tellus non eleifend. Nulla facilisi. Suspendisse elementum, felis vitae volutpat interdum. Purus urna dictum tortor, in varius magna diam ac ex. Pellentesque porta

Gambar 4.103 Tampilan Pratinjau Konten *Training* (KelolaModulTrainingView)

- Tampilan Pengelolaan Modul *Training* Bertipe Kuis

Modul Training 1

Pendahuluan

Grup Modul Delete Edit + <

Grup Modul <

Laporan Training

Kelola PIC

Arsipkan

Judul Konten

tersimpan 2 menit yang lalu Pratinjau Simpan

Konten

Pertanyaan 1

Pertanyaan?

Jawaban

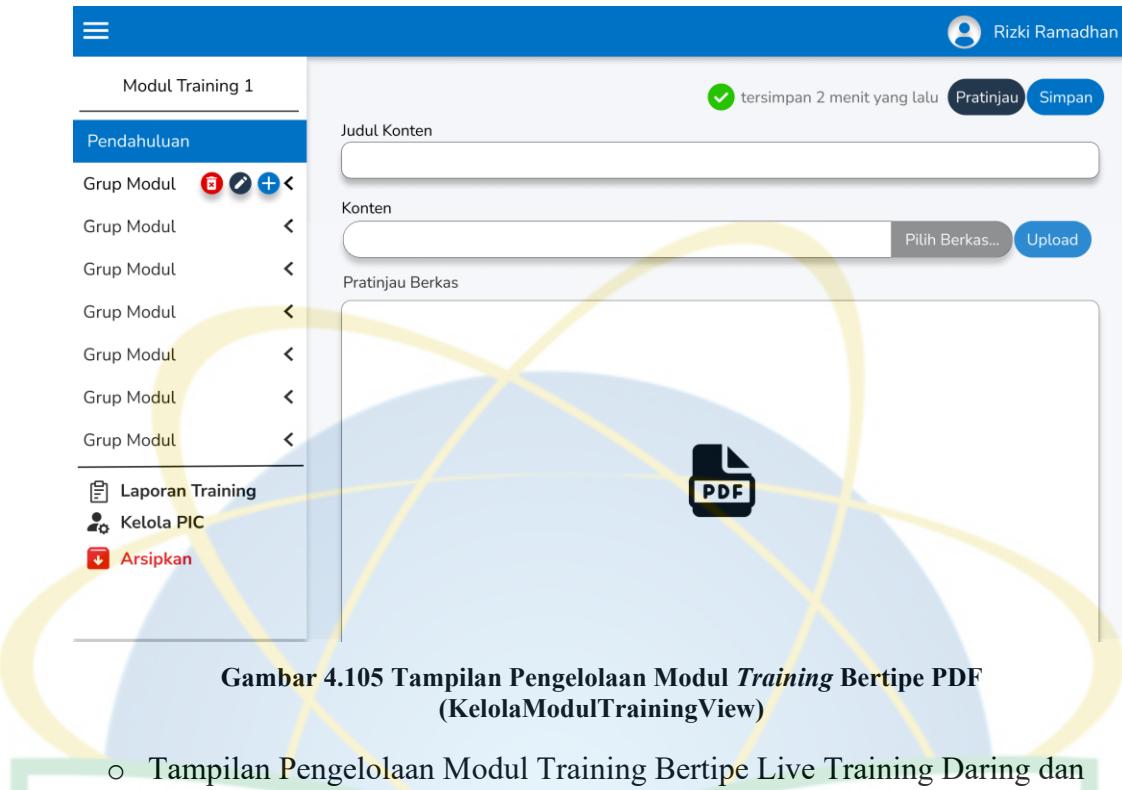
Jawaban

Jawaban

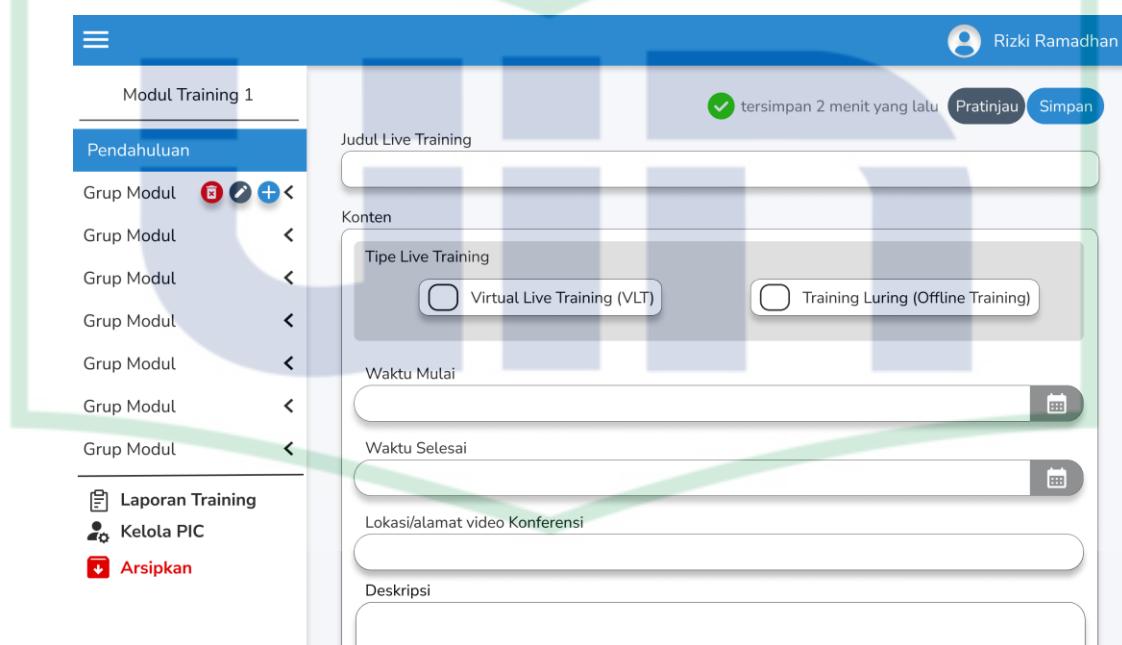
+

Gambar 4.104 Tampilan Pengelolaan Modul *Training* Bertipe Kuis (KelolaModulTrainingView)

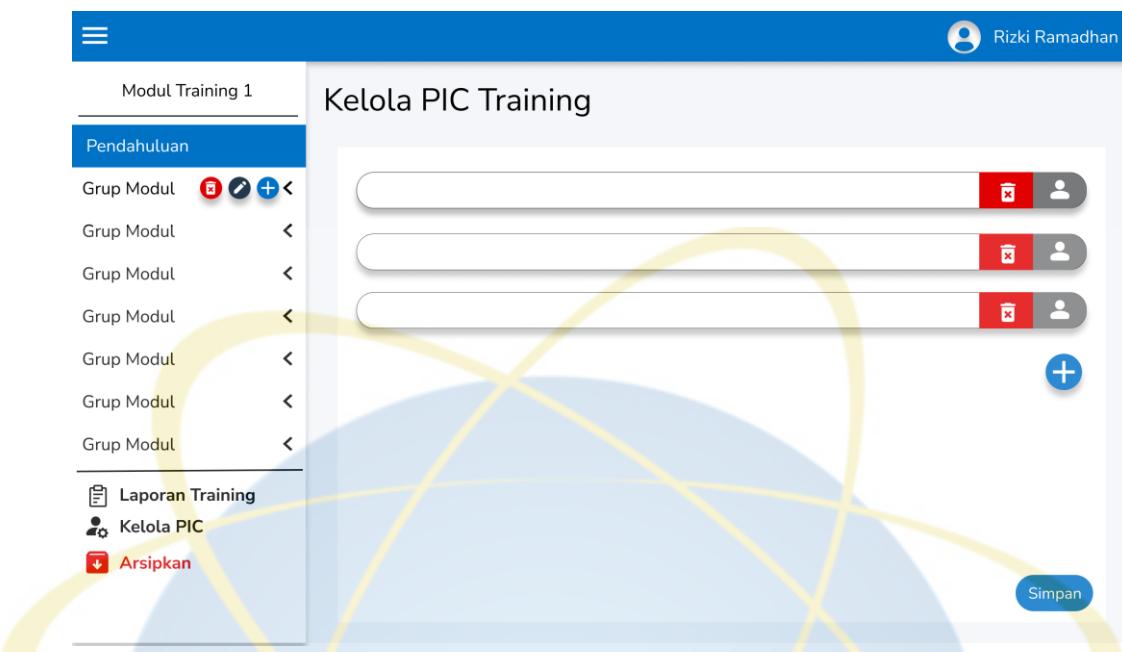
- Tampilan Pengelolaan Modul *Training* Bertipe PDF



- Tampilan Pengelolaan Modul *Training* Bertipe Live Training Daring dan Luring

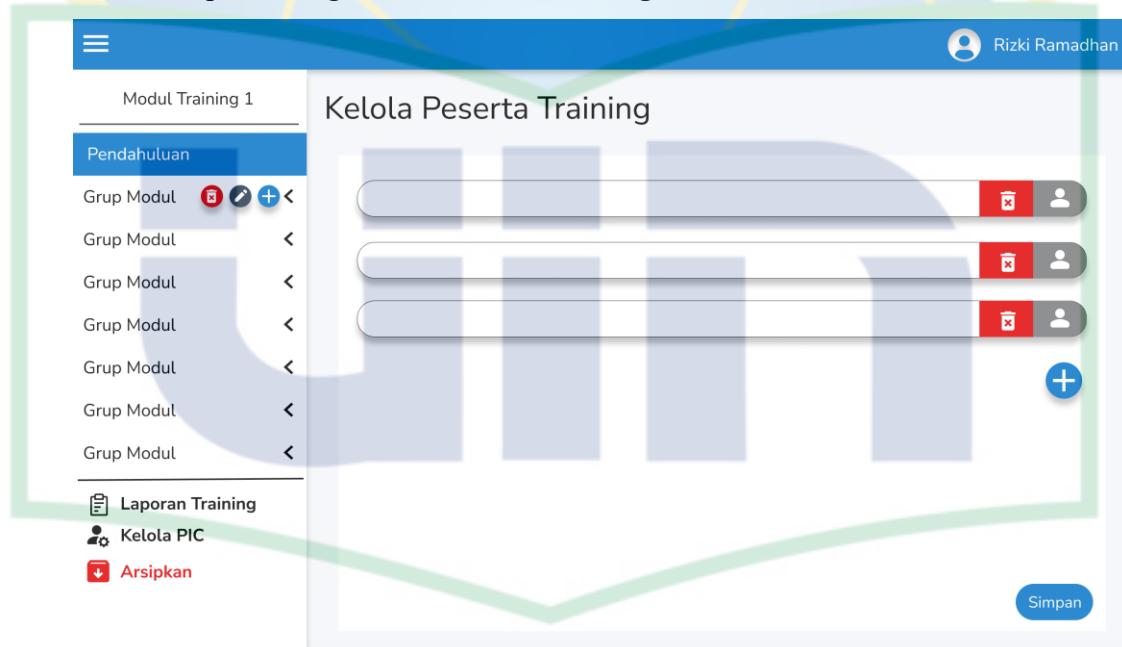


- Tampilan Pengelolaan PIC Training



Gambar 4.107 Tampilan Pengelolaan PIC Training (KelolaModulTrainingView)

- Tampilan Pengelolaan Peserta Training



Gambar 4.108 Tampilan Pengelolaan Peserta Training (KelolaModulTrainingView)

- Tampilan Laporan Training

Nama	NIP	Progres	Progres Terakhir
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30
Rizki Ramadhan	CG270819	27%	29-08-2020 14:30

Gambar 4.109 Tampilan Laporan Training (KelolaModulTraining)

- Tampilan Forum Diskusi Training

Gambar 4.110 Tampilan Forum Diskusi Training (ForumElearningView)

- Tampilan Posting Forum Diskusi *Training*

Rizki Ramadhan • 27 menit yang lalu

**Judul Forum Training**

Etiam et tortor mi. Maecenas semper nunc ac pulvinar consectetur. Donec interdum felis lorem, ac condimentum elit malesuada ac. Proin sed dapibus ligula. Sed in dolor suscipit risus mattis elementum. Fusce odio est, varius in urna vitae, ullamcorper hendrerit magna. Donec ullamcorper, augue eu mattis luctus, nibh nibh pellentesque nibh, vitae condimentum velit arcu eget velit. Aenean eu nibh et tortor scelerisque dignissim. Sed elementum consectetur mi. Nulla sollicitudin a arcu a efficitur.

#ModulTraining #KontenModulTraining

2 Balasan

Rizuki Ryumada • 1 menit yang lalu

Etiam et tortor mi. Maecenas semper nunc ac pulvinar consectetur. Donec interdum felis lorem, ac condimentum elit malesuada ac. Proin sed dapibus ligula. Sed in dolor suscipit risus mattis elementum. Fusce odio est, varius in urna vitae, ullamcorper hendrerit magna. Donec ullamcorper, augue eu mattis luctus, nibh nibh pellentesque nibh, vitae condimentum velit arcu eget velit. Aenean eu nibh et tortor scelerisque dignissim. Sed elementum consectetur mi. Nulla sollicitudin a arcu a efficitur.

**Gambar 4.111 Tampilan Posting Forum Diskusi *Training* (ForumElearningView)**

- *Popup* Pembuatan Posting Baru

Buat Posting Baru

Pastikan anda telah membaca [aturan dalam membuat postingan forum](#) sebelum membuat postingan.

Grup Konten

Konten

Judul Posting/Pertanyaan

Posting/Uraian Pertanyaan

Jelaskan uraian pertanyaan atau posting anda secara mendetail agar para pembaca dapat lebih mudah memahami pertanyaan anda. Anda juga bisa mendapatkan jawaban yang lebih cepat bila menjelaskan pertanyaan secara jelas dan detail.

Buat Posting

**Gambar 4.112 Popup Pembuatan Posting Baru**

#### 4.2.2 *Implementation/Development (Implementasi/Pengembangan)*

Penulis pada tahapan ini melakukan pengimplementasian desain yang telah dibuat sebelumnya. Namun sebelum itu, penulis mendokumentasikan kebutuhan-kebutuhan pengembangan *software* pada penelitian ini ke dalam *issues* di Github.

Penulis membahas tahapan implementasi secara runut yang dimulai dengan mengumpulkan kebutuhan pengembangan *software* sampai kepada tahapan pengodean *software* dengan melibatkan fitur Git dan Github. Penjelasan lainnya seperti alur pengembangan *software* dengan Github, penulis jelaskan pada berkas README yang penulis taruh di repositori kode agar para pengembang di masa depan dapat langsung melihat alur penulis melakukan pengembangan *software* pada penelitian ini. README ini penulis buat sebagai pemandu untuk melakukan pengembangan *software* dengan Github.

#### **A. Pengumpulan dan pencatatan kebutuhan pengembangan *e-learning* dengan fitur Github *Issues***

Sebelum mengimplementasi desain aplikasi menjadi *software*, penulis mencatat beberapa kebutuhan untuk pengembangan *e-learning*. Beberapa di antaranya adalah harapan fitur dari HC CTI. Penulis mencatatnya menjadi *issues* di Github.

Alangkah baiknya sebelum memulai pengembangan *e-learning*, penulis memahami terlebih dahulu bagaimana melakukan proyek pengembangan *software* dengan Github. Untuk itu, penulis membuat dokumen README berikut untuk menjelaskan secara lebih detail bagaimana penulis melakukannya.

<https://github.com/ryumada/centratama-hcportal-new-public/blob/main/README-github-workflow.md>

Di langkah sebelumnya, penulis telah membuat repositori sampai kepada membuat templat untuk *issues* dan PR. Penulis menggunakan templat *issue user-story* untuk membantu dalam pencatatan kebutuhan pengembangan aplikasi. Berikut beberapa *issues* yang sudah penulis buat:

Filters ▾  Labels (12) Milestones (0) New issue

⊕ 9 Open ✓ 0 Closed

<input type="checkbox"/> Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	● Add learning material support for these types <small>feature:core user-story</small>				
	#18 opened 23 minutes ago by ryumada				
<input type="checkbox"/>	● Timer for pre-test, post-test, and assessment feature. <small>feature:core user-story</small>				
	#17 opened 26 minutes ago by ryumada				
<input type="checkbox"/>	● The flexibility of the feature activation <small>feature:core user-story</small>				
	#16 opened 32 minutes ago by ryumada				
<input type="checkbox"/>	● Test the understanding of trainee after completing the courses or programs <small>feature:core user-story</small>				
	#15 opened 2 hours ago by ryumada				
<input type="checkbox"/>	● The goal of learning with Learning Path <small>feature:core user-story</small>				
	#14 opened 2 hours ago by ryumada				
<input type="checkbox"/>	● Track my learning progress <small>feature:core user-story</small>				
	#13 opened 2 hours ago by ryumada				
<input type="checkbox"/>	● Pre-test and Post-test for the learning module <small>feature:core user-story</small>				
	#12 opened 2 hours ago by ryumada				
<input type="checkbox"/>	● Forum feature to share the learned material from the training module. <small>feature:core user-story</small>				
	#11 opened 2 hours ago by ryumada				
<input type="checkbox"/>	● Access training material easily <small>feature:core user-story</small>				
	#10 opened 2 hours ago by ryumada				

**Gambar 4.113 Beberapa *issues* yang penulis buat untuk mengumpulkan kebutuhan pengembangan *e-learning***

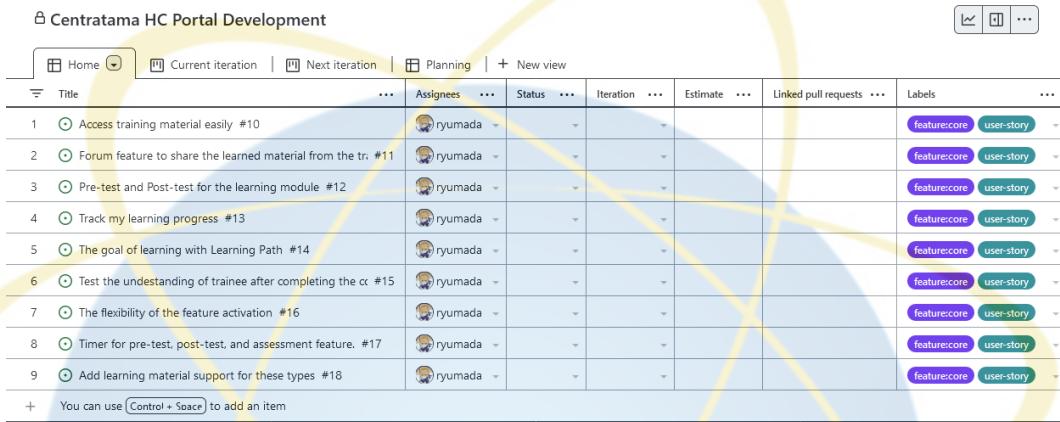
Masing-masing dari *issues* di atas penulis berikan label `feature:core` untuk menandakan bahwa beberapa *issues* tersebut merupakan fitur utama untuk *e-learning*. Penulis juga menambahkan *issues* tersebut ke dalam Github Project untuk pengelolaan manajemen tugas.

## B. Manajemen tugas dengan Github Project

Setelah mengumpulkan kebutuhan untuk pengembangan *e-learning*, penulis kemudian membuka Github Project. Terdapat beberapa bentuk tampilan pada Github Project. Dokumen README berikut penulis buat untuk memahami fungsi-fungsinya:

[https://github.com/ryumada/centratama-hcportal-new-public/blob/main/README-persiapan\\_awal/README-membuat\\_project.md](https://github.com/ryumada/centratama-hcportal-new-public/blob/main/README-persiapan_awal/README-membuat_project.md)

Tampilan Project telah berisi beberapa *issues* yang telah ditambahkan sebelumnya, namun terdapat beberapa *field* informasi yang perlu penulis tetapkan untuk melakukan manajemen proyek. Gambar berikut menampilkan tampilan *home* Project Centratama HC Portal Development.



Title	... Assignees	... Status	... Iteration	... Estimate	... Linked pull requests	... Labels	...
1 Access training material easily #10	ryumada					featurecore user-story	
2 Forum feature to share the learned material from the tr. #11	ryumada					featurecore user-story	
3 Pre-test and Post-test for the learning module #12	ryumada					featurecore user-story	
4 Track my learning progress #13	ryumada					featurecore user-story	
5 The goal of learning with Learning Path #14	ryumada					featurecore user-story	
6 Test the understanding of trainee after completing the cc #15	ryumada					featurecore user-story	
7 The flexibility of the feature activation #16	ryumada					featurecore user-story	
8 Timer for pre-test, post-test, and assessment feature. #17	ryumada					featurecore user-story	
9 Add learning material support for these types #18	ryumada					featurecore user-story	

Gambar 4.114 Tampilan awal sebelum penulis menyetel atribut *field* informasi untuk kebutuhan manajemen proyek.

Dari gambar di atas, Project memiliki empat tampilan, di antaranya:

- **Home** yang merupakan tampilan awal,
- **Current Iteration** yang merupakan tampilan iterasi pengembangan saat ini,
- **Next Iteration** yang merupakan tampilan iterasi pengembangan selanjutnya,
- **Planning** yang merupakan tampilan yang berfokus pada perencanaan iterasi.

Penulis melakukan penetapan informasi *field* Status dan *Estimate*. *Field* Status merupakan posisi progres tugas, sedangkan *field* *Estimate* merupakan penilaian kompleksitas tugas. Penulis membahas lebih lanjut terkait kedua *field* ini dalam dokumen README di atas. Untuk *field* Iterasi, penulis menetapkannya saat membuat perencanaan iterasi di tab *Planning*. Di gambar berikut ini, penulis telah menetapkan *field* Status dan *Estimate* untuk masing-masing tugas.

Centratama HC Portal Development

Title	Assignees	Status	Iteration	Estimate	Linked pull requests	Labels
1 Access training material easily #10	ryumada			9		epic featurecore user-story
2 Forum feature to share the learned material from the tr. #11	ryumada	New	Iteration 2	8		featurecore user-story
3 Add learning material support for these types #18	ryumada	New	Iteration 1	3		featurecore user-story
4 Track my learning progress #13	ryumada	New	Iteration 1	3		featurecore user-story
5 The goal of learning with Learning Path #14	ryumada	New	Iteration 1	3		featurecore user-story
6 Pre-test and Post-test for the learning module #12	ryumada	New	Iteration 1	4		featurecore user-story
7 Test the understanding of trainee after completing the cc #15	ryumada	New	Iteration 1	3		featurecore user-story
8 Timer for pre-test, post-test, and assessment feature. #17	ryumada	New	Iteration 1	3		featurecore user-story
9 The flexibility of the feature activation #16	ryumada	New	Iteration 1	4		featurecore user-story

You can use [Control + Space] to add an item

Gambar 4.115 Daftar tugas yang telah diberi informasi pada field Status dan Estimate

Dari gambar di atas, dapat dilihat *Estimate* untuk setiap tugas memiliki nilai yang berbeda. Penulis memberikannya dalam bentuk angka untuk menilai kerumitan dari masing-masing tugas. Tugas nomor 1 memiliki bobot *Estimate* 9, ternyata setelah ditilik lebih lanjut *issue* ini sebenarnya menjelaskan kebutuhan yang umum terkait *e-learning*. Tugas dari nomor 3 sampai 9 dapat kita kelompokkan ke tugas nomor 1 untuk memenuhinya. Itulah mengapa tugas nomor satu diberi label **epic** yang menjelaskan bahwa *issue* ini terdiri dari tugas-tugas kecil di dalamnya. Penulis juga membuat tab *epic* untuk memudahkan dalam melihatnya secara spesifik. *Issue* dengan label *epic* tidak perlu diberikan iterasi, cukup tugas-tugas kecilnya saja yang diberikan.

Centratama HC Portal Development

label:epic 1 ×	Home   Current iteration   Next iteration   Epics   Planning   + New view
New 1	+ Add item
centratama-hcportal-new #10 Access training material easily	+ Add item
Backlog 0	+ Add item
Ready 0	+ Add item
In progress 0	+ Add item

Gambar 4.116 Tampilan tab *epic* yang memuat satu *issue*

Tugas nomor 2 juga memiliki nilai *estimate* yang besar. Pembuatan forum merupakan fungsi yang berbeda dari materi *training*, namun mereka tetap dapat dihubungkan. Tugas ini nantinya akan dipecah menjadi tugas-tugas kecil yang menyusunnya.

Dari perencanaan iterasi yang telah dibuat, penulis merencanakan untuk melaksanakan dua iterasi dalam pengembangan *e-learning* ini. Pada iterasi pertama, penulis mengembangkan fitur-fitur dasar dari sebuah *e-learning*. Kemudian di iterasi kedua, penulis mengembangkan fitur Forum.

Title	Iteration	Assignees	Estimate
1 Add learning material support for these types #18	Iteration 1	ryumada	
2 Track my learning progress #13	Iteration 1	ryumada	
3 The goal of learning with Learning Path #14	Iteration 1	ryumada	
4 Pre-test and Post-test for the learning module #12	Iteration 1	ryumada	
5 Test the understanding of trainee after completing the courses or programs #15	Iteration 1	ryumada	
6 Timer for pre-test, post-test, and assessment feature. #17	Iteration 1	ryumada	
7 The flexibility of the feature activation #16	Iteration 1	ryumada	
+ Add item			
8 Forum feature to share the learned material from the training module. #11	Iteration 2	ryumada	
+ Add item			

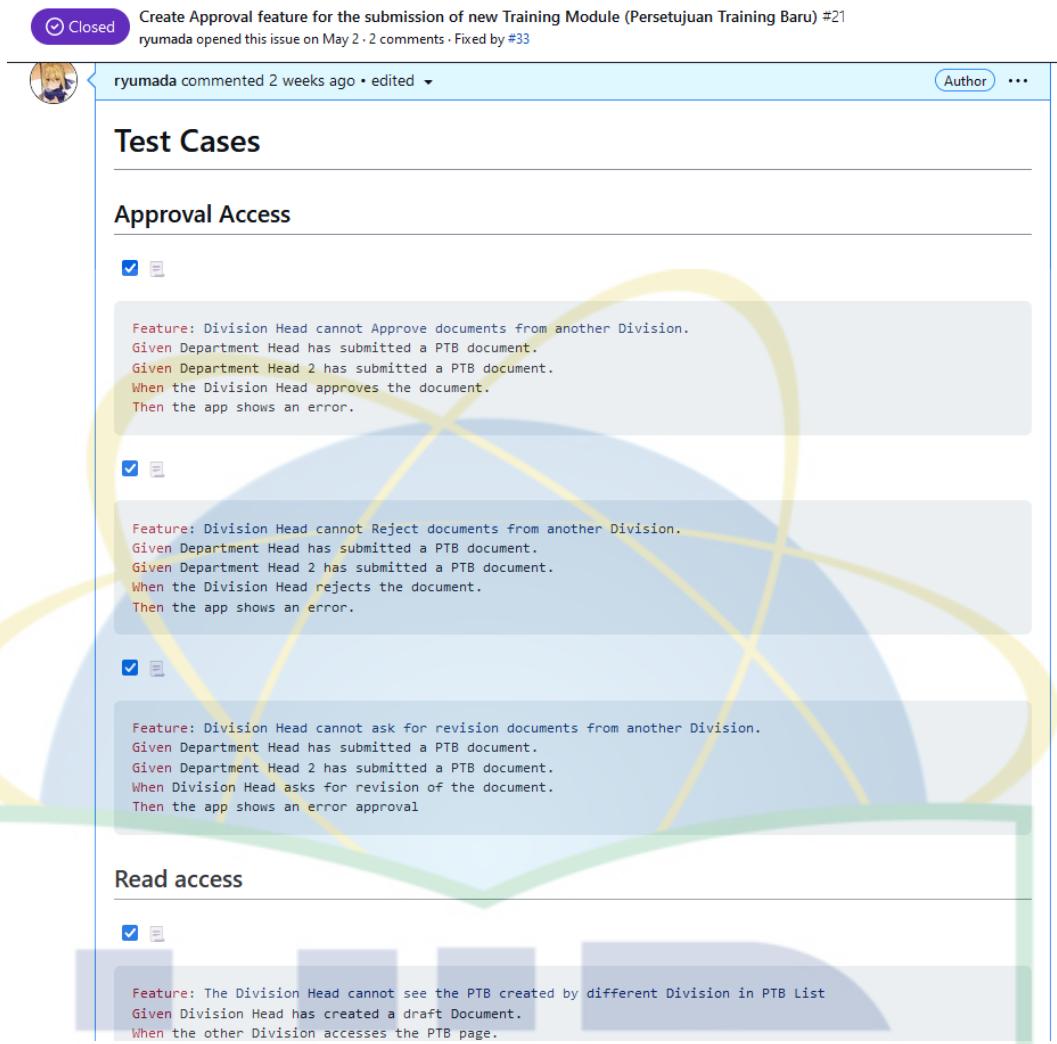
Gambar 4.117 Tampilan tab *Planning* yang memuat 2 iterasi pengembangan

### C. Pengodean (*Coding*)

Penulis telah menjelaskan bagaimana cara penulis melakukan pengodean *software* pada penelitian ini di berkas README berikut:

<https://github.com/ryumada/centratama-hcportal-new-public/blob/main/README-github-workflow.md>

Penulis memanfaatkan Github untuk membagi pengembangan menjadi fitur per fitur. Setiap fitur, penulis tuliskan kebutuhan pengembangannya pada Github *Issues*. Penulis juga menambahkan informasi tambahan pada *issues* untuk menjelaskan beberapa alur yang memerlukan percabangan seperti pada gambar berikut:



**Gambar 4.118 Penulis menambahkan informasi di Github *Issues* untuk kebutuhan teknis pengembangan**

Penulis hanya melaksanakan tahapan analisis dan desain sekali. Jika ternyata terdapat beberapa hal yang tidak dapat diimplementasikan dalam desain yang penulis buat, penulis menjelaskannya pada *Github Issues*. Saat melakukan pengembangan pun penulis juga menemukan beberapa kebutuhan teknis untuk mendukung pengembangan fitur. Hal ini membuat iterasi pengembangan yang penulis lakukan bisa lebih dari 2 iterasi.

Kemudian untuk alat-alat yang penulis gunakan untuk melaksanakan penelitian ini, seperti perangkat keras dan perangkat lunaknya. Ini telah penulis bahas di subbab 4.1.7 yang membahas persiapan awal pengembangan *Software*. Sedangkan untuk kode pengembangan *software*-nya, penulis menyimpannya pada

repositori yang telah penulis buat pada tahapan 4.1.8. Tautan berikut mengarahkan ke repositori tersebut:

<https://github.com/ryumada/centratama-hcportal-new-public>

#### 4.2.3 Testing & Revise (Revisi)

Penulis pada tahapan ini melakukan Pengujian Software *E-learning* HC Portal. Tahapan ini penulis lakukan untuk memastikan bahwa Software *e-learning* HC Portal yang telah penulis buat sesuai dengan ekspektasi mentor penulis selaku pemilik *Software* ini. Pemastian kesesuaian fungsi aplikasi HC Portal dilakukan dengan menjalankan simulasi semua prosedur yang ada pada *Use Case*. Proses pengujian ini dilaksanakan oleh penulis, mentor penulis, dan beberapa rekan kerja di lingkungan divisi HC CTI. Penjelasan tersebut dapat disimpulkan bahwa penulis di sini menggunakan teknik *Black Box Testing* untuk melakukan pengujian aplikasi.

Tabel 4.61 Pengujian *Use Case* Lihat Halaman Awal

<b>Nama Pengujian</b>	Lihat Halaman Awal
<b>Pengguna</b>	Semua Aktor
<b>Prosedur Pengujian</b>	1. Masuk ke alamat web aplikasi HC Portal.
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	Tampilan awal aplikasi HC Portal
<b>Kesimpulan</b>	OK

Tabel 4.62 Pengujian *Use Case Login*

<b>Nama Pengujian</b>	<i>Login</i>
<b>Pengguna</b>	Seluruh Karyawan CTI
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"><li>Aktor melakukan klik tombol <i>Login</i> yang tersedia pada halaman utama untuk memunculkan <i>popup Login</i>.</li><li>Memasukkan nomor induk pegawai dan kata sandinya pada kotak formulir yang tersedia.</li><li>Klik tombol “<i>Login</i>”.</li></ol>
<b>Masukan (Input)</b>	Nomor Induk Pegawai (NIP) dan kata sandi ( <i>password</i> )

<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>Tampilan utama JP ketika aktor memasukkan NIP dan kata sandi dengan benar.</li> <li>Memunculkan pesan eror ketika NIP dan kata sandi yang dimasukkan tidak sesuai dengan yang tersimpan di basis data aplikasi.</li> </ul>
<b>Kesimpulan</b>	OK

**Tabel 4.63 Pengujian Use Case Logout**

<b>Nama Pengujian</b>	<i>Logout</i>
<b>Pengguna</b>	Seluruh Karyawan CTI
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>Aktor melakukan klik pada area profil di pojok kanan baki navigasi (<i>navigation bar/navbar</i>).</li> <li>Aktor melakukan klik tombol “Logout”.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	Tampilan awal aplikasi sebelum melakukan <i>Login</i>
<b>Kesimpulan</b>	OK

**Tabel 4.64 Pengujian Use Case Kelola Profil Akun**

<b>Nama Pengujian</b>	Kelola Profil Akun
<b>Pengguna</b>	Seluruh Karyawan CTI
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>Aktor melakukan klik pada area profil di pojok kanan baki navigasi (<i>navigation bar/navbar</i>).</li> <li>Aktor melakukan klik tombol profil.</li> <li>Aktor mengedit nama dan surel di kotak formulir yang tersedia.</li> <li>Aktor melakukan klik tombol “Save” pada akhir formulir.</li> <li>Aktor memperbarui kata sandi pengguna, kemudian memasukkan kata sandi lamanya dengan benar pada kotak formulir yang tersedia.</li> <li>Aktor melakukan klik tombol “Save” pada akhir formulir.</li> </ol>

	<p>7. Aktor memperbarui kata sandi pengguna, kemudian memasukkan kata sandi lamanya dengan salah pada kotak formulir yang tersedia.</p> <p>8. Aktor melakukan klik tombol “Save” pada akhir formulir.</p>
<b>Masukan (Input)</b>	<ul style="list-style-type: none"> <li>• Perubahan nama dan surel</li> <li>• Perubahan kata sandi baru</li> <li>• Kata sandi lama</li> </ul>
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Menampilkan pesan berhasil ketika pengguna menyimpan perubahan nama dan surel.</li> <li>• Menampilkan pesan berhasil ketika pengguna mengubah kata sandinya dan memasukkan kata sandi lamanya dengan benar.</li> <li>• Menampilkan pesan eror ketika pengguna mengubah kata sandinya dan memasukkan kata sandi lamanya dengan salah.</li> </ul>
<b>Kesimpulan</b>	OK

Tabel 4.65 Pengujian Use Case Kelola Data Karyawan

<b>Nama Pengujian</b>	Kelola Data Karyawan
<b>Pengguna</b>	Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor melakukan klik menu “Settings” lalu ke submenu “Master Data”.</li> <li>2. Aktor melakukan klik tombol “Manage Employee”.</li> <li>3. Aktor melakukan klik tombol tambah karyawan di halaman tersebut.</li> <li>4. Aktor mengisi formulir penambahan data karyawan baru, kemudian melakukan klik tombol “Save” pada akhir formulir.</li> <li>5. Aktor melakukan klik tombol edit pada daftar karyawan yang tersedia pada tabel.</li> <li>6. Aplikasi menampilkan jendela <i>popup</i> untuk mengedit data karyawan.</li> <li>7. Aktor mengisi formulir edit data karyawan, kemudian melakukan klik tombol “Save” pada akhir formulir.</li> <li>8. Aplikasi menyimpan perubahan data karyawan pada basis data.</li> </ol>

	<p>9. Aktor melakukan klik tombol hapus pada daftar karyawan yang tersedia pada tabel.</p> <p>10. Aplikasi menampilkan jendela <i>popup</i> untuk meyakinkan admin dalam menghapus karyawan.</p> <p>11. Aktor melakukan klik tombol “Ya”.</p> <p>12. Aplikasi menghapus data karyawan pada basis data.</p>
<b>Masukan (Input)</b>	<ul style="list-style-type: none"> <li>• Data Karyawan Baru</li> <li>• Perubahan data karyawan</li> </ul>
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Menampilkan pesan berhasil ketika pengguna menambahkan data baru karyawan.</li> <li>• Menampilkan pesan berhasil ketika pengguna mengubah data karyawan.</li> <li>• Menampilkan pesan berhasil ketika pengguna menghapus data karyawan.</li> </ul>
<b>Kesimpulan</b>	OK

Tabel 4.66 Pengujian Akses Modul Belajar *Induction*

<b>Nama Pengujian</b>	Akses Modul Belajar <i>Induction</i>
<b>Pengguna</b>	Seluruh Karyawan CTI
<b>Prosedur Pengujian</b>	<p>1. Aktor melakukan klik modul <i>training induction</i></p> <p>2. Aktor membuka semua konten modul <i>training</i> di dalamnya.</p>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Menampilkan modul <i>Training Induction</i> dan daftar kontennya.</li> <li>• Menampilkan konten modul <i>training</i> yang dipilih aktor.</li> </ul>
<b>Kesimpulan</b>	Belum Selesai

Tabel 4.67 Pengujian Isi Formulir Pengajuan *Training* Baru (PTB)

<b>Nama Pengujian</b>	Isi Formulir Pengajuan <i>Training</i> Baru (PTB)
<b>Pengguna</b>	Kepala Departemen, Kadiv Departemen
<b>Prosedur Pengujian</b>	<p>1. Aktor melakukan klik menu Pengajuan <i>Training</i> Baru pada menu <i>sidebar</i> aplikasi <i>e-learning HC Portal</i>.</p>

	<ol style="list-style-type: none"> <li>2. Aktor melakukan klik tombol Pengajuan <i>Training</i> Baru.</li> <li>3. Aktor mengisi formulir dengan benar, lalu melakukan klik tombol <i>Submit</i>.</li> <li>4. Aktor menjalankan tiga tahap sebelumnya, hanya saja pada tahap ke-3 aktor memilih tombol <i>Save</i>.</li> </ol>
<b>Masukan (Input)</b>	Data formulir Pengajuan <i>Training</i> Baru
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Menampilkan pesan eror validasi formulir.</li> <li>• Menyimpan formulir sebagai <i>draft</i> dan menyimpannya ke dalam basis data.</li> </ul>
<b>Kesimpulan</b>	OK

**Tabel 4.68 Pengujian Persetujuan *Training* Baru**

<b>Nama Pengujian</b>	Persetujuan <i>Training</i> Baru
<b>Pengguna</b>	Kadiv Departemen, Karyawan HC, dan Kadiv HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Melakukan klik menu Pengajuan <i>Training</i> Baru pada menu <i>sidebar</i> aplikasi <i>e-learning HC Portal</i>.</li> <li>2. Melakukan klik salah satu data pengajuan <i>training</i> baru.</li> <li>3. Melakukan reviu formulir PTB dan melakukan klik tombol “Terima”.</li> <li>4. Aktor melaksanakan langkah 1 – 3 hanya saja pada langkah ke-3 aktor memilih tombol tolak.</li> <li>5. Aktor melaksanakan langkah 1 – 3 hanya saja pada langkah ke-3 aktor memilih tombol revisi.</li> </ol>
<b>Masukan (Input)</b>	Data Pengajuan <i>Training</i> Baru dengan status yang diberikan untuk aksi persetujuan.
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Memberikan status dokumen sesuai aksi yang diberikan oleh aktor.</li> <li>• Menampilkan <i>popup</i> validasi dan verifikasi untuk aksi revisi dan tolak formulir.</li> </ul>
<b>Kesimpulan</b>	OK

**Tabel 4.69 Pengujian Kelola Modul *Training*/Penambahan Modul *Training* Baru**

<b>Nama Pengujian</b>	Kelola Modul <i>Training</i> /Penambahan Modul <i>Training</i> Baru
<b>Pengguna</b>	Karyawan HC

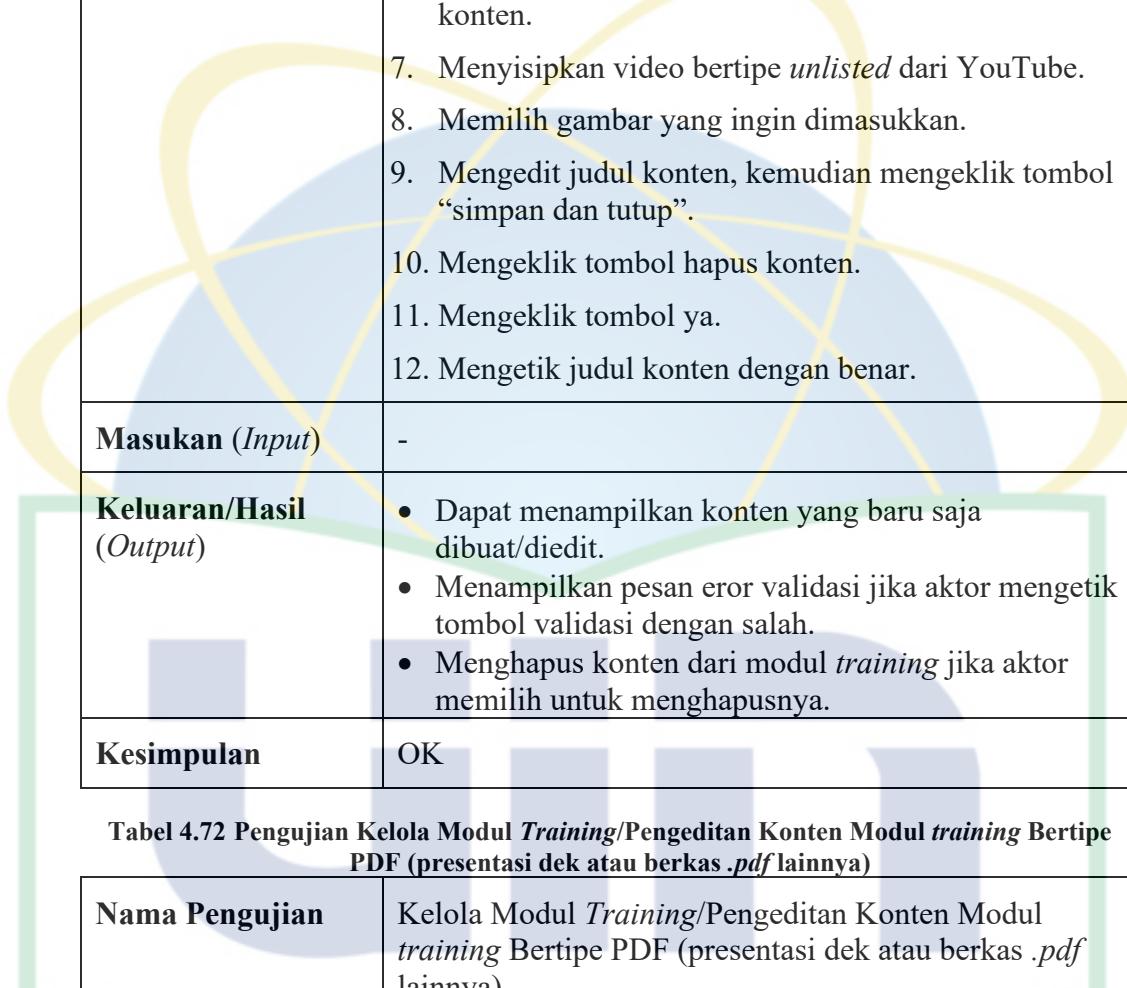
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Melakukan klik tombol penambahan modul <i>training</i> baru.</li> <li>2. Mengisi formulir pada halaman pemandu awal penambahan modul <i>training</i> dengan benar dan melakukan klik tombol “Buat modul <i>training</i> baru.</li> </ol>
<b>Masukan (Input)</b>	Data formulir pembuatan modul <i>training</i> .
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Terdapat data modul <i>training</i> baru pada basis data.</li> <li>• Menampilkan pesan eror validasi, jika data formulir pembuatan modul <i>training</i> baru tidak diisi dengan benar.</li> </ul>
<b>Kesimpulan</b>	OK

Tabel 4.70 Pengujian Kelola Modul *Training*/Pengarsipan Modul *Training*

<b>Nama Pengujian</b>	Kelola Modul <i>Training</i> /Pengarsipan Modul <i>Training</i>
<b>Pengguna</b>	Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Memilih modul <i>training</i> yang ingin diarsipkan..</li> <li>2. Memilih menu arsipkan di menu sidebar <i>e-learning</i>.</li> <li>3. Melakukan klik tombol arsip modul <i>training</i>.</li> <li>4. Mengisi konfirmasi teks dengan benar dan melakukan klik tombol ok</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Memindahkan data modul <i>training</i> dari basis data utama ke basis data arsip, sehingga modul <i>training</i> tidak terlihat pada halaman <i>E-learning</i>.</li> <li>• Menampilkan jendela <i>popup</i> validasi pengarsipan modul <i>training</i> dan menampilkan pesan eror jika validasi yang dimasukkan salah.</li> </ul>
<b>Kesimpulan</b>	Belum Selesai

Tabel 4.71 Pengujian Kelola Modul *Training*/Pengeditan Konten Modul *Training* Bertipe Tekstual, Gambar, dan Video

<b>Nama Pengujian</b>	Kelola Modul <i>Training</i> /Pengeditan Konten Modul <i>Training</i> Bertipe Tekstual, Gambar, dan Video
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Mengklik salah satu modul <i>training</i>.</li> <li>2. Memilih tombol tambah konten.</li> </ol>



	<ol style="list-style-type: none"> <li>3. Mengetik judul konten, kemudian memilih untuk menambahkan konten bertipe tekstual, gambar dan video.</li> <li>4. Memilih tombol “simpan dan tutup”.</li> <li>5. Mengklik tombol edit di konten tekstual yang baru saja ia buat.</li> <li>6. Memilih tombol penambahan gambar ke dalam konten.</li> <li>7. Menyisipkan video bertipe <i>unlisted</i> dari YouTube.</li> <li>8. Memilih gambar yang ingin dimasukkan.</li> <li>9. Mengedit judul konten, kemudian mengklik tombol “simpan dan tutup”.</li> <li>10. Mengklik tombol hapus konten.</li> <li>11. Mengklik tombol ya.</li> <li>12. Mengetik judul konten dengan benar.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Dapat menampilkan konten yang baru saja dibuat/diedit.</li> <li>• Menampilkan pesan eror validasi jika aktor mengetik tombol validasi dengan salah.</li> <li>• Menghapus konten dari modul <i>training</i> jika aktor memilih untuk menghapusnya.</li> </ul>
<b>Kesimpulan</b>	OK

**Tabel 4.72 Pengujian Kelola Modul *Training*/Pengeditan Konten Modul *training* Bertipe PDF (presentasi dek atau berkas .pdf lainnya)**



<b>Nama Pengujian</b>	Kelola Modul <i>Training</i> /Pengeditan Konten Modul <i>training</i> Bertipe PDF (presentasi dek atau berkas .pdf lainnya)
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Mengklik salah satu modul <i>training</i>.</li> <li>2. Memilih tombol tambah konten.</li> <li>3. Mengetik judul konten, kemudian memilih untuk menambahkan konten bertipe pdf.</li> <li>4. Memilih berkas PDF yang ingin diunggah lalu mengklik tombol unggah.</li> <li>5. Memilih tombol “simpan dan tutup”.</li> </ol>

	<ol style="list-style-type: none"> <li>6. Mengklik tombol edit di konten yang baru saja ia buat.</li> <li>7. Memilih berkas PDF baru, kemudian mengklik tombol unggah.</li> <li>8. Memilih tombol “simpan dan tutup”.</li> <li>9. Mengklik tombol hapus konten.</li> <li>10. Mengklik tombol ya.</li> <li>11. Mengetik judul konten dengan benar.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Dapat menampilkan konten <i>pdf</i> yang baru saja dibuat/diedit.</li> <li>• Menampilkan pesan eror validasi jika aktor mengetik tombol validasi dengan salah.</li> <li>• Menghapus konten dari modul <i>training</i> jika aktor memilih untuk menghapusnya.</li> </ul>
<b>Kesimpulan</b>	OK

**Tabel 4.73 Pengujian Kelola Modul *Training*/Pembuatan Konten Kuis**

<b>Nama Pengujian</b>	Kelola Modul <i>Training</i> /Pembuatan Konten Kuis
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Mengklik salah satu modul <i>training</i>.</li> <li>2. Memilih tombol tambah konten.</li> <li>3. Mengetik judul konten, kemudian memilih untuk menambahkan konten bertipe kuis.</li> <li>4. Mengisi formulir kuis.</li> <li>5. Memilih tombol “simpan dan tutup”.</li> <li>6. Mengklik tombol edit di konten yang baru saja ia buat.</li> <li>7. Menghapus satu pertanyaan di konten kuis yang telah dibuat dan mengubah beberapa pertanyaan pada konten kuis.</li> <li>8. Memilih tombol “simpan dan tutup”.</li> <li>9. Mengklik tombol hapus konten.</li> <li>10. Mengklik tombol ya.</li> <li>11. Mengetik judul konten dengan benar.</li> </ol>

<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>Dapat menampilkan konten kuis yang baru saja dibuat/diedit dan menjalankan waktu penghitung mundur saat tombol mulai kuis diklik.</li> <li>Menampilkan pesan eror validasi jika aktor mengetik tombol validasi dengan salah.</li> <li>Menghapus konten dari modul <i>training</i> jika aktor memilih untuk menghapusnya.</li> </ul>
<b>Kesimpulan</b>	OK

Tabel 4.74 Pengujian Penjadwalan *Training Luring*

<b>Nama Pengujian</b>	Penjadwalan <i>Training</i> Luring
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>Mengeklik salah satu modul <i>training</i>.</li> <li>Memilih tombol tambah konten.</li> <li>Mengetik judul konten, kemudian memilih untuk menambahkan konten bertipe Jadwal <i>Training</i>.</li> <li>Mengisi formulir Jadwal <i>Training</i>.</li> <li>Memilih tombol “simpan dan tutup”.</li> <li>Mengeklik konten yang baru saja ia buat.</li> <li>Mengeklik tombol “Tambahkan ke Google Calendar”.</li> <li>Mengedit data Jadwal <i>Training</i> luring.</li> <li>Memilih tombol “simpan dan tutup”.</li> <li>Mengeklik tombol hapus konten.</li> <li>Mengeklik tombol ya.</li> <li>Mengetik judul konten dengan benar.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>Dapat menampilkan konten kuis yang baru saja dibuat/diedit.</li> <li>Dapat mengarahkan pengguna ke Google Calendar untuk menambahkan jadwal <i>training</i> ke kalender personal.</li> <li>Menampilkan pesan eror validasi jika aktor mengetik tombol validasi dengan salah.</li> <li>Menghapus konten dari modul <i>training</i> jika aktor memilih untuk menghapusnya.</li> </ul>

<b>Kesimpulan</b>	OK
-------------------	----

**Tabel 4.75 Pengujian Penjadwalan VLT (*Virtual Live Training*)**

<b>Nama Pengujian</b>	Penjadwalan VLT ( <i>Virtual Live Training</i> )
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Mengklik salah satu modul <i>training</i>.</li> <li>2. Memilih tombol tambah konten.</li> <li>3. Mengetik judul konten, kemudian memilih untuk menambahkan konten bertipe Jadwal VLT.</li> <li>4. Mengisi formulir Jadwal VLT.</li> <li>5. Memilih tombol “simpan dan tutup”.</li> <li>6. Mengklik konten yang baru saja ia buat.</li> <li>7. Mengklik tombol “Tambahkan ke Google Calendar”.</li> <li>8. Mengedit data Jadwal VLT.</li> <li>9. Memilih tombol “simpan dan tutup”.</li> <li>10. Mengklik tombol hapus konten.</li> <li>11. Mengklik tombol ya.</li> <li>12. Mengetik judul konten dengan benar.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Dapat menampilkan konten kuis yang baru saja dibuat/diedit.</li> <li>• Dapat mengarahkan pengguna ke Google Calendar untuk menambahkan jadwal <i>training</i> ke kalender personal.</li> <li>• Menampilkan pesan eror validasi jika aktor mengetik tombol validasi dengan salah.</li> <li>• Menghapus konten dari modul <i>training</i> jika aktor memilih untuk menghapusnya.</li> </ul>
<b>Kesimpulan</b>	OK

**Tabel 4.76 Pengujian Kelola Akses PIC Modul *Training***

<b>Nama Pengujian</b>	Kelola Akses PIC Modul <i>Training</i>
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Mengklik salah satu modul <i>training</i>.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Memilih menu kelola PIC <i>training</i>.</li> <li>3. Memilih satu atau beberapa karyawan untuk dijadikan PIC <i>training</i>.</li> <li>4. Mengklik tombol simpan.</li> <li>5. Menghapus salah satu PIC <i>training</i> dan mengklik tombol simpan.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Menampilkan daftar PIC <i>Training</i> yang telah ditambahkan dan menghapus PIC <i>Training</i> yang telah dihapus.</li> </ul>
<b>Kesimpulan</b>	Belum Selesai

**Tabel 4.77 Pengujian Kelola Akses Peserta Modul *Training***

<b>Nama Pengujian</b>	Kelola Akses Peserta Modul <i>Training</i>
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Memilih menu kelola peserta <i>training</i>.</li> <li>2. Mengklik tombol tambah jenis peserta <i>training</i> baru.</li> <li>3. Memilih peserta <i>training</i> berdasarkan divisi, departemen, atau langsung ke karyawan yang spesifik.</li> <li>4. Mengklik tombol simpan.</li> <li>5. Menghapus salah satu jenis peserta <i>training</i> dan mengklik tombol simpan.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Menampilkan daftar Peserta <i>Training</i> yang telah ditambahkan dan menghapus Peserta <i>Training</i> yang telah dihapus.</li> </ul>
<b>Kesimpulan</b>	Belum Selesai

**Tabel 4.78 Pengujian Akses Laporan *Training***

<b>Nama Pengujian</b>	Akses Laporan <i>Training</i>
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Mengklik menu laporan <i>Training</i>.</li> <li>2. Memfilter penampil laporan <i>training</i>.</li> </ol>

	3. Memilih salah satu data peserta <i>training</i> .
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>Menampilkan laporan hasil <i>training</i> peserta.</li> </ul>
<b>Kesimpulan</b>	Belum Selesai

**Tabel 4.79 Pengujian Akses Modul *Training***

<b>Nama Pengujian</b>	Akses Modul <i>Training</i>
<b>Pengguna</b>	Peserta <i>Training</i> , Pemateri <i>Training</i>
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>Mengeklik salah satu modul <i>training</i>.</li> <li>Mengeklik konten <i>training</i> bertipe teksual, gambar, dan video.</li> <li>Mengeklik konten <i>training</i> bertipe kuis.</li> <li>Mengeklik konten <i>training</i> bertipe berkas PDF.</li> <li>Mengeklik konten <i>training</i> bertipe jadwal <i>training</i>.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>Dapat menampilkan konten sesuai dengan masing-masing tipe konten.</li> </ul>
<b>Kesimpulan</b>	OK

**Tabel 4.80 Pengujian Akses Forum *Training*/Menampilkan Posting Forum**

<b>Nama Pengujian</b>	Akses Forum <i>Training</i> /Menampilkan Posting Forum
<b>Pengguna</b>	Peserta <i>Training</i> , Pemateri <i>Training</i> , PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>Mengeklik salah satu modul <i>training</i>.</li> <li>Mengeklik salah satu konten modul <i>training</i>.</li> <li>Mengeklik tombol forum pada tampilan konten modul <i>training</i>.</li> <li>Membuka salah satu posting forum.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>Dapat menampilkan posting forum sesuai dengan konten modul <i>training</i> yang telah dipilih.</li> </ul>

<b>Kesimpulan</b>	Belum Selesai
-------------------	---------------

**Tabel 4.81 Pengujian Akses Forum *Training*/Balasan Posting Forum**

<b>Nama Pengujian</b>	Akses Forum <i>Training</i> /Balasan Posting Forum
<b>Pengguna</b>	Peserta <i>Training</i> , Pemateri <i>Training</i> , PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>Mengetik balasan pada kotak formulir yang telah tersedia di bawah posting forum <i>training</i>. Lalu, aktor mengeklik tombol <i>submit</i>.</li> <li>Mengeklik tombol kebab menu (⋮), lalu memilih menu edit balasan.</li> <li>Mengedit teks balasan forum, lalu ia mengeklik tombol simpan.</li> <li>Mengeklik tombol kebab menu (⋮), lalu memilih menu hapus balasan.</li> <li>Mengetik verifikasi penghapusan balasan forum dengan benar.</li> <li>Mengeklik tombol hapus.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>Dapat menampilkan balasan posting forum <i>training</i>, menampilkan perubahannya, lalu menandakan bahwa balasan forum <i>training</i> telah dihapus.</li> </ul>
<b>Kesimpulan</b>	Belum Selesai

**Tabel 4.82 Pengujian Akses Forum *Training*/Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul *Training***

<b>Nama Pengujian</b>	Akses Forum <i>Training</i> /Pembuatan, Pengeditan, dan Penghapusan Posting Forum Modul <i>Training</i>
<b>Pengguna</b>	Peserta <i>Training</i> , Pemateri <i>Training</i> , PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>Mengeklik tombol buat posting forum modul <i>training</i> baru yang ditandai ikon tambah (⊕).</li> <li>Mengetik formulir pembuatan posting forum dengan benar, kemudian mengeklik tombol buat posting forum baru.</li> <li>Mengeklik posting forum <i>training</i> yang telah ia buat sebelumnya.</li> </ol>

	<ol style="list-style-type: none"> <li>4. Mengklik tombol kebab menu (:), lalu memilih menu edit posting.</li> <li>5. Mengedit posting forum <i>training</i>, lalu mengklik tombol simpan.</li> <li>6. Mengklik tombol kebab menu (:), lalu memilih menu hapus posting.</li> <li>7. Mengetik verifikasi penghapusan posting forum dengan benar.</li> <li>8. Mengklik tombol hapus.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Menampilkan posting forum modul <i>training</i> yang telah dibuat, menampilkan perubahannya bila diedit, dan menandai posting apabila dihapus oleh pemiliknya.</li> </ul>
<b>Kesimpulan</b>	Belum Selesai

**Tabel 4.83 Pengujian Akses Forum *Training*/Pemberian *Voting* dan Penandaan Balasan Forum *Training* sebagai Solusi**

<b>Nama Pengujian</b>	Akses Forum <i>Training</i> /Pemberian <i>Voting</i> dan Penandaan Balasan Forum <i>Training</i> sebagai Solusi
<b>Pengguna</b>	Peserta <i>Training</i> , Pemateri <i>Training</i> , PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Memilih salah satu posting forum <i>training</i>.</li> <li>2. Memilih tombol <i>voting</i> ke atas pada salah satu balasan forum <i>training</i>.</li> <li>3. Memilih tombol <i>voting</i> ke bawah pada balasan forum <i>training</i> lainnya.</li> <li>4. Mengklik tombol kembali untuk ke halaman daftar posting forum <i>training</i>.</li> <li>5. Memilih posting forum <i>training</i> miliknya.</li> <li>6. Memilih tombol tandai sebagai solusi dengan ikon centang (<input checked="" type="checkbox"/>).</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Menampilkan balasan posting mendapatkan <i>voting</i> ke atas atau ke bawah.</li> <li>• Menampilkan tanda bahwa balasan posting merupakan solusi dari posting forum <i>training</i>.</li> </ul>

<b>Kesimpulan</b>	Belum Selesai
-------------------	---------------

**Tabel 4.84 Pengujian Moderasi Forum *Training***

<b>Nama Pengujian</b>	Moderasi Forum <i>Training</i>
<b>Pengguna</b>	PIC <i>Training</i> , Karyawan HC
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Mengeklik tombol kebab menu (:) pada posting forum. Aktor akan melihat beberapa menu tambahan untuk melakukan moderasi posting seperti edit sebagai administrator dan hapus sebagai administrator. Aktor memilih edit sebagai administrator.</li> <li>2. Mengedit teks posting forum. Kemudian, aktor mengeklik tombol simpan perubahan.</li> <li>3. Mengetik alasan perubahan dan mengetik verifikasi dengan benar, lalu aktor mengeklik tombol edit.</li> <li>4. Mengeklik tombol kebab menu (:) pada posting forum. Kemudian, aktor mengeklik menu hapus sebagai administrator.</li> <li>5. Mengetik alasan penghapusan posting forum dan mengetik verifikasi dengan benar, lalu aktor mengeklik tombol hapus.</li> <li>6. Mengeklik tombol kebab menu (:) pada balasan posting forum di bawahnya. Kemudian, aktor mengeklik menu edit sebagai administrator.</li> <li>7. Mengedit teks balasan posting forum. Kemudian, aktor mengeklik tombol simpan perubahan.</li> <li>8. Mengisi alasan perubahan dan mengetik verifikasi dengan benar, lalu aktor mengeklik tombol edit.</li> <li>9. Mengeklik tombol kebab menu (:) pada balasan posting forum sebelumnya. Kemudian, aktor mengeklik menu hapus sebagai administrator.</li> <li>10. Mengetik alasan penghapusan balasan posting forum dan mengetik verifikasi dengan benar, lalu aktor mengeklik tombol hapus.</li> </ol>
<b>Masukan (Input)</b>	-
<b>Keluaran/Hasil (Output)</b>	<ul style="list-style-type: none"> <li>• Dapat menampilkan tanda hasil dari aksi moderasi, baik itu pengeditan atau penghapusan yang dilakukan oleh administrator.</li> </ul>
<b>Kesimpulan</b>	Belum Selesai

#### **4.2.4 Ulasan untuk tahapan *Production* (Produksi) dan Implikasi Penelitian**

Penulis pada penelitian ini tidak sampai pada tahapan produksi. Seperti yang telah dijelaskan sebelumnya, tujuan penelitian ini adalah membuat solusi baru dalam melaksanakan *training* karyawan, mengembangkan solusi tersebut dalam aplikasi yang sudah dimiliki oleh HC yakni HC Portal, dan membantu mereka untuk mewujudkan pengembangan *software* yang berkelanjutan.

Solusi baru untuk melaksanakan *training* karyawan adalah dengan membuat sebuah platform untuk membantu mewadahi proses *training*. *E-learning* adalah solusinya berdasarkan riset dari penelitian sejenis yang telah dilakukan. Platform *e-learning* ini dapat menyimpan materi bahan *training* dalam 4 tipe konten, yakni bertipe tekstual, gambar, video, kuis, berkas *pdf* untuk dek presentasi dan lainnya, serta jadwal *live training*. Karyawan dapat mengakses materi *training* secara langsung melalui platform *e-learning* ini. Hal ini membuat pengaksesan materi *training* menjadi lebih mudah untuk dikelola dan digunakan oleh karyawan peserta *training*.

HC memiliki solusi *software* yang telah dikembangkan sebelumnya, yakni HC Portal. Mereka memiliki kebutuhan untuk menambahkan *e-learning* ini sebagai modul aplikasi di *software* tersebut. Di penelitian ini, penulis mengembangkannya di HC Portal. Tidak hanya itu, HC juga memiliki kebutuhan untuk mengembangkan HC Portal secara berkelanjutan. Hal ini memberikan tantangan kepada penulis untuk menyusun solusi atas kebutuhan tersebut. Penulis menggunakan Github sebagai solusinya. Penulis membuat repositori kode program dan beberapa petunjuk berupa berkas README.md. Penulis juga melaksanakan pengembangan *e-learning* ini dengan memanfaatkan penuh fitur-fitur dari Github seperti sistem Git, Issues, Pull-Request, dan Project. Penulis mencatat kebutuhan fitur pengembangan *Software* dan catatan pengembangannya. Ini penulis lakukan dengan harapan agar pengembangan selanjutnya untuk aplikasi HC Portal dapat lebih mudah untuk dilakukan.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Kesimpulan yang dapat ditarik berdasarkan hasil dan pembahasan penelitian ini dari bab-bab sebelumnya:

1. Solusi *e-learning* ini membuat proses pengelolaan *training* seperti proses pencarian dan pengumpulan data dan informasi menjadi terpusat dalam satu platform. Hal ini membuat proses pembuatan laporan hasil *training* menjadi lebih mudah untuk dilakukan.
2. Solusi *e-learning* ini mempermudah persiapan proses *training* dengan memanfaatkan fitur pengajuan *training* baru untuk kepala departemen yang ingin mengajukan *training* karyawan dan memudahkan pengelolaan materi *training* dengan fitur pengelolaan modul *training*. Serta pembuatan *e-learning* dapat memberikan cara baru yang lebih fleksibel dalam hal waktu untuk melaksanakan proses *training* karyawan selain *training* secara luring dan daring. Karyawan dapat mengakses modul *training* dengan mudah melalui solusi *e-learning* ini.
3. Solusi *e-learning* ini dikembangkan di *software* berbasis web yang telah dimiliki oleh HC, yakni HC Portal. Hal ini memberikan gambaran untuk ke depannya CTI dapat membuat satu solusi digital yang mencakup beberapa proses kerja. Hal ini juga memudahkan pengelolaan solusi digital yang telah dikembangkan di CTI. Penggunaan metode pengembangan *software iterative model* dan pembuatan repositori digital dengan platform Github mendukung pengembangan HC Portal yang berkelanjutan. Penulis membahas proses pengembangan solusi *e-learning* ini dengan memanfaatkan penuh fitur-fitur Github agar pengembangan HC Portal selanjutnya dapat dengan mudah dilakukan.

Selain kesimpulan di atas, terdapat dua hal yang penulis temukan saat melaksanakan tahapan analisis dan pengembangan *software* dengan Github.

Hal pertama terkait dengan tahapan analisis dan desain. Penulis hanya melaksanakan tahapan ini sekali pada iterasi pertama saja. Untuk iterasi

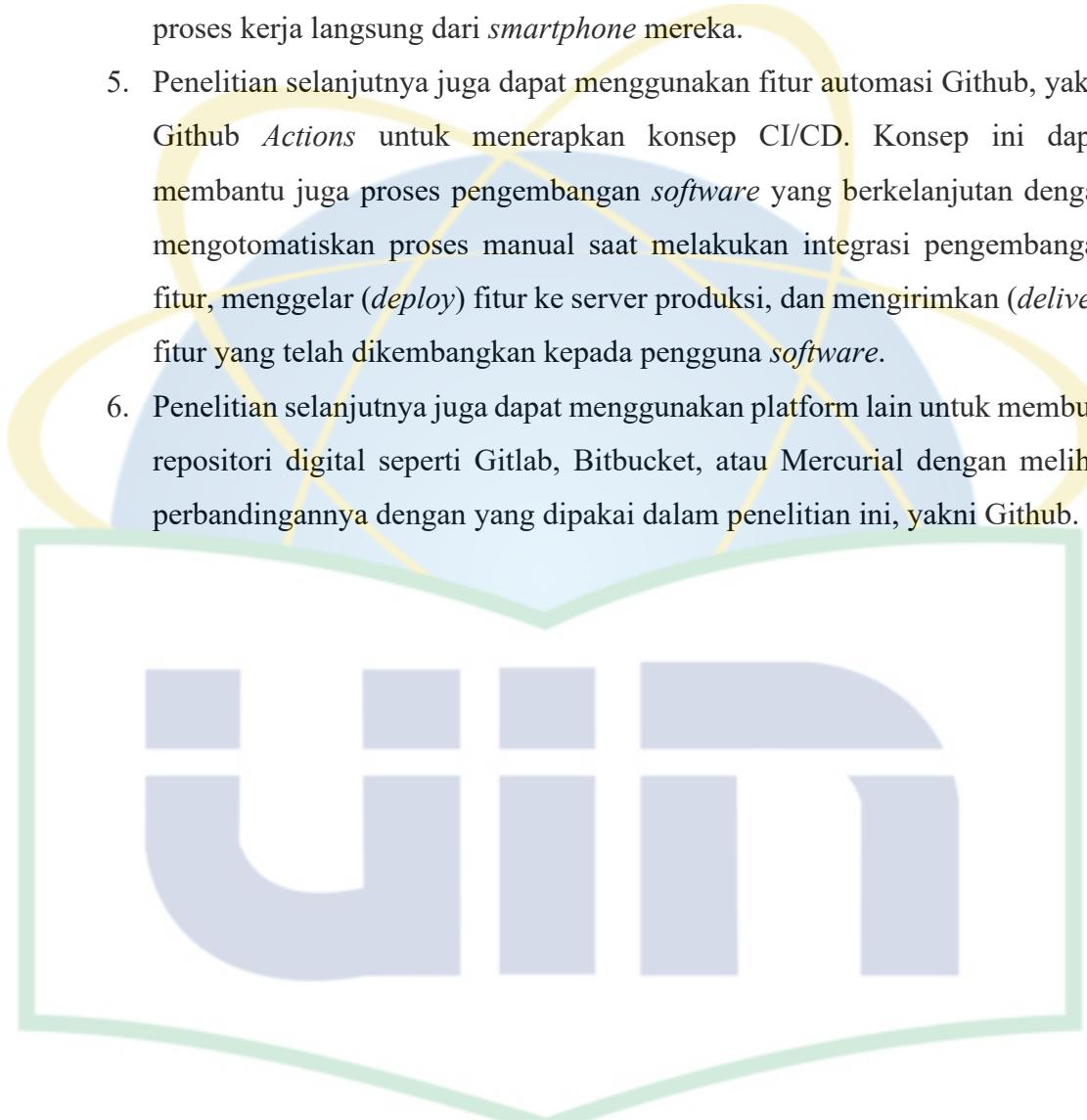
selanjutnya, penulis hanya melakukan modifikasi atau perbaikan dari analisis dan desain sistem yang telah dibuat di iterasi pertama. Ini membuat iterasi pertama memakan waktu yang sangat lama karena penulis membuat semua desain aplikasi terlebih dahulu. Setelah masuk ke iterasi selanjutnya, penulis hanya bermain dengan Github untuk merincikan kebutuhan dari pengembangan fitur *e-learning* dengan Github *Issues* pada saat memasuki tahapan *backlog* di setiap pengembangan fitur. Inilah yang penulis alami saat memasuki tahapan implementasi.

Hal kedua masih berkaitan dengan sebelumnya. Namun, kali ini membahas tentang perencanaan iterasi pengembangan yang sebaiknya dilakukan setelah tahapan analisis dan desain selesai. Hal ini karena sebelum menentukan berapa iterasi yang harus kita lakukan, kita perlu mendapatkan gambaran seperti apa bentuk fitur *software* yang harus kita buat. Namun di *iterative model* ini, kita diminta untuk melakukan review desain yang telah kita buat di iterasi berikutnya. Pembuatan desain biasanya hanya dilakukan di awal yang berikutnya hanya berupa revisi desain saja. Hal ini juga berkaitan dengan manajemen tugas di Github Project yang lebih baik dilakukan setelah tahapan analisis dan desain *software*.

## 5.2 Saran

Penulis sadar bahwa dalam penelitian ini masih terdapat kekurangan yang masih harus ditingkatkan lebih lanjut. Terdapat beberapa saran untuk penelitian selanjutnya. Berikut uraiannya:

1. Penelitian selanjutnya dapat mencoba metode pengembangan lain selain *Iterative Model* tentunya yang tidak melupakan tahapan analisis. Mungkin tahapan yang mengedepankan iterasi seperti *Extreme Programming* (XP), *Spiral Model*, ataupun Agile dapat digunakan dalam penelitian selanjutnya, hanya saja perlu dilakukan penelitian lebih lanjut terkait bagaimana cara melakukan analisis di metode pengembangan *software* tersebut.
2. Penelitian selanjutnya dapat menambahkan konsep TDD (*Test Driven Development*) untuk membantu dalam mengurangi eror fungsi aplikasi yang telah dikembangkan sebelumnya. Hal ini dapat mengurangi eror pada fitur lama aplikasi saat mengembangkan fitur baru yang memerlukan integrasi. Hal ini juga mendukung konsep pengembangan *software* yang berkelanjutan.

- 
3. Penelitian selanjutnya dapat menerapkan konsep pengembangan OOP pada topik lain selain Portal HC ataupun mengintegrasikan Portal HC dengan proses kerja di luar dari divisi HC.
  4. Penelitian selanjutnya dapat melakukan pengembangan aplikasi berbasis *mobile* agar pengguna dapat lebih mudah mengakses dan menerima notifikasi proses kerja langsung dari *smartphone* mereka.
  5. Penelitian selanjutnya juga dapat menggunakan fitur automasi Github, yakni Github *Actions* untuk menerapkan konsep CI/CD. Konsep ini dapat membantu juga proses pengembangan *software* yang berkelanjutan dengan mengotomatiskan proses manual saat melakukan integrasi pengembangan fitur, menggelar (*deploy*) fitur ke server produksi, dan mengirimkan (*deliver*) fitur yang telah dikembangkan kepada pengguna *software*.
  6. Penelitian selanjutnya juga dapat menggunakan platform lain untuk membuat repositori digital seperti Gitlab, Bitbucket, atau Mercurial dengan melihat perbandingannya dengan yang dipakai dalam penelitian ini, yakni Github.



## DAFTAR PUSTAKA

Agilealliance. (2015, Juni 29). What is Agile Software Development? [Blog].

Diambil 27 April 2021, dari Agile Alliance website:

<https://www.agilealliance.org/agile101/>

Agustin, F. E. M., Anggraini, N., & Syarifah, L. (2018). *A Mandarin language learning mobile application for indonesian-speaking using Computer Assisted Instruction (CAI) method* (hlm. 5) [Working Paper]. Jakarta: UIN Syarif Hidayatullah Jakarta. Diambil dari UIN Syarif Hidayatullah Jakarta website: <https://repository.uinjkt.ac.id/dspace/handle/123456789/46159>

Albert, W., & Tullis, T. (2013). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics* (2 ed.). Waltham: Newnes.

Diambil dari <https://books.google.co.id/books?id=bPhLeMBLEkAC>

Alshamrani, A., & Bahattab, A. (2015). A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI)*, 12(1), 106–111.

Anggraeni, E. Y. (2017). *Pengantar Sistem Informasi*. Yogyakarta: Penerbit Andi.

Diambil dari <https://books.google.co.id/books?id=8VNLDwAAQBAJ>

Artemiou, A. (2018, September 19). Benefits of Primary Keys in Database Tables [Blog]. Diambil 12 Oktober 2021, dari Tech How To's website:

<https://www.techhowtos.com/databases/benefits-of-primary-key-in-a-database/>

Black, R. (2018, Mei 1). What do you know about continuous software development? [Blog]. Diambil 22 Oktober 2022, dari SearchSoftwareQuality website:

- <https://www.techtarget.com/searchsoftwarequality/quiz/What-do-you-know-about-continuous-software-development>
- Blischak, J. D., Davenport, E. R., & Wilson, G. (2016). A Quick Introduction to Version Control with Git and GitHub. *PLOS Computational Biology*, 12(1), e1004668. <https://doi.org/10.1371/journal.pcbi.1004668>
- Budi, D. S., Siswa, T. A. Y., & Abijono, H. (2016). Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak. *Teknika*, 5(1), 24–31. <https://doi.org/10.34148/teknika.v5i1.48>
- Chacon, S., & Straub, B. (2014). *Pro Git* (2 ed.). California, US: Apress. Diambil dari <https://books.google.co.id/books?id=jVYnCgAAQBAJ>
- Chapple, M. (2021, Juli 17). What Are Schemas and What Is Their Relationship to Databases? [Blog]. Diambil 22 Oktober 2021, dari Lifewire website: <https://www.lifewire.com/definition-of-a-schema-in-a-database-1019262>
- Delisle, M. (2012). *Phpmyadminstarter* (1 ed.). Birmingham, UK: Packt Publishing Ltd. Diambil dari <https://books.google.co.id/books?id=-8Rv8BlSotgC>
- diagrams.net. (2021). About diagrams.net [Company Profile]. Diambil 10 November 2021, dari Diagrams.net website: <https://www.diagrams.net/about>
- Effendey, S. S. A. (2018). *Pengembangan aplikasi e-learning pada bimbingan belajar kiswah* (Bachelor Thesis, Universitas Islam Negeri Syarif Hidayatullah Jakarta). Universitas Islam Negeri Syarif Hidayatullah Jakarta, Jakarta. Diambil dari <https://repository.uinjkt.ac.id/dspace/handle/123456789/55767>

- Ege, R. K. (2014). *Programming in an Object-Oriented Environment*. California: Academic Press. Diambil dari  
<https://books.google.co.id/books?id=SjGjBQAAQBAJ>
- Elmasri, R., & Navathe, S. (2016). *Fundamentals of Database Systems* (7 ed.). United States: Pearson. Diambil dari  
[https://books.google.co.id/books?id=Yb\\_uoQEACAAJ](https://books.google.co.id/books?id=Yb_uoQEACAAJ)
- Enterprise, J. (2015). *Membuat Website PHP dengan CodeIgniter* (1 ed.). Jakarta: Elex Media Komputindo. Diambil dari  
<https://books.google.co.id/books?id=qE1JDwAAQBAJ>
- Fadlilah, U. (2015). Rancang Bangun Website dan E-Learning di TPQ Al-Fadhillah. *Khazanah Informatika : Jurnal Ilmu Komputer dan Informatika*, 1(1), 40–43. <https://doi.org/10.23917/khif.v1i1.1181>
- Figma. (2020). About Figma, the collaborative interface design tool. [Company Profile]. Diambil 14 November 2021, dari Figma website:  
<https://www.figma.com/about/>
- Fikry, M. (2019). Rancangan Basis Data Kependudukan Berdasarkan Aspek-Aspek Kualitas Schema Database. *TECHSI - Jurnal Teknik Informatika*, 8(2), 28–43. <https://doi.org/10.29103/techsi.v8i2.136>
- Gibb, R. (2016, Mei 31). What is a Web Application? | How a Web Application Works [Blog]. Diambil 16 Agustus 2021, dari Articles for Developers Building High Performance Systems website:  
<https://blog.stackpath.com/web-application/>
- Github. (2020). GitHub features: The right tools for the job [Company Profile]. Diambil 11 Mei 2021, dari GitHub website: <https://github.com/features>

Github. (2022). About repositories [Documentation]. Diambil 26 Juni 2023, dari GitHub Docs website: <https://ghdocs-prod.azurewebsites.net/en/repositories/creating-and-managing-repositories/about-repositories>

Greenbaum, J., & Kyng, M. (2020). *Design at Work: Cooperative Design of Computer Systems*. New Jersey: CRC Press. Diambil dari <https://books.google.co.id/books?id=KAoHEAAAQBAJ>

Grobelny, M., & Grobelna, I. (2015). UML activity diagram swimlanes in logic controller design. *AIP Conference Proceedings*, 1702, 100009. Athens, Greece: Scitation. <https://doi.org/10.1063/1.4938886>

Gultom, H. (2020, September 25). *Data-Data yang Digunakan dalam Proses Asuhan Keperawatan dan Metode Pengumpulan Data* (hlm. 7) [Article]. hlm. 7. OSF Preprints. <https://doi.org/10.31219/osf.io/efxq9>

Habsy, B. A. (2017). Seni Memahami Penelitian Kuliatatif dalam Bimbingan dan Konseling: Studi Literatur. *Jurnal Konseling Andi Matappa*, 1(2), 90–100. <https://doi.org/10.31100/jurkam.v1i2.56>

Hamilton, T. (2021, Oktober 8). What is Software Testing? Definition, Basics & Types in Software Engineering [Blog]. Diambil 3 November 2021, dari Guru99 website: <https://www.guru99.com/software-testing-introduction-importance.html>

Hillar, G. C. (2015). *Learning Object-Oriented Programming*. Mumbai: Packt Publishing Ltd. Diambil dari <https://books.google.co.id/books?id=zicxCgAAQBAJ>

- Jadhav, S. (2020, September 8). An introduction to continuous software development | Globant Blog [Blog]. Diambil 22 Oktober 2022, dari Globant website: <https://stayrelevant.globant.com/en/continuous-software-development/>
- Kalliamvakou, E., Damian, D., Blincoe, K., Singer, L., & German, D. M. (2015). Open Source-Style Collaborative Development Practices in Commercial Projects Using GitHub. *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, 1*, 574–585. Florence, Italy: IEEE Xplore. <https://doi.org/10.1109/ICSE.2015.74>
- Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., & Damian, D. (2014). The promises and perils of mining GitHub. *Proceedings of the 11th Working Conference on Mining Software Repositories*, 92–101. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2597073.2597074>
- KBBI. (2016a). Hasil Pencarian Kata “Digital” [Dictionary]. Diambil 26 Juni 2023, dari KBBI Daring website: <https://kbbi.kemdikbud.go.id/entri/digital>
- KBBI. (2016b). Hasil Pencarian Kata “Repositori” [Dictionary]. Diambil 26 Juni 2023, dari KBBI Daring website: <https://kbbi.kemdikbud.go.id/entri/repositori>
- Keller, S., Krafft, M., Fraser, G., Walkinshaw, N., Otto, K., & Sabitzer, B. (2019). Improving Scratch Programming with CRC-Card Design. *Proceedings of the 14th Workshop in Primary and Secondary Computing Education*, 1–4.

New York, NY, USA: Association for Computing Machinery.

<https://doi.org/10.1145/3361721.3362114>

Kemenkeu RI. (2022, November 16). Presiden Sebut Tiga Fokus Utama Transformasi Digital Global [News]. Diambil 12 Januari 2023, dari Berita Utama Kemenkeu website: <https://www.kemenkeu.go.id/informasi-publik/publikasi/berita-utama/Presiden-Sebut-Tiga-Fokus-Utama-Transformasi>

Kemkes RI. (2022, Desember 30). PPKM di Indonesia Resmi Dicabut [Blog].

Diambil 12 Januari 2023, dari Sehat Negeriku website:

<https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20221230/0042128/ppkm-di-indonesia-resmi-dicabut/>

Kertész, C.-Z. (2015). Using GitHub in the classroom—A collaborative learning experience. *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 381–386.

<https://doi.org/10.1109/SIITME.2015.7342358>

Kittivaraporn, J., Chokdeeanan, J., Yaophrukchai, T., & Sunetnanta, T. T. (2014). HRM portal: Human resource management portal. *2014 Third ICT International Student Project Conference (ICT-ISPC)*, 183–186. Nakhon Pathom, Thailand: IEEE. <https://doi.org/10.1109/ICT-ISPC.2014.6923246>

Kravchuk, I. (2021, Mei 19). Learn the Difference Between Software vs Application. Diambil 14 Agustus 2021, dari Tateeda.com website: <https://tateeda.com/blog/difference-between-software-vs-application>

- Laaziri, M., Benmoussa, K., Khoulji, S., & Kerkeb, M. L. (2019). A Comparative study of PHP frameworks performance. *Procedia Manufacturing*, 32, 864–871. <https://doi.org/10.1016/j.promfg.2019.02.295>
- Latief, M. (2012). Pendekatan Database untuk Manajemen Data dalam Meningkatkan Kemampuan Mahasiswa Mengaplikasikan Konsep Basisdata. *Prosiding APTEKINDO*, 6(1), 231–238.
- Lawrance, J., Jung, S., & Wiseman, C. (2013). Git on the cloud in the classroom. *Proceeding of the 44th ACM technical symposium on Computer science education*, 639–644. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2445196.2445386>
- Lutfiani, D. (2020, Desember 20). Apa yang Dimaksud dengan Repositori Institusi? [Forum]. Diambil 26 Juni 2023, dari Dictio Community website: <https://www.dictio.id/t/apa-yang-dimaksud-dengan-repositori-institusi/150457/2>
- Luthfyana, L. F., & Sediyono, E. (2021). Rancang Bangun Aplikasi Smart Training berbasis Android Menggunakan Flutter dengan Metode RAD. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 8(2), 420–437. <https://doi.org/10.35957/jatisi.v8i2.828>
- Mahmud, R., Ramamohanarao, K., & Buyya, R. (2018). Latency-Aware Application Module Management for Fog Computing Environments. *ACM Transactions on Internet Technology*, 19(1), 9:1-9:21. <https://doi.org/10.1145/3186592>

Makbul, M. (2021, Juni 15). *Metode Pengumpulan Data dan Instrumen Penelitian* (hlm. 38) [Article]. hlm. 38. Makassar: OSF Preprints.

<https://doi.org/10.31219/osf.io/svu73>

McMinn, K. (2015, Januari 22). How to write the perfect pull request [Blog].

Diambil 26 Januari 2023, dari The GitHub Blog website:

<https://github.blog/2015-01-21-how-to-write-the-perfect-pull-request/>

Mongi, N. S., & Hendry, H. (2021). Analisis Pengembangan dan Implementasi System E-learning Untuk Meningkatkan pengetahuan Agent Menggunakan Metode ADDIE Model (Study Kasus: PT.Global Infotech Solution). *Jurnal Sistem Komputer dan Informatika (JSON)*, 2(3), 269–276. <https://doi.org/10.30865/json.v2i3.2920>

Mulyani, S. (2017). *Metode Analisis dan Perancangan Sistem*. Bandung: Abdi Sistematika. Diambil dari

<https://books.google.co.id/books?id=SbrPDgAAQBAJ>

Munawar. (2018). *Analisis Perancangan Sistem Berorientasi Objek dengan UML (Unified Modelling Language)*. Bandung: Informatika Bandung.

Mustajab, D., Bauw, A., Rasyid, A., Irawan, A., Akbar, M. A., & Hamid, M. A. (2020). Working From Home Phenomenon As an Effort to Prevent COVID-19 Attacks and Its Impacts on Work Productivity. *TIJAB (The International Journal of Applied Business)*, 4(1), 13–21.

<https://doi.org/10.20473/tijab.V4.I1.2020.13-21>

Nalini, S. N. L. (2021). Dampak covid-19 terhadap Usaha Mikro, Kecil dan Menengah. *Jesya (Jurnal Ekonomi Dan Ekonomi Syariah)*, 4(1), 662–669.

<https://doi.org/10.36778/jesya.v4i1.278>

- Nixon, R. (2015). *Learning PHP, MySQL & JavaScript: With JQuery, CSS & HTML5* (4 ed.). United States of America: O'Reilly Media, Inc. Diambil dari <https://books.google.co.id/books?id=uHqrBQAAQBAJ>
- Nugroho, S., Waluyo, S. H., & Hakim, L. (2017). Comparative Analysis of Software Development Methods between Parallel, V-Shaped and Iterative. *International Journal of Computer Applications*, 169(11), 7–11.  
<https://doi.org/10.5120/ijca2017914605>
- Nyakundi, H. (2021, Desember 8). How to Write a Good README File for Your GitHub Project [Blog]. Diambil 31 Desember 2022, dari FreeCodeCamp.org website: <https://www.freecodecamp.org/news/how-to-write-a-good-readme-file/>
- Okesola, O. J., Adebiyi, A. A., Owoade, A. A., Adeaga, O., Adeyemi, O., & Odun-Ayo, I. (2020). Software Requirement in Iterative SDLC Model. Dalam R. Silhavy (Ed.), *Intelligent Algorithms in Software Engineering* (Vol. 1224, hlm. 26–34). Cham: Springer International Publishing.  
[https://doi.org/10.1007/978-3-030-51965-0\\_2](https://doi.org/10.1007/978-3-030-51965-0_2)
- Olanrewaju, R. F., Islam, T., & Ali, N. (2015). An Empirical Study of the Evolution of PHP MVC Framework. Dalam H. A. Sulaiman, M. A. Othman, M. F. I. Othman, Y. A. Rahim, & N. C. Pee (Ed.), *Proceedings of the 1st International Conference on Communication and Computer Engineering* (hlm. 399–410). Cham: Springer International Publishing.  
[https://doi.org/10.1007/978-3-319-07674-4\\_40](https://doi.org/10.1007/978-3-319-07674-4_40)

- Oliver, A. (2020, Desember 10). Web Development: Arti, Jenis, dan Proses Kerja [Blog]. Diambil 16 Agustus 2021, dari Glints Blog website: <https://glints.com/id/lowongan/web-development/>
- Pamungkas, C. A. (2017). *Pengantar dan Implementasi Basis Data* (1 ed.). Yogyakarta: Deepublish. Diambil dari <https://books.google.co.id/books?id=hKdADwAAQBAJ>
- Pamungkas, F. L. (2016). *Rancang bangun modul pelatihan Karyawan untuk peningkatan SDM pada USP Swamitra Sumberpucung Malang* (Skripsi, Institut Teknologi Nasional Malang). Institut Teknologi Nasional Malang, Malang, Jawa Timur. <https://doi.org/10.10918/172.pdf>
- Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V., ... Vizcaíno, J. A. (2016). Ten Simple Rules for Taking Advantage of Git and GitHub. *PLOS Computational Biology*, 12(7), e1004947. <https://doi.org/10.1371/journal.pcbi.1004947>
- Peterson, R. (2021, Oktober 7). SQL Commands: DML, DDL, DCL, TCL, DQL with Query Example [Blog]. Diambil 29 November 2021, dari Guru99 website: <https://www.guru99.com/sql-commands-dbms-query.html>
- Porter, J. D. (2014, Desember 18). Camel Case [Wiki]. Diambil 10 September 2021, dari WikiWikiWeb website: <https://wiki.c2.com/?CamelCase>
- Rachmawati, T. (2017). Metode Pengumpulan Data dalam Penelitian Kualitatif. *UNPAR Press. Bandung*, 1–29.
- Rajkumar, S. (2015a, Desember 6). Software Testing—Definition, Types, Methods, Approaches [Blog]. Diambil 3 November 2021, dari Software

Testing Material website:

<https://www.softwaretestingmaterial.com/software-testing/>

Rajkumar, S. (2015b, Desember 21). Levels of Testing [Blog]. Diambil 7

November 2021, dari Software Testing Material website:

<https://www.softwaretestingmaterial.com/levels-of-testing/>

Rajkumar, S. (2018, Agustus 19). Integration Testing—Big Bang, Top Down,

Bottom Up & Hybrid Integration [Blog]. Diambil 7 November 2021, dari Software Testing Material website:

<https://www.softwaretestingmaterial.com/integration-testing/>

Ram, K. (2013). Git can facilitate greater reproducibility and increased

transparency in science. *Source Code for Biology and Medicine*, 8(1), 7.

<https://doi.org/10.1186/1751-0473-8-7>

Rice, S., & McKendree, J. (2013). E-Learning. Dalam *Understanding Medical Education* (2 ed., hlm. 161–173). London, UK: John Wiley & Sons, Ltd.

<https://doi.org/10.1002/9781118472361.ch12>

Rijal, M. S., & Sholihah, N. A. (2022). Penerapan Sistem E-Learning untuk

Meningkatkan Produktifitas Kerja Karyawan di Era Pandemi Covid-19.

*Jurnal Pendidikan dan Kewirausahaan*, 10(1), 254–266.

<https://doi.org/10.47668/pkwu.v10i1.354>

Romindo, R., Niar, H., Sipayung, R., Julyanthry, J., Yendrianof, D., Pelu, M. F.

A., ... Purba, B. (2020). *Sistem Informasi Bisnis*. Medan: Yayasan Kita

Menulis. Diambil dari

<https://books.google.co.id/books?id=qrcKEAAAQBAJ>

- Rubens, N., Kaplan, D., & Okamoto, T. (2014). E-Learning 3.0: Anyone, Anywhere, Anytime, and AI. Dalam D. K. W. Chiu, M. Wang, E. Popescu, Q. Li, & R. Lau (Ed.), *New Horizons in Web Based Learning* (hlm. 171–180). Berlin, Heidelberg: Springer Berlin Heidelberg.  
[https://doi.org/10.1007/978-3-662-43454-3\\_18](https://doi.org/10.1007/978-3-662-43454-3_18)
- Sakovich, N. (2020, Desember 29). Top Ten Software Development Trends in 2020–2021 | SaM Solutions [Blog]. Diambil 5 Mei 2021, dari Sam Solutions website:  
<https://web.archive.org/web/20210111084205/https://www.sam-solutions.com/blog/software-development-trends/>
- Salo, R., Poranen, T., & Zhang, Z. (2015). Requirements management in GitHub with a lean approach. Dalam J. Nummenmaa, O. Sievi-Korte, & E. Mäkinen (Ed.), *Proceedings of the 14th Symposium on Programming Languages and Software Tools* (hlm. 164–178). Tampere, Finland: CEUR. Diambil dari <http://ceur-ws.org/Vol-1525/#paper-12>
- Sauro, S., & Zourou, K. (2019). What are the digital wilds? *Language Learning & Technology*, 23(1), 7. <https://doi.org/10.13016/m2t5ec-yena>
- Sharma, S. (2022). Bitbucket vs GitHub vs GitLab [Blog]. Diambil 26 Januari 2023, dari GeeksforGeeks website:  
<https://www.geeksforgeeks.org/bitbucket-vs-github-vs-gitlab/>
- Shenoy, A., & Sossou, U. (2014). *Learning Bootstrap* (1 ed.). Birmingham, United Kingdom: Packt Publishing Ltd. Diambil dari <https://books.google.co.id/books?id=pswGBgAAQBAJ>

Silkalns, A. (2021). Free Bootstrap Admin Template | AdminLTE.IO [Project Site]. Diambil 2 Desember 2021, dari AdminLTE website:  
<https://adminlte.io>

Somasundaram, R. (2013). *Git: Version Control for Everyone Beginner's Guide* (1 ed.). Birmingham, UK: Packt Publishing Ltd. Diambil dari  
[http://kim.marcelandkim.com/git/Git\\_-\\_Version\\_Control\\_for\\_Everyone.pdf](http://kim.marcelandkim.com/git/Git_-_Version_Control_for_Everyone.pdf)

Spurlock, J. (2013). *Bootstrap: Responsive Web Development* (1 ed.). United States of America: O'Reilly Media, Inc. Diambil dari  
<https://books.google.co.id/books?id=LZm7Cxgi3aQC>

Stephens, R. (2015). *Beginning Software Engineering*. Canada: John Wiley & Sons. Diambil dari  
<https://books.google.co.id/books?id=SyHWBgAAQBAJ>

Sudirman, A., Muttaqin, M., Purba, R. A., Wirapraja, A., Abdillah, L. A., Fajrillah, F., ... Simarmata, J. (2020). *Sistem Informasi Manajemen*. Medan: Yayasan Kita Menulis. Diambil dari  
<https://books.google.co.id/books?id=WiLwDwAAQBAJ>

Sugiarti, Y. (2013). Analisis dan Perancangan UML (Unified Modeling Language) Generated VB. 6. Yogyakarta: Graha Ilmu.

Sugiarti, Y., Suroso, A. I., Hermadi, I., Sunarti, E., & Broer, R. (2021). Knowledge Management System to Improve The Competence of Aglaonema Farmers. *2021 9th International Conference on Cyber and IT Service Management (CITSM)*, 1–4.  
<https://doi.org/10.1109/CITSM52892.2021.9588814>

- Surya, R. A. (2015). *Rancang Bangun E-HRM (Electronic-Human Resources Management) (Studi Kasus: BPRS Harta Insan Karimah Kantor Pusat)* (Bachelor Thesis, UIN Syarif Hidayatullah Jakarta). UIN Syarif Hidayatullah Jakarta, Jakarta. Diambil dari  
<https://repository.uinjkt.ac.id/dspace/handle/123456789/30157>
- Thung, F., Bissyandé, T. F., Lo, D., & Jiang, L. (2013). Network Structure of Social Coding in GitHub. *2013 17th European Conference on Software Maintenance and Reengineering*, 323–326.  
<https://doi.org/10.1109/CSMR.2013.41>
- Torlak, E. (2015, April). *UML Sequence Diagrams*. Powerpoint Presentation dipresentasikan pada CSE 403: Software Engineering, Spring 2015, University of Washington. University of Washington. Diambil dari  
<https://courses.cs.washington.edu/courses/cse403/15sp/lectures/L10.pdf>
- Tulungen, E. E. W., Saerang, D. P. E., & Maramis, J. B. (2022). Transformasi Digital: Peran Kepemimpinan Digital. *Jurnal EMBA : Jurnal Riset Ekonomi, Manajemen, Bisnis Dan Akuntansi*, 10(2), 1116–1123.  
<https://doi.org/10.35794/emba.v10i2.41399>
- Tyoso, J. S. P. (2016). *Sistem Informasi Manajemen*. Yogyakarta: Deepublish. Diambil dari  
<https://books.google.co.id/books?hl=en&lr=&id=0sKIDwAAQBAJ>
- Vuorimaa, P., Laine, M., Litvinova, E., & Shestakov, D. (2016). Leveraging declarative languages in web application development. *World Wide Web*, 19(4), 519–543. <https://doi.org/10.1007/s11280-015-0339-z>

- Wafa, A. S. F. (2019). *Rancang Bangun Sistem Operasional Budidaya Tambak Ikan Kerapu Berbasis Android (Studi Kasus: Kelompok Tani Bhakti Usaha 2)* (Sarjana, Universitas Brawijaya). Universitas Brawijaya, Malang, Jawa Timur. Brawijaya Knowledge Garden. Diambil dari <http://repository.ub.ac.id/169262/>
- Wagner, G. (2019, November 17). Deriving a Table Model from an Information Design Model [Personal Website]. Diambil 31 Oktober 2021, dari Web-engineering.info website: <https://web-engineering.info/book/InformationManagement/MakingTableModels.html>
- Wells, D. (2013, Oktober 8). Extreme Programming: A Gentle Introduction. [Personal Website]. Diambil 27 Mei 2021, dari Extreme Programming website: <http://www.extremeprogramming.org/index.html>
- Widodo, A. W., & Kurnianingtyas, D. (2017). *Sistem Basis Data* (1 ed.). Malang: Universitas Brawijaya Press. Diambil dari <https://books.google.co.id/books?id=GzVTDwAAQBAJ>
- Wu, Y., Kropczynski, J., Shih, P. C., & Carroll, J. M. (2014). Exploring the ecosystem of software developers on GitHub and other platforms. *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing*, 265–268. New York, NY, USA: Association for Computing Machinery.
- <https://doi.org/10.1145/2556420.2556483>
- Yaseen, M., Ibrahim, N., & Mustapha, A. (2019). Requirements Prioritization and using Iteration Model for Successful Implementation of Requirements.

*International Journal of Advanced Computer Science and Applications,*

*10(1), 121–127. <https://doi.org/10.14569/IJACSA.2019.0100115>*

Yusriyanto, A. (2019). *Rancang bangun aplikasi pelatihan online dengan metode blended learning system berbasis website (studi kasus: Pusat data dan informasi Kementerian Kesehatan Republik Indonesia)* (BachelorThesis, Universitas Islam Negeri Syarif Hidayatullah Jakarta). Universitas Islam Negeri Syarif Hidayatullah Jakarta, Jakarta. Diambil dari <https://repository.uinjkt.ac.id/dspace/handle/123456789/51978>

Yusriyanto, A., Arham, Z., & Nuryasin, N. (2018). Rancang Bangun Aplikasi Pelatihan Online dengan Metode Blended Learning System Berbasis Website (Studi Kasus). *Applied Information System and Management (AISM)*, 1(2), 122–125. <https://doi.org/10.15408/aism.v1i2.20109>

Zhou, W. F., Li, X. M., Lv, S. Q., & Zhang, Z. (2013). Automatic Test Case Generation for Context Based Multiplicity Checking in UML. *Applied Mechanics and Materials*, 433–435, 1643–1648.

<https://doi.org/10.4028/www.scientific.net/AMM.433-435.1643>





# Hasil Wawancara

## Detail Wawancara

**Tujuan:** Mengidentifikasi kebutuhan pengembangan *e-learning* pada Centratama Group

**Topik:** Pengembangan *e-learning*

**Narasumber :**

- Wahyudi (HCIS Specialist Centratama Group);

**Kapan wawancara dilakukan?**

Penulis melakukan wawancara pada tanggal 7 April 2021 tepatnya pada jam 19.57 – 22.13.

**Di mana wawancara dilakukan?**

Penulis melakukan wawancara melalui platform daring Telepon Video—Google Meet.

## Wawancara

Penulis melakukan wawancara dengan mengangkat 3 pembahasan. Pertama, penulis menanyakan terkait bagaimana proses dalam melakukan pembelajaran (*learning*) di Centratama Group, siapa saja aktor yang terlibat dan yang bertanggung jawab untuk melaksanakan proses kerja ini. Penulis menanyakan hal mendasar terkait pembelajaran di Centratama. Kedua, penulis menanyakan terkait masalah pembelajaran yang terjadi pada Centratama Group. Ketiga, penulis juga menanyakan tentang urgensi dari pengembangan *e-learning* ini. Penulis menyusun draf ini dalam bentuk naskah dialog dengan aktor yang disimbolkan sebagai berikut,

**M** : Mahasiswa, Penulis

**W**: Narasumber

**M:** Terkait proses pembelajaran (*learning*), apa saja pembelajaran (*learning*) yang sudah ada di Centratama?

**W:** Pembelajaran dimulai dari karyawan baru masuk. Divisi HC (*Human Capital*) yang bertugas untuk melaksanakan proses kerja di bidang HRD (*Human Resource Development*) melakukan proses pengenalan awal terhadap perusahaan, yang disebut *Induction*. Karyawan baru melaksanakan proses *Induction* ini yang ditangani oleh karyawan HC. Ada beberapa hal yang dikenalkan pada proses *Induction* ini:

1. Visi dan Misi Perusahaan
2. Aturan Perusahaan
3. *Product Knowledge*
4. Pengenalan Lingkungan Perusahaan
5. Dan pengenalan awal lainnya yang diperlukan.

HC juga menyediakan pembelajaran dalam bentuk pelatihan (*training*) untuk karyawan lama. Tidak hanya HC, divisi lain pun juga dapat mengajukan pelatihan untuk melatih bawahannya ataupun karyawan lain yang terlibat. Tetapi untuk prosesnya, divisi lain perlu

mengkoordinasikan ke pihak HC untuk diberi arahan lebih lanjut, karena semua proses pelatihan merupakan tanggung jawab HC.

**M: Bagaimana HC melakukan proses pembelajaran di lingkungan perusahaan?**

**W:** Ada 2 jenis proses pembelajaran di lingkungan perusahaan, yaitu *Induction* (pengenalan) dan *training* untuk karyawan. Proses pembelajaran dalam bentuk *Induction* yaitu sebagai berikut,

1. Karyawan baru diterima pada *Job Vacancy* (lowongan kerja) Perusahaan.
2. Karyawan pada hari pertama menemui karyawan HC untuk diberikan *Induction*. Karyawan HC melakukan presentasi untuk karyawan baru ini untuk mengenal visi misi perusahaan, *product knowledge* (produk perusahaan), dan aturan yang berlaku.
3. Karyawan HC memberi pengenalan lingkungan kepada karyawan baru untuk berkeliling ke lingkungan perusahaan.
4. Karyawan HC mengenalkan karyawan baru tersebut ke lingkungan dan rekan kerjanya.

Sementara itu, proses pembelajaran dalam bentuk *training* ada yang diadakan oleh divisi HC maupun divisi lain. *Recruitment Specialist*, posisi pada divisi HC, bertanggung jawab atas proses *training* yang diadakan divisi HC maupun divisi lain. Divisi lain meminta pengadaan *training* kepada divisi HC. Permintaan ini akan diproses oleh *Recruitment Specialist*. Apabila permintaan ini disetujui, *Recruitment Specialist* akan menyiapkan tempat dan memberitahu semua *trainee* (karyawan yang terlibat) dan PIC (orang yang bertanggung jawab untuk pelatihan).

**M: Apa masalah yang terjadi dari proses pembelajaran (*learning*) yang sudah ada?, dan apa solusi sementara yang dilakukan?**

**W:** Masalah yang terjadi yaitu pertama dari kondisi saat ini yang masih dalam keadaan pandemi Covid-19. Pemerintah membatasi tatap muka yang tentunya berpengaruh terhadap proses *training*, akibatnya banyak dari proses *training* ditunda. Untuk sementara solusi yang kami lakukan yaitu mengadakan *training* dengan metode *webinar* yaitu melakukan seminar di platform konferensi video (*video conference*). Platform yang kami gunakan yaitu Zoom. Karyawan HC ataupun pihak dari divisi lain yang melaksanakan proses *training* melakukan presentasi pada platform ini yang terhubung langsung ke para *trainee*.

Solusi sementara ini kami rasa kurang efektif, hal ini disebabkan platform konferensi video memerlukan koneksi internet yang stabil dan memadai. Koneksi internet yang dimiliki oleh para *trainee* tidak selalu stabil. Dengan metode ini pula, karyawan HC tidak dapat memantau para *trainee* apakah benar-benar mengikuti *training* atau tidak dan juga memahami materi yang disampaikan ataupun tidak.

Ada juga proses *training* yang mengharuskan untuk bertemu tatap muka. Untuk *training* dengan jenis ini, kami mengadakannya dalam beberapa kloter (*batch*) dan membatasi *trainee* di setiap kloternya. Cara ini menyebabkan proses *training* jadi memakan waktu lebih lama dari biasanya, serta memerlukan lebih banyak biaya. Banyak proses *training* yang pada akhirnya mengalami penundaan.

**M: Apakah solusi *e-learning* menjadi jawaban atas masalah yang terjadi?, Kenapa demikian? (Pembahasan lanjutan terkait masalah yang terjadi pada proses pembelajaran yang sudah ada)**

**W:** Kami rasa *e-learning* menjadi jawaban untuk masalah pembelajaran yang terjadi. Proses pembelajaran jadi lebih mudah dilakukan, karena karyawan bisa melakukannya kapan saja dan di mana saja menyesuaikan dengan keinginan para *trainee*. Proses pembelajaran bisa menyesuaikan keadaan *trainee* dengan pengaksesan *e-learning* yang tidak terbatas waktu. Tentunya akan ada jadwal untuk para *trainee* untuk sudah menyelesaikannya pada waktu tertentu, tapi untuk prosesnya bisa mengikuti keseharian para *trainee*. Para *trainee* juga dapat melaksanakan proses *learning* di mana saja. Jadi, bisa disimpulkan bahwa dengan adanya *e-learning* proses *learning* akan lebih fleksibel dalam hal waktu dan tempat.

*E-learning* memudahkan dalam melakukan penilaian tes. Divisi HC tidak perlu mencetak lembar penilaian yang akan menghemat penggunaan sumber daya ATK dan memeriksanya satu per satu untuk mendapatkan nilai dari para *trainee*. Sumber Daya ATK yang dihemat ini bisa dialihkan untuk proses lain yang lebih memerlukannya. Data penilaian yang langsung tersentralisasi pada *e-learning* memudahkan proses penilaian tes juga. *E-learning* juga memudahkan dalam melakukan pemantauan untuk melihat sampai sejauh mana para *trainee* sudah melaksanakan proses pembelajaran. Hal ini juga berkaitan dengan pembuatan laporan hasil *training*. Harapan kami adalah dengan adanya solusi *e-learning* ini bisa menghilangkan tersebarnya data dan informasi terkait proses *training* karyawan karena biasanya data dan informasi tersebar di komputer lokal kami dan penyimpanan *cloud*.

**M: Apakah ekspektasi Anda terhadap *e-learning* yang akan dikembangkan penulis nanti?, Fitur apa saja yang ingin tersedia pada *e-learning* ini?**

**W:** Kami berharap dengan pengembangan sistem *e-learning* ini bisa mencakup beberapa fitur, seperti:

1. **Forum**, fitur ini digunakan untuk melakukan proses tanya dan jawab dengan mentor dan peserta lain. Para *trainee* bisa mencari jawaban atas materi yang diikuti maupun menanyakannya langsung. Para *trainee* bisa saling bertukar pikiran juga di fitur ini.
2. **Pre-test**, para mentor bisa melakukan penilaian awal terlebih dahulu untuk menilai sejauh mana pengetahuan *trainee* sebelum mengikuti materi.
3. **Post-test**, para *trainee* bisa menilai sejauh mana pengetahuannya terhadap materi yang telah diikuti dan bisa langsung melihat hasilnya.
4. **Learning Track**, para *trainee* bisa melihat sejauh mana materi yang telah diikuti dan materi yang akan dia ikuti selanjutnya, dan mentor dapat memantau progres para *trainee*.
5. **Learning Path**, para mentor dapat membuat sebuah perjalanan *learning* untuk para *trainee* yang terlibat.
6. **Quiz**, para *trainee* bisa mengisi kuis di setiap akhir modul pembelajaran untuk mengetahui kualitas pemahaman pada satu modul materi.

Dari berbagai fitur tersebut kami juga menginginkan fleksibilitas bentuk *template* dari materi dengan menambahkan tombol untuk menggunakan fitur tersebut atau tidak. Fitur tes pada *e-learning* ini juga dibatasi oleh waktu dan akan langsung dipaksa untuk mengakhiri apabila waktu sudah habis, fitur *timer* (penghitungan waktu mundur). Sementara itu, *e-learning* diharapkan bisa mendukung materi dalam bentuk:

1. Teks modul dan Gambar,

2. Video,
3. Presentasi PowerPoint.

**M: Di manakah platform untuk mengembangkan *e-learning* ini?**

**W:** Kami sudah memiliki aplikasi Portal HC. Aplikasi ini memuat berbagai fungsi tugas dan pokok dan informasi terkait divisi HC. Walaupun modul yang dibuat pada aplikasi ini masih belum mencakup keseluruhan tugas dan pokok divisi HC, Aplikasi ini akan terus dikembangkan ke depannya seiring dari permintaan departemen di divisi HC. Untuk itu, kami mengharapkan *e-learning* ini bisa langsung dikembangkan pada aplikasi portal ini.

**M: Adakah saran untuk membuat proses baru pembelajaran apabila *e-learning* ini berhasil dilakukan?, Bagaimanakah proses baru yang diinginkan?**

**W:** Pertama hal yang harus diperhatikan adalah proses validasi. HC menjadi sentral validasi utama apabila divisi HC maupun divisi lain ingin mengadakan *training*. Adapun saran proses baru apabila mentor ingin mengadakan *training*:

1. PIC memberitahu HC untuk mengadakan *training*
2. HC membuat modul *learning*
3. PIC mengirimkan materi ke HC untuk ditambahkan aplikasi *e-learning*
4. HC membuka akses modul yang sudah berisi materi ke PIC untuk disunting dan divalidasi
5. HC melakukan validasi materi dan *Recruitment Specialist* menayangkan (*publish*) materi ke aplikasi.



**KEMENTERIAN AGAMA**  
**UIN SYARIF HIDAYATULLAH JAKARTA**  
**FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Ir. H. Juanda No. 95 Ciputat 15412 Indonesia  
Telp. (62-21) 7493606, 7493547 Fax. (62-21) 7493315

Website : fst.uinjkt.ac.id  
Email : fst@uinjkt.ac.id

Nomor : B - 2266E/F9/ KM.01 /07/2023

Jakarta, 05 Juli 2023

Lampiran : -

Perihal : Pembimbing Skripsi

Kepada Yth.

1. Yuni Sugiarti M.Kom
2. Elvi Fetrina M.IT.

*Assalamualaikum, Wr Wb*

Dengan ini diharapkan kesediaan Saudara untuk menjadi pembimbing I/II/(Materi/Teknis)\* penulisan skripsi mahasiswa:

Nama

: RIZKI RAMADHAN

NIM

: 11160930000053

Program Studi

: Sistem Informasi

Judul Skripsi

Rancang Bangun E-Learning dan Repositori Digital  
: dengan Platform Github (Studi Kasus: Centratama  
Telekomunikasi Indonesia)

Judul tersebut telah disetujui oleh Program Studi bersangkutan pada tanggal dengan outline, abstraksi dan daftar pustaka terlampir. Bimbingan skripsi ini diharapkan selesai dalam waktu 6 (enam) bulan setelah ditandatanganinya surat penunjukan pembimbing skripsi

Apabila terjadi perubahan terkait dengan skripsi tersebut selama proses pembimbingan, harap segera melaporkan kepada Program Studi bersangkutan.

Demikian atas kesediaan Saudara, kami ucapkan terima kasih.

*Wassalamu'alaikum Wr.Wb*



Jakarta, 05 Juli 2023

a.n Dekan  
Wakil Dekan Bid. Akademik



Dr. La Ode Sumarlin, M.Si.  
NIP. 197509182008011007/1

SURAT KETERANGAN  
PROGRAM PELATIHAN KERJA (MAGANG)  
Nomor: 615/MAC-HC/SKE/XII/2020

Yang bertandatangan di bawah ini:

Nama : Michael Loebis  
Jabatan : VP Human Capital

Dengan ini menyatakan bahwa:

Nama : Rizki Ramadhan  
Perguruan Tinggi : Universitas Islam Negeri Syarif Hidayatullah  
Alamat : Jl. Raya TMII No. 18, RT/RT 001/002, Kel. Ceger, Kec. Cipayung, Jakarta Timur

Mahasiswa yang bersangkutan telah melaksanakan program pelatihan kerja (magang) di PT. Mac Sarana Djaya sejak 19 Maret 2020 hingga 18 Desember 2020 di Departemen Human Capital.

Demikian surat keterangan magang ini kami buat untuk dipergunakan sebagaimana mestinya.

Jakarta, 18 Desember 2020



Michael Loebis  
VP Human Capital