**Pengembangan Sistem E-learning Industri Untuk Meningkatkan Pengalaman Belajar Yang Lebih Baik**

**PROPOSAL TUGAS AKHIR**

oleh:

**Yandra Muslim**

**3312101019**

Disusun untuk pengajuan proposal Tugas Akhir Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BATAM**

**BATAM**

**2023**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc152592176)

[DAFTAR GAMBAR 3](#_Toc152592177)

[DAFTAR TABEL 4](#_Toc152592178)

[BAB I PENDAHULUAN 5](#_Toc152592179)

[1.1. Latar Belakang 5](#_Toc152592180)

[1.2. Perumusan Masalah 6](#_Toc152592181)

[1.3. Batasan Masalah 6](#_Toc152592182)

[1.4. Tujuan 7](#_Toc152592183)

[1.5. Manfaat 7](#_Toc152592184)

[BAB II LANDASAN TEORI 8](#_Toc152592185)

[2.1. Penelitian Yang Relevan 8](#_Toc152592186)

[2.2 Penjelasan Teoritis Masing - Masing Variable 10](#_Toc152592187)

[BAB III METODE PENELITIAN 14](#_Toc152592188)

[3.1. Metode Pengembangan Perangkat Lunak 14](#_Toc152592192)

[3.2. Gambaran Umum Sistem 40](#_Toc152592193)

[DAFTAR PUSTAKA 41](#_Toc152592194)

# DAFTAR GAMBAR

[Gambar 1 : Metode penelitian waterfall 10](#_Toc152591768)

[Gambar 2 : Tabel relasi 16](#_Toc152591769)

[Gambar 3 : Use case diagram 17](#_Toc152591770)

[Gambar 4 : Halaman dashboard 32](#_Toc152591771)

[Gambar 5 : Tampilan data tabel 32](#_Toc152591772)

[Gambar 6 : Halaman tambah data tabel 33](#_Toc152591773)

[Gambar 7 : Halaman detail data 33](#_Toc152591774)

[Gambar 8 : Halaman login 35](#_Toc152591775)

[Gambar 9 : Halaman welcome 35](#_Toc152591776)

[Gambar 10 : Halaman daftar materi 36](#_Toc152591777)

[Gambar 11 : Halaman pelajaran 36](#_Toc152591778)

[Gambar 12 : Halaman teori 37](#_Toc152591779)

[Gambar 13 : Halaman soal 37](#_Toc152591780)

[Gambar 14 : Halaman review kuis 38](#_Toc152591781)

[Gambar 15 : Halaman feedback form 38](#_Toc152591782)

[Gambar 16 : Gambaran Umum Sistem 40](file:///Y:\14.%20Transfer%20File\1.%20Industrial%20Attachment\2.%20Universitas\07.%20Yandra%20Muslim\3312101019_Yandra%20Muslim_Proposal.docx#_Toc152591783)

# DAFTAR TABEL

[Tabel 1: Penelitian Yang Relevan 9](#_Toc152591786)

[Tabel 2 : Kebutuhan Fungsional 14](#_Toc152591787)

[Tabel 3 : Kebutuhan Non Fungsional 15](#_Toc152591788)

[Tabel 4 : Skenario Use case login 18](#_Toc152591789)

[Tabel 5 : Skenario Use case menambah materi 18](#_Toc152591790)

[Tabel 6: Skenario use case melihat materi 19](#_Toc152591791)

[Tabel 7 : Skenario use case mengubah materi 19](#_Toc152591792)

[Tabel 8 : Skenario use case menghapus materi 20](#_Toc152591793)

[Tabel 9 : Skenario use case menambah pelajaran 20](#_Toc152591794)

[Tabel 10 : Skenario use case melihat pelajaran 21](#_Toc152591795)

[Tabel 11 : Skenario use case mengubah pelajaran 21](#_Toc152591796)

[Tabel 12 : Skenario use case menghapus pelajaran 22](#_Toc152591797)

[Tabel 13 : Skenario use case menambah teori 22](#_Toc152591798)

[Tabel 14 : Skenario use case melihat teori 23](#_Toc152591799)

[Tabel 15 : Skenario use case mengubah teori 23](#_Toc152591800)

[Tabel 16 : Skenario use case menghapus teori 24](#_Toc152591801)

[Tabel 17 : Skenario use case menambah kuis 24](#_Toc152591802)

[Tabel 18 : Skenario use case melihat kuis 24](#_Toc152591803)

[Tabel 19 : Skenario use case mengubah kuis 25](#_Toc152591804)

[Tabel 20 : Skenario use case menghapus kuis 25](#_Toc152591805)

[Tabel 21 : Skenario use case mengerjakan kuis 26](#_Toc152591806)

[Tabel 22 : Skenario use case melihat hasil kuis 26](#_Toc152591807)

[Tabel 23 : Skenario use case menambahkan feedback form 27](#_Toc152591808)

[Tabel 24 : Skenario use case feedback form 27](#_Toc152591809)

[Tabel 25 : Skenario use case menambah data karyawan 28](#_Toc152591810)

[Tabel 26 : Skenario use case melihat data karyawan 28](#_Toc152591811)

[Tabel 27 : Skenario use case menambah data trainer 29](#_Toc152591812)

[Tabel 28 : Skenario use case melihat data trainer 29](#_Toc152591813)

[Tabel 29 : Skenario use case menghapus data trainer 30](#_Toc152591814)

[Tabel 30 : Skenario use case melihat riwayat pengerjaan user 30](#_Toc152591815)

[Tabel 31 : Skenario use case menghapus riwayat pengerjaan user 31](#_Toc152591816)

[Tabel 32 : Skenario use case logout 31](#_Toc152591817)

# BAB I PENDAHULUAN

## Latar Belakang

*Trend* pengembangan karyawan saat ini sedang mengalami pergeseran dari pelatihan tatap muka ke *e-learning*. Yang dimaksud dengan *e-learning* adalah metode pembelajaran *virtual* melalui internet, baik menggunakan instruktur manusia maupun teknologi *digital*. Para praktisi HR melihat adanya pengaruh program pengembangan dan pelatihan terhadap kualitas hubungan karyawan dan perusahaan. Pelatihan dan pengembangan karyawan tidak hanya mendorong keunggulan kompetitif organisasi, tetapi juga menaikkan loyalitas karyawan terhadap perusahaan [[1].](#satty)

*Platform* *e-learning* diperusahaan tempat saya magang telah membuktikan manfaatnya dalam memfasilitasi pembelajaran dalam skala besar, khususnya dalam konteks pelatihan karyawan baru. Namun, seperti halnya banyak *platform* *e-learning* lainnya, terdapat beberapa tantangan yang perlu diatasi agar pengalaman pembelajaran menjadi lebih efektif.

Salah satu masalah utama yang dihadapi oleh perusahaan adalah kurangnya daya tarik visual dan kurangnya interaktivitas dalam situs web *e-learning* mereka. Desain yang kurang menarik dan antarmuka yang kurang interaktif dapat menghambat kemampuan pengguna dalam memahami materi pelajaran dengan baik dan mengikuti kursus dengan efisien. Untuk menjawab tantangan ini, diperlukan perubahan signifikan dalam pengembangan *platform* *e-learning*.

Selain menjadi sarana utama untuk memberikan pelatihan kepada karyawan baru, penggunaan *platform* *e-learning* juga merupakan alat yang potensial untuk memantau dan mengukur hasil pembelajaran mereka. Mereka menyadari bahwa pengukuran hasil pembelajaran adalah langkah penting dalam memastikan bahwa karyawan yang telah mengikuti pelatihan siap diterjunkan ke lapangan dengan pengetahuan dan keterampilan yang diperlukan.

Pengukuran hasil pembelajaran ini dapat mencakup aspek-aspek seperti peningkatan pengetahuan teknis, kerjasama team, dll. Oleh karena itu, selain mengatasi tantangan daya tarik visual dan interaktivitas, perubahan dalam pengembangan *platform* *e-learning* juga diarahkan untuk menciptakan mekanisme pemantauan dan pengukuran yang lebih efektif terhadap hasil pembelajaran karyawan.

Terkait tantangan yang dihadapi oleh perusahaan ditempat saya magang perlu diatasi dengan cara mengembangkan aplikasi *e-learning* industri berbasis *website* dengan tampilan yang interaktif dan pengukuran hasil pembelajaran untuk kebutuhan perusahaan. Dibuatnya aplikasi ini diharapkan dapat meningkatkan pengalaman belajar untuk karyawan dan membantu tim *HRD* perusahaan dalam memantau hasil pembelajaran karyawan.

Dengan kebutuhan di tempat perusahaan saya magang, maka judul tugas akhir ini yaitu “Pengembangan Sistem *E-learning* Industri Untuk Meningkatkan Pengalaman Belajar Yang Lebih Baik” dengan tujuan utama yaitu meningkatkan pengalaman belajar lebih baik dan pengukuran hasil pembelajaran karyawan sehingga karyawan yang telah mengikuti pelatihan siap diterjunkan ke lapangan dengan pengetahuan dan keterampilan yang diperlukan dari perusahaan ditempat saya magang.

## Perumusan Masalah

Berdasarkan uraian latar belakang, maka perumusan masalah pada proposal tugas akhir ini yaitu bagaimana cara mengembangkan *e-learning* industri berbasis *website* untuk meningkatkan pengalaman pembelajaran yang lebih baik.

## Batasan Masalah

Berdasarkan uraian masalah di atas, sistem yang dibuat memiliki batasan masalah yaitu :

1. Pengembangan ini hanya mendukung *website*.
2. *Platform* *e-learning* hanya dapat digunakan oleh perusahaan industri terkait.

## Tujuan

Adapun tujuan peneliti membuat penelitian ini adalah :

1. Membuat *platform* *e-learning* Industri versi baru dan berguna untuk pengalaman pembelajaran karyawan baru.
2. Mengembangkan sistem pemantauan dan pengukuran hasil pembelajaran karyawan baru yang efektif untuk memastikan bahwa karyawan yang telah mengikuti pelatihan siap diterjunkan ke lapangan dengan pengetahuan dan keterampilan yang diperlukan.

## Manfaat

Dari tujuan di atas, maka dapat diperoleh manfaat dalam pengaplikasian sistem tersebut

adalah:

1. Meningkatkan *platform* *e-learning* di perusahaan tempat saya magang, yang akan menghasilkan pengalaman pembelajaran yang lebih baik bagi karyawan baru dan membantu perusahaan dalam mencapai tujuan pelatihan.
2. Membantu perusahaan dalam memantau dan mengukur hasil pembelajaran karyawan baru dengan lebih efektif.
3. Menyediakan panduan dan wawasan yang dapat digunakan oleh semua karyawan perusahaan.

# BAB II LANDASAN TEORI

* 1. Penelitian Yang Relevan

1. Kukuh, S, et al. (2020) dalam penelitiannya mengenai Pengembangan *E-learning* Berbasis *Website* menggunakan Metode *Waterfall*. Dengan *e-learning* ini dapat menghemat biaya pembelajaran/pendidikan.
2. Mario, S., & Johanes, S.A. (2020) dalam penelitiannya mengenai Pengembangan Aplikasi *E-learning* Berbasis Web Menggunakan Model *Waterfall* Pada SMK STRADA 2 JAKARTA. *E-learning* dipilih karena dinilai mampu sebagai bentuk peningkatan per*form*a dan informasi yang lebih dalam proses pembelajaran.
3. Clara, F, et al. (2020) dalam penelitiannya mengenai Perancangan *E-learning* Sebagai Media Pembelajaran. Dengan adanya *e-learning* dirasa dapat mempermudah dalam penyampaian informasi. Karena dengan *e-learning* penyampaian materi yang tadinya dengan sistem konvesional berubah menjadi sistem yang didukung dengan media internet.
4. Fahri, R., & Linda, S.D. (2022) dalam penelitiannya mengenai Sistem Informasi Pembelajaran *Online* Berbasis WEB di SMKN 1 Depok. Kegiatan belajar mengajar dapat lebih mudah karena dapat diakses kapan saja serta memudahkan bagi siswa dan guru melihat pengumuman serta proses absensi dan rekap absensi tidak perlu lagi dilakukan secara manual. Dengan sebuah *e-learning*, maka dapat diambil kesimpulan bahwa melalui sistem informasi pembelajaran *online* proses pembelajaran dapat dilakukan dengan lebih efektif dan efisien.
5. Ahmad, L., & Shofiyah. (2023) dalam penelitiannya mengenai Sistem Informasi *E-learning* Berbasis Web Menggunakan PHP dan *MYSQL. Denga*n adanya *e-learning* ini lebih memudahkan dalam proses pembelajaran dan juga suntuk meningkatkan mutu pendidikan dalam menghadapi revolusi industri 5.0.

Tabel 1: Penelitian Yang Relevan

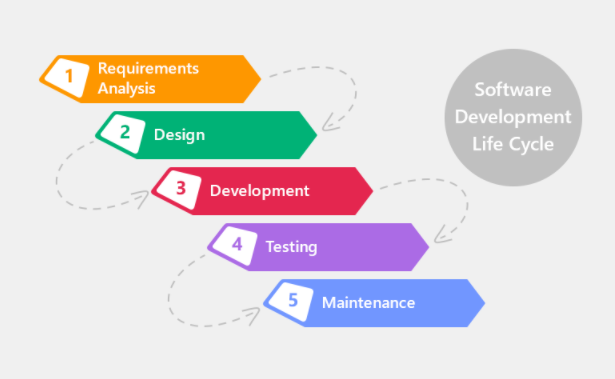
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Penulis | Tahun | Judul | Tujuan | Metode | Hasil |
| Kukuh, S., Mustofa, A.H., & Didik, A. | 2020 | Pengembangan *E-learning* Berbasis *Website* | Mengembangkan sistem pembelajaran *e-learning* berbasis *website* dan mengetahui kelayakan produk yang dihasilkan | *Waterfall* | (1) tingkat kelayakan produk *website* *e-learning* diperoleh nilai skor ratarata sebesar 74,6 dengan kategori “sangat layak”. (2) hasil uji coba penggunaan *website* *e-learning* oleh dosen dan  staff diperoleh nilai skor rata-rata sebesar 90,3 dengan kategori “sangat layak”. (3) hasil uji coba penggunaan  *website* *e-learning* oleh mahasiswa diperoleh nilai skor rata-rata sebesar 85 dengan kategori “sangat layak” |
| Mario, S., & Johanes, F.A. | 2020 | Pengembangan Aplikasi *E-learning* Berbasis Web Menggunakan Model *Waterfall* Pada SMK Strada 2 Jakarta. | Pengembangan *e-learning* sebagai model pembelajaran yang baru, sehingga peningkatan performa dari guru maupun siswa dapat terjadi, serta informasi yang disampaikan menjadi lebih jelas dan tidak terbatas | *Waterfall* | Sebuah aplikasi elearning  berbasis *website* yang dapat memfasilitasi kebutuhan proses pembelajaran antara guru dan siswa secara *online*. |
| Clara, F., Adjat, S., & Retno, S. | 2020 | Perancangan *E-learning* Sebagai Media Pembelajaran | Mempermudah proses belajar mengajar dan meningkatkan komunikasi antara siswa dan guru | *Waterfall* | Sebuah aplikasi *e-learning* yang dapat membantu proses belajar mengajar jarak jauh serta mampu menjangkau penyebaran informasi secara luas, cepat dan tepat dan memudahkan para pengajar dalam kegiatan belajar mengajar tanpa harus membuang waktu yang lama. |
| Fahri, R. **&** Linda, S.D. | 2022 | Sistem Informasi Pembelajaran *Online* Berbasis WEB di SMKN 1 Depok | Kegiatan belajar mengajar dapat lebih mudah karena dapat diakses kapan saja serta memudahkan bagi siswa dan guru melihat pengumuman serta proses absensi dan rekap absensi tidak perlu lagi dilakukan secara manual. | *Waterfall* | sebuah sistem informasi pembelajaran *online* yang berbasis web, maka dapat diambil kesimpulan bahwa melaluisistem informasi pembelajaran *online* prosespembelajaran dapat dilakukan dengan lebih efektif dan efisien. |
| Ahmad, L., &Shofiyah | 2023 | Sistem Informasi *E-learning* Berbasis Web Menggunakan PHP dan *MYSQL* | Memudahkan dalam proses pembelajaran dan juga untuk meningkatkan mutu pendidikan dalam menghadapi revolusi industri 5.0 | *Waterfall* | Sebuah aplikasi *e-learning* yang membuat siswa bisa belajar dirumah tanpa harus dating ke sekolah untuk mendapatkan materi atau latihan latihan soal |

1. Penjelasan Teoritis Masing - Masing Variable
2. *E-learning*

*E-learning* adalah pembelajaran secara elektronik atau proses belajar mengajar yang dilakukan secara *online*. Jika dilihat dari susunan katanya, *e-learning* merupakan gabungan dua kata yakni electronic dan learning. Maka dapat disimpulkan bahwa metode pembelajaran ini memanfaatkan teknologi informasi untuk mendistribusikan materi ke penggunanya [[2].](#nomor1)

1. *Waterfall*

Gambar 1 : Metode penelitian waterfall



Metode *waterfall* adalah salah satu jenis model pengembangan aplikasi dan termasuk ke dalam *class*ic *life cycle* (siklus hidup klasik), yang mana menekankan pada fase yang berurutan dan sistematis. Untuk model pengembangannya, dapat dianalogikan seperti air terjun, dimana setiap tahap dikerjakan secara berurutan mulai dari atas hingga ke bawah [[3]](#nomor2).

1. *HTML*

Hypertext Markup Language atau *HTML* adalah bahasa markup standar yang digunakan untuk membuat halaman *website* dan aplikasi web. Sejarah *HTML* diciptakan oleh Tim Berners-Lee, seorang ahli fisika di lembaga penelitian CERN yang berlokasi di Swiss.Versi pertama *HTML* dirilis oleh Tim Berners Lee pada tahun 1991, yang memiliki 18 *tag*. Sejak saat itu, setiap kali ada versi barunya, pasti akan selalu ada *tag* dan attribute (*tag* modifier) yang juga baru. Ketika bekerja dengan bahasa markup ini, Anda menggunakan struktur kode sederhana (*tag* dan attribute) untuk mark up halaman *website*. Misalnya, Anda membuat sebuah paragraf dengan menempatkan *enclosed text* di antara *tag* pembuka <p> dan *tag* penutup </p> [[4]](#nomor3).

1. *CSS*

*CSS* adalah bahasa Cascading Style Sheet dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti *HTML*. *CSS* berfungsi untuk memisahkan konten dari tampilan visualnya di situs. *CSS* dibuat dan dikembangkan oleh *W3C* (*World Wide Web Consortium*) pada tahun 1996 untuk alasan yang sederhana. Dulu *HTML* tidak dilengkapi dengan *tags* yang berfungsi untuk mem*format* halaman. Anda hanya perlu menulis markup untuk situs [[5]](#nomor4).

1. *TailwindCSS*

Jika membaca pada halaman dokumentasinya dia menyebut dirinya “A *utility-first* *CSS* *framework* for *rapidly* *building* *custom* *designs*“. Apa maksudnya *utility-first*? Jadi *Tailwind* *CSS* memiliki pendekatan yang berbeda dengan *framework* *CSS* yang sudah biasa kita lihat seperti *framework*, *Foundation* atau *Bulma*. Jika *framework* lainnya menyediakan komponen yang utuh dan tinggal kita gunakan, beda dengan *Tailwind* *CSS* yang memberikan *class* *class* kecil yang bisa kita gunakan dengan sangat fleksibel untuk membangun antarmuka yang sesuai keinginan tanpa harus *override* *styles* seperti di *framework* *CSS* lainnya [[6]](#nomor5).

1. *Javascript*

*JavaScript* adalah bahasa skrip yang digunakan untuk membuat konten halaman web dinamis, berfungsi untuk membuat elemen yang mampu meningkatkan interaksi pengunjung seperti menu *drop-down*, animasi, dan warna background dinamis. Sejarah *JavaScript* dimulai pada tahun 1995, diciptakan oleh Brendan Eich di Netscape Communications. Awalnya, *JavaScript* hanya digunakan untuk keperluan internal dengan *web browser* perusahaan tersebut, Netscape Navigator. Dulunya, *JavaScript* bernama *LiveScript*, yang kemudian diubah menjadi *JavaScript* agar bisa menjadi “teman” bagi bahasa pemrograman Java milik mitra mereka, *Sun Microsystems*. Kami akan menjelaskan perbedaan *Java* dan *JavaScript* nanti. *JavaScript* terus berkembang bersama *browser web* baru seperti *Mozilla Firefox* dan *Google Chrome* sejak saat itu. Bahkan, saat ini sedang dikembangkan mesin *JavaScript* modern pertama, yaitu *V8*, yang bertugas untuk mengompilasi *bytecode* menjadi kode mesin asli [[7]](#nomor6).

1. *ReactJs*

*ReactJs* adalah sebuah pustaka/*library* *javascript* yang bersifat *open source* untuk membangun *User* Interface yang dibuat oleh Facebook. *ReactJs* hanya mengurusi semua hal yang berkaitan dengan tampilan dan logika di sekitarnya. *ReactJs* dapat mendesain tampilan sederhana untuk setiap level dalam aplikasi, sehingga dapat digunakan untuk membuat dan mengembangkan pembuatan aplikasi berbasis web. Popularitasnya dapat diukur oleh aplikasi – aplikasi yang menggunakannya seperti *Facebook*, *WhatsApp, Netflix, Instagram, Airbnb, American Express, Dropbox, Ebay*, dan ratusan penyedia jasa pembuatan aplikasi berbasis *web* memanfaatkan kemampuan *ReactJs* [[8]](#nomor7).

1. *MySQL*

*MySQL* adalah sebuah *database* management system (manajemen basis *data*) menggunakan perintah dasar *SQL (Struc*tured Query Language) yang cukup terkenal. *Database management system* (*DBMS*) *MySQL multi* pengguna dan multi alur ini sudah dipakai lebih dari 6 juta pengguna di seluruh dunia [[9]](#nomor8).

1. *Express.js*

*Express.js* adalah *framework* web app untuk *Node.js* yang ditulis dengan bahasa pemrograman *JavaScript*. *Framework* *open source* ini dibuat oleh TJ Holowaychuk pada tahun 2010 lalu. *Express.js* adalah *framework* *backend*. Artinya, ia bertanggung jawab untuk mengatur fungsionalitas *website*, seperti pengelolaan *routing* dan *session*, permintaan *HTTP*, penanganan *error*, serta pertukaran *data* di *server* [[10]](#nomor9). *Framework* yang satu ini punya arsitektur *MVC (Model View Controller).* Dengan begitu, setiap *data* diolah pada *Model*, dihubungkan melalui *Controller*, lalu ditampilkan menjadi informasi melalui *View*.

1. *UML*

*UML* adalah bahasa pemodelan visual yang digunakan untuk merancang dan memodelkan sistem perangkat lunak. Dilansir dari *Visual Paradigm*, *UML* mencakup notasi grafis yang digunakan untuk merepresentasikan berbagai aspek dari sistem, termasuk struktur sistem, perilaku sistem, interaksi antara objek, dan lingkungan di mana sistem beroperasi. Menurut Grady Booch, salah satu ahli yang terlibat dalam pengembangan *Unified Modeling Language*, *UML* adalah “bahasa pemodelan visual yang dapat digunakan untuk merepresentasikan sistem perangkat lunak yang berbeda, mulai dari sistem yang sederhana hingga sistem yang kompleks.” *Unified Modeling Language* terdiri dari berbagai *diagram* yang masing-masing merepresentasikan aspek tertentu dari sistem. Beberapa *diagram* *Unified Modeling Language* yang paling umum digunakan termasuk *class* *diagram*, *activity diagram*, dan lain sebagainya [[11]](#nomor10).

1. *ERD*

*ERD* (*Entity Relationship Diagram*) adalah model atau rancangan untuk membuat *database*, supaya lebih mudah dalam menggambarkan *data* yang memiliki hubungan atau relasi dalam bentuk sebuah desain. Dengan adanya *ER diagram*, maka sistem *database* yang terbentuk dapat digambarkan dengan lebih terstruktur dan terlihat rapi [[12]](#nomor11).

1. *IIS*

*IIS* merupakan singkatan dari *Internet Information Server*. *IIS* adalah sebuah *HTTP server* yang digunakan pada *server* dengan *OS windows*. *IIS* pertama kali dirilis sebagai *addon* dari layanan berbasis internet yang terdapat pada *Windows NT 3.51*.

Beberapa waktu kemudian, *IIS* melakukan update ke versi 2.0 dengan tujuan agar sistem *IIS* dapat *compatibel* untuk sistem operasi *Windows* NT 4.0 yang baru dirilis saat itu. Kemudian *IIS* kembali melakukan *update* *IIS* ke versi 3.0n *Update* ini dilakukan agar *IIS* mendukung lingkungan skrip dinamis Active *Server* *Pages* (*ASP*) [[13]](#nomor12).

# BAB III METODE PENELITIAN

3. 1. Metode Pengembangan Perangkat Lunak

Pengembangan perangkat lunak yang dipakai yaitu sesuai dengan gambar 1 tahapannya adalah:

1. *Requirements,* tahapan ini dilakukan dengan cara melakukan observasi dan juga terhadap proses pembelajaran untuk karyawan baru dan juga *e-learning* versi lama yang ada diindustri terkait. Tujuannya adalah untuk mendefinisikan kebutuhan terkait *platform* yang akan dibuat dan menggambarkannya menjadi kebutuhan fungsional dan non-fungsional sebagai berikut:

Tabel 2 : Kebutuhan Fungsional

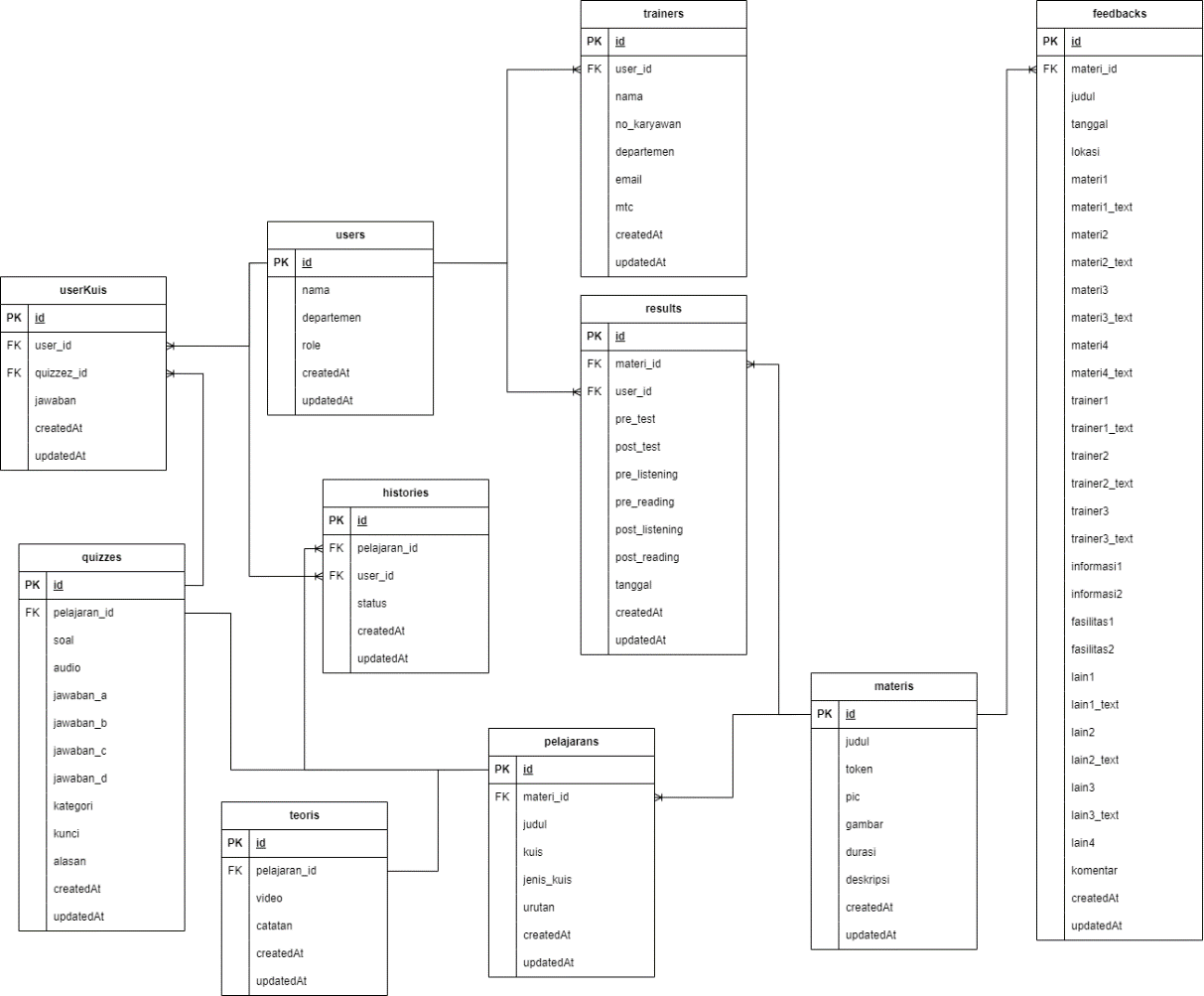
|  |  |
| --- | --- |
| **NO** | **Kebutuhan Fungsional** |
| F01 | *User* dapat melakukan *login* |
| F02 | *User* dapat melihat daftar materi |
| F03 | *User* dapat memasukkan *token* |
| F04 | *User* dapat melihat daftar pelajaran (teori & kuis) |
| F05 | *User* dapat melihat daftar *status* pelajaran, seperti pelajaran yang harus dikerjakan, pelajaran yang telah dikerjakan, dan pelajaran yang belum dikerjakan |
| F06 | *User* dapat melihat *video* & rangkuman yang ada didalam pelajaran (teori) |
| F07 | *User* dapat melihat soal yang ada didalam pelajaran |
| F08 | *User* dapat mengirim jawaban dari soal yang telah dikerjakan |
| F09 | *User* dapat melihat hasil dari soal yang yang dikerjakan pada isi pelajaran (kuis) |
| F10 | *User* dapat melihat kunci jawaban benar jika jawaban yang dimasukkan salah (kuis) |
| F11 | *User* dapat memainkan *audio* dari soal (kuis) |
| F12 | *User* dapat mengisi *feedback form* pada akhir pelajaran |
| F13 | *User* dapat melakukan logout |
| F14 | *Admin* dapat melakukan *login* |
| F15 | *Admin* dapat menambah, melihat, mengupdate, menghapus *data* materi |
| F16 | *Admin* dapat mengubah *status* *website* menjadi dalam perbaikan |
| F17 | *Admin* dapat menambahkan *token* pada setiap materi |
| F18 | *Admin* dapat menambah, melihat, mengupdate, menghapus *data* pelajaran (teori & kuis) |
| F19 | *Admin* dapat menambahkan, melihat, mengubah, menghapus *data* *video* dan rangkuman yang ada di dalam pelajaran (teori) |
| F20 | *Admin* dapat menambahkan, melihat, mengubah, menghapus *data* soal, jawaban, beserta kuncinya yang ada di dalam pelajaran (kuis) |
| F21 | *Admin* dapat melihat hasil jawaban yang dikerjakan oleh *user* (kuis) |
| F22 | *Admin* dapat melihat *status* pengerjaan/riwayat materi yang sedang dikerjakan oleh *user* |
| F23 | *Admin* dapat menambahkan, melihat, mengubah, menghapus *data* trainer |
| F24 | *Admin* dapat menambahkan *data* karyawan |
| F25 | *Admin* dapat melihat hasil *feedback form* dari *user* |
| F26 | *Admin* dapat melihat j*uml*ah *user* yang mengerjakan materi perbulannya |
| F27 | *Admin* dapat mengexport *data* result dan *data* feedback menjadi excel |
| F27 | *Admin* dapat melakukan logout |

Tabel 3 : Kebutuhan Non Fungsional

|  |  |
| --- | --- |
| **NO** | **Kebutuhan Non Fungsional** |
| NF01 | Apikasi dibuat dengan tampilan yang interaktif dan mudah dipahami *user* |
| NF02 | Aplikasi menggunakan bahasa pemrograman *javascript* |
| NF03 | Aplikasi hanya dapat dijalankan di*server* lokal |
| NF04 | Karyawan hanya dapat mengerjakan materi 1 kali |

1. *Design,* tahapan ini dilakukan dengan cara pembuatan relasi tabel, *use case* *diagram*, ERD (Entity Relationship *Diagram*), dan rancangan antarmuka atau *user* interface.
   1. Tabel Relasi

Gambar 2 : Tabel relasi



Gambar 2 diatas merupakan tabel relasi yang memiliki 10 tabel, yaitu tabel quizzes yang menyimpan informasi kuis, userKuis menyimpan jawaban kuis dari user, *users* menyimpan data *user*, histories menyimpan riwayat pengerjaan dari user, teoris menyimpan video dan juga catatan untuk user, pelajarans menyimpan untuk pelajaran, results yang menyimpan hasil jawaban atau nilai dari user, trainers menyimpan data pelatih dari perusahaan terkait, materis untuk menyimpan data materi, dan feedbacks yang menyimpan kritik dan saran dari user. Tabel pelajaran memiliki relasi one to one terhadap teoris dan juga quizzes. Sedangkan tabel yang lain berleasi one to many dan juga many to one.

* 1. *Use Case* *Diagram*

Gambar 3 : Use case diagram



Gambar 3 ditas merupakan *use case* yang memiliki 2 aktor, yaitu aktor user dan aktor admin. Aktor user memiliki beberapa akses seperti, bisa melakukan login, melihat daftar materi, melihat daftar pelajaran, melihat teori, melihat kuis, mengerjakan kuis, mengisi feedback form, melihat data trainer, dan logout. Lalu aktor admin memiliki akses yang lebih banyak, yaitu bisa melakukan login, melihat, menambah, mengubah, menghapus materi. Melihat, menambah, mengubah, menghapus pelajaran. Melihat, menambah, mengubah, menghapus teori. Melihat, menambah, mengubah, menghapus kuis. Melihat hasil kuis, menambahkan feedback form, menambah data user, melihat data user, menambah trainer, melihat trainer, menghapus trainer, melihat riwayat pengerjaan user, menghapus riwayat pengerjaan user, dan logout.

* 1. Skenario *Use Case*
     1. Skenario *Use case* *login*

Tabel 4 : Skenario Use case login

|  |  |
| --- | --- |
| **Nama *Use Case*** | *Login* |
| **Deskripsi** | Aktor mendapatkan hak akses |
| **Aktor** | *User* & *Admin* |
| **Kondisi awal** | Aktor mengisi *login* *form* |
| **Kondisi akhir** | Aktor berhasil *login* |
| **Skenario** | 1. Aktor mengisi *login* *form* 2. Sistem akan menampilkan notifikasi berhasil *login* 3. Sistem melakukan validasi *redirect* aktor |
| **Skenario alternatif** | 2.a[nomor karyawan salah] kondisi ini akan menampilkan notifikasi nomor karyawan tidak ditemukan. Lalu Kembali lagi ke nomor 1.  3.a[aktor *admin*] kondisi ini akan mengarahkan aktor *admin* ke halaman dashboard.  3.b[aktor karyawan] kondisi ini akan mengarahkan aktor ke halaman *welcome*. |

* + 1. Skenario *Use case* menambah materi

Tabel 5 : Skenario Use case menambah materi

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menambah materi |
| **Deskripsi** | Aktor dapat melakukan tambah materi |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Materi berhasil ditambahkan |
| **Skenario** | 1. Aktor mengisi *form* pada halaman tambah materi 2. Sistem melakukan validasi 3. Sistem berhasil menambah materi |
| **Skenario alternatif** | 2.a[materi sudah ada] Kondisi ini akan mengharuskan aktor untuk mengubah judul materi, karena sudah dibuat sebelumnya. Lalu Kembali ke nomor 1. |

* + 1. Skenario *Use case* melihat materi

Tabel 6: Skenario use case melihat materi

|  |  |
| --- | --- |
| **Nama *Use Case*** | Melihat materi |
| **Deskripsi** | Aktor dapat melihat materi |
| **Aktor** | Admin & User |
| **Kondisi awal** | Aktor berhasil login |
| **Kondisi akhir** | Materi berhasil dilihat |
| **Skenario** | 1. Aktor masuk ke halaman materi 2. Sistem menampilkan materi 3. Aktor berhasil melihat daftar materi |
| **Skenario alternatif** | 3.a[gagal menampilkan materi] Kondisi ini akan menampilkan notifikasi gagal pada aktor |

* + 1. Skenario *Use case* mengubah materi

Tabel 7 : Skenario use case mengubah materi

|  |  |
| --- | --- |
| **Nama *Use Case*** | Mengubah materi |
| **Deskripsi** | Aktor dapat mengubah materi |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Materi berhasil diubah |
| **Skenario** | 1. Aktor masuk halaman detail materi yang akan diubah 2. Aktor menekan tombol update pada halaman detail materi 3. Aktor mengubah *data* 4. Sistem melakukan validasi 5. Sistem berhasil mengubah *data* materi |
| **Skenario alternatif** | 4.a[materi sudah ada] Kondisi ini akan mengharuskan aktor untuk mengubah judul materi, karena sudah ada sebelumnya. Lalu Kembali ke nomor 3. |

* + 1. Skenario *Use case* menghapus materi

Tabel 8 : Skenario use case menghapus materi

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menghapus materi |
| **Deskripsi** | Aktor dapat menghapus materi |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Materi berhasil dihapus |
| **Skenario** | 1. Aktor masuk halaman detail materi yang akan diubah 2. Aktor menekan tombol hapus pada halaman detail materi 3. Aktor menekan tombol hapus pada *pop up* 4. Sistem berhasil menghapus *data* |
| **Skenario alternatif** | 3.a[aktor menekan tombol batal] Kondisi ini membuat sistem membatalkan penghapusan pada materi yang dipilih |

* + 1. Skenario *Use case* menambah pelajaran

Tabel 9 : Skenario use case menambah pelajaran

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menambah pelajaran |
| **Deskripsi** | Aktor dapat menambah pelajaran |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Pelajaran berhasil ditambahkan |
| **Skenario** | 1. Aktor mengisi *form* pada halaman pelajaran 2. Sistem melakukan validasi 3. Sistem berhasil menambah pelajaran |
| **Skenario alternatif** | 2.a[pelajaran sudah ada] Kondisi ini akan mengharuskan aktor untuk mengubah nama pelajaran, karena sudah dibuat sebelumnya. Lalu Kembali ke nomor 1. |

* + 1. Skenario *Use case* melihat pelajaran

Tabel 10 : Skenario use case melihat pelajaran

|  |  |
| --- | --- |
| **Nama *Use Case*** | Melihat pelajaran |
| **Deskripsi** | Aktor melihat pelajaran |
| **Aktor** | Admin dan User |
| **Kondisi awal** | Aktor berhasil login |
| **Kondisi akhir** | Aktor berhasil melihat pelajaran |
| **Skenario** | 1. Aktor masuk ke halaman pelajaran 2. Sistem menampilkan pelajaran 3. Aktor berhasil melihat daftar pelajaran |
| **Skenario alternatif** | 3.a[gagal menampilkan pelajaran] Kondisi ini akan menampilkan notifikasi gagal pada aktor |

* + 1. Skenario *Use case* mengubah pelajaran

Tabel 11 : Skenario use case mengubah pelajaran

|  |  |
| --- | --- |
| **Nama *Use Case*** | Mengubah pelajaran |
| **Deskripsi** | Aktor dapat mengubah pelajaran |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Pelajaran berhasil diubah |
| **Skenario** | 1. Aktor masuk ke halaman detail pelajaran 2. Aktor menekan tombol update pada halaman detail pelajaran 3. Aktor mengubah *data* 4. Sistem melakukan validasi 5. Sistem berhasil mengubah *data* pelajaran |
| **Skenario alternatif** | 4.a[pelajaran sudah ada] Kondisi ini akan mengharuskan aktor untuk mengubah judul pelajaran, karena sudah ada sebelumnya. Lalu Kembali ke nomor 3. |

* + 1. Skenario *Use case* menghapus pelajaran

Tabel 12 : Skenario use case menghapus pelajaran

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menghapus pelajaran |
| **Deskripsi** | Aktor dapat menghapus pelajaran |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Pelajaran berhasil dihapus |
| **Skenario** | 1. Aktor masuk halaman detail pelajaran yang akan diubah 2. Aktor menekan tombol hapus pada halaman detail pelajaran 3. Aktor menekan tombol hapus pada *pop up* 4. Sistem berhasil menghapus *data* pelajaran |
| **Skenario alternatif** | 3.a[aktor menekan tombol batal] Kondisi ini membuat sistem membatalkan penghapusan pada pelajaran yang dipilih |

* + 1. Skenario *Use case* menambah teori

Tabel 13 : Skenario use case menambah teori

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menambah teori |
| **Deskripsi** | Aktor dapat menambah teori |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Teori berhasil ditambahkan |
| **Skenario** | 1. Aktor mengisi *form* pada halaman teori 2. Sistem melakukan validasi 3. Sistem berhasil menambah teori |
| **Skenario alternatif** | 2.a[teori sudah ada] Kondisi ini akan mengharuskan aktor untuk mengubah judul teori, karena sudah dibuat sebelumnya. Lalu Kembali ke nomor 1.  2.b[teori tidak dimasukkan *video*] Kondisi ini akan mengharuskan aktor untuk memasukkan *video* kedalam *inputan* yang tersedia. |

* + 1. Skenario *Use case* melihat teori

Tabel 14 : Skenario use case melihat teori

|  |  |
| --- | --- |
| **Nama *Use Case*** | Melihat teori |
| **Deskripsi** | Aktor melihat teori |
| **Aktor** | Admin dan User |
| **Kondisi awal** | Aktor berhasil login |
| **Kondisi akhir** | Aktor berhasil melihat teori |
| **Skenario** | 1. Aktor masuk ke halaman pelajaran 2. Sistem menampilkan pelajaran 3. Aktor berhasil melihat daftar pelajaran |
| **Skenario alternatif** | 3.a[gagal menampilkan teori] Kondisi ini akan menampilkan notifikasi gagal pada aktor |

* + 1. Skenario *Use case* mengubah teori

Tabel 15 : Skenario use case mengubah teori

|  |  |
| --- | --- |
| **Nama *Use Case*** | Mengubah teori |
| **Deskripsi** | Aktor dapat mengubah teori |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Teori berhasil diubah |
| **Skenario** | 1. Aktor masuk ke halaman detail teori 2. Aktor menekan tombol update pada halaman detail teori 3. Aktor mengubah *data* 4. Sistem melakukan validasi 5. Sistem berhasil mengubah *data* teori |
| **Skenario alternatif** | 4.a[teori tidak dimasukkan *video*] Kondisi ini akan mengharuskan aktor untuk memasukkan *video* kedalam *inputan* yang tersedia. |

* + 1. Skenario *Use case* menghapus teori

Tabel 16 : Skenario use case menghapus teori

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menghapus teori |
| **Deskripsi** | Aktor dapat menghapus teori |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Teori berhasil dihapus |
| **Skenario** | 1. Aktor masuk halaman detail teori yang akan dihapus 2. Aktor menekan tombol hapus pada halaman detail teori 3. Aktor menekan tombol hapus pada *pop up* 4. Sistem berhasil menghapus *data* teori |
| **Skenario alternatif** | 3.a[aktor menekan tombol batal] Kondisi ini membuat sistem membatalkan penghapusan pada pelajaran yang dipilih |

* + 1. Skenario *Use case* menambah kuis

Tabel 17 : Skenario use case menambah kuis

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menambah kuis |
| **Deskripsi** | Aktor dapat menambah kuis |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | *Data* kuis berhasil ditambahkan |
| **Skenario** | 1. Aktor mengisi *form* pada halaman kuis 2. Sistem melakukan validasi 3. Sistem berhasil menambah kuis |
| **Skenario alternatif** | 2.a[jawaban tidak diisi] Kondisi ini mengharuskan aktor untuk mengisi field yang tersedia agar tidak kosong. Lalu Kembali ke nomor 1 |

* + 1. Skenario *Use case* melihat kuis

Tabel 18 : Skenario use case melihat kuis

|  |  |
| --- | --- |
| **Nama *Use Case*** | Melihat kuis |
| **Deskripsi** | Aktor dapat melihat kuis |
| **Aktor** | Admin dan User |
| **Kondisi awal** | Aktor berhasil login |
| **Kondisi akhir** | Aktor berhasil melihat kuis |
| **Skenario** | 1. Aktor masuk ke halaman kuis 2. Sistem menampilkan kuis 3. Aktor berhasil melihat daftar kuis |
| **Skenario alternatif** | 3.a[gagal menampilkan kuis] Kondisi ini akan menampilkan notifikasi gagal pada aktor |

* + 1. Skenario *Use case* mengubah kuis

Tabel 19 : Skenario use case mengubah kuis

|  |  |
| --- | --- |
| **Nama *Use Case*** | Mengubah kuis |
| **Deskripsi** | Aktor dapat mengubah kuis |
| **Aktor** | *Admin* |
| **Kondisi awal** | *Admin* berhasil *login* |
| **Kondisi akhir** | *Data* kuis berhasil diubah |
| **Skenario** | 1. Aktor masuk ke halaman detail kuis 2. Aktor menekan tombol update pada halaman detail kuis 3. Aktor mengubah *data* kuis 4. Sistem melakukan validasi 5. Sistem berhasil mengubah *data* kuis |
| **Skenario alternatif** | 4.a[jawaban tidak diisi] Kondisi ini mengharuskan aktor untuk mengisi field yang tersedia agar tidak kosong. Lalu Kembali ke nomor 1 |

* + 1. Skenario *Use case* menghapus kuis

Tabel 20 : Skenario use case menghapus kuis

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menghapus kuis |
| **Deskripsi** | Aktor dapat menghapus kuis |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil melakukan *login* |
| **Kondisi akhir** | *Data* kuis berhasil dihapus |
| **Skenario** | 1. Aktor masuk halaman detail kuis yang akan dihapus 2. Aktor menekan tombol hapus pada halaman detail kuis 3. Aktor menekan tombol hapus pada *pop up* 4. Sistem berhasil menghapus *data* kuis |
| **Skenario alternatif** | 3.a[aktor menekan tombol batal] Kondisi ini membuat sistem membatalkan penghapusan pada kuis yang dipilih |

* + 1. Skenario *Use case* mengerjakan kuis

Tabel 21 : Skenario use case mengerjakan kuis

|  |  |
| --- | --- |
| **Nama *Use Case*** | Mengerjakan kuis |
| **Deskripsi** | Aktor dapat mengerjakan kuis |
| **Aktor** | *User* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Aktor berhasil mengirim jawaban kuis |
| **Skenario** | 1. Aktor menekan kuis yang tersedia 2. Aktor memilih *option* a, b, c, dan d yang telah tersedia pada semua soal 3. Aktor menekan tombol simpan 4. Aktor berhasil mengirim jawaban |
| **Skenario alternatif** | 1.a[*status* kuis] kuis hanya bisa ditekan sesuai dengan urutan pengerjaan yang sudah diselesaikan kuis sebelumnya  3.a[total jawaban sudah sama dengan soal] Kondisi ini akan mengubah tombol simpan menjadi tombol *submit*  3.b[*option* masih belum di isi] Kondisi ini akan menampilkan notifikasi bahwa jawaban belum di isi |

* + 1. Skenario *Use case* melihat hasil kuis

Tabel 22 : Skenario use case melihat hasil kuis

|  |  |
| --- | --- |
| **Nama *Use Case*** | Melihat hasil kuis |
| **Deskripsi** | Aktor dapat melihat hasil kuis yang telah dikerjakan |
| **Aktor** | *User* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Aktor berhasil melihat hasil kuis |
| **Skenario** | 1. Sistem melakukan validasi dari jawaban yang dikirim 2. Sistem menampilkan semua kuis 3. Aktor berhasil melihat hasil kuis yang benar dan salah |
| **Skenario alternatif** | 2.a[jawaban benar] Kondisi ini akan membuat mark berwarna hijau dan centang pada tampilan *option*  2.b[jawaban salah] Kondisi ini akan membuat mark berwarna merah dan X pada tampilan *option* |

* + 1. Skenario *Use case* menambahkan *feedback form*

Tabel 23 : Skenario use case menambahkan feedback form

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menambahkan *feedback form* |
| **Deskripsi** | Aktor dapat menambahkan *feedback form* |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | *Feedback form* berhasil ditambahkan |
| **Skenario** | 1. Aktor mengisi *form* pada tambah pelajaran 2. Aktor menambahkan feedback pada pelajaran 3. Sistem berhasil menambahkan *feedback form* |
| **Skenario alternatif** | 2.a[menekan tombol batal] Pada kondisi ini Aktor akan dialihkan ke halaman pelajaran tanpa menambahkan *feedback form* yang baru. |

* + 1. Skenario *Use case* mengisi *feedback form*

Tabel 24 : Skenario use case feedback form

|  |  |
| --- | --- |
| **Nama *Use Case*** | Mengisi *feedback form* |
| **Deskripsi** | Aktor dapat mengisi *feedback form* |
| **Aktor** | *User* |
| **Kondisi awal** | Aktor dapat melakukan *login* |
| **Kondisi akhir** | Feedback berhasil dikirim |
| **Skenario** | 1. Aktor mengisi *form* pada *feedback form* 2. Sistem melakukan validasi 3. Feedback berhasil dikirim |
| **Skenario alternatif** | 2.a[field masih belum diisi] Kondisi ini akan membuat *user* ke field yang masih kosong dan diharuskan untuk mengisi field tersebut.  2.b[field berhasil di kirim] Kondisi ini akan membuat *user* otomatis terlogout dalam 30detik. |

* + 1. Skenario *Use case* menambah *data* karyawan

Tabel 25 : Skenario use case menambah data karyawan

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menambah *data* karyawan |
| **Deskripsi** | Aktor dapat menambah *data* karyawan |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | *Data* karyawan berhasil ditambahkan |
| **Skenario** | 1. Aktor memilih *data* *file* *data* karyawan untuk di *import* 2. Sistem melakukan validasi 3. Sistem menyimpan *data* karyawan |
| **Skenario alternatif** | 2.a[*Format* tidak sesuai] Kondisi ini menampilkan notifikasi gagal *import*. Lalu Kembali ke nomor 1. |

* + 1. Skenario *Use case* melihat *data* karyawan

Tabel 26 : Skenario use case melihat data karyawan

|  |  |
| --- | --- |
| **Nama *Use Case*** | Melihat data karyawan |
| **Deskripsi** | Aktor melihat data karyawan |
| **Aktor** | Admin |
| **Kondisi awal** | Aktor berhasil login |
| **Kondisi akhir** | Aktor berhasil melihat data karyawan |
| **Skenario** | 1. Aktor masuk ke halaman karyawan 2. Sistem menampilkan data karyawan 3. Aktor berhasil melihat data karyawan |
| **Skenario alternatif** | 3.a[gagal menampilkan data karyawan] Kondisi ini akan menampilkan notifikasi gagal pada aktor |

* + 1. Skenario *Use case* menambah *data* trainer

Tabel 27 : Skenario use case menambah data trainer

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menambah *data* trainer |
| **Deskripsi** | Aktor dapat menambah *data* trainer |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | *Data* trainer berhasil ditambahkan |
| **Skenario** | 1. Aktor mengisi *form* pada halaman trainer 2. Sistem melakukan validasi 3. Sistem berhasil menambah trainer |
| **Skenario alternatif** | 2.a[Trainer sudah ada] Kondisi ini akan menampilkan notifikasi trainer sudah tersedia. Lalu Kembali ke nomor 1. |

* + 1. Skenario *Use case* melihat *data* trainer

Tabel 28 : Skenario use case melihat data trainer

|  |  |
| --- | --- |
| **Nama *Use Case*** | Melihat data trainer |
| **Deskripsi** | Aktor dapat melihat data trainer |
| **Aktor** | Admin |
| **Kondisi awal** | Aktor berhasil login |
| **Kondisi akhir** | Aktor berhasil melihat data trainer |
| **Skenario** | 1. Aktor masuk ke halaman trainer 2. Sistem menampilkan data trainer 3. Aktor berhasil melihat data trainer |
| **Skenario alternatif** | 3.a[gagal menampilkan data trainer] Kondisi ini akan menampilkan notifikasi gagal pada aktor |

* + 1. Skenario *Use case* menghapus *data* trainer

Tabel 29 : Skenario use case menghapus data trainer

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menghapus *data* trainer |
| **Deskripsi** | Aktor dapat menghapus *data* trainer |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | *Data* trainer berhasil dihapus |
| **Skenario** | 1. Aktor masuk halaman trainer yang akan dihapus 2. Aktor menekan tombol hapus pada halaman trainer 3. Aktor menekan tombol hapus pada *pop up* 4. Sistem berhasil menghapus *data* trainer |
| **Skenario alternatif** | 3.a[aktor menekan tombol batal] Kondisi ini membuat sistem membatalkan penghapusan pada trainer yang dipilih |

* + 1. Skenario *Use case* melihat riwayat pengerjaan *user*

Tabel 30 : Skenario use case melihat riwayat pengerjaan user

|  |  |
| --- | --- |
| **Nama *Use Case*** | Melihat Riwayat pengerjaan *user* |
| **Deskripsi** | Aktor dapat melihat riwayat pengerjaan *user* |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Aktor dapat melihat Riwayat pengerjaan |
| **Skenario** | 1. Aktor pergi ke halaman histori 2. Sistem berhasil menampilkan daftar *user* sedang mengerjakan training |
| **Skenario alternatif** | 2.a[sistem gagal menampilkan daftar *user* sedang mengerjakan training] Kondisi ini akan menampilkan notifikasi gagal pada aktor |

* + 1. Skenario *Use case* menghapus riwayat pengerjaan *user*

Tabel 31 : Skenario use case menghapus riwayat pengerjaan user

|  |  |
| --- | --- |
| **Nama *Use Case*** | Menghapus Riwayat pengerjaan *user* |
| **Deskripsi** | Aktor dapat menghapus riwayat pengerjaan *user* |
| **Aktor** | *Admin* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Aktor dapat menghapus Riwayat pengerjaan |
| **Skenario** | 1. Aktor pergi ke halaman histori 2. Aktor menekan tombol hapus pada halaman histori 3. Aktor menekan tombol hapus pada *pop up* 4. Sistem berhasil menghapus *data* |
| **Skenario alterinatif** | 3.a[aktor menekan tombol batal] Kondisi ini akan membuat sistem membatalkan penghapusan pada *data* |

* + 1. Skenario *Use case* logout

Tabel 32 : Skenario use case logout

|  |  |
| --- | --- |
| **Nama *Use Case*** | Logout |
| **Deskripsi** | Aktor dapat melakukan logout |
| **Aktor** | *Admin* & *User* |
| **Kondisi awal** | Aktor berhasil *login* |
| **Kondisi akhir** | Aktor berhasil logout |
| **Skenario** | 1. Aktor menekan tombol logout 2. Aktor berhasil logout |
| **Skenario alternatif** | 2.a[aktor gagal logout] Kondisi ini akan menampilkan notifikasi gagal pada aktor |

* 1. Rancangan Antar Muka
     1. *Admin*
        1. Halaman *Dashboard*

Gambar 4 : Halaman dashboard

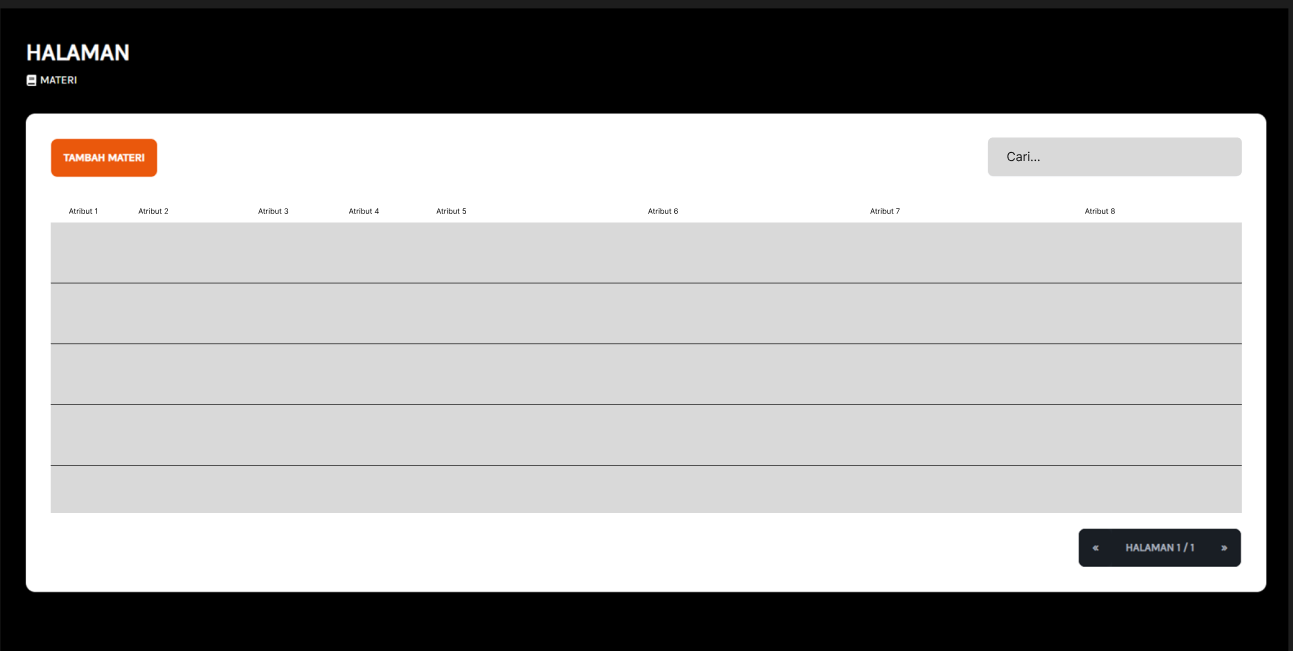
A screenshot of a computer

Description automatically generated

Gambar 4 diatas merupakan halaman dashboard untuk admin. Pada halaman tersebut terdapat chart yang nantinya akan menghitung jumlah peserta yang sudah mengikuti training settiap bulannya. Lalu diatas chart terdapat informasi jumlah materi, jumlah karyawan, jumlah trainer, dan juga jumlah riwayat pengerjaan yang sedang dilakukan oleh user. Lalu pada tampilan sebelah kiri terdapat sidebar yang menavigasikan ke halaman lain dan juga tombol logout untuk keluar dari halaman.

* + - 1. Tampilan *Data* Tabel

Gambar 5 : Tampilan data tabel



Gambar 5 diatas merupakan tampilan admin yang menampilkan data. Tampilan ini nantinya akan diterapkan ke semua halaman yang memiliki data untuk ditampilkan.

* + - 1. Halaman Tambah *Data* Tabel

Gambar 6 : Halaman tambah data tabel



Gambar 6 diatas merupakan tampilan admin untuk kebutuhan menambahkan data. Tampilan ini nantinya akan diterapkan ke semua halaman yang memiliki fitur tambah data.

* + - 1. Halaman *Detail* *Data*

Gambar 7 : Halaman detail data

A screenshot of a computer

Description automatically generated

Gambar 7 diatas merupakan tampilan admin untuk menampilkan detail data. Pada tampilan ini admin nantinya juga dapat mengupdate dan juga menghapus data dengan cara meneken tombol yang sudah disediakan. Tampilan ini nantinya akan diterapkan ke semua halaman yang memiliki data untuk ditampilkan detailnya.

* + 1. *User*
       1. Halaman *Login*

Gambar 8 : Halaman login

*A screenshot of a video game

Description automatically generated*

Gambar 8 diatas merupakan halaman login untuk aktor user dan juga admin. Pada halaman ini juga akan menampilkan notifikasijika data yang dimasukkan tidak sesuai.

* + - 1. Halaman *Welcome*

Gambar 9 : Halaman welcome

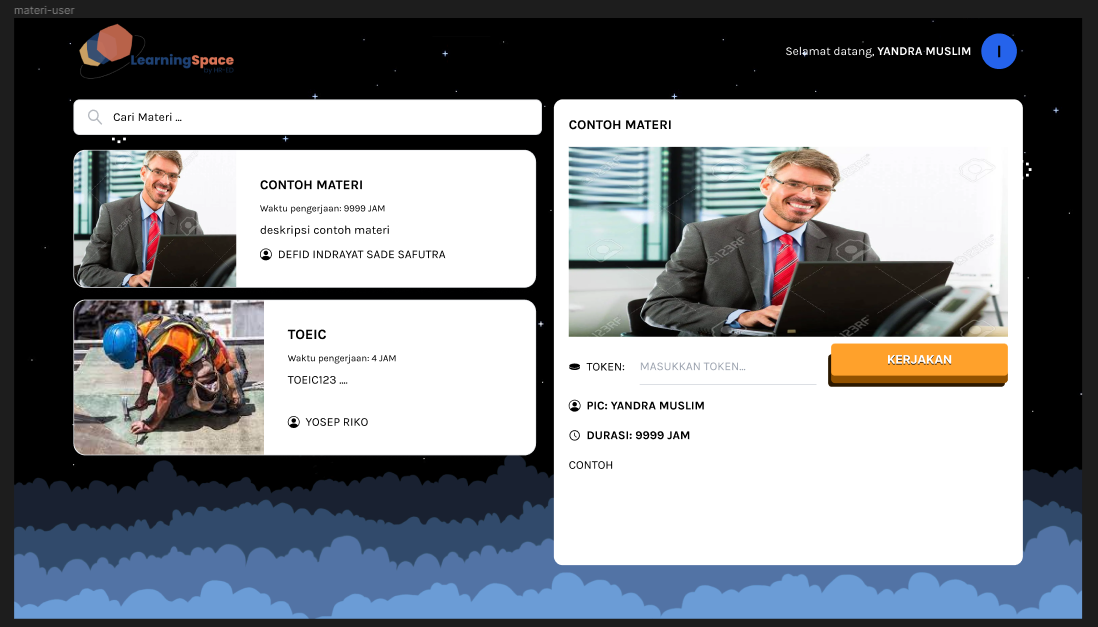
A screenshot of a video game

Description automatically generated

Gambar 9 diatas merupakan halaman *welcome* untuk aktor user. Halaman ini merupakan halaman pertama untuk user ketika sudah berhasil login. Lalu user diarahkan untuk menekan tombol jelajah agar masuk ke halaman berikutnya.

* + - 1. Halaman Daftar Materi

Gambar 10 : Halaman daftar materi



Gambar 10 diatas merupakan halaman materi untuk aktor user. Halaman ini akan menampilkan data materi yang telah disediakan oleh admin. Lalu untuk masuk ke dalam pelajaran dari materi tersebut user diharuskan untuk memasukkan token ditombol input yang sudah disediakan.

* + - 1. Halaman Pelajaran

Gambar 11 : Halaman pelajaran

A video game screen with stars and space objects

Description automatically generated

Gambar 11 diatas merupakan halaman pelajaran untuk aktor user. Halaman ini akan menampilkan pelajaran yang telah di sediakan oleh admin. Pada halaman ini terdapat 3 status yang berupa gambar. Jika gambar roket, artinya pelajaran bisa dibuka/dikerjakan, jika gambar alien maka pelajaran tidak bisa ditekan, dan jika gambar astronaut artinya pelajaran sudah dikerjakan.

* + - 1. Halaman Teori

Gambar 12 : Halaman teori

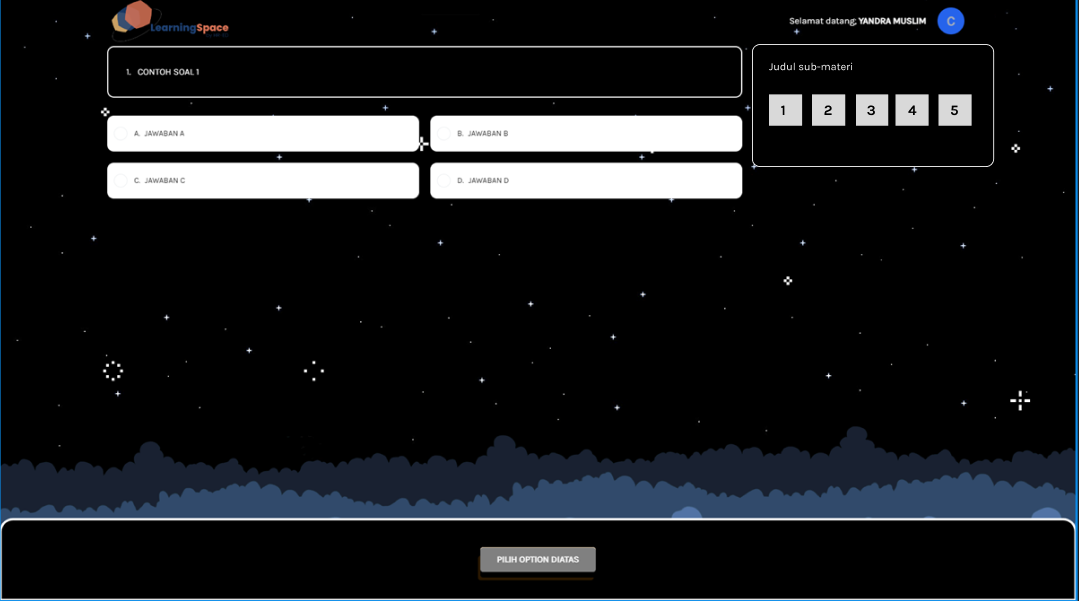
A screenshot of a video

Description automatically generated

Gambar 12 diatas merupakan halaman teori untuk aktor user. Halaman ini akan menampilkan teori yang telah di sediakan oleh admin. Pada halaman ini terdapat 2 tampilan yaitu video pada sebelah kiri, dan catatan/rangkuman pada sebelah kanan.

* + - 1. Halaman Soal

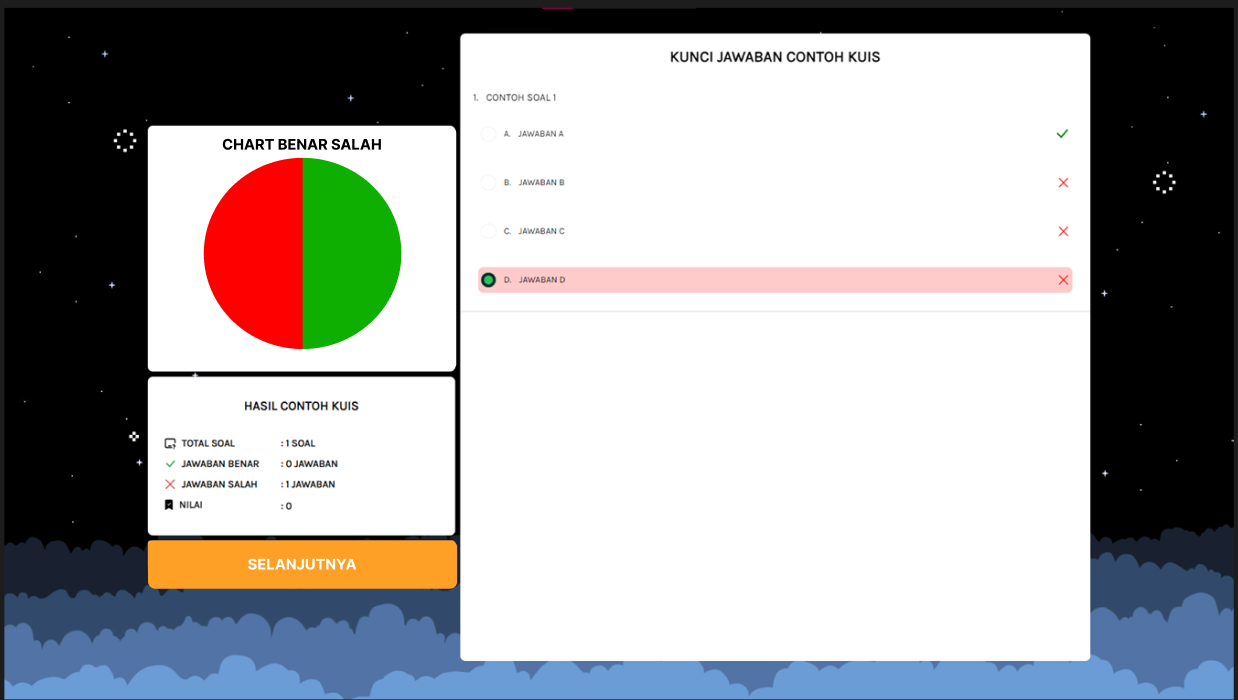
Gambar 13 : Halaman soal



Gambar 13 diatas merupakan halaman kuis untuk aktor user. Halaman ini akan menampilkan kuis yang telah di sediakan oleh admin. Pada halaman ini jika option kuis sudah ditekan, makan tombol akan berubah menjadi warna kuning dengan tulisan simpan dan diarahkan ke soal berikutnya. Ketika semua soal sudah terisi maka tombol akan berubah menjadi submit.

* + - 1. Halaman review kuis

Gambar 14 : Halaman review kuis



Gambar 14 diatas merupakan halaman review kuis untuk aktor user. Halaman ini akan menampilkan hasil jawaban yang sudah dikerjakan oleh user. Pada halaman ini akan menampilkan jumlah soal, jumlah jawaban benar, jumlah jawaban salah, nilai, dan juga koreksi benar dan salah dari kunci jawaban.

* + - 1. Halaman *Feedback Form*

Gambar 15 : Halaman feedback form

A screenshot of a computer

Description automatically generated

Gambar 15 diatas merupakan halaman feedback form untuk aktor user. Halaman ini akan menampilkan form yang telah di sediakan oleh admin. Ketika user sudah mensubmit maka user akan dibuat logout otomatis dalam 30detik.

1. *Development,* Pada tahap ini, pengembang menggunakan teknologi dan *tools* seperti *ReactJS* untuk frontend, ExpressJS untuk backend, dan *MySQL sebaga*i basis *data*. Proses *development* dilakukan selama kurang lebih 4 bulan menggunakan browser Chrome dan Visual Studio Code (VSCode) sebagai Integrated *Development* *Environment* (IDE). Penggunaan teknologi tersebut sesuai dengan kebutuhan fungsional dan non-fungsional yang telah didefinisikan pada tahap Requirements.
2. *Testing,* tahap ini dilakukan selama kurang lebih 1 minggu. Testing mencakup pengujian secara manual terhadap semua fungsi aplikasi. Tim *HRD* dari perusahaan juga turut serta dalam proses pengujian untuk memastikan bahwa aplikasi *e-learning* memenuhi kebutuhan dan harapan mereka. Pengujian manual ini melibatkan simulasi penggunaan aplikasi dalam situasi nyata untuk mengidentifikasi dan memperbaiki potensi *bug* atau masalah lainnya.
3. *Maintenance,* pada tahap ini *maintenance* dilakukan secara proaktif, termasuk perbaikan *bug*, peningkatan fungsionalitas, dan penerapan perubahan berdasarkan umpan balik pengguna. Pemeliharaan juga mencakup pemantauan kinerja aplikasi secara berkala dan peningkatan keamanan sistem. *Maintenance* ini dijalankan untuk memastikan bahwa aplikasi tetap berjalan optimal dan memenuhi kebutuhan pengguna seiring berjalannya waktu.
   1. Gambaran Umum Sistem



Gambar 16 : Gambaran Umum Sistem

*Platform* *E-learning* yang akan di kembangkan nantinya dirancang khusus untuk karyawan baru menjadi alat penting bagi tim *HRD* dalam penyelenggaraan pelatihan. Halaman utama memberikan sambutan hangat dan memberikan gambaran umum tentang pelatihan yang tersedia, sementara itu *e-learning* ini menyajikan materi teori, kuis, dan kuis *toeic* dengan evaluasi otomatis. Bagian kuis menawarkan variasi soal dan teori dalam *format* *video* dan catatan untuk meningkatkan pengalaman pembelajaran. Melalui halaman *admin*, *administrator* dapat dengan mudah mengelola dan menyusun ulang isi pelatihan sesuai kebutuhan. Profil karyawan menyajikan kemajuan dan pencapaian individu, dan pusat bantuan menyediakan dukungan langsung.

# DAFTAR PUSTAKA

[1] Anggita, D., (2021). 5 Manfaat e-Learning untuk Pelatihan Karyawan. Employers.glints. https://employers.glints.com/id-id/blog/5-manfaat-e-learning-untuk-pelatihan-karyawan/

[2]Dewaweb, T, (2022). Apa itu *E-learning*? Pengertian, Jenis, Manfaat dan Contohnya. Dewaweb. https://www.dewaweb.com/blog/pengertian-*e-learning*/

[3] Muhammad, R.A., (2020). Pengertian, Sejarah, Manfaat & Tahapanya Metode *Waterfall*: Pengertian, Sejarah, Manfaat & Tahapanya. Sekawanmedia. https://www.sekawanmedia.co.id/blog/metode-*waterfall*/

[4] Faradilla, A., (2023). Apa Itu *HTML*? Fungsi dan Cara Kerja *HTML*. Hostinger. https://www.hostinger.co.id/tutorial/apa-itu-*html*

[5] Ariata C., (2022). Apa Itu *CSS*? Pengertian, Fungsi, dan Cara Kerjanya. Hostinger. https://www.hostinger.co.id/tutorial/apa-itu-*css*

[6] Muhammad, A.I., (2020). Apa Itu *Tailwind* *CSS* ?. Kawankoding. https://www.kawankoding.id/apa-itu-*tailwind*-*css*/

[7] Ariata, C., (2023). Apa Itu *JavaScript*? Apa Perbedaannya dengan Java?. Hostinger. https://www.hostinger.co.id/tutorial/apa-itu-*javascript*

[8] HANRY HAM., S.KOM., M.ENG, (2019). Apa itu React.Js?. Socs.binus. https://socs.binus.ac.id/2019/12/30/apa-itu-react-js/

[9] Yasin, K., (2022). Pengertian *MySQL, Fungs*i, dan Cara Kerjanya (Lengkap). Niagahoster. https://www.niagahoster.co.id/blog/*mysql-adalah*/

[10] Ariffud, M., (2021). Mengenal *Express.js*: Pengertian, Cara Kerja, Keunggulan, Tutorial. Niagahoster. https://www.niagahoster.co.id/blog/express-js-adalah/

[11] Aulia, R.F., (2023). Apa itu *UML*? Ini Pengertian, Fungsi, dan Contohnya. Sekawanmedia. https://www.sekawanmedia.co.id/blog/apa-itu-*uml*/

[12] Muhammad, R.A., (2021). *ERD*: Pengertian, Jenis, Komponen & Cara membuatnya. Sekawanmedia. https://www.sekawanmedia.co.id/blog/apa-itu-erd/

[13] Syahrizal, W., (2021). Apa itu *IIS*? Pengertian, Kelebihan dan System Requirement. Rumahweb. https://www.rumahweb.com/journal/apa-itu-*iis*-adalah/