

Analisis dan Pengujian Aplikasi Ideabox dan Eventeer

Matthew Christopher^{#1}, Diana Trivena Yulianti^{*2}

[#]Program Studi S1 Sistem Informasi, Fakultas Teknologi dan Rekayasa Cerdas, Universitas Kristen Maranatha
Jl. Prof. Drg. Surya Sumantri No. 65, Sukawarna, Bandung, Indonesia

¹2173005@maranatha.ac.id

²diana.trivena@it.maranatha.edu

Abstract— This research explores the role of System Analysts and Quality Assurance in designing Entity-Relationship Diagrams (ERD) to represent database structures and conducting functional and performance testing to ensure system reliability. The study involves updating system documentation, including ERD, Use Case Diagrams, and Activity Diagrams, as well as verifying application functionality through structured testing. These processes were applied to the Ideabox and Eventeer applications, resulting in refined ERD documentation, workflow diagrams, test plans, and performance evaluation outcomes for Eventeer.

Keywords— Entity Relationship Diagram, Functional Testing, Performance Testing, Quality Assurance, System Analyst

I. PENDAHULUAN

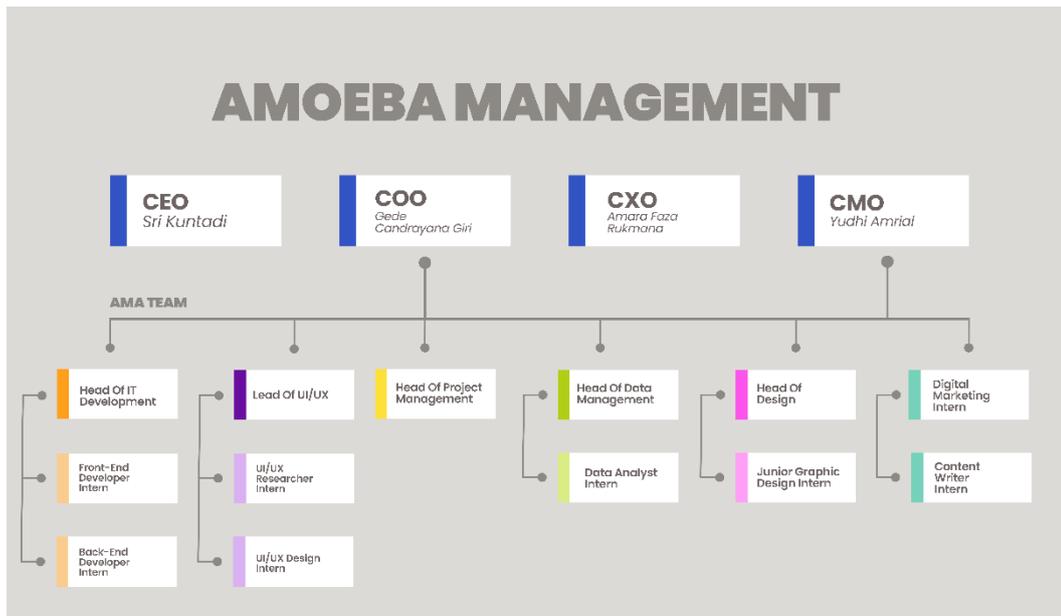
Digital Amoeba merupakan program yang berfokus pada *Research and Innovation* di PT Telekomunikasi Indonesia. Program ini memiliki beberapa *project* yang sedang berjalan, seperti Ideabox dan Eventeer Telkom Group. Ideabox merupakan *single platform* yang memfasilitasi karyawan Telkom Group dalam menyampaikan ide dan berkolaborasi [1]. Saat ini, Ideabox masih menggunakan CodeIgniter 3, yang memiliki kekurangan dari segi *security*. Dengan bertambahnya jumlah *user*, diperlukan penyesuaian ke framework Laravel. Namun, dokumentasi sistem seperti *Entity Relationship Diagram* dan *Database Schema* belum dilakukan secara berkala karena keterbatasan sumber daya manusia dan beban kerja yang tinggi.

Eventeer merupakan *All-In-One platform* yang dirancang untuk membantu organisasi, *community manager*, dan *event organizer* dalam mengelola serta mengembangkan komunitas [2]. Meskipun telah digunakan di lingkungan internal Telkom, dashboard admin masih terpisah dari web utama, sehingga perlu dikembangkan lebih lanjut dalam persiapan *Go To Market*. Selain itu, dalam persiapan penggunaan Eventeer untuk instansi SPBE, diperlukan *performance testing* guna menilai kesiapan aplikasi. Penelitian ini membahas cara mendokumentasikan *Entity Relationship Diagram* (ERD) agar dapat menggambarkan struktur data dalam sistem basis data dengan lebih jelas. Selain itu, penelitian ini mencakup pengujian fungsional dan kinerja sistem untuk memastikan kesesuaiannya dengan rancangan yang telah dibuat.

II. PROFIL PERUSAHAAN

PT Telekomunikasi Indonesia (Telkom Indonesia) adalah BUMN yang bergerak di bidang layanan teknologi informasi, komunikasi, dan telekomunikasi digital di Indonesia. Didirikan pada Juli 1965, Telkom Indonesia merupakan hasil pemisahan dari Perusahaan Negara Pos dan Telekomunikasi (PN Postel) yang berdiri sejak 1961 [5]. Visinya adalah menjadi digital telco pilihan utama untuk memajukan masyarakat, yang didukung oleh misi mempercepat pembangunan infrastruktur dan *platform* digital cerdas, mengembangkan talenta digital unggulan, serta mengorkestrasi ekosistem digital untuk memberikan pengalaman digital terbaik bagi pelanggan [3].

Digital Amoeba merupakan *Corporate-Innovation Lab* milik Telkom Indonesia yang berada di bawah *Research & Innovation Management System (RIMS)*. Digital Amoeba didirikan pada 2017, bertujuan untuk mempercepat transformasi digital di Telkom Indonesia dengan menciptakan solusi bagi permasalahan yang dihadapi perusahaan. Visi Digital Amoeba adalah “*We want to be the real game changer for Telkom and Indonesia to be the best,*” yang didukung dengan misi mengembangkan talenta inovasi digital, produk dan layanan digital, serta sistem inovasi yang relevan. Sejak 2017 hingga 2021, lebih dari 3.900 ide inovasi telah diajukan dan lebih dari 10.500 talenta terlibat dalam program ini. Beberapa produk inovasi yang telah dikembangkan antara lain Ideabox, Eventeer, *Omni Communication Assistant (OCA)*, Worki, Bigbox, Sprint Think, dan Smarteye. Slogan Digital Amoeba adalah “*We don’t create products, we build people who create products,*” dengan prinsip *Build People, Build Business*, dan *Build System* sebagai pilar utama dalam mendorong inovasi dan pengembangan talenta di lingkungan Telkom Indonesia [4].



Gambar 1. Struktur Organisasi Digital Amoeba

III. LANDASAN TEORI

A. Entity Relationship Diagram

Entity Relationship Diagram merupakan salah satu model data yang dapat dimanfaatkan untuk menggambarkan hubungan antar entitas (objek) dalam sebuah sistem informasi[5]. Dalam *Entity Relationship Diagram* menggambarkan secara detail relasi antar entitas atau objek berikut dengan atribut-atributnya. Dengan memanfaatkan *Entity Relationship Diagram* dapat mempermudah dalam melakukan *debugging database* apabila terjadi masalah pada *database* dikarenakan *Entity Relationship Diagram* dapat memodelkan *database scheme* dengan lebih terstruktur dan rapih [6].

1) *Elemen Dasar Entity Relationship Diagram*: Terdapat tiga elemen dasar dalam *Entity Relationship Diagram* meliputi entitas, atribut dan relasi. Entitas merupakan objek atau konsep yang ditentukan dan memiliki data yang perlu disimpan dalam basis data. Pada umumnya entitas dapat digambarkan dalam bentuk persegi panjang kemudian diberi label nama. Contoh dari entitas seperti mahasiswa, produk atau pesanan. Atribut merupakan bagian dari suatu entitas yang menjelaskan karakteristik dari entitas tersebut. Atribut pada umumnya digambarkan dengan bentuk oval kecil kemudian diberi label nama. Contoh dari atribut pada entitas Mahasiswa seperti tanggal lahir, Nama, NIM, dll. Relasi merupakan hubungan antara dua entitas atau lebih yang ada pada *database*. Relasi digambarkan oleh garis kemudian diberi label nama. Terdapat tiga jenis relasi yaitu *One-to-One*, *One-to-Many* dan *Many-to Many* [5].

2) *Jenis-jenis Entity Relationship Diagram*: *Entity Relationship Diagram* memiliki tiga jenis yaitu ERD Konseptual, ERD Logis, dan ERD Fisik. *Entity Relationship Diagram* Konseptual merupakan jenis ERD yang digunakan untuk menjelaskan hubungan atau relasi antar entitas secara konseptual, dengan artian tidak memperhatikan bagaimana cara implementasi secara teknis. Pada umumnya ERD konseptual dimanfaatkan dalam perancangan konsep *database*. *Entity Relationship Diagram* Logis merupakan jenis ERD yang dimanfaatkan untuk menjelaskan hubungan atau relasi antar entitas secara teknis dan berkaitan dengan *database*. Pada umumnya *Entity Relationship Diagram* logis dimanfaatkan untuk perancangan model *database* secara logis. *Entity Relationship Diagram* Fisik merupakan representasi teknis dari implementasi ERD logis. ERD fisik biasanya dimanfaatkan dalam pembuatan dan implementasi *database* secara fisik[5].

B. Unified Modeling Language

Unified Modeling Language atau yang biasa disingkat menjadi UML adalah salah satu standart bahasa yang banyak dimanfaatkan pada dunia industri dalam menentukan kebutuhan, menganalisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [7].

1) *Jenis-Jenis Diagram UML*: Pada *Unified Modeling Language* terdapat beberapa jenis diagram yang membantu dalam menggambarkan sebuah sistem informasi seperti *use case diagram*, *class diagram*, *sequence diagram*, dan *activity*

diagram. *Use case diagram* merupakan sebuah diagram yang menjelaskan persyaratan secara *functional* dalam sebuah sistem. Dalam *use case diagram* lebih menekankan pada penjelasan terkait “apa” yang dilakukan oleh sistem, bukan pada “bagaimana”. Manfaat dari penggunaan *use case diagram*, membantu dalam menentukan kebutuhan-kebutuhan yang ada pada sebuah sistem serta menjadi media komunikasi dengan klien dalam menjelaskan sebuah rancangan sistem. Sedangkan *Activity diagram* merupakan sebuah diagram yang menjelaskan bagaimana alur aktivitas yang ada pada sebuah sistem informasi yang sedang dirancang. *Activity diagram* digunakan untuk memberikan gambaran bagaimana aktivitas user dan juga respon dari sistem serta menjelaskan keputusan yang mungkin terjadi dan penjelasan bagaimana berakhirnya aktivitas tersebut [7].

C. Identifier

Identifier atau pengenal dalam pemrograman merupakan penggunaan nama untuk mengidentifikasi entitas seperti objek, fungsi dan variabel pada dalam program. Fungsi dari *identifier* yaitu label yang diberikan sebagai identitas unik kepada entitas tersebut dalam konteks pemrograman [8].

D. Software Quality Assurance

Software Quality Assurance atau yang biasa disingkat menjadi SQA merupakan sekumpulan aktivitas dalam mendefinisikan dan mengevaluasi kecukupan proses dari perangkat lunak untuk dapat memberikan bukti yang bisa membangun kepercayaan dari proses perangkat lunak tersebut sudah sesuai dan mampu menghasilkan produk perangkat lunak berkualitas yang sesuai dengan tujuan yang diharapkan. Fungsi *software quality assurance* juga dapat diatur secara independen dari proyek dalam artian terlepas dari tekanan teknis, manajerial, atau finansial yang berasal dari proyek tersebut [9].

E. Software testing

Software testing merupakan sebuah metode yang digunakan untuk memastikan aplikasi yang dikembangkan oleh developer sudah sesuai dengan rancangan dan dapat berjalan dengan baik serta dapat mencapai tujuan dari perusahaan. Dengan melakukan software testing perusahaan dapat memastikan standar kualitas serta melihat kekurangan atau kesalahan dari hasil pengembangan aplikasi dengan maksud mencapai hasil yang maksimal [10].

F. Test Case

Test case adalah serangkaian pengujian yang dilakukan berdasarkan kondisi dan hasil yang sudah ditetapkan sebelumnya [11].

G. Black Box Testing

Black box testing merupakan metode pengujian untuk spesifikasi fungsional. Dimana pada *black box testing* digunakan untuk mengetahui terkait fungsi-fungsi mulai dari *input* dan *output* dari sebuah perangkat lunak sudah sesuai dengan spesifikasi yang dibutuhkan atau belum [7]. Pada *black box testing* berfokus pada *output*-nya atau reaksi dari sistem. Penguji tidak perlu mengetahui bagaimana proses secara detail yang ada pada perangkat lunak.

IV. HASIL PEKERJAAN

A. Tahapan Implementasi

Digital Amoeba memanfaatkan *framework* yaitu scrum sebagai metodologi untuk memastikan setiap *project* dalam berjalan dengan lebih terstruktur dan sistematis. Ketika memanfaatkan scrum, terdapat tahapan-tahapan yang harus dilalui ketika menjalani sprint. Diawali dengan *Sprint Planning* bersama dengan mentor, *Scrum Master*, *Product Owner* dan *developer*. Dimana pada *Sprint Planning*, *Product Owner* akan menugaskan *developer* untuk mengembangkan fitur berdasarkan *user story point*. Pada Digital Amoeba, durasi satu *sprint* yaitu selama dua minggu, terhitung dari sesi *Sprint Planning* dan diakhiri dengan *Sprint Review* bersama dengan *stakeholder* maupun mentor.

Setiap hari rabu dan jumat terdapat *checkpoint meeting* secara daring bersama dengan mentor, untuk mentor bisa memonitoring pengerjaan dari *developer* sudah sesuai berdasarkan tugas yang diberikan langsung oleh *Product Owner*. Kemudian apabila terdapat kendala yang dihadapi langsung oleh *developer* dalam menjalankan tugas yang diberikan, bisa langsung berdiskusi dengan mentor pada sesi *checkpoint meeting*.

Setiap harinya pada pagi hari pukul 10.00, penulis melakukan *daily scrum* bersama dengan tim, baik *project eventeer* dan *ideabox* untuk membahas terkait *progress* dari *sprint* yang sedang berjalan. *Scrum Master* yang akan memimpin *daily scrum* dan menanyakan terkait apa yang sudah dikerjakan hari sebelumnya, apa yang mau dilakukan hari ini dan apabila terdapat kendala dalam melakukan tugas dan tanggung jawab.

Terdapat sesi *handover design* aplikasi yang sudah dibuat oleh UI/UX yang dihadiri oleh *Product Owner*, *System Analyst* dan juga mentor. Dimana pada saat sesi *handover*, seorang *System Analyst* dibantu mentor akan menentukan apakah *design* yang sudah dibuat sebelumnya memungkinkan untuk diimplementasi atau tidak. Pada sesi *handover design* juga merupakan momen untuk bernegosiasi apabila terdapat *design* yang kurang memungkinkan untuk diimplementasikan dengan mempertimbangan kapasitas, kompetensi developer, serta aspek teknis. *System Analyst* juga bisa memberikan masukan apabila terdapat *flow* atau tampilan yang mungkin bisa ditambahkan untuk meningkatkan *user experience*. Penulis memberikan masukan terkait *flow* aplikasinya untuk setiap *action* alangkah baiknya akan ada *popup message* untuk meningkatkan *user experience*. Serta memberikan masukan untuk menyeragamkan bentuk *popup message*, dapat berupa *toast* maupun menggunakan *sweet alert*.

Setiap *sprint*, penulis ditugaskan oleh *Product Owner* untuk melakukan pengujian dari hasil pengembangan yang sudah selesai dikembangkan pada *sprint* sebelumnya, serta membuat *test case* dari fitur yang sedang dikembangkan pada *sprint* yang sedang berjalan. Adapun beberapa kali terdapat juga tugas yang diberikan langsung oleh mentor ketika *sprint* sedang berjalan seperti diminta untuk melakukan *update* dokumen *Entity Relationship Diagram* untuk aplikasi Eventeer dan Ideabox.

Ketika penulis mendapatkan tugas untuk melakukan pengujian aplikasi, penulis akan melihat dari tugas yang diberikan sudah ada dokumentasi untuk *flow chart* aplikasi atau belum. Apabila belum ada, maka penulis akan mempelajari secara mandiri terkait dari *flow* yang akan dikembangkan, seperti pada *project* Ideabox. Pada *project* Ideabox, penulis mempelajari *flow* aplikasi secara mandiri dengan melihat aplikasi yang sudah ada sebelumnya dikarenakan, *project* Ideabox merupakan *project* revamp dari aplikasi yang sudah ada dengan menerapkan laravel sebagai *framework*. Pada *project* Eventeer sudah terdapat *design system* dari UI/UX, maka penulis akan mempelajari *flow* aplikasi dari dokumentasi yang sudah dibuat.

Setelah mempelajari terkait *flow* aplikasi, penulis akan mulai membuat *test case* berdasarkan dokumentasi yang diberikan. Dalam membuat *test case*, penulis menerapkan *case positive*, *negative* dan *destructive* dalam upaya pengujian ke aplikasi untuk melihat bagaimana reaksi dari aplikasi apabila dilakukan *case* tersebut. Apabila dokumen *test case* sudah selesai dibuat, penulis akan meminta persetujuan dan nantinya mentor akan mengecek hasil pengerjaan penulis.

Dari hasil *review document*, mentor akan menentukan apakah dari *test case* yang dibuat sudah cukup dan sudah mencakup keseluruhan pengujian untuk fitur yang dikembangkan atau masih perlu dilakukan penyesuaian. Jika sudah mendapatkan persetujuan dari mentor maka penulis dapat melanjutkan pengujian aplikasi. Apabila perlu penyesuaian maka penulis akan melakukan penyesuaian terlebih dahulu berdasarkan masukan dari mentor dan akan di-*review* kembali oleh mentor.

Dalam pengujian aplikasi, penulis menjalankan *test case* yang sudah dibuat sebelumnya dan memilih status berdasarkan reaksi aplikasi apakah sudah sesuai dengan harapan atau belum. Adapun status yang dapat dipilih yaitu:

- *Passed Production*: status yang dapat dipilih ketika hasil pengujian atau reaksi dari aplikasi sudah memenuhi harapan yang sudah dibuat pada saat dilakukan pengujian di *environment production*.
- *Passed Development*: status yang dapat dipilih ketika hasil pengujian atau reaksi dari aplikasi sudah memenuhi harapan yang sudah dibuat pada saat dilakukan pengujian di *environment development*.
- *Pass Constraint*: status yang dapat dipilih ketika hasil pengujian atau reaksi dari aplikasi sudah memenuhi batasan yang ditetapkan sebelumnya.
- *Failed*: status yang dapat dipilih ketika hasil pengujian atau reaksi dari aplikasi tidak sesuai harapan pada saat dilakukan pengujian baik di *environment production* maupun *development*.

Setiap hasil pengujian yang memiliki status *failed*, selalu diberikan keterangan oleh penulis untuk *developer* dalam menjelaskan kesalahan yang terjadi ketika menjalankan *test case* tersebut. Setelah selesai pengujian aplikasi, penulis akan memindahkan *issues* yang ditemukan ke *board* yang digunakan bersama dengan *developer* supaya bisa langsung menugaskan kepada *developer* yang akan memperbaiki *issue* yang ditemukan. Pencatatan *issue* yang ditemukan pada *board* di-*assign* kepada *developer* sesuai dengan permasalahan yang ada baik dari *front-end* atau *back-end*. Pada kondisi khusus setelah melakukan pengujian aplikasi dan didapati banyak *issue* yang ditemukan, penulis berkolaborasi dengan *product owner* untuk menentukan prioritas *issue* dari hasil pengujian.

Pada bulan Oktober 2024, penulis mendapatkan tugas untuk melakukan pengujian kinerja dari aplikasi Eventeer dalam upaya persiapan untuk digunakan oleh instansi SPBE untuk media learning. Hal yang pertama dilakukan penulis yaitu melakukan instalasi *tools* yaitu Taurus dan membuat konfigurasi pengujian kinerja untuk aplikasi Eventeer. Dimana penulis mempelajari dari dokumentasi yang diberikan oleh mentor untuk cara penggunaan Taurus dan mempelajari dari internet dalam pembuatan konfigurasi untuk pengujian kinerja menggunakan Taurus. Penulis mengajukan persetujuan terkait konfigurasi pengujian yang telah dibuat berdasarkan *list API* yang diberikan oleh mentor untuk dilakukan pengujian. Setelah mendapatkan persetujuan dari mentor, penulis melakukan pengujian kinerja untuk aplikasi Eventeer pada jam-jam sibuk atau jam kantor. Pengujian dilakukan kurang lebih 8 kali. Kemudian hasil dari pengujian kinerja aplikasi Eventeer disimpan dan dihitung nilai rata-rata untuk tingkat kesuksesannya.

B. Produk/Jasa yang Dihasilkan

Hasil karya yang dibuat selama magang yaitu dokumen *Entity Relationship Diagram*, *Test Plan Document*, *Activity Diagram*, dan *performance testing*. Terdapat total sembilan *s* yang dilalui oleh penulis, diantaranya terdapat enam *sprint* untuk *project* Ideabox dan tiga *sprint* untuk *project* Eventeer. Untuk detail dari *project-project* yang telah dikerjakan oleh penulis sebagai *system analyst* dan *quality assurance* dapat dilihat pada Tabel I dan Tabel II.

TABEL I
HASIL PEKERJAAN SYSTEM ANALYST

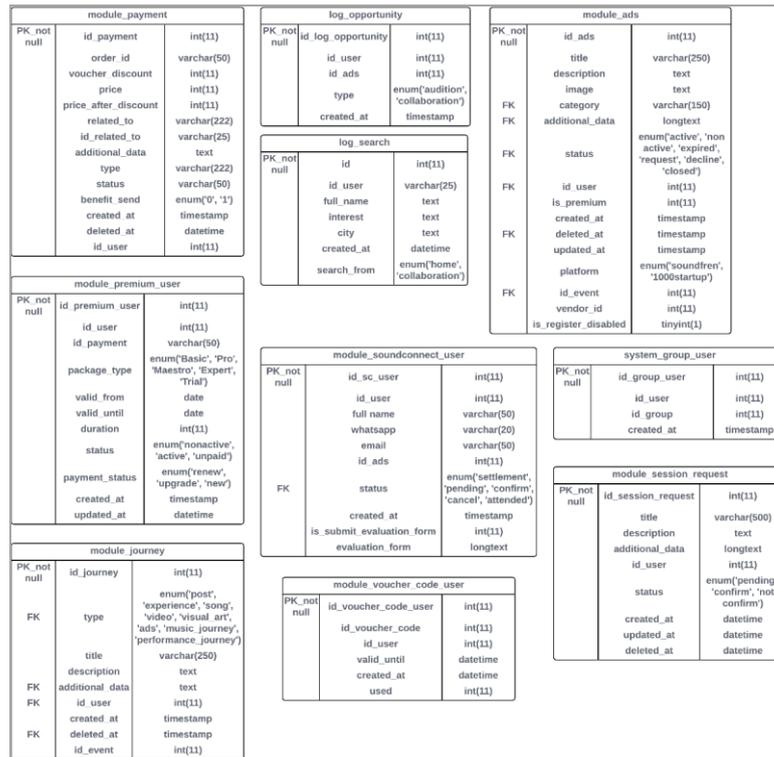
No	Nama Project	Deskripsi
1	Eventeer	Update dokumen <i>Entity Relationship Diagram</i> dan membuat <i>activity diagram</i> dari fitur <i>Finding Founder</i> .
2	Ideabox	Membuat <i>Entity Relationship Diagram</i> dengan menerapkan <i>snake case</i> sebagai standar penulisan pada <i>database</i> .

TABEL II
HASIL PEKERJAAN QUALITY ASSURANCE

No	Nama Project	Deskripsi
1	Eventeer	Membuat <i>test case</i> dan melakukan pengujian fitur <i>Community Dashboard</i> serta melakukan <i>performance testing</i> .
2	Ideabox	Membuat <i>test case</i> dan melakukan pengujian fitur yang dikembangkan pada keseluruhan aplikasi Ideabox.

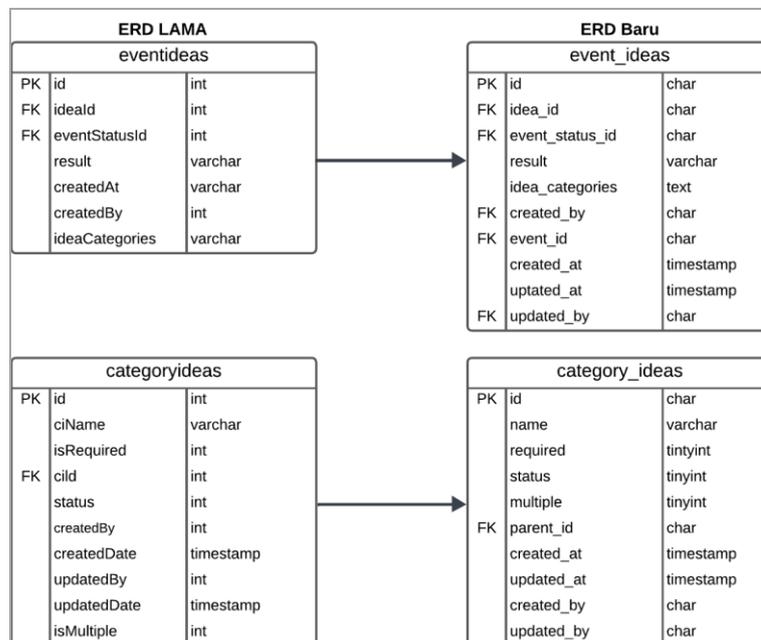
1) *Entity Relationship Diagram* Eventeer: Kondisi saat ini *Entity Relationship Diagram* Eventeer sudah terdapat standarisasi untuk penamaan *identifier* menggunakan *snake case* tetapi masih kurang lengkap, maka dari itu penulis pemutahiran untuk *Entity Relationship Diagram* Eventeer sesuai dengan kondisi yang terkini. Penulis menambahkan beberapa tabel baru yang belum ada pada *Entity Relationship Diagram* Eventeer yang lama. *Entity Relationship Diagram* pada aplikasi Eventeer termasuk kedalam bentuk *Entity Relationship Diagram* fisik untuk menggambarkan struktur data pada sistem basis data. Arsitektur pada aplikasi Eventeer menggunakan MySQL sebagai *database* dalam menyimpan dan mengelola data. Terdapat total 68 table yang merepresentasikan berbagai entitas dan fitur pada aplikasi Eventeer.

Penulis melakukan pemutahiran dokumen *Entity Relationship Diagram* untuk aplikasi Eventeer dengan memastikan kesesuaian penamaan *identifier* baik nama tabel dan nama kolom serta tipe data sudah sesuai dengan *database*. Dengan melakukan pemutahiran dokumen *Entity Relationship Diagram* membantu user mengetahui kondisi *database* Eventeer mulai dari penamaan tabel, kolom dan tipe data tanpa harus melihat pada *database* secara langsung.



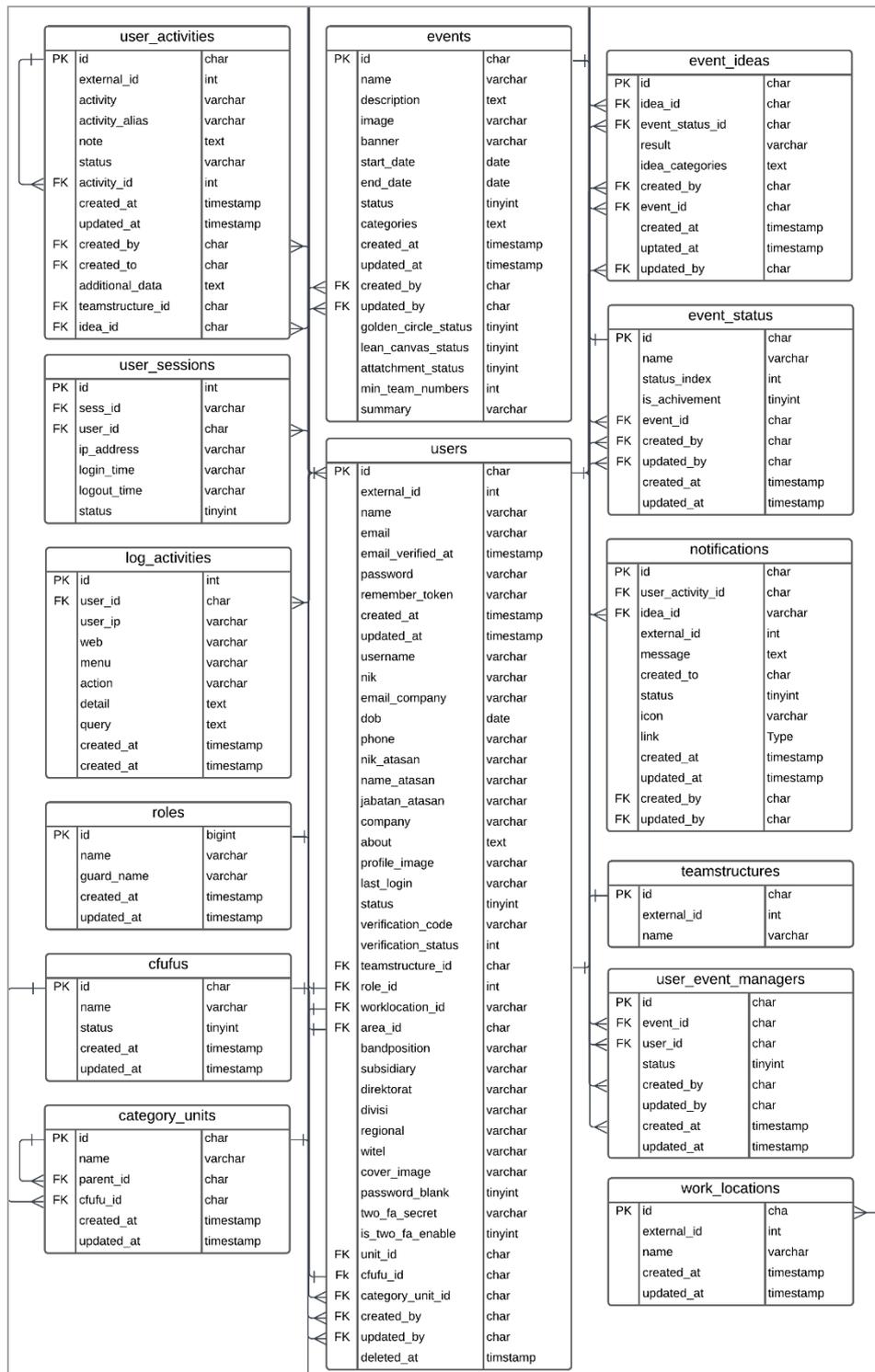
Gambar 2. Penambahan Tabel pada ERD Eventeer

2) *Entity Relationship Diagram* Ideabox: Kondisi *Entity Relationship Diagram* yang lama tidak memiliki standar penamaan untuk *identifier* seperti nama tabel dan nama kolom. Karena tidak memiliki penamaan *identifier* pada *Entity Relationship Diagram* aplikasi Ideabox yang lama, menjadikan banyak sekali perbedaan gaya penulisan *identifier* yang menghambat developer dalam mengembangkan aplikasi. Penulis menerapkan standar penulisan *identifier* menggunakan *snake case*. Standar penamaan *identifier* menggunakan *snake case* dengan menggunakan garis bawah (_) untuk memisahkan antar kata yang digambarkan pada Gambar 3.



Gambar 3. Perbandingan ERD Ideabox Lama dan ERD Baru

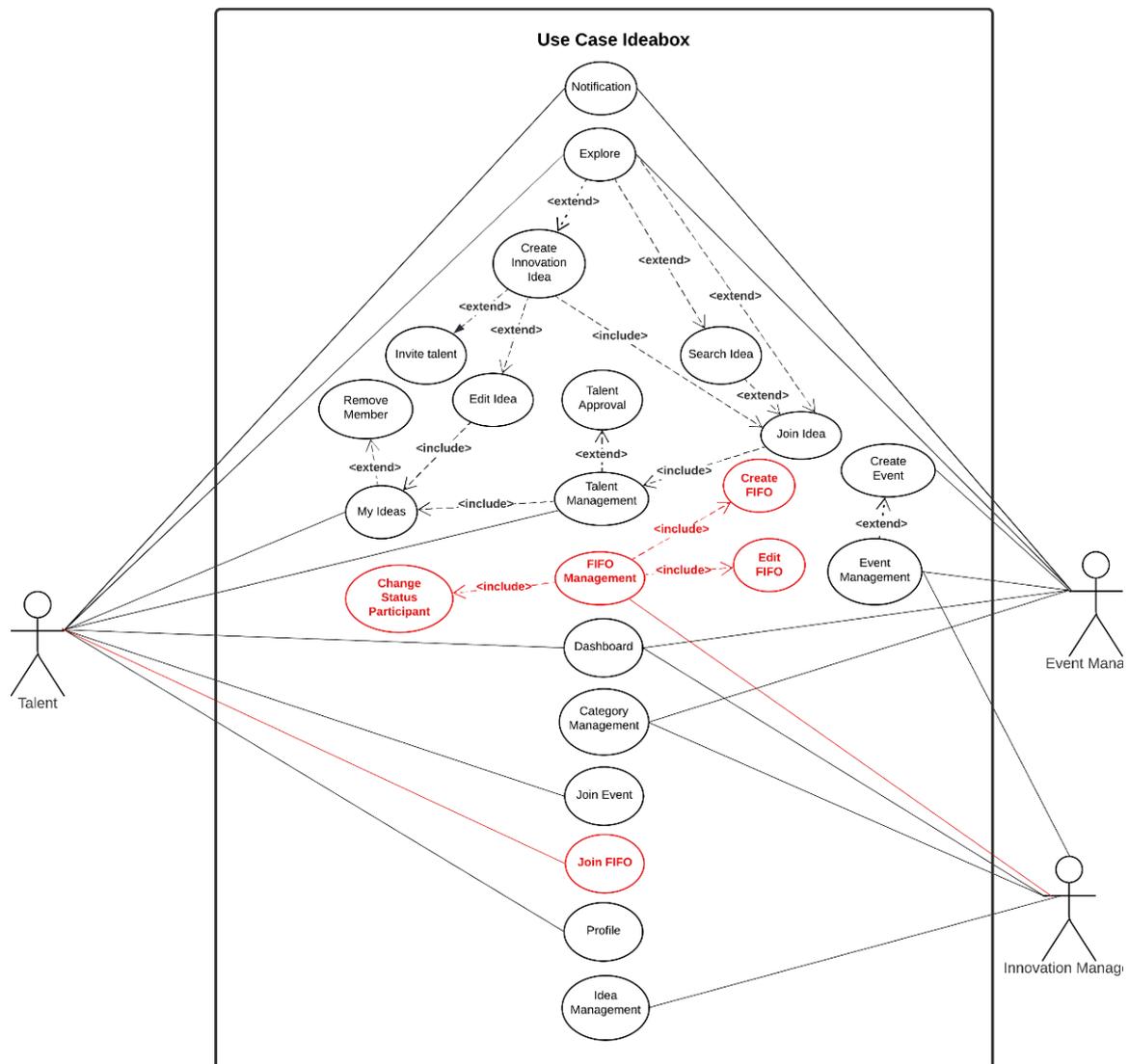
langsung. Arsitektur pada aplikasi Ideabox menggunakan MySQL sebagai database dalam menyimpan dan mengelola data. Terdapat 29 table yang merepresentasikan berbagai entitas dan fitur pada aplikasi ideabox. Bentuk Entity Relationship Diagram pada aplikasi Ideabox termasuk dalam Entity Relationship Diagram Fisik.



Gambar 5. ERD Ideabox B

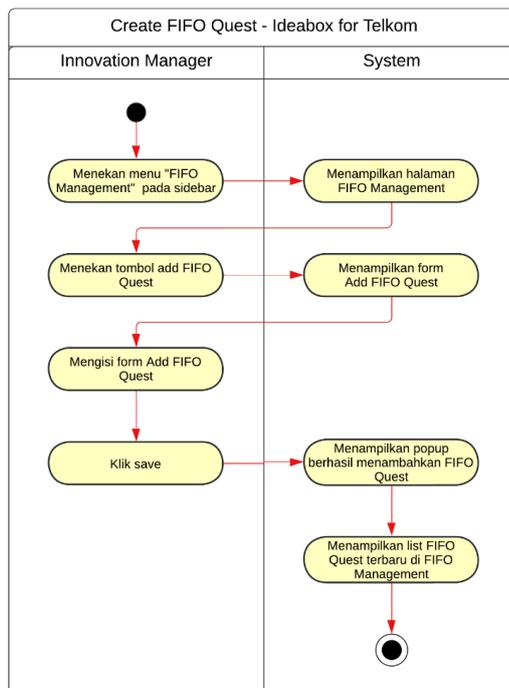
3) *Penambahan Dokumentasi Finding Founder*: Pada aplikasi Ideabox terdapat fitur baru yaitu fitur *Finding Founder* atau yang biasa disingkat menjadi FIFO. *Finding Founder* merupakan fitur terbaru pada *project* Ideabox yang dirancang untuk mempermudah *Innovation Manager* dalam menemukan *founder* dari *idea* yang telah terpilih, sehingga bisa melakukan pendekatan untuk mengembangkan *idea* yang ada. Terdapat fitur *FIFO Management* untuk mengelola *FIFO* yang didalamnya terdapat fitur *create FIFO* dan *edit FIFO*, Serta terdapat fitur *join FIFO* untuk para *talent* bisa berpartisipasi pada pencarian *founder* yang dilakukan oleh *Innovation Manager*. Dikarenakan fitur *Finding Founder* merupakan fitur yang baru, maka penulis melakukan *update* dokumentasi untuk *use case diagram* dan juga membuat *activity diagram* untuk aktivitas *join*, *create*, *edit* dan perubahan status partisipan untuk *Finding Founder*.

Aplikasi Ideabox sudah terdokumentasi dengan baik, salah satunya terdapat dokumentasi *use case diagram* yang membantu dalam penjelasan fitur-fitur yang ada pada aplikasi Ideabox. Gambar 6 merupakan *use case diagram* dari aplikasi ideabox, dimana penulis melakukan *update* dokumentasi *use case diagram* untuk menambahkan dokumentasi terkait fitur terbaru yaitu *Finding Founder* yang diberikan tanda berwarna merah.



Gambar 6. Use Case Diagram Ideabox

4) *Activity Diagram Create FIFO*: Keseluruhan proses *Finding Founder* dikelola oleh *Innovation Manager*, salah satunya yaitu membuat *FIFO Quest*. Proses dimulai dari *Innovation Manager* menekan menu “*FIFO Management*” yang ada pada *sidebar* yang akan mengalihkan ke halaman *FIFO Management*. Pada halaman *FIFO Management*, *Innovation Manager* akan menekan tombol *Add FIFO Quest* dan akan berpindah ke halaman *create fifo quest* yang berisikan form yang perlu diisi. Dilanjutkan dengan *Innovation Manager* mengisi keseluruhan form dengan data yang valid dan menekan tombol *save*. Kemudian sistem akan menampilkan *pop-up* yang menandakan sudah berhasil menambahkan *fifo quest* dan langsung pindah halaman ke *FIFO Management* dengan kondisi data *fifo quest* muncul pada tabel *fifo quest list*. Keseluruhan aktivitas digambarkan pada *activity diagram* seperti pada Gambar 7.



Gambar 7. Activity Diagram Create FIFO

5) *Performance Testing Eventeer*: *Performance testing* dilakukan dengan memanfaatkan *tools* yang bernama Taurus. *Performance testing* dilakukan dengan memanfaatkan *tools* taurus untuk pengujian *API Login*, *Get Event List*, *Get Timeline*, *Get List Ads*. Penulis membuat konfigurasi menetapkan jumlah *user* aktif diwaktu yang bersamaan yaitu 250 *user* dari jumlah *user* SPBE yaitu 500 *user*. Penulis menetapkan waktu untuk mencapai jumlah *user* keseluruhan aktif dalam waktu 5 menit dan menetapkan setiap *user* akan aktif selama 10 menit. *Performance testing* dilakukan oleh penulis pada jam kantor atau waktu-waktu sibuk. Adapun penulis melakukan beberapa kali pengujian untuk beberapa waktu seperti:

- Pukul 09.20 WIB
- Pukul 10.00 WIB
- Pukul 11.00 WIB
- Pukul 13.30 WIB
- Pukul 14.00 WIB
- Pukul 15.00 WIB
- Pukul 15.30 WIB
- Pukul 16.15 WIB

Untuk hasil pengujian yang dilakukan pada jam sibuk, berikut merupakan perhitungan yang menjadi indikator penilaian kelayakan dari kinerja aplikasi Eventeer. Penulis memastikan untuk setiap *API* tingkat keberhasilannya tidak dibawah 90% yang merupakan angka standar kesuksesan sesuai SOP yang berlaku. Dari beberapa hasil pengujian, penulis membuat rata-rata tingkat keberhasilannya untuk memastikan tingkat keberhasilan tidak dibawah 90%.

Dari hasil *performance testing* untuk aplikasi eventeer, keseluruhan nilai rata-rata mulai dari *API login*, *get timeline*, *get list ads*, *get event list* masih didapati berada diatas 90% yang merupakan persentase dari kelayakan performa aplikasi. Dapat dikatakan untuk aplikasi Eventeer sudah siap untuk digunakan oleh client SPBE. Adapun detail dari masing-masing nilai rata-rata seperti yang bisa dilihat pada Tabel III.

TABEL III
NILAI RATA-RATA HASIL PENGUJIAN API

API	Nilai Rata- RataTingkat Kesuksesan (%)
Login	98.63
Get Event List	98.42
Get Timeline	99.92
Get List Ads	99.40

6) *Pengujian Aplikasi Eventeer*: Pada tahap pengujian *Dashboard Community Manager* aplikasi Eventeer, penulis membuat total 102 *test case* dengan menerapkan skenario pengujian positif, negatif, dan destruktif. Namun, dalam penelitian ini, hanya sebagian *test case* yang ditampilkan karena data *test case* bersifat konfidensial.

Media learning merupakan fitur untuk pengelolaan konten konten untuk pembelajaran seperti video dan artikel. Pengujian halaman *media learning* dilakukan dengan memastikan halaman bisa dibuka, setiap fitur seperti fitur *search*, filter dan *action button* dapat berjalan sesuai dengan rancangan yang sudah dibuat sebelumnya.

TABEL IV
TEST CASE MEDIA LEARNING

\	Uji Kasus	Langkah-Langkah	Hasil Yang Diharapkan
1	Membuka halaman <i>Media learning</i>	Login dengan data yang valid Menekan menu “ <i>Event Activity</i> ” pada <i>sidebar</i> Menekan tab “ <i>Media Learning</i> ”	Halaman <i>Media learning</i> dapat diakses dan menampilkan <i>list data media learning</i> .
2	Mencari <i>media learning</i> dengan data valid pada fitur <i>search media learning</i>	Login dengan data yang valid Menekan menu “ <i>Event Activity</i> ” pada <i>sidebar</i> Menekan tab “ <i>Media Learning</i> ” Menekan <i>searchbar</i> Input salah satu nama <i>media learning</i> yang valid	Menampilkan data berdasarkan keyword pada tabel.
3	Mencari <i>media learning</i> dengan data tidak valid pada fitur <i>search media learning</i>	Login dengan data yang valid Menekan menu “ <i>Event Activity</i> ” pada <i>sidebar</i> Menekan tab “ <i>Media Learning</i> ” Menekan <i>searchbar</i> Input salah satu nama <i>media learning</i> yang tidak valid	Tidak muncul data apapun pada tabel.
4	Filtering data menggunakan fitur filter	Login dengan data yang valid Menekan menu “ <i>Event Activity</i> ” pada <i>sidebar</i> Menekan tab “ <i>Media Learning</i> ” Menekan button “ <i>Filter</i> ” Melakukan filtering data berdasarkan <i>category/publish date/status</i> dan menekan button “ <i>Apply</i> ”	Data pada tabel <i>media learning list</i> muncul berdasarkan filtering data.
5	memeriksa functional <i>action button</i>	Login dengan data yang valid Menekan menu “ <i>Event Activity</i> ” pada <i>sidebar</i> Menekan tab “ <i>Media Learning</i> ” Menekan dots horizontal pada kolom “ <i>Action</i> ” Menekan <i>action button</i> “ <i>view</i> ” Menekan <i>action button</i> “ <i>Edit</i> ”	Muncul <i>drop down view, edit, unpublish/publish, delete</i> . Direct ke halaman <i>view detail media learning</i> . Direct ke halaman <i>edit media learning</i> .

Halaman new video detail merupakan halaman untuk membuat *media learning video* yang berisikan form dan terdapat *section preview* yang akan menampilkan *preview* dari *video media learning*. Dimana *user* dapat *upload video file* pembelajaran dengan format .MP4, file *thumbnail* dengan format .jpg/.png/.jpeg. Keseluruhan form wajib untuk *user* isi untuk dapat membuat *video media learning*. Pengujian halaman *create video media learning* dilakukan dengan memastikan

halaman bisa diakses, bisa menampilkan video dan data pada *section preview*, bisa membuat data baru *video media learning* menggunakan data yang valid, dan memastikan validasi setiap form upload. Adapun pengujian halaman *create media learning* dengan menjalankan *test case* yang sudah dibuat sebelumnya.

TABEL V
TEST CASE CREATE NEW VIDEO DETAIL

s	Uji Kasus	Langkah-Langkah	Hasil Yang Diharapkan
1	Membuka halaman <i>new video detail</i>	<i>Pre Condition:</i> sudah login dengan data yang valid dan berada di halaman <i>media learning</i> Menekan button " <i>create media learning</i> " Memilih <i>icon video</i>	Bisa mengakses halaman <i>create new video</i> dengan kondisi form kosong
2	Membuat <i>video media learning</i> dengan data yang valid dengan skema <i>publish now</i> .	<i>Pre Condition:</i> sudah berada pada halaman <i>new video detail</i> . Upload file .MP4 (<i>max size</i> 200MB) pada <i>field upload video file</i> Upload <i>thumbnail file</i> .PNG (<i>max size</i> 3MB) pada <i>field upload thumbnail file</i> Input <i>title media learning</i> Input <i>description media learning</i> Menekan button " <i>Publish now</i> " Memilih <i>taget audience</i> Memilih <i>category</i> Menekan button " <i>publish</i> "	Berhasil membuat <i>video media learning</i> .
3	Cek validasi <i>upload file video</i> wajib format .MP4 dan tidak lebih dari 200MB	<i>Pre Condition:</i> sudah berada pada halaman <i>create video media learning</i> . Upload file .MP4 > 200MB pada <i>field upload video file</i> . Upload file .pdf, .xlsx, .docs, .jpg pada <i>field upload video file</i> .	Upload <i>video field</i> berubah warna merah, muncul informasi " <i>Please upload a file size no longer than the limit</i> " dan muncul informasi wajib <i>upload</i> format file .MP4

7) *Pengujian Aplikasi Ideabox For Telkom Group:* Pada tahap pengujian aplikasi *Ideabox Revamp*, penulis membuat total 263 *test case* dengan menerapkan skenario pengujian positif, negatif, dan destruktif. Namun, dalam laporan penelitian ini, hanya sebagian *test case* yang ditampilkan karena data *test case* bersifat konfidensial.

Halaman login dari aplikasi *Ideabox*, dilakukan pengujian *authentication*, dengan *user* dapat login dengan data yang valid dengan melakukan skenario-skenario *positive*, *negative* dan *destructive*.

TABEL VI
TEST CASE LOGIN

No	Uji Kasus	Langkah-Langkah	Hasil Yang Diharapkan
1	Mengakses halaman <i>Login</i> .	Membuka <i>url ideabox development</i> Menekan tab " <i>Login General</i> "	Bisa mengakses halaman <i>login ideabox</i> dan form <i>login</i> pada tab <i>Login Telkom</i> dan <i>Login General</i> .
2	<i>Login</i> dengan data yang valid untuk skema <i>login general</i> .	<i>Pre Condition:</i> sudah berada pada halaman <i>Login Ideabox</i> Menekan tab " <i>Login General</i> " Input <i>email</i> dengan data valid. Input <i>password</i> dengan data valid Menekan button " <i>Log In</i> " Input <i>OTP</i> Menekan button " <i>Confirm</i> "	Muncul <i>pop-up</i> untuk <i>input OTP</i> dan berhasil <i>login</i> .

No	Uji Kasus	Langkah-Langkah	Hasil Yang Diharapkan
3	Login dengan data yang tidak valid untuk skema login general.	<i>Pre Condition:</i> sudah berada pada halaman <i>Login Ideabox</i> Menekan tab “Login General” Input invalid data Email: TestingAja@gmail.com Password: PasswordSalah	Muncul <i>pop-up message</i> “The password you entered is invalid. Please try again”.

Sebagai *talent*, dapat mengumpulkan ide inovasi pada aplikasi Ideabox. Dimana *talent* melakukan memasukan nama ide inovasi, memilih kategori ide, dan memberikan deskripsi, *story behind*, *lean canvas*, dan memasukan file-file pendukung ide tersebut. Dilakukan pengujian fitur *create idea* dengan memastikan halaman dapat diakses, terdapat validasi dari *input user*, memastikan seluruh *button* atau *tab* berfungsi sesuai rancangan yang ada.

TABEL VII
TEST CASE CREATE IDEA

No	Uji Kasus	Langkah-Langkah	Hasil Yang Diharapkan
1	Mengakses halaman <i>create idea</i>	Login dengan data yang valid Menekan <i>button create idea</i>	Berhasil mengakses halaman <i>create idea</i> dan menampilkan form <i>idea description</i> .
2	Mencoba keseluruhan tab pada halaman <i>create idea</i> .	<i>Pre Condition:</i> sudah berada pada halaman <i>create idea</i> . Menekan <i>tab story behind</i> Menekan <i>tab lean canvas</i> Menekan <i>tab teams</i> Menekan <i>tab additional attachment</i> .	Menampilkan form sesuai dengan <i>tab</i> yang dipilih.
3	Membuat ide dengan cara langsung menekan <i>button “next”</i>	<i>Pre Condition:</i> sudah berada pada halaman <i>create idea</i> . Menekan <i>button next</i> tanpa mengisi form terlebih dahulu.	Muncul informasi untuk wajib mengisi form.

8) *Hasil Pengujian Aplikasi:* Setelah melalui pengujian aplikasi untuk pengujian dari segi fungsionalitas dan kinerja dengan membandingkan hasil pengembangan dengan rancangan yang ada, kemudian dilakukan perbaikan oleh *developer* ketika didapati terdapat fitur yang perlu dilakukan penyesuaian, didapati hasil pengujian aplikasi ideabox *revamp* dan *dashboard admin* untuk aplikasi eventeer dapat dikatakan sudah siap untuk *push* ke *environment production*. Tabel VIII merupakan hasil dari pengujian aplikasi dan status dari tiap halamannya. Secara keseluruhan untuk aplikasi atau fitur yang dikembangkan, sudah siap untuk digunakan oleh *end user*. Dalam artian seluruh fitur atau aplikasi yang dikembangkan baik dari aplikasi Ideabox dan Eventeer sudah sesuai dengan *requirement* atau rancangan yang ada dan siap untuk *go live*.

TABEL VIII
HASIL PENGUJIAN APLIKASI

No	Aplikasi	Halaman	Status
1	Eventeer	Media Learning List	Pass Development
2		New Video Detail	Pass Development
3		Create New Article	Pass Development
4		Article Detail	Pass Development
5		Video Detail	Pass Development
6		Edit Article	Pass Development
7		Edit Video Detail	Pass Development
8		Announcement Banner List	Pass Development
9		Create New Announcement Banner	Pass Development
10		Edit Announcement Banner	Pass Development
11		Announcement Banner Detail	Pass Development
12		Push Notification List	Pass Development
13		Create New Notification	Pass Development

No	Aplikasi	Halaman	Status	
14		<i>Push Notification Detail</i>	<i>Pass Development</i>	
15		<i>Email Broadcast List</i>	<i>Pass Development</i>	
16		<i>Create Email Broadcast</i>	<i>Pass Development</i>	
17		<i>Email Broadcast Detail</i>	<i>Pass Development</i>	
18		<i>Feed Topic Lis</i>	<i>Pass Development</i>	
19		<i>Create New Feed Topic</i>	<i>Pass Development</i>	
20		<i>Edit Feed Topic</i>	<i>Pass Development</i>	
21		<i>Community Settings</i>	<i>Pass Development</i>	
22		Ideabox	<i>Login</i>	<i>Pass Development</i>
23			<i>Sign Up</i>	<i>Pass Development</i>
24			<i>Explore</i>	<i>Pass Development</i>
25			<i>Create Idea</i>	<i>Pass Development</i>
26			<i>List Ideas</i>	<i>Pass Development</i>
27			<i>Talent Approval</i>	<i>Pass Development</i>
28			<i>Dashboard</i>	<i>Pass Development</i>
29			<i>Finding Founder Quest List</i>	<i>Pass Development</i>
30			<i>Finding Founder Quest Detail</i>	<i>Pass Development</i>
31			<i>FIFO Management</i>	<i>Pass Development</i>
32			<i>Create FIFO Quest</i>	<i>Pass Development</i>
33			<i>Edit FIFO Quest</i>	<i>Pass Development</i>
34			<i>Idea Management</i>	<i>Pass Development</i>
35	<i>Edit Idea</i>		<i>Pass Development</i>	
36	<i>Events</i>		<i>Pass Development</i>	
37	<i>Event Detail</i>		<i>Pass Development</i>	
38	<i>Event Management</i>		<i>Pass Development</i>	
39	<i>Create Event</i>		<i>Pass Development</i>	
40	<i>Edit Event</i>		<i>Pass Development</i>	
41	<i>Category Management</i>		<i>Pass Development</i>	
42	<i>Announcement Management</i>		<i>Pass Development</i>	
43	<i>User Management</i>		<i>Pass Development</i>	
44	<i>Profile</i>		<i>Pass Development</i>	

V. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil dari tugas akhir yang telah dibuat penulis maka dapat disimpulkan bahwa dalam *Entity Relationship Diagram* harus dapat menggambarkan struktur data pada sistem basis data dengan cara menuliskan tipe data pada setiap atributnya. Dengan kata lain apabila seseorang ingin mengetahui struktur data pada sebuah sistem basis data cukup dengan membaca dokumen *Entity Relationship Diagram*.

Pengujian aplikasi dapat berupa pengujian fungsional maupun kinerja untuk memastikan kesiapan aplikasi untuk bisa digunakan oleh *end user*. Pengujian fungsional dilakukan dengan memastikan aplikasi yang sudah selesai dikembangkan oleh *developer* sudah sesuai dengan rancangan yang sudah dibuat sebelumnya. Pengujian kinerja aplikasi dilakukan untuk memastikan aplikasi siap untuk digunakan oleh *end user* dengan jumlah yang sesuai dengan *requirement*. Dari hasil pengujian yang sudah dilakukan pengujian dari segi fungsional dan kinerja oleh penulis serta penyesuaian yang dilakukan oleh *developer* dapat disimpulkan untuk aplikasi *Ideabox Revamp* dan fitur *Dashboard Admin* untuk aplikasi *Eventeer* sudah sesuai dengan *requirement* atau rancangan yang ada dan siap untuk *go live*.

B. Saran

Sebelum pemegang terjun secara langsung pada proyek-proyek yang sedang berjalan, ada baiknya diberikan pelatihan terlebih dahulu. Pelatihan dapat diberikan kepada pemegang sebagai bekal pemegang sebelum nantinya terjun secara langsung ke *project* yang ada. Adapun pelatihannya dapat berupa memberi pengetahuan terkait standar aplikasi yang ada pada *Digital Amoeba* dan pelatihan terkait cara pengujian aplikasi yang baik untuk memastikan keseluruhan aplikasi sudah teruji. Serta memberikan pelatihan untuk menstandarisasi pemahaman sebagai *System Analyst* yang baik. Dengan adanya pelatihan mempermudah pemegang dalam menjalankan tugas dan tanggung jawab sebagai *System Analyst* maupun *Quality Assurance*.

UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa atas limpahan berkat dan rahmat-Nya, sehingga penulis dapat menyelesaikan laporan penelitian ini. Pada kesempatan ini pemegang ingin menyampaikan rasa terima kasih kepada:

1. Tuhan Yang Maha Esa yang telah memberikan rahmat, berkah, dan hidayah-Nya
2. Orang tua pemegang yang selalu mendoakan dan memberikan dukungan baik secara moril maupun materil.
3. PT Telekomunikasi Indonesia, Tbk yang telah memberikan izin dan mempercayakan kepada pemegang agar dapat melakukan kegiatan magang.
4. Ibu Diana Trivena Yulianti, S.Kom.,M.T, selaku pembimbing utama yang telah memberikan semangat, masukkan dan bimbingan kepada penulis selama proses pengerjaan penelitian ini.
5. Bapak Abul Nizam Faisal selaku mentor selama magang di Digital Amoeba.
6. Rekan-rekan magang atas semangat dan dukungannya untuk menyelesaikan penelitian ini.

DAFTAR PUSTAKA

- [1] Telkom Group, "IDEABOX Telkom Group." Accessed: Oct. 14, 2024. [Online]. Available: <https://ideabox.telkom.co.id/>
- [2] Eventeer, "Eventeer Customer Success Playbook Preface," 2022.
- [3] Telkom Group, "Sekilas tentang Telkom Indonesia." Accessed: Nov. 10, 2024. [Online]. Available: https://www.telkom.co.id/sites/profil-telkom/id_ID/page/profil-dan-riwayat-singkat-22
- [4] Amoeba Management, *Journey Digital Amoeba 2021*. 2021.
- [5] A. Gunawan, S. Ningsih, and D. A. Lantana, *PENGANTAR BASIS DATA*. 2023. [Online]. Available: www.penerbitlitnus.co.id
- [6] R. Setiawan, "Memahami ERD, Model Data, dan Komponennya." Accessed: Nov. 26, 2024. [Online]. Available: <https://www.dicoding.com/blog/memahami-erd/>
- [7] O. Fitria, N. Hasanah, M. Pd, and R. S. Untari, *REKAYASA PERANGKAT LUNAK*. 2020.
- [8] I. R. Mukhlis, "BUKU AJAR LOGIKA & ALGORITMA," 2023. [Online]. Available: <https://www.researchgate.net/publication/372826383>
- [9] "IEEE Standard for Software Quality Assurance Processes," Mar. 27, 2014, *IEEE, Piscataway, NJ, USA*. doi: 10.1109/IEEESTD.2014.6835311.
- [10] "APA ITU SOFTWARE TESTING? BERIKUT PENGERTIAN, JENIS, CARA KERJA DAN MANFAATNYA." Accessed: May 29, 2024. [Online]. Available: <https://cdcbpsdmi.kemenperin.go.id/article/detail/apa-itu-software-testing-berikut-pengertian-jenis-cara-kerja-dan-manfaatnya>
- [11] Romeo, *TESTING DAN IMPLEMENTASI SISTEM*. 2003.