BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Objek penelitian merupakan tempat dimana penelitian tersebut akan di lakukan, adapun tempat penelitian yang penulis lakukan adalah PT. eBdesk Teknologi.

3.1.1. Sejerah Singkat Perusahaan

PT eBdesk didirikan pada tahun 1998 sebagai perusahaan perangkat lunak untuk mengembangkan produk portal perusahaan berdasarkan teknologi internet. Selain menerima permintaan aplikasi dari konsumen perusahaan ini juga membuat beberapa produk seperti *Expander Corporate Portal*, *Expedition Workflow*, *Encylo Document Management*, *expert Knowledge Management*. Ebdesk tetap fokus untuk mengembangkan produk-produk yang berkelanjutan sesuai dengan perubahan teknologi dan tren. Ebdesk adalah salah perusahaan pertama yang mengembangkan corporate portal, yang mana corporate portal itu sendiri adalah sebuah konsep baru ketika eBdesk didirikan.

Ebdesk Corporate Portal mengintegrasikan berbagai informasi dan aplikasi, dari perusahaan di dalam atau di luar, di antarmuka web pribadi tunggal. Ini adalah *platform* untuk menyebarkan sepenuhnya portal web interface yang memberikan konek simulus antara, karyawan, manajer, *pemilik*, mitra, dan pelanggan. Pada mulanya semua produk Ebdesk dikembangkan dengan bahasa C /

C++ untuk mencapai kinerja tinggi, ketahanan dan kompatibilitas lintas *platform*. Namun dengan bahasa Java yang memberikan banyak kelebihan diantaranya ukurannya kecil, sederhana, dan portable (dapat dipindah-pindahkan diantara bermacam platform dan sistem operasi), sehingga saat ini produk Ebdesk dikembangkan dengan bahasa Java. Ebdesk adalah anggota Asia Software Alliance (SAA), merupakan koalisi dari pemenang-pemenang penghargaan perusahaan software dan jasa. SAA adalah aliansi berbasis ICT global pertama dari jenisnya yang menawarkan solusi yang komprehensif end-to-end dan layanan untuk berbagai segmen industri.

3.1.2. Visi dan Misi Perusahaan

Setiap perusahaan sudah pasti mempunyai visi dan misi yang berisikan maksud dan tujuan yang ingin dicapai perusahaan baik itu untuk jangka pendek, maupun jangka panjang , berikut visi dan misi PT. eBdesk Teknologi :

Visi :

Menjadi perusahaan yang handal dalam melakukan eletronik bisnis.

Misi :

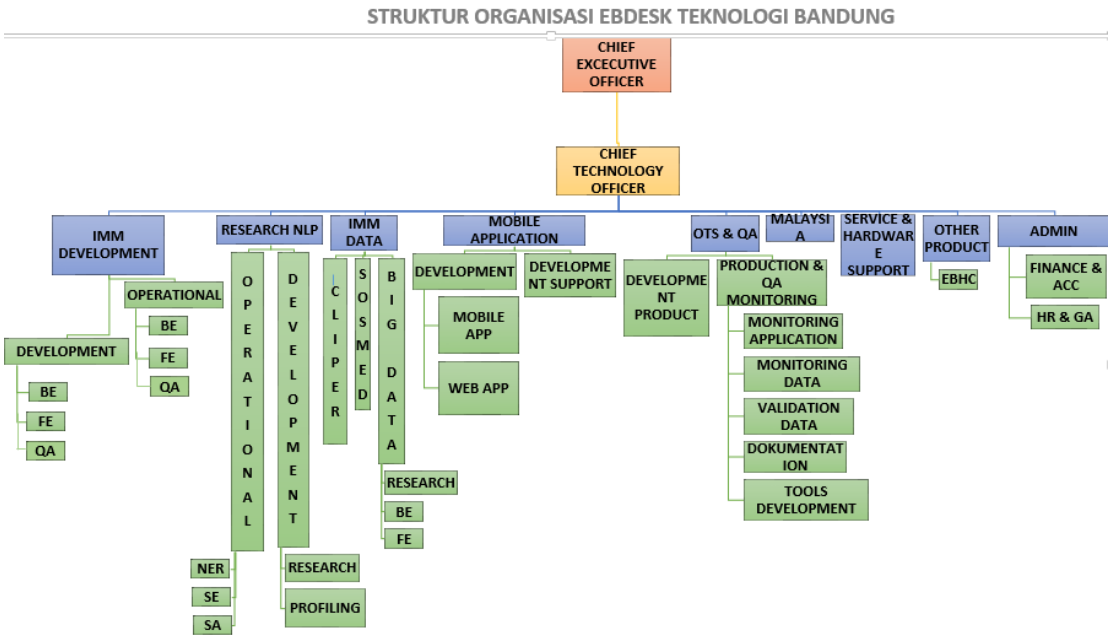
Untuk mencapai visi tersebut, PT. eBdesk Teknologi telah menerapkan misi sebagai berikut :

1. Membangun *platform portal* sebagai dasar dari semua produk, layanan dan solusi.

2. Memberikan produk terintegrasi, layanan dan solusi untuk memungkinkan perusahaan melakukan e-bisnis.
3. Pelanggan harus mendapatkan nilai tambah dari investasi mereka.
4. Menjaga hubungan yang berkesinambungan dengan pelanggan.
5. Mencapai posisi sebagai pemain global diseluruh dunia.

3.1.3. Struktur Organisasi PT eBdesk Teknologi

Struktur organisasi perusahaan merupakan bangunan fungsi bagian-bagian manajemen yang tersusun dari suatu kesatuan hubungan yang menunjukan tingkatan fungsi, tugas, wewenang dan tanggung jawab dalam manajemen perusahaan. Penerapan struktur organisasi di lingkungan PT eBdesk berbentuk jabatan dan, dimana wewenang dari pimpinan dilimpahkan kepada satuan semua bidang staf pekerjaan bantuan.Struktur organisasi PT eBdesk sebagai berikut :



Gambar 1.1 Struktur Organisasi PT. eBdesk Teknologi

3.1.4 Deskripsi Kerja

a. CEO (*Chief Executive Offiser*) mempunyai tugas memimpin perusahaan dan bertanggung jawab untuk kestabilan perusahaan.
b. CTO (*Chief Technology Officer*) mempunyai tugas mengupdate perkembangan teknologi lalu mengembangkan untuk digunakan perusahaan, menjadi koordinator pengelolaan pengembangan produk.
c. BE (*Back End*) mempunyai tugas mengacu pada program dan skrip yang bekerja pada server di balik layar untuk membuat halaman aplikasi yang dinamis dan interaktif
d. FE (*Front End*) mempunyai tugas mengacu pada setiap aspek dari proses desain yang muncul dalam atau berhubungan langsung dengan browser yang berkaitan dengan desain antarmuka applikasi.
e. QA (*Quality Assurance*) mempunyai tugas untuk menjamin produk yang dihasilkan mempunyai kualitas yang bagus.
f. Reaserch NLP (Neuro-linguistic Programming) mempunyai tugas membuat mesin yang mampu mengerti dan memahami makna bahasa manusia lalu memberikan respon yang sesuai.
g. NER (*Named-entity Recognition*) mempunyai tugas mengidentifikasikan kata benda menjadi tiga kelas, yakni nama orang, lokasi, dan organisasi.
h. SE (*Statement Extraction*) mempunyai tugas mengidentifikasi pernyataan yang diambil dari social media sesuai dengan person.
i. SA (*Sentiment Analyst*) mempunyai tugas menentukan sentimen negative, positif na netral dari setiap pernyataan untuk dianalisis.

j. *Mobile* APP mempunya tugas membuat atau mengembangkan produk dalam fomat *mobile*

k. *Profiling* mempunyai tugas menganalisa suatu kasus atau perilaku manusia yang nantinya akan dikelompokan.

l. IMM Clipper mempunyai tugas menyajikan data dari news untuk diolah sesuai kebutuhan

m. IMM Sosmed mempunyai tugas menyajikan data dari sosial media untuk diolah sesuai kebutuhan

n. IMM Big Data mempunya tugas menyajikan data dalam jumlah banyak untuk diolah sesuai kebutuhan

o. OTS (*Operational Technical Support*) mempunyai tugas memperbaiki suatu sistem.

p. *Monitoring Application* mempunyai tugas memonitor aplikasi agar terhindar dari sistem error/ human error.

q. *Monitoring Data* mempunyai tugas memastikan data dari news dan sosial media masuk kedalam sistem secara real time.

r. *Validation Data* mempunyai tugas memastikan data yang masuk kedalam sistem sesuai dengan permintaan.

s. *Documentation* mempunyai tugas mengumpulkan, menyusun, menyelidiki/meneliti dan mengolah serta memelihara bahan, guna menyiapkan informasi yang bermanfaat.

t. *Tools Development* mempunyai tugas membuat suatu aplikasi guna mempermudah divisi lain.

u. *Service & Hardware Support* yaitu dimana produk diperbaiki, dilakukan pengecekan dan pemeliharaan produk agar lebih baik lagi kerjanya.

v. *EBHC (eB Hardcopy)* mempunyai tugas memindai koran untuk keperluan sistem.

w. HR (*Human Resource*) mempunyai tugas untuk menjaga kesimbangan hak dan kewajiban antara karyawan dengan perusahaan.

x. *Finance* mempunyai tugas untuk mencatat setiap transaksi perusahaan dan akan disajikan didalam laporan keuangan tiap periode waktu tertentu.

3.2. Metode Penelitian

Pada sub bab ini penulis akan menerangkan metode penelitian yang akan digunakan dalam penelitian ini. Metode penelitian ini meliputi desain penelitian, jenis dan metode pengumpulan data, serta metode pendekatan dan pengembangan system. Desain penelitian yang digunakan oleh penulis adalah *Action Research.* Sedangkan jenis pengumpulan data menggunakan premier dan data sekunder. Metode pendekatan sistem yang penulis pilih untuk perancangan berorientasi objek dan metode pengembangan sistem yang digunakan adalah *Prototype.*

3.2.1 Desain Penelitian

Penelitian ini menggunakan metode *Action Research* (penelitian tindakan). Metode *Action Research* atau tindakan merupakan penelitian langsung, disertai dengan praktek dilapangan. Membuat suatu program yang akan dilaksanakan secara sistematis dan terencana, serta mempunyai nilai perbaikan signifikan. Penelitian tindakan ini lebih efektif, karena akan terlihat langsung hasilnya. Salah

satu syarat dalam melakukan penelitian tindakan adalah adanya keinginan dari orang yang memiliki masalah untuk mengidentifikasi masalah yang ada dan mempunyai keinginan untuk memecahkannya.

3.3.1. Jenis Penelitian dan Metode Pengumpulan Data

Metode pengumpulan data merupakan suatu teknik atau cara untuk memperoleh, mencari, mengumpulkan dan mencatat data yang digunakan untuk menyusun karya. Adapun pengumpulan data itu sendiri terbagi menjadi dua yaitu data premier dan data sekunder.

3.3.2.1. Sumber Data Primer

Data Premier adalah data yang hanya kita peroleh dari sumber asli (nara sumber) / responden penelitian baik melalui pengamatan maupun pencatatan terhadap objek penelitian.[18]

Dalam penelitian ini penulis mengumpulkan data premier dengan menggunakan teknik :

1. Pengamatan Langsung (observasi)

   Observasi merupakan suatu cara pengumpulan data melalui pengamatan dan pencatatan oleh pengumpul data terhadap gejala atau peristiwa yang diselidiki pada objek penelitian secara langsung tetapi tidak ada interaksi dengan objek yang diteliti dengan cara melakukan tinjauan langsung ke lapangan.[18]

Penulis melakukan tinjauan langsung ke lapangan untuk mencari data-data yang diperlukan terhadap objek yang diteliti dibagian *Operational Development*.

2. Wawancara

   Wawancara merupakan proses memperoleh keterangan untuk tujuan penelitian dengan cara tanya jawab, sambal bertatap muka antara penanya atau pewawancara dengan penjawab atau responden. Dalam penelitian ini penulis melakukan wawancara secara langsung dengan mengajukan berbagai pertanyaan kepada pihak yang ikut terlibat langsung. .[18]

3.3.2.2. Sumber Data Sekunder

Data Sekunder adalah data yang digunakan untuk mendukung data primer, merupakan jenis data yang sudah diolah terlebih dahulu oleh pihak pertama, data sekunder diambik secara tidak langsung dari objek penelitian misalnya data ini diperoleh dari buku-buku, jurnal, tutorial, internet dan lain-lain.

Dokumentasi adalah teknik pengumpulan data dengan cara mengumpulkan dokumen-dokumen yang berhubungan dengan objek penelitian. Dalam hal ini, dokumen yang diperoleh akan dianalisis agar diperoleh data yang sesuai dengan penelitian.

3.2.3 Metode Pendekatan dan Pengembangan Sistem

Dalam merancang suatu sistem dibutuhkan sebuah metode pendekatan dan pengembangan sistem untuk membantu dalam proses perancangan dan

pembuatannya. Berikut adalah penjelasan metode pendekatan dan pengembangan sistem yang akan digunakan.

3.2.3.1 Metode Pendekatan Sistem

Pada dasarnya saat ini pengembangan sistem dapat dikategorikan dalam 2 pendekatan pengembangan, yaitu pengembangan secara terstruktur dan pengembangan secara *object oriented* (berorientasi objek). Metode Pendekatan yang penulis ambil adalah pendekatan berorientasi objek yang divisualisasikan dengan UML (*Unified Modeling Language*).

Dalam jurnal yang ditulis Hendrik, Fathul Wahid yang berjudul "Pengembangan Learning Management System" menerangkan bahwa :

"Metode berorientasi objek (*object oriented method*) merupakan suatu cara pengembangan perangkat lunak dan sistem informasi berdasarkan abstraksi objek-objek yang didunia nyata. Brooks (1987) – dalam Nugroho (2002) menyatakan bahwa bagian tersulit dari pengembangan perangkat lunak dan / atau sistem informasi adalah tahap analisis, yang didalamnya abstraksi dilakukan. Dalam metode berorientasi objek kejadian atau hubungan antar entitas dalam dunia nyata direprensentasikan dalam objek-objek."

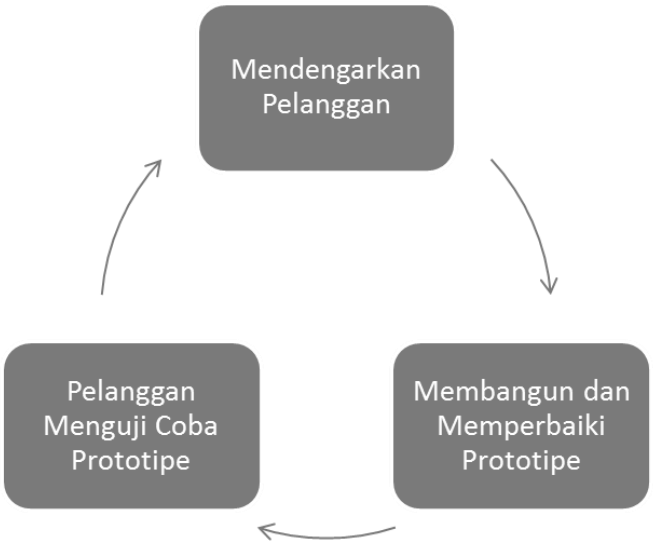
3.2.3.2 Metode Pengembangan Sistem

Metode pengembangan yang digunakan yaitu model *prototype*. Model *prototype* cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan seringkali kesulitan menyampaikan kebutuhannya secara detail tanpa melihat gambaran yang jelas.

Model *prototype* dimulai dari pengumpulan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototype agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototype ini biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program prototype ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan user atau pelanggan.

Secara umum tahapan pada model *prototyping* dapat dilihat pada gambar yang mengacu pada sistem yang akan dibuat, Maka dapat digambarkan pada diagram sebagai berikut :

(Sumber : Roger, S. Pressman, Ph.D. Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7 : Buku 1, 2012)



Gambar 3.2 Metode Pengembangan Sistem

3.2.3.3 Alat Bantu Analisis dan Perancangan

Pada rancangan sistem informasi *Task Management* ini, penulis menggunakan metode pendekatan berorientasi objek, terdapat banyak alat bantu yang dapat digunakan dalam analisis dan desain berorientasi objek menyebabkan munculnya ketidakjelasan alat bantu apa yang paling unggul. Para pengguna harus memilih diantara bahasa pemodelan dan alat bantu pemodelan yang beragam. Untuk mengatasi masalah tersebut maka *Object Management Group* (OMG) kemudian mengeluarkan UML, dimana dengan adanya UML ini diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selamai ini terjadi dalam lingkungan industri perangkat lunak. [23]

a. Diagram *use-case (use case diagram)*, Diagram ini menunjukan sekumpulan kasus fungsional dan aktor (jenis kelas khusus) dan keterhubungannya.
b. Skenario *use case* adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem.
c. Diagram aktivitas (*activity diagram*), Diagram ini untuk menunjukan aliran aktivitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem. Diagram ini penting untuk memodelkan fungsi sistem dan menekankan pada aliran kendali di antara objek-objek.
d. Diagram sekuen (*sequence diagram*), Diagram ini menunjukan interaksi yang terjadi antara objek. Diagram ini merupakan pandangan dinamis terhadap siste. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi.

e. Diagram kelas *(Class diagram),* Diagram yang menunukan sebuah kelas, *Interface* dan kolaborasi dan keterhubungannya. Diagram kelas ditunjukan untuk pandangan statik terhadap sistem.

f. Diagram komponen (*Component diagram*), Diagram ini menunjukan organisasi dan kebergantungan diantara sekumpulan komponen. Diagram ini merupakan pandangan statik terhadap implementasi sistem.

g. Diagram *deployment (Deployment diagram),* Diagram ini menunjukan konfigurasi pemrosesan saat jalan dan komponen-komponen yang di dalamnya. Diagram ini merupakan pandangan static dari arsitektur.

3.2.4 Pengujian Sistem

Menurut Roger S Pressman dalam bukunya Rekayasa Perangkat Lunak : "Pengujian *Black-Box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian *Black-Box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program". Pengujian *Black-Box* bukan merupakan alternative dari teknik *White-Box,* tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu menggunakan kelas kesalahan daripada metode *white-box*. Pengujian *Black–Box* berusaha menemukan kesalahan dalam kategori sebagai berikut [19]:

Pada pengujian black box, mencoba beragam masukan dan memeriksa keluaran yang dihasilkan. Teknik pengujian black box juga dapat digunakan untuk pengujian berbasis skenario, dimana isi dalam sistem mungkin tidak tersedia untuk

diinspeksi tapi masukan dan keluaran yang didefinisikan dengan use case dan informasi analisis yang lain.

Beberapa jenis kesalahan yang dapat diidentifikasi :

1. Fungsi tidak benar atau hilang
2. Kesalahan antar muka
3. Kesalahan pada struktur data (pengaksesan basis data)
4. Kesalahan inisialisasi dan akhir program
5. Kesalahan performansi.

3.3. Analisis Sistem yang Sedang Berjalan

Setelah penulis mengadakan penelitian dan menganalisa sistem prosedur yang berjalan di PT. eBdesk Teknologi serta melihat langsung kegiatan yang berhubungan dengan pembuatan aplikasi baru yang ada di perusahaan tersebut. Maka Pada analisis prosedur ini, harus di ketahui kegiatan apa saja yang dilalui, untuk perancangan kedepannya, apa yang menjadi kebutuhan pemakai serta keluaran yang nantinya dihasilkan.

3.3.1 Analisis Prosedur Pembelajaran yang Sedang Berjalan

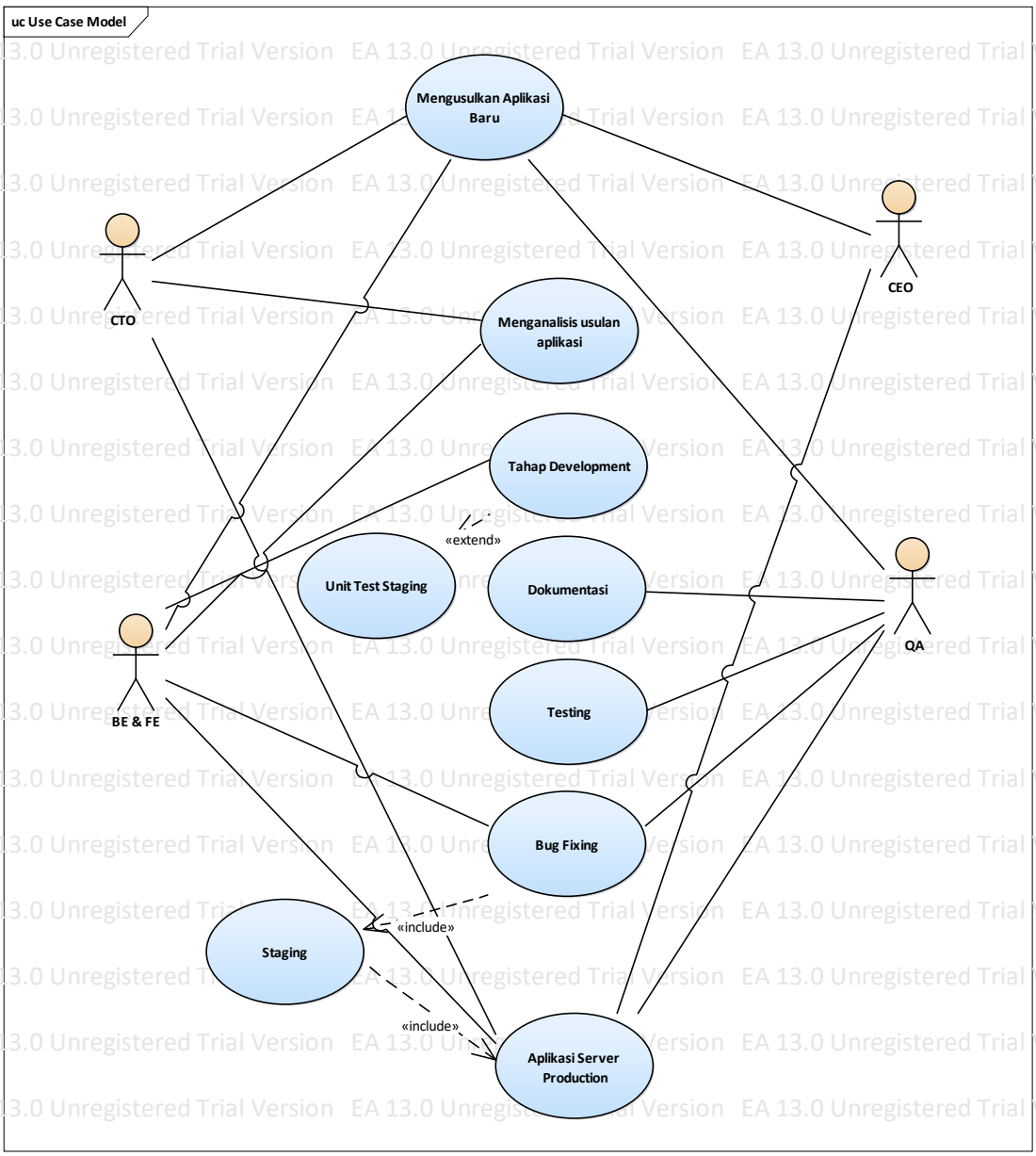
Berikut ini hasil prosedur Pembuatan Aplikasi Baru yang sedang berjalan :

1. CEO, CTO dan *leader* dari setiap divisi *development* melakukan meeting dikantor pusat.
2. CTO menganalisis usulan aplikasi baru yang diberikan oleh CEO.

3. CTO memberikan hasil analisis berupa rancangan antarmuka atau pemakaian code kepada leader BE dan leader FE.
4. Leader BE dan leader FE memberika tugas-tugas kepada *programmer* BE dan FE.
5. *Programmer* BE dan FE mengerjakan tugas yang telah diberikan oleh *leader*.
6. *Programmer* BE dan FE mengecek aplikasi yang telah mereka kerjakan.
7. Setelah selesai implementasi oleh *programmer* masuk kedalam tahap pembuatan dokumen oleh QA.
8. Setelah selesai membuat dokumen, QA melakukan testing secara keseluruhan.
9. Apabila QA menemukan bug, segera dilaporkan kepada *programmer* untuk diperbaiki.
10. CEO melihat aplikasi yang sudah jadi yang bebas dari bug.
11. Tahap akhir dari Aplikasi baru dengan melakukan pemeliharaan aplikasi.

3.3.2 *Use Case Diagram*

Use case diagram memperlihatkan pada kita hubungan-hubungan yang terjadi antara aktor-aktor dengan *use case-use case* dalam sistem. Adapun *use case* diagram yang berjalan pada PT. eBdesk Teknologi adalah sebagai berikut:



Gambar 3.3 Use Case Yang Sedang Berjalan

3.3.2.1 Definisi Aktor dan Deskripsinya

Diagram *use case* memperlihatkan pada kita hubungan-hubungan yang terjadi antara aktor-aktor dengan *use case-use case* dalam sistem. Adi Nugroho (2005: 59). Berikut ini merupakan Definisi Aktor berserta Deskripsi tentang *Use Case* Diagram Pembuatan *Aplikasi Baru* yang sedang berjalan di PT. eBdesk Teknologi.

Table 3.1 Definisi Aktor dan Deskripsinya

No	Aktor	Deskripsi
1	CEO	Merupakan Aktor mempunyai tugas melahirkan ide atau membuat suatu aplikasi baru sesuai dengan perkembangan IT
2	CTO	Merupakan Aktor mempunyai tugas untuk mengembangkan produk sesuai kebutuhan pengguna.
3	BE&FE	Merupakan Aktor mempunyai tugas untuk membuat program aplikasi.
4	QA	Merupakan Aktor yang terlibat dalam Proses Pengecekan Sesuai Prosedur sebuah aplikasi baru sebelum masuk ke dalam bagian Production System.

3.3.2.2 Definisi *Use Case* dan Deskripsinya

Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal *Frase* nama *Use Case*. *Use Case* mendeskripsikan apa yang sistem (atau subsistem, kelas, atau antarmuka) kerjakan tetapi ia tidak menspesifikasikan (dan tidak memiliki kompetensi untuk menspesifikasi) bagaimana ia melakukannya.

Table 1.2 Definisi Use Case dan Deskripsinya

No	Use Case	Deskripsi
1	Mengusulkan Aplikasi Baru	Merupakan proses pengusulan/ terlahirnya ide untuk membuat aplikasi baru
2	Menganalisis usulan aplikasi	Merupakan proses meneliti/ menganalisa *request* aplikasi baru
3	Tahap *Development*	Merupakan proses pengerjaan aplikasi baru
4	Unit Test	Merupakan Proses pengecekan aplikasi oleh programmer sesuai dengan task yang diberikan oleh CTO
8	Dokumentasi	Merupakan proses Pembuatan dokumen aplikasi baru (Test Script dan Test Result)
4	*Testing*	Merupakan proses pengujian aplikasi sesuai dengan dokumen
5	*BugFixing*	Merupakan proses perbaikan aplikasi yang menjamin sudah tidak ditemukan lagi kesalahan

6	*Staging*	Merupakan Proses pengecekan apabila sudah selesai proses *bug fixing* dan sebelum masuk server production
7	Aplikasi *Server Production*	Merupakan proses yang dilakukan setelah tahapan testing selesai, kini tiba waktunya untuk aplikasi mulai dioperasikan serta bekerja sesuai fungsinya

3.3.3 Sekenario *Use Case*

Selanjutnya, setelah diagram *use case* selesai, setiap use case harus dirinci lagi dalam bentuk skenario *use case* sehingga jelas bagaimana urutan aksi aktor dan reaksi sistem. Skenario *use case* merupakan narasi tentang aktivitas dalam suatu *use case* diagram. Adapun skenario *use case* dalam sistem yang sedang berjalan adalah sebagai berikut:

Table 3.3 Use Case Skenario Mengusulkan Aplikasi Baru

Identifikasi		
No Usecase	1	
Nama	Mengusulkan Aplikasi Baru	
Tujuan	Membuat aplikasi baru sesuai dengan perkembangan IT yang ide/ usulannya dari CEO	
Deskripsi	Membuat Aplikasi baru sesuai dengan perkembangan IT	
Aktor	CEO, CTO, BE&FE, QA	
Skenario utama		
Kondisi awal	CEO mempunyai usulan membuat aplikasi baru	
Kondisi akhir	Seluruh Tim terkait mengetahui usulan dari CEO	
Aksi Aktor		**Reaksi**
1. CEO memanggil semua pimpinan terkait untuk melakukan rapat		
2. CEO melakukan rapat dengan semua pimpinan terkait untuk menyampaikan ide nya		
		3. Seluruh Pimpinan menerima usulan dari CEO perihal pembuatan Aplikasi baru

Table 3.4 Use Case Skenario Menganalisis usulan Aplikasi

Identifikasi	
No Usecase	2
Nama	Menganalisis usulan Aplikasi
Tujuan	Menentukan design/ code yang tepat
Deskripsi	Meneliti/ menganalisa aplikasi baru untuk menentukan design/ code yang tepat
Aktor	CTO, BE & FE
Skenario utama	
Kondisi awal	Usulan Aplikasi baru belum dianalisis
Kondisi akhir	Usulan Aplikasi Baru sudah dianalisis dan siap untuk dikerjakan
Aksi Aktor	**Reaksi**
1. CTO membuka kembali catatan hasil rapat	
2. CTO menganalisa usulan aplikasi baru dari CEO	
	3. CTO dan Programmer berdiskusi menentukan desain/ code yang tepat untuk aplikasi baru
4. CTO membagikan *task-task* kepada BE&FE	
5. CTO memberikan estimated time pengerjaan	

Table 3.5 Use Case Skenario Tahap Development

Identifikasi	
No Usecase	3
Nama	Tahap Development
Tujuan	Membagikan tugas-tugas Kepada Programmer dari CTO
Deskripsi	Merupakan proses pengerjaan request aplikasi baru
Aktor	Programmer BE&FE
Skenario utama	
Kondisi awal	Request masih dalam tahap analisis
Kondisi akhir	Request sudah masuk dalam tahap pengerjaan

Aksi Aktor	Reaksi
1. Programmer menerima task-task yang sudah dianalisis dari CTO	
	2. Programmer mulai mengerjakan task-task untuk di implementasikan
	3. Programmer mengetes program yang sudah mereka kerjakan sesuai dengan task sebelum masuk dalam tahap testing keseluruhan
4. Programer memberitahu QA bahwa aplikasi sudah siap testing	

Table 3.5 Use Case Skenario Unit Test Staging

Identifikasi	
No Usecase	3
Nama	Unit Test Staging
Tujuan	Memastikan tidak ada kesalahan dalam tahap development
Deskripsi	Merupakan proses pengecekan setelat tahap development selesai
Aktor	Programmer BE&FE
Skenario utama	
Kondisi awal	Belum dilakukan pengecekan setelah tahap development
Kondisi akhir	Sudah dilakukan pengecekan setelah tahap development
Aksi Aktor	**Reaksi**
1. Programmer telah selesai dalam pengerjaan tahap development	
	2. Programmer kembali mengecek task yang sudah dikerjakan
	3. Apabila sudah benar dan lolos unit test, maka programmer akan memberi tahu QA untuk di test
4. Programer memberitahu QA bahwa aplikasi sudah siap testing	

Table 3.6 Use Case Skenario Dokumentasi

Identifikasi		
No Usecase	3	
Nama	Dokumentasi	
Tujuan	Membatu QA dalam hal pengetesan aplikasi baru	
Deskripsi	Merupakan proses pembuatan dokumen untuk aplikasi baru	
Aktor	QA	
Skenario utama		
Kondisi awal	Aplikasi baru belum ada dokumennya	
Kondisi akhir	Aplikasi baru sudah dibuat dokumennya	
Aksi Aktor		**Reaksi**
1. QA membuka aplikasi baru yang berada pada server staging		
		2. QA membuat 2 dokumen diantara dokumen Test Script dan Dokumen Test Result

Table 3.7 Use Case Skenario Testing

Identifikasi		
No Usecase	5	
Nama	Testing	
Tujuan	Membuat aplikasi baru bebas bug/ error	
Deskripsi	Merupakan proses pengujian aplikasi sesuai dengan dokumen	
Aktor	QA, Programmer BE&FE	
Skenario utama		
Kondisi awal	Aplikasi baru masih dalam tahap development	
Kondisi akhir	Aplikasi baru sudah siap testing secara keseluruhan	
Aksi Aktor		**Reaksi**
1. QA melihat dokumen untuk panduan mengetes aplikasi		
		2. QA mulai melakukan pengujian secara keseluruhan terhadap aplikasi

	baru
3. QA mencatat apabila menemukan bug	
4. QA Melaporkan hasil testing kepada Programmer	

Table 3.8 Use Case Skenario Bug Fixing

Identifikasi	
No Usecase	6
Nama	Bug Fixing
Tujuan	Bug yang ditemukan bisa segera diperbaiki dan kualitas perangkat lunak terjaga
Deskripsi	Merupakan proses perbaikan aplikasi yang menjamin sudah tidak ditemukan lagi kesalahan
Aktor	Programmer BE&FE, QA
Skenario utama	
Kondisi awal	Aplikasi baru masih terdapat bug/ error
Kondisi akhir	Aplikasi baru bebas dari bug
Aksi Aktor	**Reaksi**
1. Programmer BE&FE Menerima laporan error dari QA	
2. Programmer BE&FE melakukan pengecekan dan melakukan perbaikan di server staging	
	3. QA melakukan pengecekan kembali sesuai dengan dokumen
	4. Bila sudah sesuai, maka QA akan melaporkannya kepada Programmer
5. Bila sudah tidak ada bug, Programmer akan bersiap untuk melakukan deploy ke server production	

Table 3.8 Use Case Staging

Identifikasi		
No Usecase	6	
Nama	Staging	
Tujuan	Memastikan aplikasi bebas dari error	
Deskripsi	Menguji perangkat lunak, di lingkungan produksi yang serupa sebelum ditetapkan secara langsung.	
Aktor	Programmer BE&FE	
Skenario utama		
Kondisi awal	Aplikasi baru belum diterapkan di server staging	
Kondisi akhir	Aplikasi baru sudah diterapkan di server staging	
Aksi Aktor		**Reaksi**
1. Programmer BE&FE menerapkan aplikasi baru pada server staging		
		2. Programmer BE&FE mengecek apakah sudah sesuai atau belum
3. Apabila sudah sesuai Aplikasi siap diluncurkan pada server production		

Table 3.9 Use Case Skenario Aplikasi Server Production

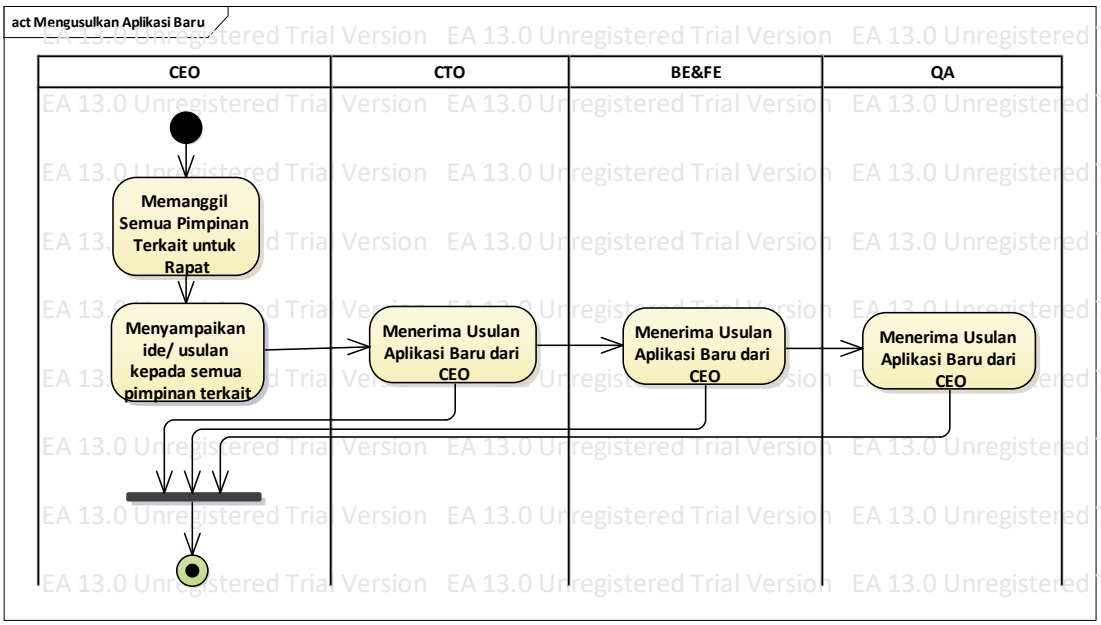
Identifikasi		
No Usecase	7	
Nama	Aplikasi Server Production	
Tujuan	Aplikasi bisa diakses atau dipergunakan dimanapun berada	
Deskripsi	Merupakan proses yang dilakukan setelah tahapan testing selesai, kini tiba waktunya untuk aplikasi mulai dioperasikan serta bekerja sesuai fungsinya	
Aktor	CEO, CTO, BE&FE, QA	
Skenario utama		
Kondisi awal	Aplikasi baru masih pada Staging server	
Kondisi akhir	Aplikasi baru sudah pada Production system	
Aksi Aktor		**Reaksi**
1. CTO mengecek aplikasi yang sudah		

selesai pada server staging	
2. Bila sudah dapat approval dari CTO, maka aplikasi dilihat oleh CEO	
	3. Bila sudah sesuai maka Programmer akan melakukan deploy ke Server Production
	3. QA mengecek kembali aplikasi yang sudah masuk kedalam server production guna memastikan tidak ada error
4. CEO siap menggunakan aplikasi baru untuk dipasarkan	

3.3.4 *Acrivity Diagram*

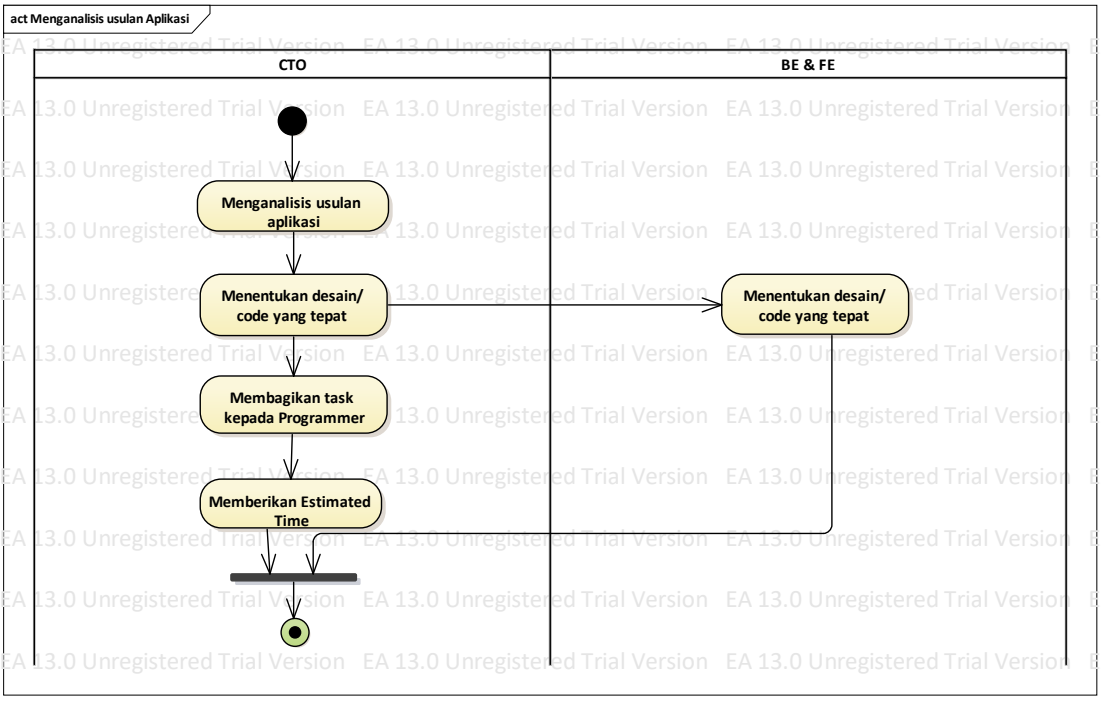
Di dalam *activity diagram* menjelaskan aktivitas-aktivitas dari fungsi pada sistem yang berjalan berikut diagram aktivitasnya :

3.3.4.1 *Activity Diagram* Mengusulkan Aplikasi Baru



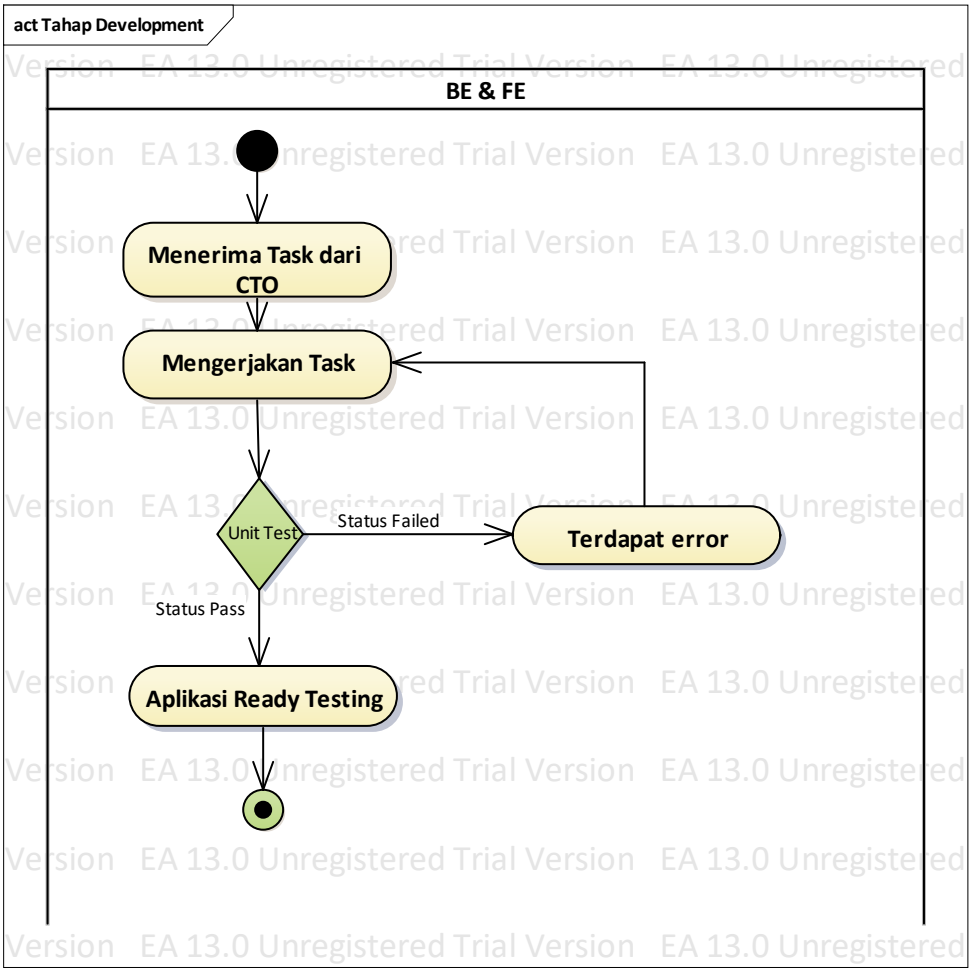
Gambar 3.4 Activity Diagram Mengusulkan Aplikasi Baru

3.3.4.2 *Activity Diagram* Menganalisis Usulan Aplikasi



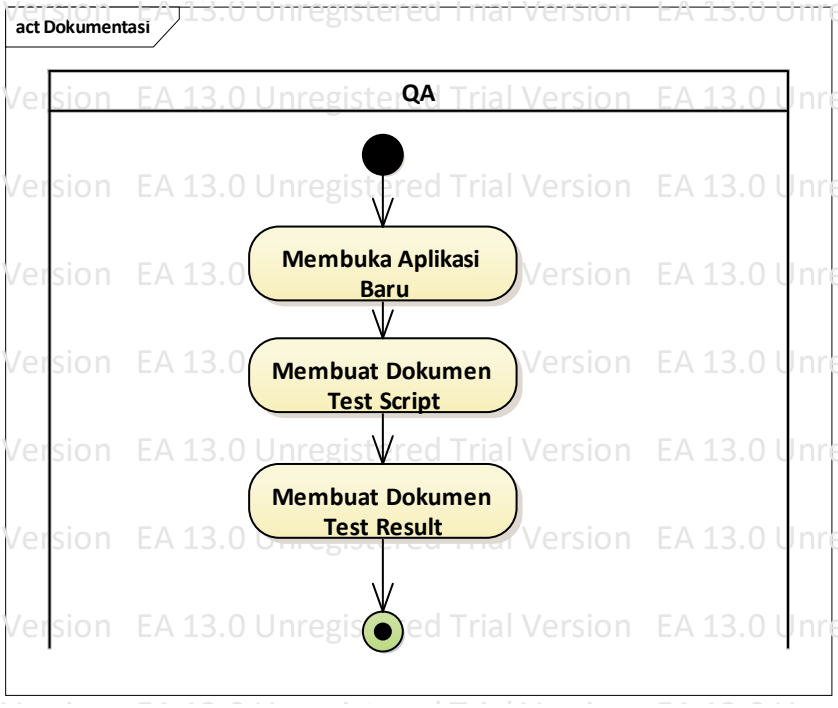
Gambar 3.5 Activity Diagram Menganalisis Usulan Aplikasi

3.3.4.3 *Activity Diagram* Tahap *Development*



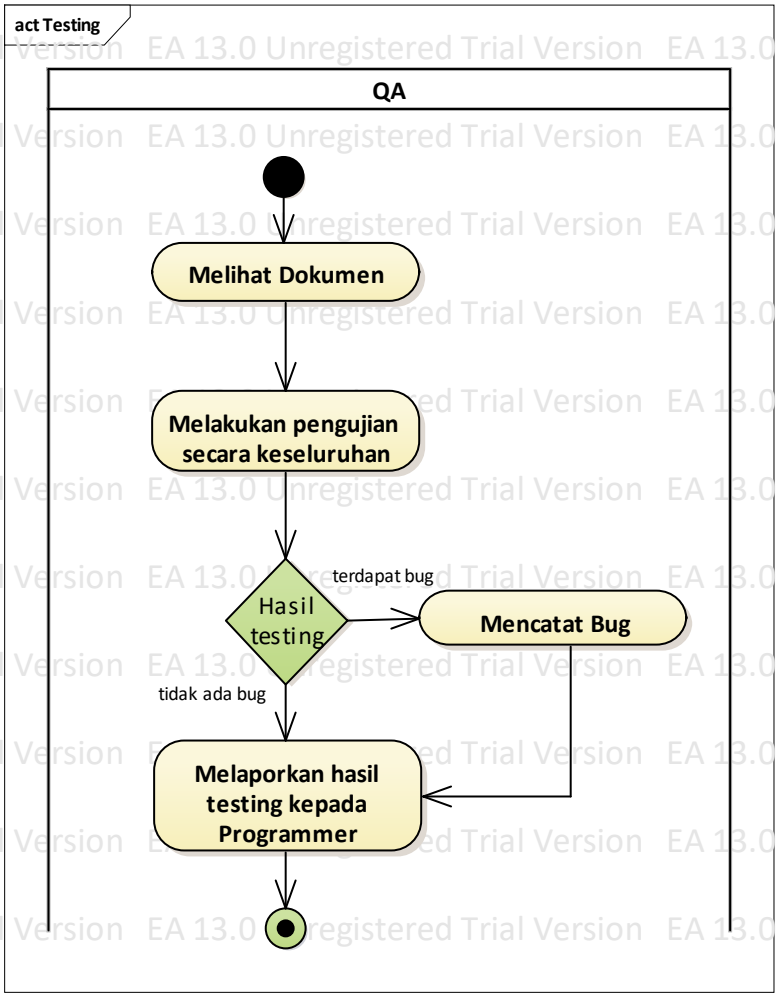
Gambar 3.6 Activity Diagram Tahap Development

3.3.4.4 *Activity Diagram* Dokumentasi



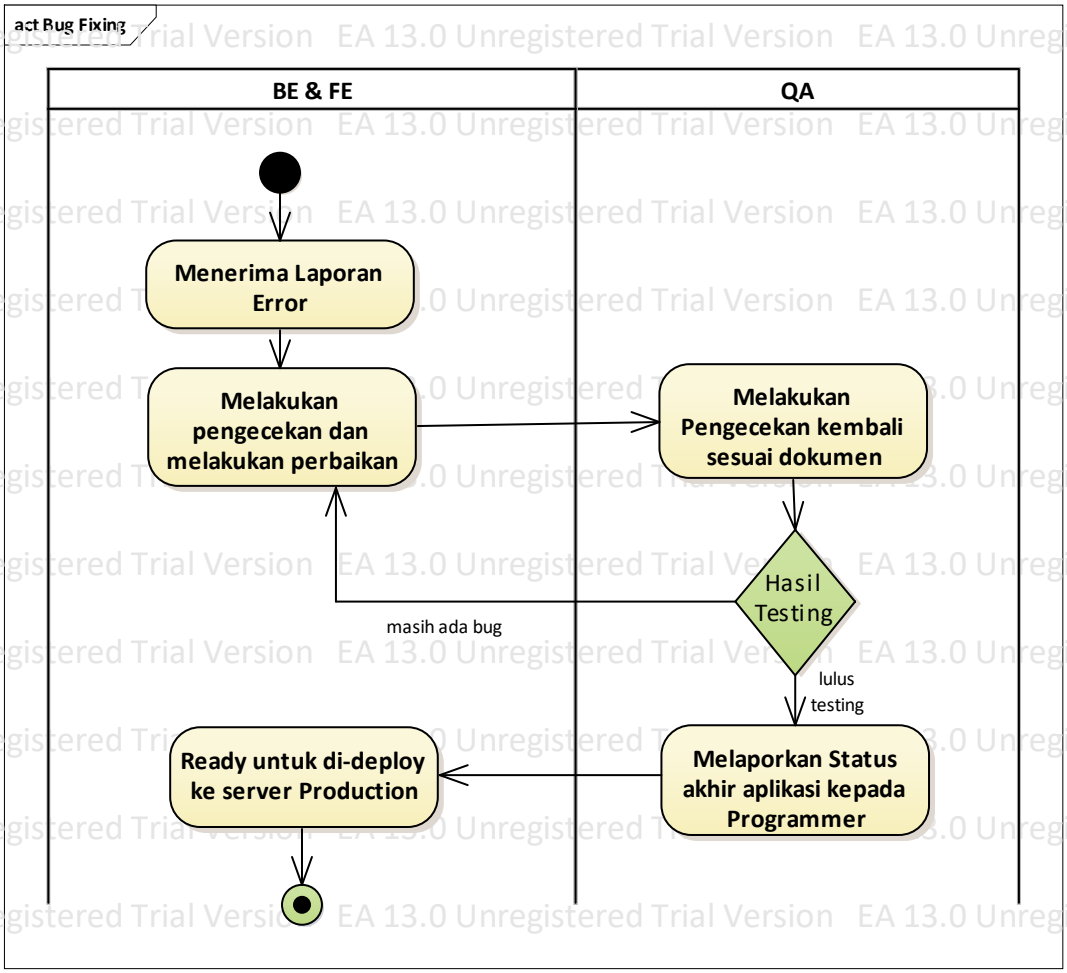
Gambar 3.7 Activity Diagram Dokumentasi

3.3.4.5 *Activity Diagram Testing*



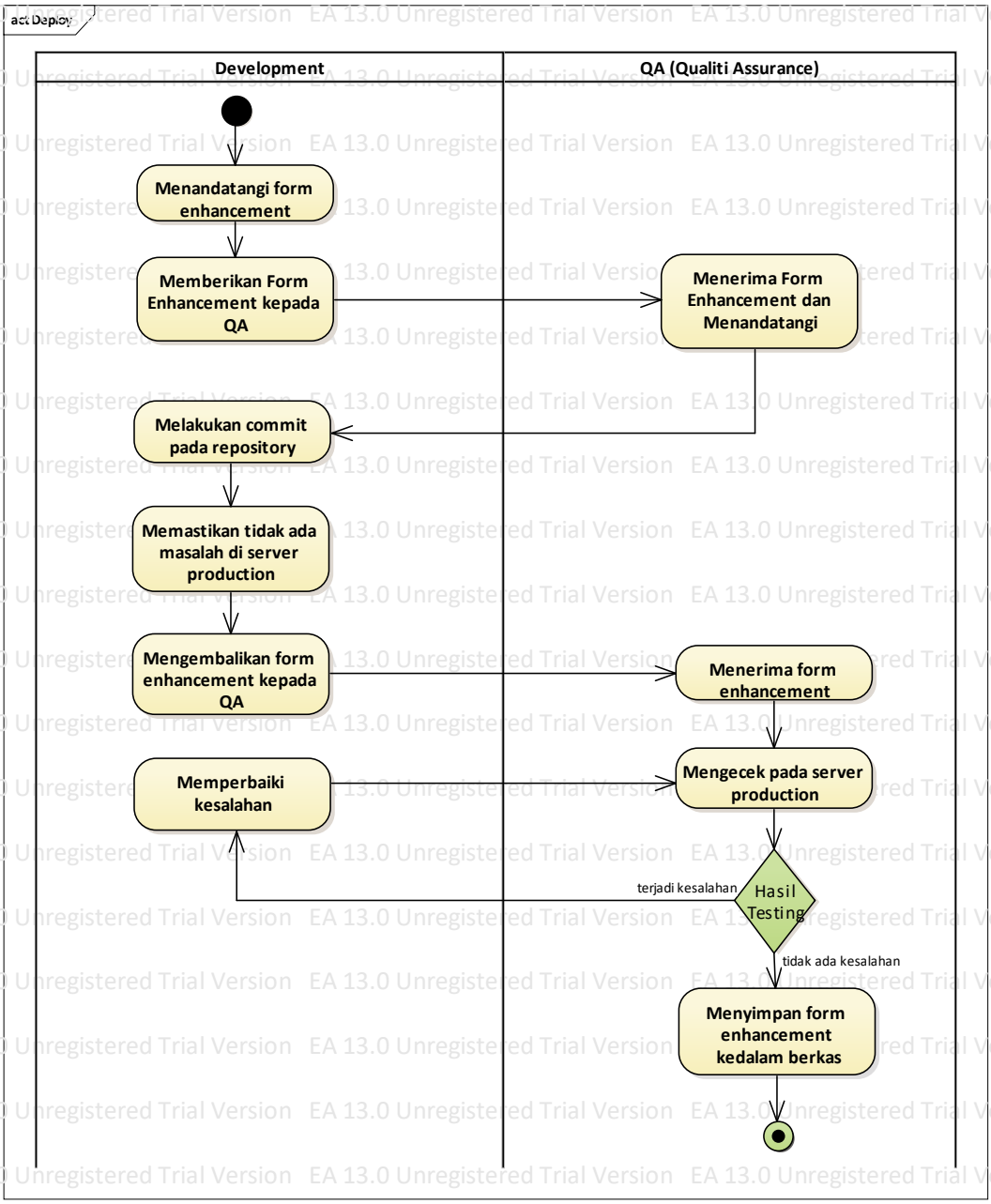
Gambar 3.8 Activity Diagram Testing

3.3.4.6 *Activity Diagram Bug Fixing*



Gambar 3.9 Activity Diagram Bug Fixing

3.3.4.7 *Activity Diagram* Aplikasi *Server Production*



Gambar 3.9 Activity Diagram Bug Fixing

3.3.5 Evaluasi yang Sedang Berjalan

Evaluasi sistem yang berjalan adalah proses menemukan kesalahan, kekurangan, hambatan-hambatan yang terjadi pada sistem yang sedang berjalan. Sehingga dapat diusulkan pemecahan suatu masalah agar dapat menghasilkan, mengoptimalkan dan membantu proses pengeloaan sistem secara lebih baik, cepat dan akurat.

Table 3.10 2Evaluasi yang Sedang Berjalan

No	Permasalahan	Pemecahan	Bagian
1.	Mengeluarkan banyak *resource* seperti waktu, kertas dan printer	Pembuatan Aplikasi *task management* secara *online*, sehingga bisa menghemat *resource*	All Development Team
2.	Kesulitan dalam mencari atau menyusun dokumen	Pembuatan fasilitas untuk menyimpan dan mencari data secara online.	QA Development
3.	Kesulitan dalam mendistribusikan *task*	Pembuatan fasilitas Backlog (mendistribusikan *task)*	Programmer Development
4.	Kesulitan melakukan koordinasi antar team untuk implementasi sistem	Pembuatan fasilitas status progress feature	All Development Team