

**MEMBANGUN SISTEM INFORMASI PELATIHAN MAHASISWA BERBASIS WEB
MENGUNAKAN *FRAMEWORK CODEIGNITER* PADA *CAREER DEVELOPMENT
CENTRE (CDC) UNIVERSITAS TELKOM***

Ferdy Sibuea¹, Irfan Darmawan², Warih Puspitasari³

Program Studi Sistem Informasi Fakultas Rekayasa Industri Universitas Telkom Jl.

Telekomunikasi No. 1, Terusan Buahbatu, Bandung 40257 Indonesia

pbeh_sibuea@yahoo.com¹, irfandarmawan@telkomuniversity.ac.id², warihpuspita@gmail.com³

Abstrak

CDC Telkom University merupakan bagian pengembangan karir yang ada di bawah Direktorat Pusat Pengembangan Karir dan Pengelolaan Alumni yang mempunyai tugas utama, yaitu fokus pada kegiatan yang memfasilitasi mahasiswa dan calon alumni untuk mendapatkan konseling dan pelatihan *softskill* yang bertujuan agar mahasiswa dan calon alumni siap menghadapi dunia kerja.

CDC Telkom University masih menggunakan sistem manual pada proses pendaftaran peserta dimana pihak **CDC Telkom University** masih menggunakan layanan yang disediakan oleh *Google* seperti *Google Docs* maupun *Google Spreadsheet*. Dengan memanfaatkan layanan dari *Google* ini masih sering terjadi kendala seperti *redundancy* data peserta pelatihan pada saat melakukan pendaftaran. Kendala lain seperti mahasiswa lupa jadwal pelatihan juga masih sering terjadi. Pada proses sertifikasi, sistem akan diubah dari pencetakan sertifikat secara manual menjadi otomatis sebagai tindakan preventif untuk menghindari kesalahan input data seperti kesalahan pengetikan dan sebagainya.

Oleh karena itu, dalam penelitian ini dibangun sebuah sistem informasi pelatihan *softskill* dimana di dalam sistem yang dibangun terdapat fungsi-fungsi yang dapat digunakan untuk mengatasi kendala-kendala seperti *redundancy* data, lupa jadwal, dan otomatisasi proses sertifikasi. Penelitian ini dilakukan dengan menggunakan metode *Prototyping-Oriented Software* yang didukung dengan pengujian *beta testing*. Setiap fungsionalitas dari sistem yang dibangun diuji oleh *user* dari sistem ini sendiri sehingga sistem yang dibangun sesuai dengan yang diharapkan oleh *user*.

Hasil dari penelitian ini berupa sebuah aplikasi *website* untuk **CDC Telkom University** dimana dengan adanya sistem ini diharapkan dapat mempermudah beban kerja dari **CDC Telkom University** dalam melakukan perencanaan pelatihan *softskill*.

Kata kunci: *CDC Telkom University, Softskill, Prototyping-Oriented Software, Beta Testing.*

Abstract

CDC Telkom University is part of career development that is under the Directorate of Career Development and Management Center Graduates who have the main task, namely to focus on activities that facilitate prospective students and graduates to get counseling and *softskill* training that aims to make students and prospective graduates ready to face the world of work.

CDC Telkom University still using manual systems in the registration process which the **CDC Telkom University** still use the services provided by *Google* such as *Google Docs* or *Google Spreadsheet*. By utilizing the services provided by *Google*, *redundancy* data still occur in the registration process. On the other hand, students occasionally forget the training schedule. In the certification process, the system will be changed from printing certificates manually into automated as a preventive measures to avoid data input errors like typos and so on.

Therefore, an information system is built in this research to prevent data *redundancy*, forgot the schedule, and to automate the certification process.. The research was conducted using the method *Prototyping-Oriented Software* that is supported by testing *beta testing*. Each functionality of the system built tested by the user of the system itself so that the system is built in accordance with what is expected by the user.

Result from this research is a *website* application for **CDC Telkom University** where the presence of this system is expected to ease the workload of **CDC Telkom University** in the planning of *soft skill* training.

Keywords: *CDC Telkom University, Softskill, Prototyping-Oriented Software, Beta Testing.*

I. PENDAHULUAN

Career Development Centre (CDC) Universitas Telkom merupakan bagian kemahasiswaan yang khusus menangani pengembangan karir mahasiswa Universitas Telkom. Salah satu bentuk pengembangan karir yang dilakukan oleh CDC adalah memberikan pelatihan kepada mahasiswa. Jenis-jenis pelatihan yang diberikan kepada mahasiswa antara lain *Personal Goal Setting*, *Self-Awareness*, *Basic Study Skill*, dan *MOCK Interview*, selain itu ada beberapa pelatihan yang dilakukan bersama pihak perusahaan. *Personal Goal Setting*, *Basic Study Skill*, dan *Self-Awareness* dikhususkan untuk mahasiswa baru, sedangkan untuk mahasiswa aktif lainnya hanya pelatihan umum. Untuk alumni sendiri biasanya dilakukan *profiling* alumni dan *MOCK Interview*.

Metode penelitian yang digunakan dalam penelitian ini adalah metode *prototype* yang dipadukan dengan *beta testing* dalam melakukan uji coba sistem. *Testing* dapat dilakukan oleh *user* setiap terbentuk *prototype* baru dari sistem yang dibangun. Dengan demikian, *user* akan lebih mudah untuk menyampaikan apa saja yang perlu diperbaiki dari sistem yang dibangun ini sehingga sistem ini nantinya akan sesuai dengan apa yang diharapkan oleh *user*. *User* yang dimaksud disini terdiri dari *admin*, *trainer* (unit pengembangan karir CDC), dan mahasiswa (peserta pelatihan).

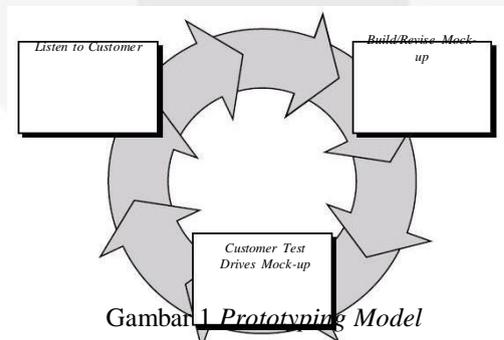
II. LANDASAN TEORI

A. Framework CodeIgniter (CI)

Framework merupakan suatu kerangka kerja yang di dalam bidang pemrograman kumpulan kelas (*class*) dan fungsi (*function*, *method*) yang disusun secara sistematis berdasarkan kegunaan atau fungsionalitas tertentu untuk mempermudah pembuatan atau pengembangan suatu aplikasi. Secara umum, *framework* yang banyak digunakan saat ini adalah *framework* yang menggunakan konsep *Object-Oriented Programming* (OOP). Keuntungan yang diperoleh dari penggunaan *framework* ini adalah waktu kerja yang menjadi lebih efisien dalam penulisan kode program (*coding*) dan pengaturan berkas-berkas kode (*code*) karena tidak perlu menuliskan kode program untuk fungsionalitas tertentu yang sudah tersedia. Berkas kode dapat disusun secara sistematis sesuai dengan struktur yang ditawarkan oleh *framework* tersebut (Sidik, 2012).

B. Metode Prototyping-Oriented Software

Metode *Prototyping-Oriented Software* adalah proses pengembangan sebuah perangkat lunak secara berulang yang memberikan ide bagi pengembang sistem untuk menyajikan gambaran yang lengkap. *Prototype* yang dibangun pada tahap awal merupakan bagian dari produk secara keseluruhan yang mengekspresikan logika maupun fisik pada antarmuka eksternal yang ditampilkan. Konsumen menggunakan *prototype* tersebut untuk memberikan masukan kepada tim pengembang sebelum pengembangan produk dalam skala besar dimulai (Simarmata, 2010). Secara ideal metode pengembangan *prototype* berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan dari sebuah perangkat lunak (Pressman, 2009). Metode cocok digunakan dalam pembangunan sistem informasi yang inovatif berdasarkan perspektif pemakai. Proses metode *Prototyping* ini digambarkan dalam gambar berikut ini:



Gambar 1 *Prototyping Model*

Keterangan:

1. *Listen to Customer*

Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengarkan keluhan dari pelanggan. Untuk membuat suatu sistem yang sesuai kebutuhan, maka harus

diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

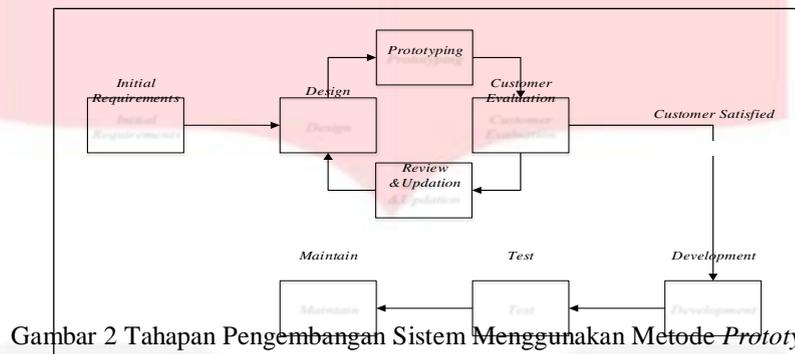
2. *Build/Revise Mock-up*

Pada tahap ini, dilakukan perancangan dan pembuatan *prototype* sistem. *Prototype* yang dibuat disesuaikan dengan kebutuhan sistem yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.

3. *Customer Test Drives Mock-up*

Pada tahap ini, *Prototype* dari sistem di uji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki *Prototype* yang ada.

Tahapan pengembangan sistem dengan metode *prototyping* digambarkan sebagai berikut.



Gambar 2 Tahapan Pengembangan Sistem Menggunakan Metode *Prototyping*

C. UML (Unified Modeling Language)

Unified Modelling Language (UML) merupakan sebuah bahasa visual untuk menangkap pola dan desain perangkat lunak. UML dapat diterapkan ke berbagai *domain* berbeda dengan dasar pengembangan perangkat lunak (Pitman, 2005).

UML merupakan standar terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka. Banyak perusahaan OMG dibentuk untuk membuat standar-standar yang mendukung *interoperabilitas*, khususnya *interoperabilitas* system berorientasi objek OMG yang dikenal dengan standar-standar CORBA (*Common Object Request Broker Architecture*).

Pada umumnya metode-metode yang ditujukan untuk pembangunan aplikasi berorientasi objek menggunakan UML untuk pembangunan aplikasi berorientasi objek menggunakan UML untuk memodelkan berbagai artefak dari perangkat lunak. UML adalah sekumpulan simbol dan diagram untuk memodelkan *software* dalam bentuk simbol dan diagram yang kemudian dapat diterjemahkan menjadi kode program.

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram*, untuk memodelkan proses bisnis.
2. *Activity Diagram*, untuk memodelkan perilaku *Use Case* dan objek di dalam sistem.
3. *Class Diagram*, untuk memodelkan struktur *class*.
4. *Object Diagram*, untuk memodelkan struktur objek.
5. *Conceptual Diagram*, untuk memodelkan aplikasi konsep-konsep yang ada di dalam aplikasi.
6. *Sequence Diagram*, untuk memodelkan pengiriman *message* antar objek.
7. *Collaboration Diagram*, untuk memodelkan interaksi antar objek.
8. *State Diagram*, untuk memodelkan perilaku objek dalam sistem.
9. *Component Diagram*, untuk memodelkan komponen objek.
10. *Deployment Diagram*, untuk memodelkan distribusi aplikasi.

D. *Entity Relationship Diagram*

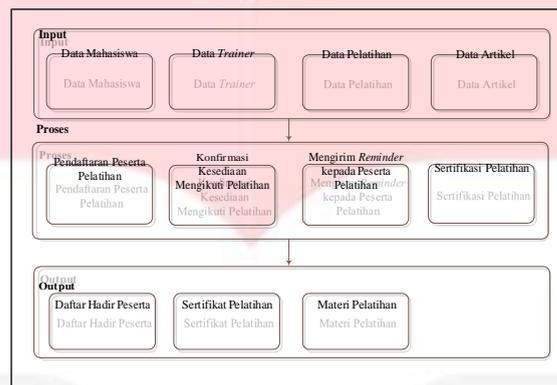
Entity Relation Diagram (ERD) merupakan model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara penyimpanan. Model data sendiri merupakan sekumpulan cara dan peralatan untuk mendeskripsikan data-data yang hubungannya satu sama lain serta batasan konsistensi. Model data terdiri dari model hubungan entitas dan model relasional.

Diagram hubungan entitas ditemukan oleh Peter Chen dalam buku *Entity Relational Model-Toward a Unified of Data*. ERD merupakan suatu model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi objek-objek dasar yang dinamakan Entitas yang serta memiliki Relasi antar entitas-entitas tersebut (Fathansyah, 2007).

III. METODOLOGI PENELITIAN

A. Model Konseptual

Model konseptual adalah konsep pemikiran yang membantu peneliti untuk merumuskan pemecahan masalah dan membantu dalam merumuskan solusi permasalahan yang ada. Berikut adalah model konseptual penelitian ini.



Gambar 3 Model Konseptual

Berdasarkan model konseptual di atas, dapat dilihat bahwa *input* dari sistem informasi yang akan dibangun berupa data mahasiswa, data *trainer*, data pelatihan, dan data artikel. Data mahasiswa ini berisi data-data mahasiswa seperti nama, NIM, nomor telepon, alamat, dan *e-mail*. Sedangkan data *trainer* berisi data-data *trainer*, yaitu nama, *email*, alamat, dan nomor telepon. Kemudian mahasiswa yang ingin mengikuti pelatihan harus melakukan pendaftaran pada sistem yang akan dibangun. Peserta harus melakukan konfirmasi ulang pada sistem yang menyatakan bahwa mahasiswa yang bersangkutan bersedia mengikuti pelatihan pada jadwal yang telah ditentukan. Peserta juga akan mendapatkan SMS sebagai *reminder* terhadap pelaksanaan pelatihan. Setelah peserta selesai mengikuti kegiatan pelatihan, peserta tersebut kemudian dapat melakukan sertifikasi pelatihan pada sistem. *Output* dari ke tiga proses ini berupa daftar hadir peserta pelatihan, materi pelatihan, dan sertifikat pelatihan.

B. Sistematika Penelitian

Pada penelitian ini terdapat empat fase yang akan dilakukan, yaitu fase *initial requirements*, fase *prototyping*, fase *customer evaluation* dan *review & updation*, serta fase implementasi dan *maintenance*.

1. Initial Requirements

Pengumpulan data pada fase ini dimulai dengan proses studi literatur yang terkait metode pengumpulan data dan metode *prototype* dari sumber-sumber seperti *paper* ataupun buku. Selain berdasarkan studi literatur, pengumpulan data juga dilakukan dengan *interview* kepada pihak CDC. Setelah itu akan dilakukan perumusan masalah serta menentukan tujuan, manfaat, dan batasan penelitian. Kemudian dilakukan analisis kebutuhan sistem.

2. Fase Prototyping

Fase ini akan dilakukan apabila analisis kebutuhan sistem sudah teridentifikasi. Apabila kebutuhan sistem belum teridentifikasi, maka akan dilakukan analisis ulang terhadap kebutuhan sistem. Pada fase ini akan dilakukan perancangan awal terhadap sistem yang akan dibangun kemudian rancangan sistem tersebut dibangun menjadi sebuah *prototype*.

3. Fase Customer Evaluation, Riview & Updation

Pada fase ini akan dilakukan evaluasi terhadap *prototype* yang sudah dibangun. Evaluasi ini dilakukan oleh *user* untuk memastikan apakah *prototype* yang ada sudah sesuai dengan keinginan

user. Jika *prototype* yang dibangun masih belum sesuai dengan keinginan *user*, maka perlu dilakukan revisi terhadap *prototype* yang dibangun. Revisi yang dilakukan akan dimulai pada tahap perancangan sistem.

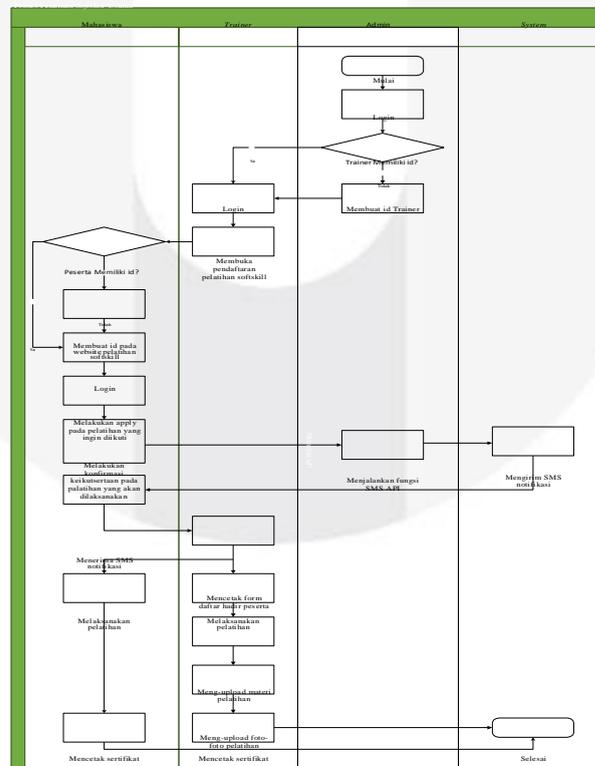
4. Fase Implementasi dan *Maintenance*

Implementasi terhadap sistem akan dilakukan apabila hasil dari evaluasi *user* terhadap *prototype* yang dibangun sudah sesuai harapan. Setelah sistem diimplementasikan, sistem juga akan mendapatkan *maintenance* berkala.

IV. ANALISIS DAN PERANCANGAN APLIKASI

A. Proses Bisnis Usulan

Gambar berikut menjelaskan mengenai proses pelaksanaan pelatihan *softskill* usulan dimulai dari admin *login* ke sistem kemudian membuat *id* untuk *trainer*. Setelah *id* untuk *trainer* tersedia, *trainer* kemudian dapat *login* ke sistem untuk membuka pendaftaran pelatihan *softskill*. Mahasiswa yang tidak memiliki *id* tidak mempunyai akses untuk masuk ke dalam *website* yang dibangun. Untuk itu mahasiswa perlu melakukan registrasi pada sistem untuk membuat *id* mahasiswa tersebut. Setelah *id* mahasiswa berhasil didaftarkan, mahasiswa akan memiliki akses untuk *login* ke dalam sistem. Untuk dapat mengikuti pelatihan yang akan dilaksanakan, mahasiswa harus melakukan *apply* pada pelatihan yang akan dilaksanakan. Menu *apply* ini tersedia pada halaman *trainings*. Proses ini baru akan selesai ketiga mahasiswa melakukan konfirmasi pada halaman profil. Batas waktu untuk melakukan konfirmasi adalah sehari sebelum pelatihan dilaksanakan. Apabila mahasiswa belum melakukan konfirmasi sampai batas waktu yang telah ditentukan, maka data keikutsertaan mahasiswa dalam pelatihan tersebut akan dihapus dari *database* dan dianggap tidak akan mengikuti pelatihan yang akan dilaksanakan. Kemudian admin akan menjalankan fungsi SMS API sebagai *reminder*. SMS ini akan dikirim secara otomatis oleh sistem kepada *trainer* dan mahasiswa yang menjadi peserta pelatihan. *Trainer* dapat langsung mencetak *form* daftar hadir peserta sebelum melaksanakan pelatihan. Setelah kegiatan pelatihan selesai dilaksanakan, *trainer* dapat meng-*upload* materi pelatihan agar dapat di *download* oleh mahasiswa, *upload* foto-foto pelatihan, dan yang terakhir adalah mencetak sertifikat pelatihan untuk mahasiswa yang mengikuti kegiatan pelatihan *softskill* yang telah dilaksanakan.



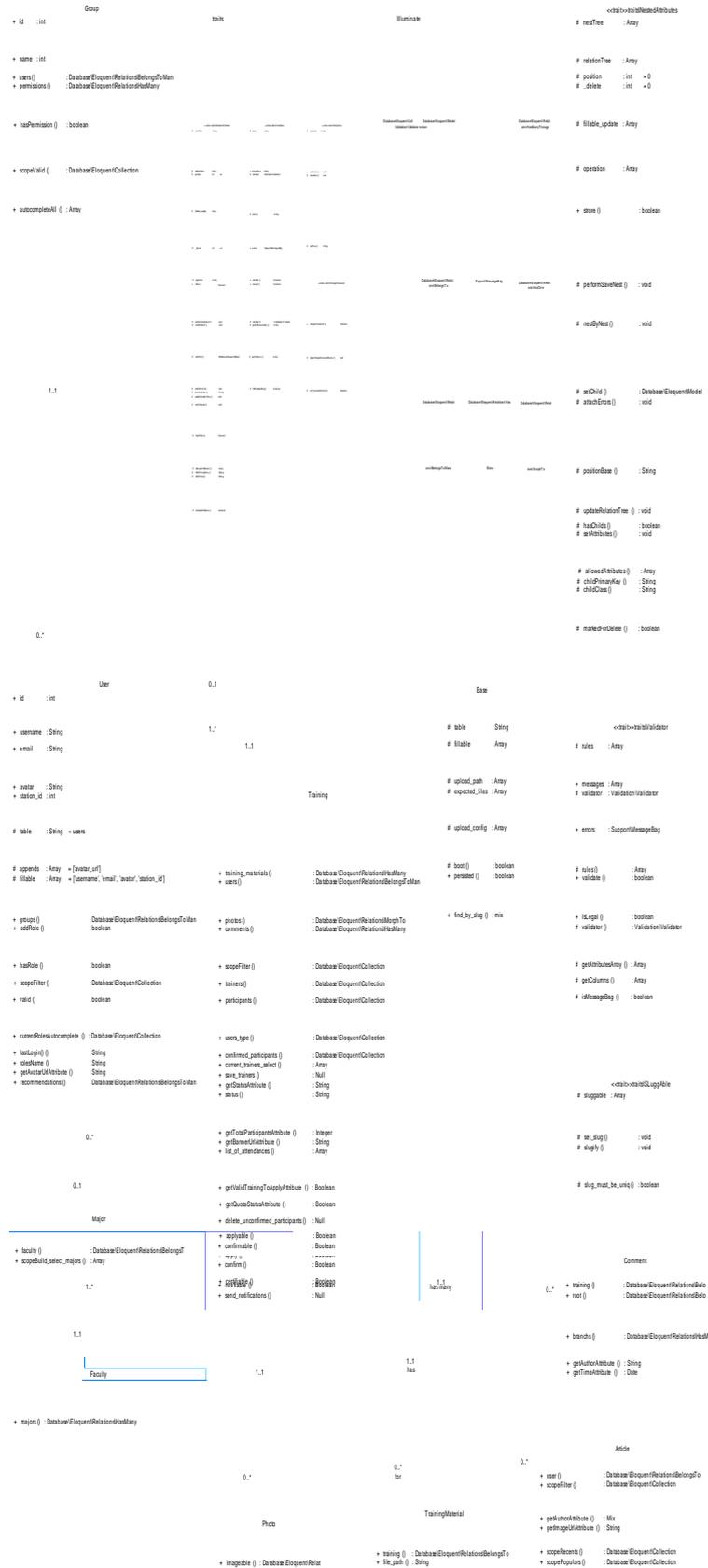
Gambar 4 Proses Bisnis Usulan

B. Perancangan Aplikasi

Berikut ini adalah rancangan aplikasi menggunakan UML (*Unified Modelling Language*) yang akan diusulkan.

1. Class Diagram

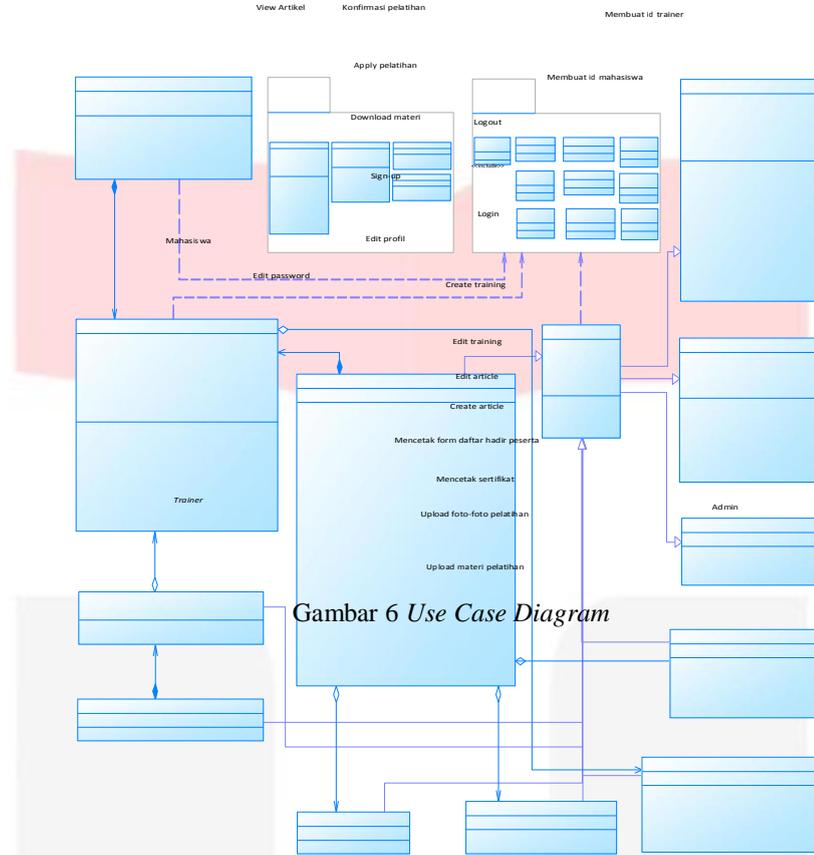
Class diagram menunjukkan kelas dan bagaimana kelas itu berhubungan satu sama lain. Kelas yang berisi objek-objek yang akan saling berinteraksi dalam aplikasi memiliki atributnya masing-masing. Atribut adalah property yang dimiliki setiap objek. Berikut adalah class diagram pada sistem yang dibangun.



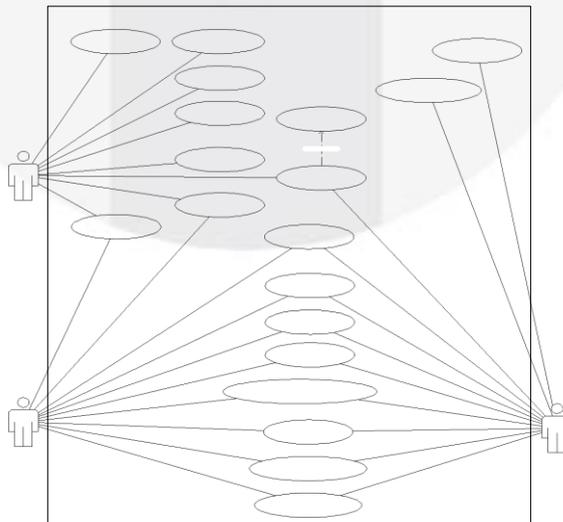
Gambar 5 Class Diagram

2. Use Case Diagram

Use case diagram adalah diagram yang menggambarkan interaksi antara pengguna (user) dengan sistem. Pengguna sistem dibagi menjadi tiga aktor, yaitu Admin, Trainer, dan Mahasiswa. Interaksi tersebut digambarkan sebagai berikut.

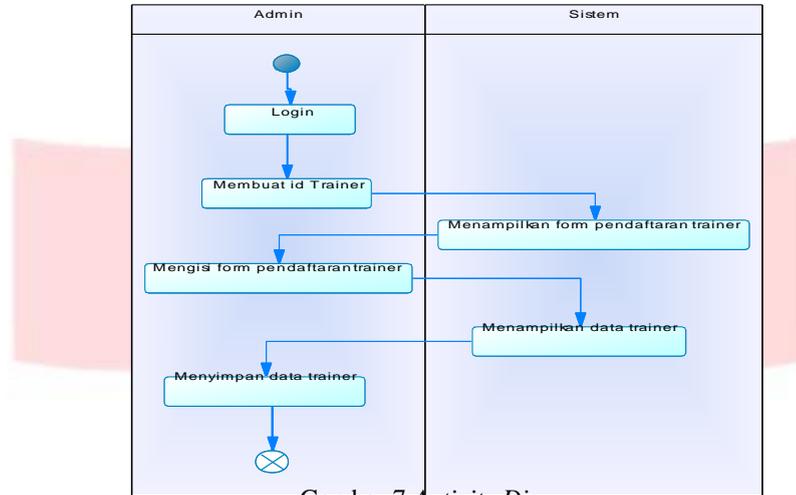


Gambar 6 Use Case Diagram



3. Activity Diagram

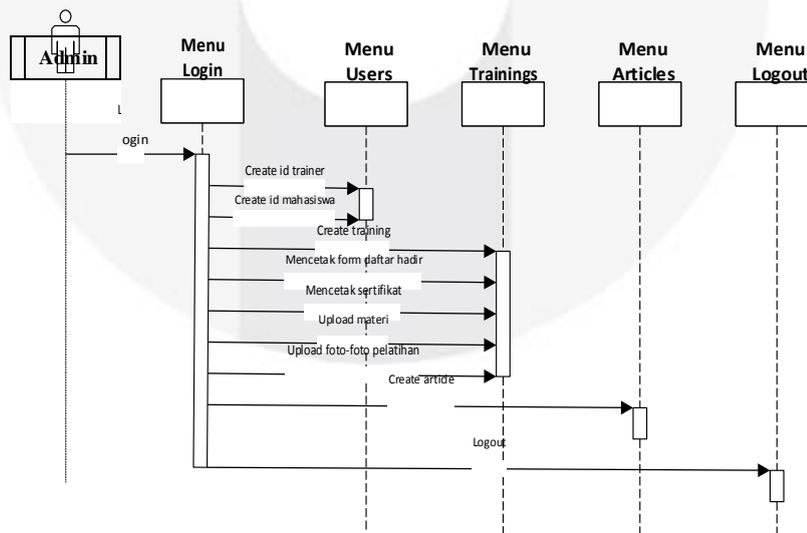
Admin harus terlebih dahulu *login* ke dalam sistem dan masuk ke halaman *users* untuk membuat id *trainer*. Kemudian sistem akan menampilkan form pendaftaran *trainer*. Selanjutnya admin mengisi form pendaftaran *trainer* dan sistem akan menampilkan data yang telah dibuat kemudian admin dapat menyimpan data *trainer*.



Gambar 7 Activity Diagram

4. Sequence Diagram

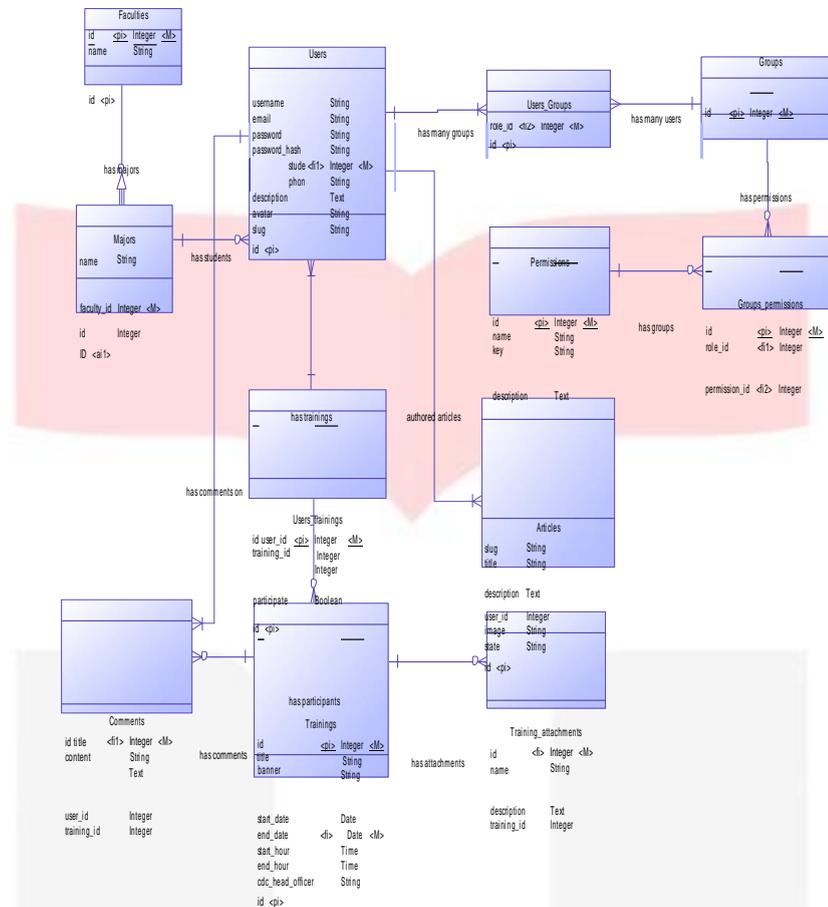
Admin harus *login* terlebih dulu pada sistem melalui menu *login* yang tersedia. Setelah *login* pada sistem, admin dapat berinteraksi dengan tiga menu utama, yaitu menu *users*, *trainings*, dan *articles*. Dalam menu *users*, admin dapat membuat id untuk *trainer* dan mahasiswa. Dalam menu *trainings*, admin dapat membuka pendaftaran pelatihan, mencetak form daftar hadir, mencetak sertifikat, *upload* materi, dan *upload* foto-foto pelatihan. Ketika pelatihan telah selesai dilaksanakan, status pelatihan akan berubah dari *ongoing* menjadi *completed*. Setelah status pelatihan ini berubah menjadi *completed*, admin dapat menjalankan fungsi mencetak sertifikat, *upload* materi, dan *upload* foto-foto pelatihan. Dalam menu *articles*, admin dapat membuat artikel yang nantinya dapat dilihat oleh *trainer* dan mahasiswa. Selanjutnya, admin dapat mengakhiri interaksi dengan sistem dengan melakukan *logout* dari sistem.



Gambar 8 Sequence Diagram

5. Entity Relationship Diagram

Entity Relationship Diagram menggambarkan hubungan antar entitas-entitas yang ada di dalam sistem. Berikut ini adalah diagram relasi antar entitas yang terdapat dalam aplikasi Pelatihan Softskill CDC Universitas Telkom.



Gambar 9 Entity Relationship Diagram

V. IMPLEMENTASI DAN PENGUJIAN

A. Implementasi

Aplikasi pelatihan *softskill* yang telah dibangun ini diharapkan dapat mengurangi beban kerja CDC Telkom University dalam perencanaan pelaksanaan pelatihan *softskill* serta dapat mengatasi masalah *redundancy* data pendaftaran peserta pelatihan yang selama ini terjadi ketika menggunakan layanan *documents* dan *spreadsheets* yang disediakan oleh Google.

B. Pengujian

Pengujian yang dilakukan yaitu pengujian fungsionalitas dan *user acceptance*. Berdasarkan hasil pengujian fungsionalitas yang dilakukan, semua fungsi yang digunakan dalam *website* sistem informasi pelatihan mahasiswa Universitas Telkom dapat berjalan dengan baik. Sedangkan berdasarkan pengujian *user acceptance* diperoleh hasil sebagai berikut.

1. Sebesar 100% dari 5 responden mengatakan bahwa tulisan di dalam website sistem informasi pelatihan mahasiswa dapat dibaca dengan mudah.
2. Sebesar 80% dari 5 responden mengatakan bahwa menu dan fungsi pada website ini dapat dimengerti dan berjalan dengan baik.
3. Sebesar 80% dari 5 responden mengatakan bahwa tidak mengalami loading yang lama ketika mengakses situs ini.
4. Sebesar 80% dari 5 responden mengatakan bahwa tertarik untuk menggunakan website ini.
5. Sebesar 100% dari 5 responden mengatakan bahwa website sistem informasi pelatihan mahasiswa Universitas Telkom memberikan informasi baru bagi penggunaannya.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan yang didapat dari pembuatan Sistem Informasi Pelatihan Mahasiswa ini adalah sebagai berikut.

1. Aplikasi ini dapat memberikan informasi jadwal pelatihan dengan memanfaatkan SMS gateway menggunakan fitur SMS.
2. Aplikasi ini telah berhasil mempermudah dan mempercepat proses bisnis di CDC.
3. Aplikasi CDC ini telah berhasil dibangun dengan nilai *acceptance* sebesar 88%.

B. Saran

Adapun saran-saran yang diberikan untuk pengembangan selanjutnya mengenai Sistem Informasi Pelatihan Mahasiswa ini adalah sebagai berikut.

1. Sistem CDC dapat diintegrasikan dengan sistem iGracias Universitas Telkom untuk memanfaatkan fitur *single account*.
2. Sistem dapat memberikan informasi melalui *email*.

DAFTAR PUSTAKA

- Bonnie, Soeherman. 2008. *Designing Information System Concept & Cases with Visio*. Jakarta: PT Elex Media Komputindo.
- Fathansyah. 2007. *Basis Data*. Bandung: Informatika.
- Fine, Michael. 2002. *Beta Testing For Better Software*. Indianapolis: Wiley Publishing, Inc.
- Hanifah R., Mita. 2012. *Sistem Informasi Manajemen konseling di CDC IT Telkom*. Bandung: Institut Teknologi Telkom.
- Husni, Thamrin. 2015. *Aplikasi Sertifikasi Kendaraan Angkutan untuk Mempersingkat Waktu Layanan*. Surakarta: Universitas Muhammadiyah Surakarta.
- Janner, S. 2006. *Menggunakan PHP dan MySQL*. Yogyakarta: CV. Andi Offset.
- Nidhra, S., & Dondeti, J. 2012. *Black Box and White Box Testing Techniques - A Literature Review*. International Journal of Embedded Systems and Applications (IJESA), 2(2), 1-2.
- Nugroho, B. 2008. *Membuat Sistem Informasi Penjualan Berbasis Web dengan PHP dan MySQL*. Yogyakarta: Gave Media.
- Pitman, D.P. 2005. *UML 2.0 in a Nutshell*. USA: O'Reilly Media.
- Pressman, R. 2009. *Software Engineering: A Practitioner's Approach, 7th International edition*. McGraw-Hill; 7th edition.
- Sabale, R. G. 2012. *Comparative Study of Prototype Model for Software Engineering with Development Life Cycle*. IOSR Journal of Engineering (IOSRJEN).
- Sidik, B. 2012. *Framework CodeIgniter*. Bandung: Informatika Bandung.
- Simarmata, J. 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi Offset.
- Quadri, S. M. K, & Farooq, Sheikh Umar. 2010. *Software Testing-Goals, Principles, and Limitations*. International Journal of Computer Applications, 6(9), 1.
- Zahra. 2010. *Sistem Pendaftaran Pelatihan di Lembaga Pengembangan Pendidikan (LPP) di Universitas Sebelas Maret Menggunakan SMS Gateway*. Surakarta: Universitas Sebelas Maret.