**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : BAYU AJI MAHENDRA PUTRA**

**NRP : 5110100013**

**DOSEN WALI : Daniel Oranova Siahaan, S.Kom, M.Sc., PDEng**

**DOSEN PEMBIMBING :1. Dr. Ir. Siti Rochimah, MT.**

**2. Rizky Januar Akbar, S.Kom, M.Eng**

# JUDUL TUGAS AKHIR

**“GITCODE: MANAJEMEN TEMPAT PENYIMPANAN KODE SUMBER BERBASIS GIT UNTUK LINGKUNGAN PERKULIAHAN”**

# LATAR BELAKANG

Beberapa lingkungan perkuliahan menerapkan sebuah proyek mahasiswa sebagai sebuah tugas terakhir yang menjadi salah satu aspek penilaian. Sebuah proyek yang diberikan biasanya merupakan proyek besar yang harus dikerjakan secara berkelompok dan memerlukan adanya kolaborasi antar anggota. Tentunya dengan adanya proyek besar yang dikerjakan berkelompok, pengajar perlu mengetahui kemajuan pekerjaan sehingga apabila terdapat permasalahan, pengajar dapat segera mengetahuinya berdasarkan perkembangan proyek. Ketika pengerjaan telah selesai, maka merupakan tugas pengajar untuk memberikan penilaian berdasarkan kerjasama kelompok dan kontribusi tiap individu sehingga penilaian dapat lebih objektif. Tentunya proyek besar yang dibuat oleh mahasiswa dapat dikembangkan lebih lanjut dan perlu adanya pengarsipan proyek berupa manajemen tempat penyimpanan kode sumber sehingga untuk dapat mengembangkan lebih lanjut dapat lebih mudah karena beberapa informasi seperti hasil akhir proyek tersebut seperti apa dan apa saja fitur yang terdapat dalam proyek tersebut dapat diketahui.

Di lingkungan Teknik Informatika sendiri, perangkat yang dapat menghubungkan tiap anggota kelompok juga dapat dimanfaatkan oleh para pengajar karena dapat memantau pengerjaan proyek tiap kelompok dengan lebih detail. Pengajar dapat melihat pekerjaan tiap individu di dalam kelompok tersebut serta dapat menilai dengan lebih mudah berdasarkan info-info yang didapat dari proses pengerjaan mahasiswa. Dengan kemudahan seperti ini tentu dapat mengurangi kesalahan penilaian tiap individu dalam kelompok.

Kebutuhan seperti ini sebenarnya bisa didapat dengan menggunakan sebuah *Version Control System* (VCS), seperti Git, Subversion, Mercurial dan VCS lainnya. Hanya saja untuk mempermudah dan dapat menyesuaikan keperluan di lingkungan perkuliahan, tentu saja kita memerlukan sebuah tampilan yang dapat memberikan informasi lebih seperti sebuah perangkat lunak yang dapat menginformasikan segala macam *history* yang ada di dalam tempat penyimpanan yang dibuat. Nantinya informasi yang didapat dari tempat penyimpanan ini dapat digunakan oleh pengajar maupun kelompok untuk melihat perkembangan proyek yang dikerjakan.

Git adalah salah satu contoh dari VCS yang menerapkan *Distributed Version Control System* (DVCS). Git dapat dimanfaatkan di lingkungan perkuliahan untuk menyimpan hasil pekerjaan mahasiswa. Selain itu Git juga memiliki kelebihan yang salah satunya adalah mampu menyimpan data tidak hanya di *server* tetapi juga di komputer *client* sehingga kemungkinan untuk kehilangan data ketika *server* mengalami gangguan.[1]

Berdasarkan kebutuhan-kebutuhan di lingkungan perkuliahan seperti yang disebutkan diatas, seperti fungsionalitas proyek, keaktifan individu di dalamnya dan lain-lain dapat ditampilkan dengan lebih mudah dengan sebuah perangkat lunak yang dapat membantu pengarsipan proyek mahasiswa dan manajemennya. Perangkat lunak yang dimaksud dapat berupa web sehingga pengguna dapat mengaksesnya dengan mudah seperti Gitcode yang diharapkan dapat membantu mengatasi berbagai macam masalah yang terkait dengan manajemen informasi proyek yang di simpan di tempat penyimpanan arsip proyek sehingga memudahkan pengajar maupun dosen untuk mendapatkan info proyek dan dosen juga dapat lebih mudah memberikan penilaian individu kelompok. Berbeda dengan GitHub, BitBucket, dan lainnya, Gitcode dapat digunakan untuk memantau fitur apa saja yang telah ditambahkan sehingga sesuai dengan kebutuhan perkuliahan ketika pengajar atau anggota kelompok ingin memantau hasil perkembangan pengerjaan tiap anggota lainnya. Dengan mengetahui kontribusi tiap anggota, maka pengajar dapat memberikan penilaian individu lebih objektif.

# RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut :

1. Bagaimana membuat web manajemen kode sumber yang dapat memfasilitasi pekerjaan proyek perangkat lunak mahasiswa secara kolaboratif.
2. Bagaimana membuat sebuah web manajemen kode sumber yang dapat memfasilitasi dosen pengajar dalam memantau kemajuan proyek perangkat lunak mahasiswa di perkuliahan.

# BATASAN MASALAH

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut :

1. Bahasa pemrograman menggunakan ASP.
2. Menggunakan *browser* untuk menjalankan perangkat lunak.
3. Manajemen sistem mengikuti kebutuhan perkuliahan.
4. *Client* dapat diakses melalui Git Client yang telah tersedia.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari tugas akhir ini adalah sebagai berikut :

1. Membuat perangkat lunak yang dapat membantu pekerjaan yang dikerjakan secara berkelompok.
2. Membuat perangkat lunak yang dapat mempermudah pengawasan pembagian pekerjaan mahasiswa.
3. Membuat perangkat lunak yang dapat membantu proses penilaian oleh pengajar di lingkungan perkuliahan.
4. Perangkat lunak dapat mewadahi pekerjaan mahasiswa sehingga dapat digunakan di kemudian hari.

# MANFAAT TUGAS AKHIR

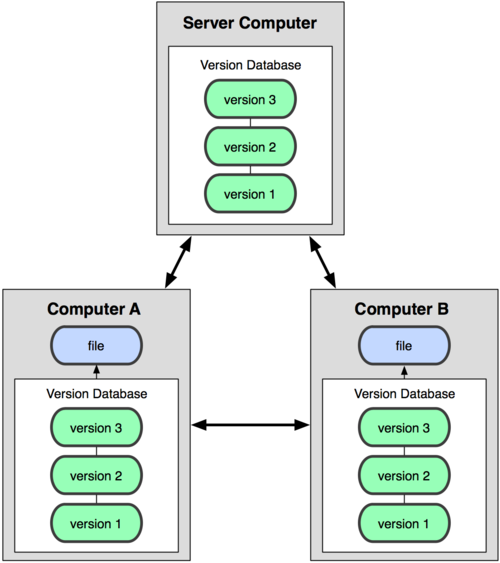
Manfaat yang diharapkan dari tugas akhir ini adalah :

1. Mempermudah pengerjaan secara berkelompok di Teknik Informatika.
2. Mempermudah penilaian secara perorangan dalam sebuah kelompok oleh pengajar.
3. Mempermudah pengawasan kemajuan pekerjaan mahasiswa.
4. Meningkatkan kemampuan mahasiswa bekerja dalam kelompok dan bertanggung jawab dengan tugasnya masing-masing.

# TINJAUAN PUSTAKA

* 1. **Distributed Version Control System (DVCS)**

Dalam sebuah DVCS, seluruh *client* tidak hanya akan melakukan *check out* dari *snapshot* terakhir sebuah dokumen, tetapi *client* juga akan memiliki info yang sama dengan yang terdapat di dalam *server.* Sistem seperti ini akan membantu *client* untuk mendapatkan dokumen terakhir yang disimpan ketika *server* sedang mengalami masalah. Beberapa contoh DVCS adalah Git, Mercurial, Bazaar atau Darcs. Berikut adalah gambaran dari cara kerja DVCS. [1]



Gambar 1. *Distributed Version Control Diagram*

* 1. **Git**

Git adalah salah satu contoh perangkat yang mengatur penyimpanan tempat penyimpanan dari version sebuah data. Berbeda dengan Sentralize Version Control System (SVCS), Git berjalan dengan dengan DVCS sehingga seluruh data yang terdapat di *server* akan disimpan di *client.* Perbedaan paling penting dari sebuah Git dengan VCS lain adalah dalam hal pengiriman data yang berubah. Pada VCS lain, data yang dikirim atau disimpan adalah data secara keseluruhan, sedangkan pada Git, data yang dikirim atau disimpan hanyadata yang mengalami perubahan, kemudian data yang tidak mengalami perubahan hanya akan disimpan *link* data tersebut. Cara Git untuk menyimpan informasi file juga berbeda dengan VCS lain. Perubahan yang disimpan di Git hanya sebuah *snapshot* atau sebuah informasi kecil dari file, sedangkan VCS lain menyimpan segala perubahan yang terjadi. [1]

Hampir keseluruhan lingkungan kerja di Git dilakukan secara local, sehingga tidak perlu mengkhawatirkan masalah koneksi internet. Apabila pada VCS lain memerlukan koneksi ke *server* apabila ingin melakukan commit file, commit pada Git dapat dilakukan kapan saja bahkan ketika tidak ada koneksi internet sehingga informasi perubahan file akan tercatat secara local pula. [1]

* 1. **GitSharp API**

GitSharp adalah sebuah *library* yang digunakan pada proyek-proyek.NET. *Library* ini digunakan untuk membantu proyek berhubungan dengan tempat penyimpanan Git. GitSharp dapat membantu pengguna untuk membuat sebuah proyek yang berhubungan dengan pengambilan data pada sebuah tempat penyimpanan Git. Salah satu proyek yang menggunakan GitSharp untuk menghubungkan Git dengan perangkat lunaknya adalah Bonobo Git Server,[2]

* 1. **ASP .NET MVC**

ASP .NET MVC adalah bagian dari kerangka kerja aplikasi web ASP .NET yang merupakan satu dari dua model pemprograman untuk membuat aplikasi web ASP .NET. Berikut adalah penjelasan dari *Model*, *View* dan *Controller* yang diterapkan. [3]

* *Model* : *Model* mengandung informasi inti untuk aplikasi. *Model* mengandung data dan aturan-aturan validasi seperti akses data dan logika agregasi.
* *View*: *View* membungkus tampilan dari aplikasi, dan didalamnya secara umum berisi markup HTML.
* *Controller*: *Controller* berisi logika pengaturan alur (*control-flow*) aplikasi. *Controller* ini yang menghubungkan antara *Model* dan *Views* untuk mengatur alur informasi dan jalannya aplikasi.

# RINGKASAN ISI TUGAS AKHIR

Perangkat lunak ini adalah perangkat yang berjalan pada aplikasi *browser.* Perangkat lunak ini dibuat untuk membantu *client* mengatur pengerjaan proyek mereka. Tujuan lain adalah agar pekerjaan-pekerjaan yang dikerjakan oleh mahasiswa maupun dosen di lingkungan Teknik Informatika ITS dapat tersimpan dengan baik dan dapat digunakan di lain waktu. Selain itu juga para pengguna lain dapat mengunduh proyek-proyek yang dipublikasikan oleh pengguna lainnya. Dengan adanya perangkat lunak ini, diharapkan nantinya pengerjaan kelompok-kelompok mahasiswa dapat dinilai dengan lebih baik melihat dari seberapa aktif individu dari tiap kelompok. Hal ini dapat membantu pengajar untuk menilai dengan lebih mudah karena dapat melihat keaktifan tiap mahasiswanya. Manfaat lain juga dapat membantu mahasiswa mempublikasikan hasil pekerjaan mereka karena terdapat fitur yang nantinya dapat menampilkan hasil pengerjaan proyek mahasiswa.

Dalam penggunaannya, ada beberapa aktor yang dapat menggunakan perangkat lunak ini, seperti dosen, mahasiswa dan pengguna umum. Pengguna umum nantinya merupakan pengguna tanpa harus login terlebih dahulu untuk dapat menggunakan perangkat lunak ini, sehingga pengguna ini hanya dapat menggunakan fitur minimum seperti mengunduh proyek yang dipublikasikan. Selain pengguna umum, ada beberapa pengguna yang harus login terlebih dahulu sebelum dapat memanfaatkan fitur-fitur lain seperti dosen dan mahasiswa. Mereka dapat menggunakan fitur perangkat lunak secara menyeluruh terkecuali fitur pembuatan kelas yang hanya dimiliki oleh dosen.



Gambar 2. Diagram kasus penggunaan Gitcode

Penjelasan fitur aplikasi yang ada adalah sebagai berikut :

1. **Melakukan perubahan profil**

Fitur ini berguna bagi pengguna yang telah terdaftar untuk dapat melakukan perubahan kata sandi mereka dan hal-hal lain yang nantinya berhubungan dengan perorangan.

1. **Melihat *history* tempat penyimpanan**

Fitur ini digunakan untuk melihat apa saja yang telah terjadi di dalam tempat penyimpanan tersebut, seperti perubahan-perubahan yang terjadi di dalam tempat penyimpanan.

1. **Menampilkan hasil pengerjaan.**

Fitur ini berguna untuk mempublikasikan hasil dari proyek yang telah dibuat oleh pengguna. Nantinya fitur ini akan menampilkan beberapa gambaran hasil akhir sehingga pengguna lain dapat melihat hasil pengerjaan dari proyek tersebut.

1. **Mengunduh data yang dipublikasikan.**

File yang telah diselesaikan oleh pengguna dan di publikasikan untuk umum dapat diunduh oleh pengguna lainnya. Nantinya aka ada beberapa file yang dapat di publikasikan atau tidak oleh pengguna.

1. **Membuat kelompok**

Kelompok ini adalah kumpulan pengguna yang terdapat di dalam sebuah kelas dengan mentor yang merupakan pengajar di kelas tersebut. Mentor nantinya akan secara otomatis dimasukkan ke dalam kelompok agar dapat melihat pengerjaan kelompok tersebut.

1. **Melihat detail history perorangan pada history tempat penyimpanan**

Fungsi dari fitur ini adalah agar pengajar dapat melihat bagaimana performa setiap individu di dalam kelompok sehingga dapat memudahkan pengajar dalam memberikan penilaian.

1. **Membuat *issue***

Fitur ini sama dengan fitur komentar dari pengguna termasuk segala macam masalah yang ada di dalam proyek tersebut, sehingga ketika permasalahan telah diselesaikan, pengguna lain dapat mengkonfirmasi komentar tersebut.

1. **Membuat *tempat penyimpanan***

Membuat tempat penyimpanan adalah fungsi penting agar para pengguna dapat terhubung dengan pengguna lain dengan menggunakan Git Client yang telah ada.

1. **Membuat *job desk* untuk user lainnya di dalam kelompok yang telah dibuat**

Membuat *job desk* di sini adalah posisi individu di dalam kelompok tersebut, seperti manajer proyek, *designer*, *programmer* atau *tester*. Pemberian *job desk* seperti ini juga dapat memudahkan pengajar dalam memberikan penilaian apakah individu dalam kelompok tersebut telah menyelesaikan tugasnya dengan baik.

1. **Memasukkan fitur yang harus dipenuhi dalam pengerjaan**

Fitur ini nantinya dapat digunakan untuk menentukan tingkat keberhasilan pembuatan perangkat lunak oleh pengajar kepada setiap mahasiswanya. Fitur ini memberikan penjelasan apa saja yang dapat dilakukan pengguna dengan perangkat lunak yang telah dibuat.

1. **Membuat informasi mengenai pekerjaan yang sedang dibuat**

Fitur ini memberikan sedikit info mengenai perangkat lunak yang akan dibangun oleh pengguna, sehingga pengguna lain dapat mengetahui manfaat atau deskripsi lain dari perangkat lunak tersebut.

1. **Menambahkan kelas**

Fitur ini hanya dapat digunakan oleh pengajar. Nantinya pengajar akan menambahkan sebuah kelas dan mahasiswa akan masuk ke dalam kelas tersebut.

# METODOLOGI

## Penyusunan proposal tugas akhir

Tahap ini merupakan tahap perumusan masalah hingga penyelesaian yang ditawarkan. Selain itu tahap ini akan memberikan gambaran perangkat lunak yang akan dibangun untuk menyelesaikan permasalahan tersebut. Disamping itu juga ada beberapa metodologi yang dijelaskan untuk mengerjakan solusi yang ditawarkan.

## Studi literatur

Pengejaan tugas akhir ini memerlukan beberapa poin yang harus dimengerti sebelum mengerjakannya. Beberapa poin tersebut adalah mengerti jenis perangkat lunak yang dapat menyesuaikan dengan lingkungan Teknik Informatika. Setelah mengerti bagaimana perangkat lunak yang tepat untuk diaplikasikan di Teknik Informatika, tahapan berikutnya adalah mempelajari bahasa pemrograman yang digunakan untuk membuat perangkat lunak. Setelah itu mengerti bagaimana melakukan komunikasi antara *server* dengan perangkat lunak yang dibuat.

## Analisis dan desain perangkat lunak

Tahap ini nantinya akan menghasilkan desain yang berasal dari analisa permasalahan yang didapat beserta jalan keluarnya. Analisa nantinya akan meliputi bagaimana sistem yang cocok diterapkan di Teknik Informatika. Hasil analisa yang telah dilakukan akan menghasilkan desain perangkat lunak yang nantinya akan membantu dalam melakukan implementasi.

## Implementasi perangkat lunak

Berikut beberapa hal yang diperlukan dalam implementasi :

* 1. Bahasa yang digunakan adalah ASP .NET.
  2. IDE yang digunakan adalah Visual Studio 2012.
  3. Rangka kerja yang digunakan adalah ASP .NET
  4. Desain perangkat lunak yang menyesuaikan dengan Teknik Informatika.
  5. Menggunakan API GitSharp.

## Pengujian dan evaluasi

Proses pengujian akan dilakukan dengan mengambil beberapa mahasiswa Teknik Informatika untuk menggunakan perangkat lunak dalam mencoba keberhasilan fungsional sistem.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka
7. **JADWAL KEGIATAN**

Tabel 1. Jadwal Kegiatan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan |  |  | |  | |  | | 2014 | | | | | | | | | | | | | |
| Februari | | | | | | | | Maret | | | | April | | | | Mei | | | |
| Analisa kebutuhan dan studi literature |  | |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan system |  | |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  | |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji coba dan evaluasi |  | |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  | |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |  |

1. **DAFTAR PUSTAKA**

[1] Chacon, S. (2010). *Pro Git.*

[2] *Eqqon*. (2011, May 26). Retrieved from http://www.eqqon.com/index.php/GitSharp

[3] Jon Galloway, P. H. (2012 ). *PROFESSIONAL ASP.NET MVC 4.* Canada: John Wiley & Sons, Inc.