

**APLIKASI PENENTUAN PENGAJAR DAN
PENGAMPU**

TUGAS AKHIR

Oleh :

DIAH SHINTA RINI

3311201004

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2015**

HALAMAN PENGESAHAN

APLIKASI PENENTUAN PENGAJAR DAN PENGAMPU

Oleh :

Diah Shinta Rini (3311201004)

Tugas Akhir ini telah diterima dan disahkan
sebagai persyaratan untuk memperoleh gelar

Ahli Madya

di

**PROGRAM STUDI DIPLOMA 3 TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM**

Batam, Agustus 2015

Disetujui oleh;

Pembimbing,

Hilda Widyastuti

NIK. 19770512 201212 2 001

HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : Diah Shinta Rini

Nama : 3311201004

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

APLIKASI PENENTUAN PENGAJAR DAN PENGAMPU

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, Agustus 2015

Diah Shinta Rini
3311201004

KATA PENGANTAR

Alhamdulillah puji syukur kehadirat Allah SWT, karena atas segala rahmat dan karunia-Nya penulis dapat menyelesaikan penyusunan Tugas Akhir ini. Adapun Tugas Akhir ini yang berjudul "Penentuan Pengajar dan Pengampu" sebagai syarat untuk memperoleh nilai dan juga sebagai salah satu alat ukur untuk mengetahui kompetensi produksi mahasiswa pada jurusan Teknik Informatika Politeknik Negeri Batam.

Dalam menyusun Tugas Akhir ini, begitu banyak rintangan dan halangan yang dihadapi penulis selama proses penyelesaiannya, akan tetapi berkat banyaknya bantuan dan dukungan dari berbagai pihak sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Oleh karena itu dalam kesempatan ini, penulis ingin mengucapkan terima kasih kepada :

1. Kedua Orang Tua dan keluarga yang selalu memberikan dukungan, do'a, dan motivasi.
2. Ibu Hilda Widyastuti selaku dosen pembimbing yang telah memberi bantuan dan masukan selama penyelesaian Tugas Akhir ini.
3. Pak Riwinoto selaku dosen penguji pada Tugas Akhir ini.
4. Seluruh rekan-rekan penulis baik dari dalam maupun luar kampus Politeknik Negeri Batam khususnya jurusan Teknik Informatia angkatan 2012 yang tidak bisa penulis sebutkan namanya satu persatu yang telah membantu dan memberikan semangat kepada penulis dalam penyelesaian Tugas Akhir ini.

Semoga laporan ini dapat berguna bagi pembaca, memberikan pemikiran baru yang berguna bagi pembaca, yang dapat disumbangkan untuk pengembangan ilmu dan memberikan manfaat bagi pihak yang membutuhkannya.

Batam, Agustus 2015

Penulis

ABSTRAK

APLIKASI PENENTUAN PENGAJAR DAN PENGAMPU

Aplikasi Penentuan Pengajar dan Pengampu ini mengembangkan aspek *user interface* interaktif pada tugas akhir sebelumnya "Sistem Informasi Identifikasi Kompetensi Dosen Prodi Teknik Informatika" oleh (Yogi & Siregar, 2013) dimana sistem informasi kompetensi dosen ini dapat melakukan identifikasi kompetensi dosen dengan cara menentukan tingkat pemahaman dengan angka pada silabus yang dikuasai, pengelolaan user, memasukkan data silabus, menyeleksi dosen pengampu dan pengajar dan menampilkan *summary* dan nilai tingkat pemahaman dosen pada masing-masing silabus.

Dari Penelitian sebelumnya dapat dilihat masih ada masalah pada seleksi dosen yang belum optimal maka dari itu aplikasi sistem informasi identifikasi kompetensi dosen dikembangkan dengan menambahkan fitur-fitur untuk meningkatkan fungsionalitas dengan menyediakan fitur menginput data jabatan dosen, menginput beban minimal dan maksimal dosen berdasarkan jabatan, melakukan *setting* kelas, menghasilkan *output* file *csv*, dan memilih pengampu dan pengajar. Aspek *user interface* interaktif diukur dengan menggunakan 10 daya guna *heuristik* sehingga aplikasi dapat memberikan kemudahan bagi user dalam menggunakan fitur aplikasi. Aplikasi ini sudah memenuhi 8 dari 10 daya guna *heuristik*. Aplikasi ini dikembangkan dengan menggunakan *Yii Framework*, *MySQL* sebagai *database* dan *JavaScript*.

Kata Kunci: *Yii Framework*, *User Interface*, Aplikasi

ABSTRACT

DETERMINATION OF TEACHING AND SUPPORTING LECTURER APPLICATIONS

Applications Determination of Teachers and supporting lecturer developing interactive user interface aspects of the department head at the end of the previous task "Identification Information Systems Informatics Engineering Department Lecturer Competence" (Yogi & Siregar, 2013), where the information system competence of lecturers is able to identify the competence of lecturers by determining the level understanding the numbers on the syllabus controlled, user management, enter data syllabus, lecturers and teachers select and display summary and lecturer at the level of understanding the value of each syllabus.

In the previous study can be seen there are still problems in the selection of lecturers is not optimal, therefore the application identification information systems faculty competence is developed by adding features to increase functionality by providing lectureship features of input data, input the minimum and maximum load lecturer based office, classroom setting, generating output csv file, and choose supporting lecturer and teachers. Aspects of interactive user interface is measured using the usability heuristics 10 so that the application can make it easy for the user to use the application features. This application has met 8 of 10 usability heuristics. This application is developed using Yii Framework, MySQL as database and JavaScript.

Keywords: Yii Framework, User Interface, Application

DAFTAR ISI

KATA PENGANTAR	4
DAFTAR ISI.....	7
DAFTAR GAMBAR	9
DAFTAR TABEL.....	10
BAB I PENDAHULUAN	11
1.1 Latar Belakang.....	11
1.2 Rumusan Masalah.....	13
1.3 Batasan Masalah	13
1.4 Tujuan.....	13
1.5 Sistematika Penulisan	13
BAB II LANDASAN TEORI.....	15
2.1 Penelitian Terkait.....	15
2.2 Dosen	16
2.3 Antarmuka Pengguna.....	17
2.3.1 Prinsip Utama Mendesain Antarmuka.....	18
2.4 Yii Framework	26
2.5 HTML.....	26
2.6 PHP.....	27
2.7 MySQL	27
2.8 JavaScript.....	29
2.9 Cascading Style Sheets (CSS)	29
BAB III ANALISIS DAN PERANCANGAN.....	31
3.1 Deskripsi Umum Sistem.....	31
3.2 Analisis Sistem	32
3.3 Use case	33
3.3.1 Diagram <i>Use case</i>	34
3.3.2 Skenario <i>Use case</i>	34
3.4 Sequence Diagram	36
3.4.1 Sequence Diagram <i>Login</i>	36
3.4.2 Sequence Diagram Menginput Data Jabatan Dosen.....	37
3.4.4 Sequence Diagram Setting Kelas.....	38
3.4.5 Sequence Diagram Memilih Pengampu	38
3.4.6 Sequence Diagram Memilih Pengajar	39

3.4.7	Sequence Diagram Mendownload File.....	39
3.5	Class Diagram.....	39
3.6	ER Diagram	41
3.7	Spesifikasi Tabel.....	42
3.7.1	Spesifikasi Tabel <i>User</i>	42
3.7.2	Spesifikasi Tabel Data Dosen.....	42
3.7.3	Spesifikasi Tabel Beban	42
3.7.4	Spesifikasi Tabel Mata kuliah	43
3.7.5	Spesifikasi Tabel Silabus.....	43
3.7.6	Spesifikasi Tabel Pengajar.....	44
3.7.7	Spesifikasi Tabel Pengampu.....	44
3.7.8	Spesifikasi Tabel Kelas	44
3.8	Perancangan Antarmuka.....	45
BAB IV IMPELEMANISASI DAN PENGUJIAN.....		51
4.1	Implementasi Antarmuka.....	51
4.1.1	Implementasi Antar Muka <i>Login</i>	51
4.1.2	Implementasi Data Jabatan Dosen.....	52
4.1.3	Implementasi Tampilan Beban	52
4.1.4	Implementasi Setting Kelas	53
4.1.5	Implementasi Memilih Pengajar.....	53
4.1.6	Implementasi Memilih Pengampu	54
4.1.7	Implementasi <i>Download</i>	55
4.2	Implementasi Konsep <i>User Interface</i>	56
4.3	Hasil Pengujian.....	59
BAB V KESIMPULAN DAN SARAN.....		65
5.1	Kesimpulan.....	65
5.2	Saran	65
DAFTAR PUSTAKA		66

DAFTAR GAMBAR

Gambar 3. 1 Deskripsi Umum Sistem Aplikasi	31
Gambar 3. 2 Diagram <i>Use case</i> Aplikasi Penentuan Pengajar dan Pengampu	34
Gambar 3. 3 Sequence Diagram <i>Login</i>	36
Gambar 3. 4 Sequence Diagram Menginput Jabatan Dosen	37
Gambar 3. 5 Sequence Diagram Input Beban	37
Gambar 3. 6 Sequence Diagram Setting Kelas	38
Gambar 3. 7 Sequence Diagram Memilih Pengampu	38
Gambar 3. 8 Sequence Diagram Memilih Pengajar	39
Gambar 3. 9 Diagram Sequence Mendownload <i>file</i>	39
Gambar 3. 10 Class Diagram	40
Gambar 3. 11 ER Diagram	41
Gambar 3. 12 Perancangan tampilan <i>login</i>	45
Gambar 3. 13 Perancangan Tampilan <i>input</i> jabatan dosen	46
Gambar 3. 14 Perancangan tampilan input beban	46
Gambar 3. 15 Perancangan tampilan <i>setting</i> kelas	47
Gambar 3. 16 Perancangan Antarmuka memilih pengampu	48
Gambar 3. 17 Perancangan tampilan memilih pengajar	49
Gambar 4. 1 Antarmuka <i>Login</i>	51
Gambar 4. 2 Antarmuka Input Jabatan Dosen	52
Gambar 4. 3 Antarmuka Beban	53
Gambar 4. 4 Antarmuka <i>Setting</i> Kelas	53
Gambar 4. 5 Antarmuka Memilih Pengajar	54
Gambar 4. 6 Antarmuka Memilih Pengajar	55
Gambar 4. 7 Antarmuka <i>Download</i>	55

DAFTAR TABEL

Tabel 3. 1 Kebutuhan fungsional dan non fungsional.....	33
Tabel 3. 2 Spesifikasi <i>User</i>	42
Tabel 3. 3 Spesifikasi Data Dosen	42
Tabel 3. 4 Spesifikasi Beban	43
Tabel 3. 5 Spesifikasi Matakuliah	43
Tabel 3. 6 Spesifikasi Silabus	43
Tabel 3. 7 Spesifikasi Pengajar	44
Tabel 3. 8 Spesifikasi Pengampu	44
Tabel 3. 9 Spesifikasi Kelas	44
Tabel 3. 10 Deskripsi Tampilan <i>login</i>	45
Tabel 3. 11 Deskripsi Tampilan <i>input</i> jabatan dosen.....	46
Tabel 3. 12 Deskripsi tampilan <i>input</i> beban.....	47
Tabel 3. 13 Deskripsi tampilan <i>setting</i> kelas	48
Tabel 3. 14 Deskripsi tampilan memilih pengampu	49
Tabel 3. 15 Deskripsi tampilan memilih pengajar	50
Tabel 4.1 Implementasi <i>User Interface</i> pada aplikasi	56
Tabel 4. 2 Hasil Pengujian	59

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dosen merupakan seseorang yang berdasarkan pendidikan dan keahliannya diangkat oleh penyelenggara perguruan tinggi dengan tugas utama sebagai pengajar. Menurut Undang–Undang Guru dan Dosen nomor 14 tahun 2005, dosen adalah pendidik profesional dari ilmuwan dengan tugas utama mentransformasikan, mengembangkan dan menyebarluaskan ilmu pengetahuan, teknologi dan seni melalui pendidikan, penelitian dan pengabdian masyarakat. Kinerja dosen diukur berdasarkan beban kerja dosen mencakup kegiatan pokok yaitu merencanakan pembelajaran, melaksanakan proses pembelajaran, melakukan evaluasi pembelajaran, membimbing dan melatih, melakukan penelitian, melakukan pengabdian pada masyarakat dan melakukan tugas tambahan. Beban kerja paling sedikit dengan 12 (dua belas) SKS dan paling banyak 16 (enam belas) SKS pada setiap semester sesuai dengan kualifikasi akademik. Dari beban yang sudah ditentukan tentunya sudah diperkirakan untuk dosen agar beban yang ditentukan sesuai dengan kegiatan dan kinerja dosen sehari-hari, maka dari itu diharapkan dosen tidak lagi melakukan proses penentuan pengajar dan pengampu dengan proses manual.

Dalam melaksanakan kegiatan belajar mengajar pada jurusan Teknik Informatika Politeknik Negeri Batam ada beberapa bagian seperti Tata Usaha (TU-IF) adalah suatu badan yang membantu Kepala Jurusan (Kajur). Bagian TU-IF melaksanakan identifikasi kompetensi dosen dan hasilnya akan diberikan kepada Kajur untuk menunjuk dosen pengampu dan pengajar berdasarkan mata kuliah yang dikuasai sesuai dengan panduan batas minimal dan maksimal. Dosen memiliki batas minimal dan maksimal sesuai jabatan strukturalnya misalnya dosen yang menjabat kudir, beban maksimal dan minimalnya berbeda dengan dosen biasa. Kajur akan menempatkan dosen untuk mengajar kelas yang ada.

Selanjutnya akan dilakukan perekapan data matakuliah dan dosen untuk mengetahui dosen yang sesuai dengan mata kuliah yang diajarkan dengan menggunakan *Microsoft Excel*. Dengan kemampuan manusia yang terbatas, kesalahan dalam proses kerja tentu tidak dapat dihindari dengan banyaknya kinerja dosen yang dilakukan sehingga beban dosen dalam bekerja meningkat, kesalahan yang terjadi dapat menyebabkan penyajian informasi yang tidak akurat.

Dari sekian banyak permasalahan dalam kinerja dosen dengan proses manual tentunya dibutuhkan aplikasi, tetapi permasalahan banyaknya aplikasi dengan desain *interface* yang membuat *user* merasa asing dalam menggunakan aplikasi, sehingga *user* terkadang membutuhkan waktu yang lebih lama untuk beradaptasi dengan aplikasi tersebut. Dunia teknologi berkembang sangat pesat, banyak dari perusahaan dan instansi pemerintahan yang berlomba-lomba untuk merancang sebuah aplikasi yang mudah dalam penggunaannya.

Berdasarkan permasalahan diatas muncul ide untuk membangun perancangan yang memberikan informasi untuk menentukan pengajar dan pengampu. Pada Tugas Akhir sebelumnya sudah pernah dilakukan oleh (Yogi & Siregar, 2013) dengan judul Sistem Informasi Identifikasi Kompetensi Dosen.

Dilihat dari penelitian sebelumnya dapat dikembangkan pada aspek *user interface* Kajar sehingga penelitian ini bertujuan untuk mengembangkan *user interface* yang interaktif. Suatu sistem interaktif adalah *user* mencapai suatu tujuan tertentu dalam suatu domain aplikasi. Untuk mencapai tujuan tersebut, sebuah sistem interaktif harus dapat di daya gunakan (*usable*). Perancangan *User Interface* yang Interaktif diharapkan agar *user* yang menggunakan aplikasi dapat dengan mudah memahami maksud dari tata letak berbagai macam navigasi yang ada pada aplikasi.

1.2 Rumusan Masalah

Perumusan masalah yang dapat diidentifikasi dalam permasalahan di atas adalah :

1. Bagaimana merancang *user interface* yang interaktif
2. Bagaimana merancang dan membuat aplikasi penentuan pengajar dan pengampu.

1.3 Batasan Masalah

Adapun rumusan masalah dalam penelitian ini adalah Aplikasi penentuan pengajar dan pengampu ini hanya digunakan untuk keperluan Politeknik Negeri Batam dan berlaku sesuai dengan prosedur yang berlaku di Politeknik Negeri Batam.

1.4 Tujuan

Adapun tujuan yang diperoleh dalam penelitian ini adalah :

1. Merancang dan membuat *user interface* yang interaktif
2. Merancang dan membangun aplikasi penentuan pengajar dan pengampu

1.5 Sistematika Penulisan

Sistematika penulisan ini ditulis untuk memberikan kemudahan informasi bagi pembaca dan memberikan gambaran dalam mempelajari dan memahami isi dari Tugas Akhir.

BAB 1 PENDAHULUAN

Bab ini menguraikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan diadakan penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menguraikan tentang teori yang melandasi pengertian dosen, *user interface*, penelitian terkait dan uraian tentang tinjauan singkat perangkat lunak digunakan.

BAB III ANALISIS DAN PERANCANGAN

Bab ini menguraikan tentang deskripsi umum sistem, perancangan *use case diagram*, *sequence diagram*, dan *class diagram* yang dibuat untuk menjelaskan analisa serta penjelasan pada program tugas akhir ini.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini menguraikan tentang hasil dari implementasi, pengujian dan pembahasan pada aplikasi yang dibuat.

BAB V KESIMPULAN DAN SARAN

Bab ini menguraikan tentang kesimpulan atas aplikasi yang telah diselesaikan dan saran yang bermanfaat untuk hasil aplikasi tersebut.

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Penelitian yang berhubungan dengan penelitian penentuan pengajar dan pengampu sudah pernah dilakukan. Seperti yang telah dibuat pada Tugas Akhir sebelumnya (Yogi & Siregar, 2013) dengan judul “Sistem Informasi Identifikasi Kompetensi Dosen Prodi Teknik Informatika Politeknik Negeri Batam”. Sistem informasi identifikasi kompetensi dosen dapat menampilkan *summary* nilai tingkat pemahaman dengan angka pada silabus yang dikuasai, mengidentifikasi kompetensi dosen dengan cara dosen akan menentukan tingkat pemahaman dengan memilih angka pada silabus yang dikuasai, identifikasi kompetensi kelola silabus dan seleksi dosen tersebut akan menampilkan nilai dan matrik kompetensi. Serta KPS dapat menyeleksi dosen untuk ditunjuk sebagai pengampu materi pengajar. Berdasarkan dari referensi tersebut maka dibuatlah aplikasi “Penentuan Pengajar dan Pengampu” untuk meningkatkan kemampuan fitur *user interface* yang interaktif dan menambahkan beberapa fungsionalitas sebagai berikut :

1. Menambahkan fungsionalitas TU menginput jabatan dosen.
2. Menambahkan fungsionalitas TU yang dapat menginput beban minimal dan maksimal berdasarkan jabatan.
3. Menambahkan fungsionalitas TU dapat melakukan *setting* kelas.
4. Menghasilkan aplikasi yang dapat menghasilkan *output* berbentuk *file* Csv yang bisa langsung *diimport* ke aplikasi FET (Aplikasi Penjadwalan).
5. Menghasilkan *user interface* interaktif yang akan menampilkan pesan pemberitahuan bila batas pengajar dan pengampu yang dipilih Kajar telah melebihi batas atau kurang dari batas minimal.
6. Kajar dapat memilih pengajar dan pengampu.

2.2 Dosen

Dosen adalah salah satu komponen esensial dalam suatu sistem pendidikan di perguruan tinggi, peran, tugas dan tanggungjawab dosen sangat penting dalam mewujudkan tujuan pendidikan nasional, yaitu mencerdaskan kehidupan bangsa, meningkatkan kualitas manusia Indonesia, yang meliputi kualitas iman/takwa, akhlak mulia, dan penguasaan ilmu pengetahuan, teknologi dan seni, serta mewujudkan masyarakat Indonesia yang maju adil, makmur, dan beradab. Untuk melaksanakan fungsi, peran dan kedudukan yang sangat strategis tersebut, diperlukan dosen yang profesional. Tugas utama dosen adalah melaksanakan tridharma perguruan tinggi dengan beban kerja paling sedikit dengan 12 (dua belas) SKS dan paling banyak 16 (enam belas) SKS pada setiap semester sesuai dengan kualifikasi akademik. Sedangkan profesor atau guru besar adalah dosen dengan jabatan akademik tertinggi pada satuan pendidikan tinggi dan mempunyai tugas khusus menulis buku dan karya ilmiah serta menyebarkan luaskan gagasannya untuk mencerahkan masyarakat. Pelaksanaan tugas utama dosen ini perlu dievaluasi dan dilaporkan secara periodik sebagai bentuk akuntabilitas kinerja dosen kepada para pemangku kepentingan. Menurut Undang-Undang nomor 14 tahun 2005 bab 1 pasal 1, dosen adalah pendidik profesional dan ilmuwan dengan tugas utama mentransformasikan, mengembangkan, dan menyebarkan ilmu pengetahuan, teknologi, dan seni melalui pendidikan, penelitian, dan pengabdian kepada masyarakat. Dosen memiliki kedudukan sebagai tenaga profesional pada jenjang pendidikan tinggi yang diangkat sesuai dengan peraturan perundang-undangan yang dikemukakan dalam Undang-undang 14 tahun 2005 bab 2 pasal 3.

Kinerja dosen diukur berdasarkan beban kerja dosen mencakup kegiatan pokok yaitu merencanakan pembelajaran, melaksanakan proses pembelajaran, melakukan evaluasi pembelajaran, membimbing dan melatih, melakukan penelitian, melakukan pengabdian pada masyarakat dan melakukan tugas tambahan. Kinerja dosen merupakan salah satu faktor penentu keberhasilan proses belajar mengajar di perguruan tinggi. Kelebihan beban kerja dapat membuat

seseorang menjadi stres, stres merupakan keadaan wajar karena merupakan salah satu respon manusia di kehidupan sehari-hari. Beban kerja seorang dosen meliputi beban kerja secara fisik maupun secara mental. Menurut Menpan (Endang Hendrayanti, 2010:16), pengertian beban kerja adalah sekumpulan atau sejumlah kegiatan yang harus diselesaikan oleh suatu organisasi atau pemegang jabatan dalam jangka waktu tertentu. Akibat beban kerja yang terlalu berat atau kemampuan fisik yang terlalu lemah, terkadang dapat mengakibatkan seorang karyawan mengalami gangguan atau hambatan untuk melaksanakan pekerjaan sesuai dengan prosedur yang telah ditetapkan. Akan tetapi beban kerja dapat didukung dengan lingkungan kerja yang nyaman sehingga akan berdampak positif bagi orang tersebut.

Terkadang dosen-dosen memiliki jabatan lain di dalam struktur organisasi sebuah universitas, selain menjabat sebagai dosen, orang tersebut juga menjabat sebagai ketua jurusan. Hal ini menyebabkan tanggung jawab yang diterima lebih besar sehingga menyebabkan suatu beban kerja. Beban kerja yang ditanggung oleh seorang dosen berbeda dengan beban yang ditanggung oleh dosen lain. Sebuah pencapaian seorang dosen tidak dapat diukur secara umum karena harus disesuaikan dengan beban kerja yang ditanggung oleh setiap dosen tersebut.

2.3 Antarmuka Pengguna

Bidang ilmu interaksi manusia dan komputer adalah ilmu yang mempelajari tentang bagaimana mendesain, mengevaluasi, dan mengimplementasikan sistem komputer yang interaktif sehingga dapat digunakan oleh manusia dengan mudah. Pengertian interaksi adalah komunikasi dua arah antara manusia (pengguna) dan sistem komputer. Interaksi menjadi maksimal apabila kedua belah pihak mampu memberikan stimulan dan respon (aksi dan reaksi) yang saling mendukung, jika salah satu tidak bisa, maka interaksi akan mengalami hambatan atau bahkan menuju pembiasan tujuan. Manusia baru menyadari proses interaksi tersebut saat menemukan masalah dan tidak menemukan solusi pemecahannya. Biasanya manusia menyalahkan antarmuka

yang kurang inovatif, kurang menarik, kurang komunikatif. Interaksi bisa dikatakan dialog antara pengguna dengan komputer.

2.3.1 Prinsip Utama Mendesain Antarmuka

Pada aplikasi ini akan diterapkan prinsip-prinsip desain antarmuka. Berikut ini beberapa hal yang menjadi prinsip utama mendesain antarmuka yang baik dengan memperhatikan karakteristik manusia dan komputer (Deborah, 1992):

1. User compatibility

Antarmuka merupakan sebuah pintu gerbang masuk ke sistem dengan diwujudkan ke dalam sebuah aplikasi *software*. Oleh karena itu sebuah *software* seolah-olah mengenal penggunanya, mengenal karakteristik penggunanya, dari sifat sampai kebiasaan manusia secara umum. Desainer harus mencari dan mengumpulkan berbagai karakteristik serta sifat dari pengguna karena antarmuka harus disesuaikan dengan pengguna yang jumlahnya bisa jadi lebih dari satu dan mempunyai karakter yang berbeda. Hal tersebut harus terpikirkan oleh desainer dan tidak dianjurkan merancang antarmuka dengan didasarkan pada dirinya sendiri. *Survey* adalah hal yang paling tepat.

2. Product compatibility

Sebuah aplikasi yang bertopengkan antarmuka harus sesuai dengan sistem aslinya. Seringkali sebuah aplikasi menghasilkan hasil yang berbeda dengan sistem manual atau sistem yang ada. Hal tersebut sangat tidak diharapkan dari perusahaan karena dengan adanya aplikasi *software* diharapkan dapat menjaga produk yang dihasilkan dan dihasilkan produk yang jauh lebih baik. Contoh: Aplikasi sistem melalui antarmuka diharapkan menghasilkan *report* atau laporan serta informasi yang detail dan akurat dibandingkan dengan sistem manual.

3. Task compatibility

Sebuah aplikasi yang bertopengkan antarmuka harus mampu membantu para pengguna dalam menyelesaikan tugasnya. Semua pekerjaan serta tugas-tugas pengguna harus diadopsi di dalam aplikasi tersebut melalui antarmuka. Sebisa mungkin pengguna tidak dihadapkan dengan kondisi memilih dan berpikir, tapi pengguna dihadapkan dengan pilihan yang mudah dan proses berpikir dari tugas-tugas pengguna dipindahkan dalam aplikasi melalui antarmuka. Contoh : Pengguna hanya klik *set up*, tekan tombol *next, next, next, finish, ok* untuk menginstal suatu *software*.

4. *Work flow compatibility*

Sebuah aplikasi sistem sudah pasti mengadopsi sistem manualnya dan didalamnya tentunya terdapat urutan kerja dalam menyelesaikan pekerjaan. Dalam sebuah aplikasi, *software engineer* harus memikirkan berbagai runutan-rununtan pekerjaan yang ada pada sebuah sistem. Jangan sampai pengguna mengalami kesulitan dalam menyelesaikan pekerjaannya karena pengguna mengalami kebingungan ketika urutan pekerjaan yang ada pada sistem manual tidak ditemukan pada *software* yang dihadapinya. Selain itu pengguna jangan dibingungkan dengan pilihan-pilihan menu yang terlalu banyak dan semestinya menu-menu merupakan urutan dari runutan pekerjaan. Sehingga dengan *work flow compatibility* dapat membantu seorang pengguna dalam mempercepat pekerjaannya.

5. *Consistency*

Sebuah sistem harus sesuai dengan sistem nyata serta sesuai dengan produk yang dihasilkan. Banyak perusahaan dalam menjalankan sistemnya menggunakan aplikasi sistem yang berbeda di setiap divisi dalam perusahaan tersebut. Ada pula yang menggunakan aplikasi yang sama di divisi yang berbeda seringkali keseragaman dalam menjalankan sistem tidak diperhatikan. Oleh karena itu *software engineer* harus memperhatikan hal-hal yang bersifat konsisten pada saat merancang

aplikasi khususnya antarmuka, contoh : penerapan warna, struktur *menu*, *font*, format desain yang seragam pada antarmuka di berbagai bagian, sehingga pengguna tidak mengalami kesulitan pada saat berpindah posisi pekerjaan atau berpindah lokasi dalam menyelesaikan pekerjaan. Hal itu didasarkan pada karakteristik manusia yang mempunyai pemikiran yang menggunakan analogi serta kemampuan manusia dalam hal memprediksi. Contoh : keseragaman tampilan toolbar pada Word, Excell, PowerPoint, Access hampir sama.

6. *Familiarity*

Sifat manusia mudah mengingat dengan hal-hal yang sudah sering dilihatnya atau didapatkannya. Secara singkat disebut dengan familiar. Antarmuka sebisa mungkin didesain sesuai dengan antarmuka pada umumnya, dari segi tata letak, model, dan sebagainya. Hal ini dapat membantu pengguna cepat berinteraksi dengan sistem melalui antarmuka yang *familiar* bagi pengguna.

7. *Simplicity*

Kesederhanaan perlu diperhatikan pada saat membangun antarmuka. Tidak selamanya antarmuka yang memiliki menu banyak adalah antarmuka yang baik. Kesederhanaan disini lebih berarti sebagai hal yang ringkas dan tidak terlalu berbelit. Pengguna akan merasa jengah dan bosan jika pernyataan, pertanyaan dan menu bahkan informasi yang dihasilkan terlalu panjang dan berbelit. Pengguna lebih menyukai hal-hal yang bersifat sederhana tetapi mempunyai kekuatan atau bobot.

8. *Direct Manipulation*

Pengguna berharap aplikasi yang dihadapinya mempunyai media atau tools yang dapat digunakan untuk melakukan perubahan pada antarmuka tersebut. Pengguna ingin sekali aplikasi yang dihadapannya bisa disesuaikan dengan kebutuhan, sifat dan karakteristik pengguna tersebut.

Selain itu, sifat dari pengguna yang suka merubah atau mempunyai rasa bosan. Contoh : Tampilan warna sesuai keinginan (misal *pink*) pada window bisa dirubah melalui *desktop properties*, tampilan skin winamp bisa dirubah, dan lain-lain.

9. *Control*

Prinsip *control* ini berkenaan dengan sifat pengguna yang mempunyai tingkat konsentrasi yang berubah-ubah. Hal itu akan sangat mengganggu proses berjalannya sistem. Kejadian salah ketik atau salah *entry* merupakan hal yang biasa bagi seorang pengguna. Akan tetapi hal itu akan dapat mengganggu sistem dan akan berakibat sangat fatal karena salah memasukkan data 1 digit atau 1 karakter saja informasi yang dihasilkan sangat dimungkinkan salah. Oleh karena itu *software engineer* haruslah merancang suatu kondisi yang mampu mengatasi dan menanggulangi hal-hal seperti itu. Contoh : “illegal command”, “can’t recognize input” sebagai portal jika terjadi kesalahan.

10. WYSIWYG

WYSIWYG (*what you see is what you get*) atau apa yang didapat adalah apa yang dilihatnya. Contoh : Apa yang tercetak di *printer* merupakan informasi yang terkumpul dari data-data yang terlihat di layar monitor pada saat mencari data. Hal ini juga perlu menjadi perhatian *software engineer* pada saat membangun antarmuka. Informasi yang dicari atau diinginkan harus sesuai dengan usaha dari pengguna pada saat mencari data dan juga harus sesuai dengan data yang ada pada aplikasi sistem (*software*). Jika sistem mempunyai informasi yang lebih dari yang diinginkan pengguna, hendaknya dibuat pilihan (*optional*) sesuai dengan keinginan pengguna. Bisa jadi yang berlebihan itu justru tidak diinginkan pengguna. Yang mendasar disini adalah harus sesuai dengan kemauan dan pilihan dari pengguna.

11. *Flexibility*

Fleksibel merupakan bentuk dari dari solusi pada saat menyelesaikan masalah. *Software engineer* dapat membuat berbagai solusi penyelesaian untuk satu masalah. Sebagai contoh adanya menu, hotkey, atau model dialog yang lainnya.

12. *Responsiveness*

Setelah memberikan *inputan* atau memasukkan data ke aplikasi sistem melalui antarmuka, sebaiknya sistem langsung memberi tanggapan atau respon dari hasil data yang *diinputkan*. Selain teknologi komputer semakin maju sesuai dengan tuntutan kebutuhan manusia, *software* yang dibangun pun harus mempunyai reaksi tanggap yang cepat. Hal ini didasari pada sifat manusia yang semakin dinamis atau tidak mau menunggu.

13. *Invisible Technology*

Secara umum, pengguna mempunyai keingintahuan sebuah kecanggihan dari aplikasi yang digunakannya. Untuk itu aplikasi yang dibuat hendaknya mempunyai kelebihan yang tersembunyi. Bisa saja kelebihan itu berhubungan dengan sistem yang melingkupinya atau bisa saja kecanggihan atau kelebihan itu tidak ada hubungannya. Contoh : sebuah aplikasi mempunyai *voice recognize* sebagai media *inputan*, pengolah kata yang dilengkapi dengan *language translator*.

14. *Robustness*

Interaksi manusia dan komputer (pembangunan antarmuka) yang baik dapat berupa frase-frase menu atau *error handling* yang sopan. Kata yang digunakan harus dalam kondisi bersahabat sehingga nuansa *user friendly* akan dapat dirasakan oleh pengguna selama menggunakan sistem . Contoh yang kurang baik: You False !!!, Bad Files!!!, Floppy Error, dan sebagainya. Akan lebih baik jika Bad Command Or Files Names, Disk Drive Not Ready, dan lain-lain.

15. *Protection*

Suasana nyaman perlu diciptakan oleh *software engineer* di antarmuka yang dibangunnya. Nyaman disini adalah suasana dimana pengguna akan betah dan tidak menemui suasana kacau ketika pengguna salah memasukkan data atau salah eksekusi. Seorang pengguna akan tetap merasa nyaman ketika dia melakukan kesalahan, misal ketika pengguna melakukan deleting atau menghapus *files* tanpa sengaja tidaklah menjadi kekacauan yang berarti karena misal ada *recovery tools seperti undo, recycle bin*, dan lain-lain atau “are you sure....” Proteksi disini lebih menjaga kenyamanan pengguna ketika menggunakan aplikasi sistem khususnya data-data berupa *file*.

16. *Ease Of Learning And Ease Of Use*

Kemudahan dalam mengoperasikan *software* hanya dengan memandangi atau belajar beberapa jam saja. Kemudahan dalam memahami *icon*, menu-menu, alur data *software*, dan sebagainya. Sesudah mempelajari, pengguna dengan mudah dan cepat menggunakan *software* tersebut. Jika sudah memahami tentunya akan membantu proses menjalankan sistem dengan cepat dan baik.

Menurut ISO 1998, salah satu faktor yang digunakan untuk mengukur sejauh mana penerimaan pengguna terhadap sistem adalah daya guna. Ada 9 daya guna *heuristik* untuk mengukur sistem sejauh mana diterima oleh pengguna sebagai berikut:

1) Dialog yang sederhana dan alami

User interface harus ringkas mungkin dan bersifat natural. Setiap dialog seharusnya menghindari perintah-perintah yang tidak perlu dan tidak ada hubungannya dengan interface, karena untuk setiap ciri atau elemen baru yang ditambahkan berarti satu masalah baru yang harus dipelajari oleh pengguna.

Pendekatan yang harus digunakan adalah :

- Hanya menampilkan perintah yang diperlukan

- Bentuk elemen grafik dalam user interface modern
- Penggunaan warna yang baik dan tidak berlebihan
- Desain layar dalam bentuk yang lebih ringkas
- Dialog yang natural

2) Berbicara dengan bahasa pengguna

Dialog yang seharusnya digunakan adalah bahasa yang dipahami oleh user. Perintah-perintah yang berorientasi mesin mudah dipahami dan dihindari multitafsir kata sehingga membuat user keliru. Objek yang tampil di layar, jenis perintah, jenis interaksi pengguna, cara sistem memberikan *feedback* dan sebagainya adalah berdasarkan frasa yang biasa digunakan, misalnya *desktop*, *icon*, *menu*, *cut*, *copy*, *paste*.

3) Mengurangi beban ingatan user

User seharusnya tidak dibebani untuk mengingat atau menghafal pada saat berinteraksi dengan sistem. Sebagai contoh penggunaan menu dapat mengurangi beban user dibandingkan penggunaan baris perintah. Aplikasi yang menggunakan menu lebih memuaskan dan fleksibel. Dalam kasus-kasus tertentu format perintah perlu disampaikan dengan jelas, misalnya perintah DOS untuk menghapus dengan *del* dan membuat duplikasi dengan perintah *copy*.

4) Konsisten

Ciri-ciri konsisten adalah dapat menghindarkan user dari rasa was-was atau ragu-ragu di saat menggunakan suatu perintah atau fungsi. Disamping itu juga dapat mempercepat interaksi, misalnya perintah cetak dari windows dengan *File > Print*.

5) Sistem timbal balik

Sistem seharusnya memberitahu pengguna segala aktifitas yang sedang berlaku atau status dari sistem. Status sistem menunggu input dari pengguna,

memproses input, menampilkan output, dan sebagainya. Proses ini juga akan memberitahu status suatu sistem jika terjadi suatu kerusakan.

6) Jalan keluar yang jelas

Sistem seharusnya dapat memberikan penjelasan tentang kondisi dan solusi untuk menghindari user terperangkap dalam tampilan-tampilan yang tidak diinginkan, aktivitas atau situasi dalam berinteraksi dengan sistem. Apabila user melakukan kesalahan dalam memilih perintah maka ia dapat keluar dari kesalahan tanpa ada masalah, misalnya perintah *Undo*.

7) Jalan pintas

Demi kemudahan dan kecepatan interaksi di dalam menggunakan suatu sistem maka sudah seharusnya bila tersedia *shortcut* yang berguna untuk membantu user agar dapat menggunakan berbagai fungsi dengan mudah.

8) Pesan-pesan kesalahan yang baik

Menyediakan mekanisme pemberitahuan kesalahan dan memerhatikan beberapa pesan sebagai berikut :

- Pesan kesalahan yang digunakan harus jelas dan mudah dipahami, disampaikan dalam bentuk teks, frasa atau konsep yang mudah dipahami
- Pesan yang disampaikan bersifat khusus
- Pesan kesalahan yang disampaikan sebaiknya menyediakan cadangan penyelesaian atas kesalahan
- Penyampaian kesalahan dilakukan secara sopan.

9) Mencegah kesalahan

Rekayasa interface yang baik seharusnya mampu membuat user menghindari kesalahan, misalnya interaksi dengan menggunakan menu.

10) Bantuan dan dokumentasi

Bantuan dan dokumentasi merupakan kemudahan yang diberikan sistem untuk menjelaskan cara penggunaan, ciri-ciri khusus sistem dan membolehkan user untuk mengendalikan sistem dengan lebih baik, misalnya bantuan *Help*.

2.4 Yii Framework

Nama *Yii* adalah kependekan dari *Yes, it is*, dan dibaca Yee atau ji. *Yii* memiliki performa tinggi, *component-based, framework* untuk aplikasi web yang ditulis dalam bahasa PHP5. Nama *Yii* juga diartikan berdasarkan sifatnya, seperti *easy, efficient, dan extensible*. *Yii Framework* (Kerangka kerja) adalah PHP berbasis komponen, berkinerja tinggi untuk pengembangan aplikasi Web berskala besar. *Yii* menyediakan *reusability* maksimum dalam pemrograman Web dan mampu meningkatkan kecepatan pengembangan secara signifikan. *Project* ini pertama kali dikembangkan oleh Qiang Xue yang memulai pembuatan open source *framework* ini pada 1 Januari 2008. Qiang sebelumnya telah membangun dan memelihara PRADO *Framework* selama bertahun-tahun sebelum memulai *Yii*. Bertahun-tahun pengalamannya dan umpan balik pengguna yang diperoleh dari proyek PRADO untuk mengembangkan *Yii* agar lebih mudah digunakan, mudah dikembangkan dan lebih efisien PHP5-based *framework* serta memenuhi perkembangan yang diharapkan dari pengembangan aplikasi. *Yii* versi alpha secara resmi dirilis pada Oktober 2008. *Yii* memiliki keunggulan yang mengesankan ketika dibandingkan dengan PHP5-based *framework* lainnya dan banyak perhatian-perhatian positif yang datang.

2.5 HTML

HTML (*Hypertext Markup Language*) adalah bahasa markup yang digunakan untuk menggantikan text dokumen kedalam halaman-halaman web. HTML memberikan penulis untuk mengidentifikasi elemen yang membuat dokumen terstruktur, seperti heading, paragraf, penomoran, dan banyak lagi. Elemen yang lain bertindak sebagai mekanisme untuk

penambahan tautan hypertext, form interaktif, dan media seperti audio dan video kedalam halaman web.

XHTML adalah reformula dari HTML di XML. Dengan kata lain, ia digunakan pada arti yang sama (elemen dan atribute yang sama) sebagai HTML, tetapi aturan sintaks diambil dari XML.

2.6 PHP

Menurut Welling dan Thompson (2003, p2), PHP adalah bahasa *scripting* untuk sisi *server* yang dirancang secara khusus untuk *web*. Dalam halaman HTML dapat dimasukkan kode-kode PHP yang akan dijalankan setiap kali halaman tersebut dieksekusi. Kode-kode PHP akan diinterpretasikan pada *server web* dan menghasilkan HTML atau *output* lainnya yang akan dilihat oleh pengunjung *web*.

PHP disusun tahun 1994 dan merupakan hasil kerja keras satu orang, Rasmus Lerdorf. Kemudian dilanjutkan oleh orang-orang lain dan telah melewati tiga kali penyusunan ulang secara besar untuk memberikan hasil produk yang matang seperti yang ada sekarang ini. Pada januari 2001, PHP digunakan hampir lima juta daerah di seluruh dunia, dan jumlah ini terus bertambah.

PHP merupakan produk *open source*. PHP awalnya berarti *Personal Home Page*, tetapi diubah dengan penamaan konvensi rekursif GNU dan sekarang PHP ialah *PHP Hypertext Preprocessor*. Beberapa dari pesaing PHP ialah *Perl*, *Microsoft Active Server Pages (ASP)*, *Java Server Pages (JSP)* dan *Allaire Cold Fusion*.

2.7 MySQL

Menurut Hardjono (2006, p181) basis data MySQL merupakan sistem manajemen berbasis data SQL yang sangat terkenal dan bersifat *open source*. MySQL dibangun, didistribusikan dan didukung oleh MySQL AB. MySQL AB merupakan perusahaan komersial yang dibiayai oleh pengembang (*developer*) MySQL. MySQL dapat didefinisikan sebagai :

- MySQL merupakan sistem manajemen *database*. *Database* merupakan struktur penyimpanan data. Untuk menambah, mengakses, dan memproses

data yang disimpan dalam sebuah *database* komputer. Diperlukan sistem manajemen *database* seperti MySQL Server.

- MySQL merupakan sistem manajemen *database* atau basis data terhubung (*relational database management system*). *Database* terhubung menyimpan data pada tabel-tabel terpisah. Hal tersebut akan menambah kecepatan dan fleksibilitasnya. Kata SQL pada MySQL merupakan singkatan dari *Structured Query Language*. SQL merupakan bahasa standar yang digunakan untuk mengakses *database* dan ditetapkan oleh ANSI/ISO SQL Standard.
- MySQL merupakan *software open source*. *Open Source* berarti semua orang diijinkan menggunakan dan memodifikasi model. Semua orang dapat *download software* MySQL dari internet dan menggunakannya tanpa membayar. Anda dapat mempelajari *Source Code* dan menggunakannya sebagai kebutuhan.
- *Server database* MySQL mempunyai kecepatan akses tinggi, mudah digunakan, dan andal. MySQL dikembangkan untuk menangani *database* yang besar secara cepat dan telah sukses digunakan selama bertahun-tahun. Konektivitas, kecepatan, dan keamanannya membuat *server* MySQL cocok untuk mengakses *database* di internet.

Fitur utama MySQL adalah:

- Ditulis dalam bahasa C dan C++
- Bekerja dalam berbagai platform (misalnya Mac Os X, Solaris, Sun OS, Unix, Novel Netware, Windows, dan lain-lain)
- Menyediakan mesin penyimpanan (*engine storage*) transaksi dan nontransaksi *server* tersedia sebagai program yang terpisah untuk digunakan pada lingkungan jaringan klien/*server*.
- MySQL mempunyai *library* yang dapat ditempelkan pada aplikasi yang berjalan sendiri, sehingga aplikasi tersebut dapat digunakan pada komputer yang tidak mempunyai jaringan.
- Mempunyai sistem *password* yang fleksibel dan aman

- Dapat menangani basis data dalam skala besar, basis data dalam *server* MySQL dapat berisi 50 juta *record*.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Pada MySQL sebuah *database* terdiri atas tabel-tabel. Sebuah tabel terdiri atas tabel dan kolom.

2.8 JavaScript

JavaScript adalah *object-based* dan *prototype-based* bahasa pemrograman yang dimulai sejak 1995 dibuat oleh Brendan Eich pada Netscape. Awalnya dinamai “Mocha” dan kemudian menjadi “Live Script” sebelum membingungkan hal ini kembali diputuskan namanya menjadi JavaScript. Perlu diingat bahwa *JavaScript* bukanlah Java meskipun keduanya mirip dengan bahasa pemrograman C yang memiliki sintaks yang sama. Bahasa pemrograman *JavaScript* telah distandarisasi pada tahun 1996, ketika itu diserahkan kepada European Computer Manufacturer’s Association (ECMA) untuk pemeliharaan dan pembangunan. *JavaScript* merupakan bahasa pengkodean *client-side*, yang artinya dia berjalan pada mesin *client* bukan pada mesin *server*. Ini merupakan perbedaan yang penting yang mana implementasinya bergantung pada kemampuan dan konfigurasi *browser* ketika membuka halaman.

2.9 Cascading Style Sheets (CSS)

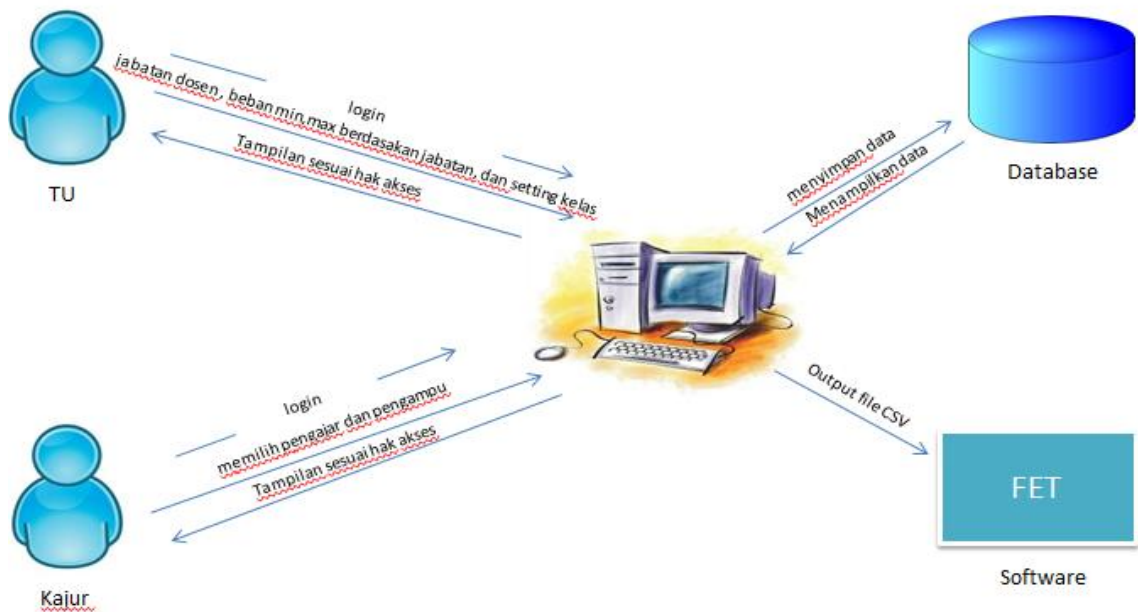
Cascading Style Sheets (CSS) adalah standar W3C untuk mendefinisikan presentasi dari dokumen web. Presentasi artinya cara dokumen ditampilkan atau dikirimkan kepada pengguna, apakah kepada monitor komputer, tampilan seluler *phone*, dibacakan oleh pembaca tampilan. CSS adalah mekanisme untuk tipe style instructions kepada dokumen yang telah ditandai dengan XHTML, HTML, atau bahasa XML lainnya. Yang terpenting, CSS tetap menjaga presentasi tersebut terpisah dari konten dan struktural dan semantic markup.

Keuntungan CSS antara lain:

- Tipografi terbaik dan pengaturan tata letak dapat menspesifikasi fitur tipografi tradisional yang tidak akan dapat dilakukan oleh HTML sendiri.
- Kerja minimal memungkinkan merubah tampilan seluruh dokumen hanya dengan satu dokumen CSS.
- Dokumen berpotensi kecil tulisan tag dan tabel dengan satu dokumen terpusat dapat mengurangi ukuran dokumen HTML.

BAB III ANALISIS DAN PERANCANGAN

3.1 Deskripsi Umum Sistem



Gambar 3. 1 Deskripsi Umum Sistem Aplikasi

Deskripsi sistem pada gambar 3.1 menjelaskan proses kerja sistem aplikasi penentuan pengajar dan pengampu yang meliputi :

1. TU melakukan hak akses ke aplikasi. TU menginput data jabatan dosen, menginput beban minimal, maksimal dosen berdasarkan jabatan dan melakukan *setting* kelas.
2. Kajar melakukan hak akses ke aplikasi. Kajar memilih pengajar dan pengampu. Dalam pemilihan pengajar dan pengampu dibuat sebuah *user interface* interaktif yang akan menampilkan pesan bahwa pengajar dan pengampu yang dipilih Kajar telah melebihi batas mengajar dosen berdasarkan beban dosen tersebut.
3. Aplikasi menghasilkan *output* berbentuk *file* CSV yang bisa langsung di *import* ke *software* FET (*software* penjadwalan).

Pengguna Aplikasi Penentuan Pengajar dan pengampu dikelompokkan menjadi 3 yaitu :

1. Tata Usaha (TU)

Yaitu pengguna yang memiliki hak akses menginput data jabatan dosen, menginput beban minimal, maksimal dosen berdasarkan jabatan dan melakukan *setting* kelas

2. Dosen

Yaitu pengguna yang memiliki hak akses mengidentifikasi kompetensi dengan cara memilih angka tingkatan pemahaman pada masing-masing silabus.

3. Kepala Jurusan (Kajur)

Yaitu pengguna yang memiliki hak akses untuk memilih pengajar dan pengampu. Tugas Akhir ini difokuskan untuk mengembangkan *user interface* agar menjadi *user interface* yang interaktif. Pada saat Kajur menentukan pengampu dan pengajar maka Kajur harus memperhatikan syarat-syarat tertentu, yaitu :

- 1) Pengajar harus memenuhi syarat minimal kompetensi tertentu.
- 2) Pengampu harus memenuhi syarat minimal kompetensi tertentu.
- 3) Dalam pemberian penugasan, Kajur harus memperhatikan jabatan struktural dan beban kerja tidak boleh melebihi batas maksimal atau kurang dari batas minimal.

Penggunaan *user interface* yang interaktif dapat membantu Kajur untuk memenuhi syarat-syarat tersebut dengan membuat suatu *user interface* interaktif .

3.2 Analisis Sistem

Analisis sistem sangat dibutuhkan dalam mendukung kinerja sistem, apakah sistem yang dibuat telah sesuai dengan kebutuhan atau belum, karena kebutuhan sistem akan mendukung tercapainya dalam fungsionalitas dan tujuan pembuatan sistem. Untuk mempermudah analisis sistem dalam menentukan keseluruhan kebutuhan secara lengkap, analisis membagi kebutuhan sistem menjadi dua jenis, yaitu kebutuhan fungsional, dan kebutuhan nonfungsional.

Analisis kebutuhan fungsional dan non fungsional dapat ditunjukkan pada tabel 3.1.

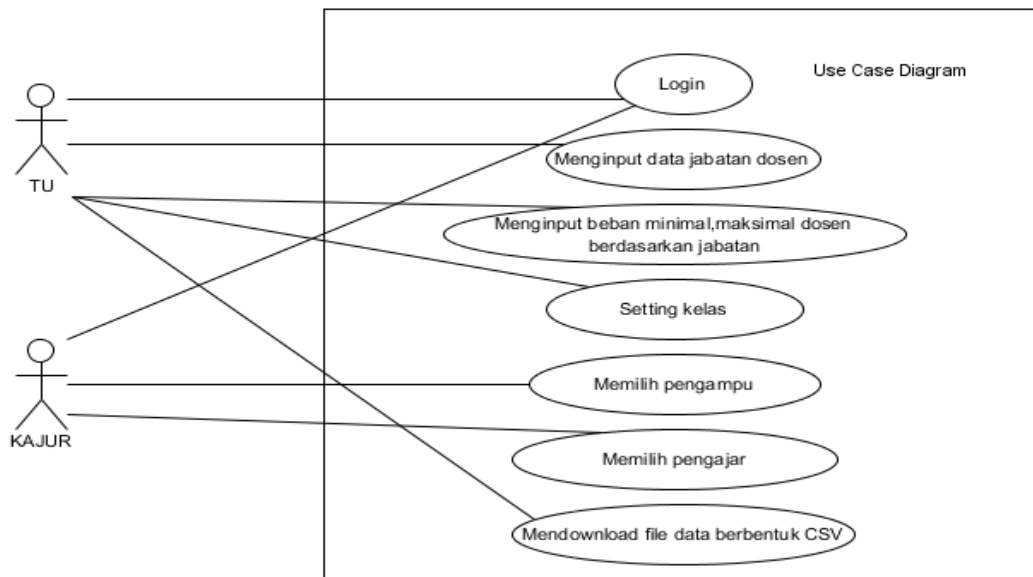
Tabel 3. 1 Kebutuhan fungsional dan non fungsional

Kode	Kebutuhan Fungsional
F-001	Aplikasi dapat menangani <i>login</i> dan <i>logout</i> untuk memastikan bahwa hanya pengguna yang sudah terdaftar saja yang dapat mengakses aplikasi
F-002	Aplikasi dapat menampilkan hasil <i>output</i> data dengan <i>download</i>
F-003	Pengguna dapat menginput data jabatan dosen
F-004	Pengguna dapat menginput beban minimal, maksimal dosen berdasarkan jabatan
F-005	Pengguna dapat melakukan <i>Setting</i> Kelas
F-006	Pengguna dapat memilih pengampu
F-007	Pengguna dapat memilih pengajar
Kode	Kebutuhan Non Fungsional
NF-001	Aplikasi ini menggunakan Bahasa Inggris

3.3 Use case

Use case menjelaskan interaksi yang terjadi antara *user* dengan fungsi aplikasi yang ada, seperti yang terlihat pada gambar 3.2. Sebuah *Use case* dipresentasikan dengan urutan langkah yang sederhana.

3.3.1 Diagram Use case



Gambar 3. 2 Diagram Use case Aplikasi Penentuan Pengajar dan Pengampu

3.3.2 Skenario Use case

Adapun skenario *use case* pada perancangan *use case* diagram diatas sebagai berikut :

1. Skenario Use case Login

User : Dosen, TU dan Kajur

Kondisi awal : Belum melakukan *login*

Kondisi Akhir : Tampilan halaman utama

Skenario : Mengisi *username* dan *password*, menekan tombol *login*.

2. Skenario Use case menginput jabatan dosen

User : TU

Kondisi awal : Data jabatan dosen belum ada

Kondisi akhir : Data jabatan dosen telah disimpan

Skenario : TU memasukkan jabatan dosen

3. Skenario Use case Menginput beban minimal dan maksimal dosen

User : TU

Kondisi awal : Data beban minimal, maksimal dosen belum ada

Kondisi Akhir : Data beban minimal, maksimal dosen telah disimpan

Skenario : Memasukkan beban minimal, maksimal dosen berdasarkan jabatan

4. Skenario *Use case* Setting Kelas

User : TU

Kondisi awal : TU belum melakukan *setting* kelas

Kondisi akhir : Kelas telah *disetting* dan disimpan

Skenario : TU melakukan *setting* kelas dengan memperkirakan berapa kelas yang ada dari mahasiswa yang mendaftar dan mengubah kelas jika ada pengurangan mahasiswa.

5. Skenario *Use case* Memilih pengampu

User : Kajor

Kondisi awal : Dosen pengampu belum dipilih

Kondisi akhir : Dosen pengampu telah disimpan

Skenario : Memilih dosen pengampu dan menekan tombol simpan.

6. Skenario *Use case* Memilih pengajar

User : Kajor

Kondisi awal : Dosen pengajar belum dipilih

Kondisi akhir : Dosen pengajar telah disimpan

Skenario : Memilih dosen pengajar. Jika pemilihan sudah selesai maka Kajor akan menekan tombol finish

7. Skenario *Use case* Mendownload data file CSV

User : TU

Kondisi awal : Kajor sudah menekan tombol finish

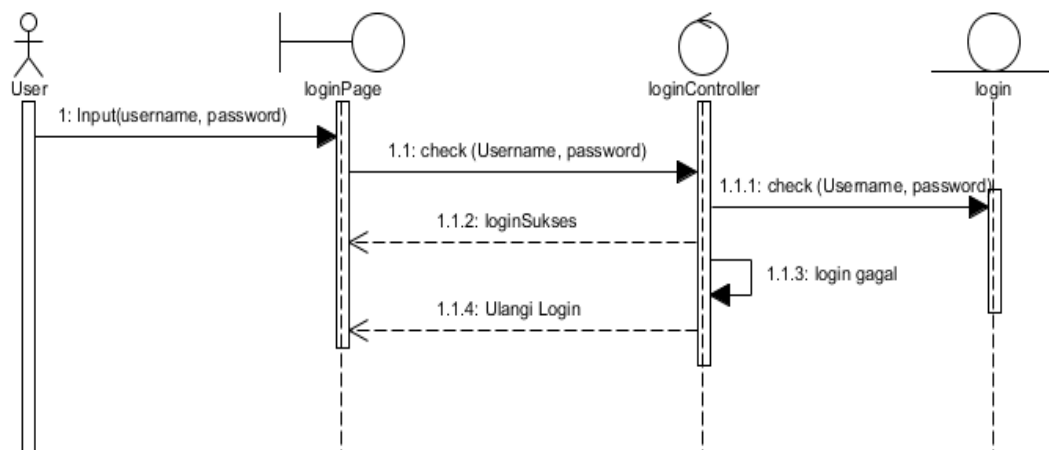
Kondisi Akhir : Data telah *download*

Skenario : TU *download* data file dalam bentuk CSV

3.4 Sequence Diagram

Sequence diagram mendeskripsikan bagaimana entitas dalam sistem berinteraksi, termasuk pesan yang digunakan saat interaksi. Semua pesan dideskripsikan dalam urutan dari eksekusi. *Sequence diagram* berhubungan erat dengan *use case* diagram karena satu *use case* akan menjadi satu *sequence diagram*

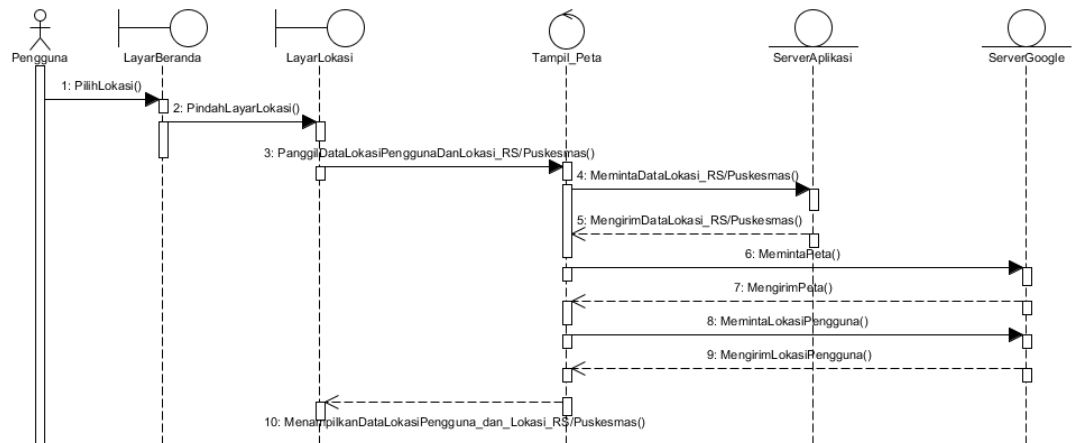
3.4.1 Sequence Diagram *Login*



Gambar 3. 3 Sequence Diagram *Login*

Pada gambar 3.3 dijelaskan bahwa *user* masuk ke *website* aplikasi penentuan pengajar dan pengampu dan melakukan *login* dengan memasukkan *username* dan password yang sesuai. Jika *username* dan password tidak sesuai maka proses *login* akan gagal (ulangi *login*).

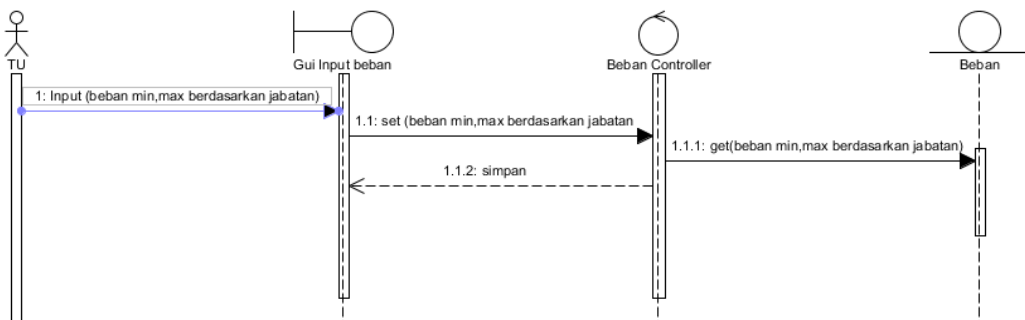
3.4.2 Sequence Diagram Menginput Data Jabatan Dosen



Gambar 3. 4 Sequence Diagram Menginput Jabatan Dosen

Pada gambar 3.4 dijelaskan proses TU mengakses Gui Jabatan dosen dan melakukan Input data jabatan dosen dengan memasukkan jabatan dosen pada data dosen.

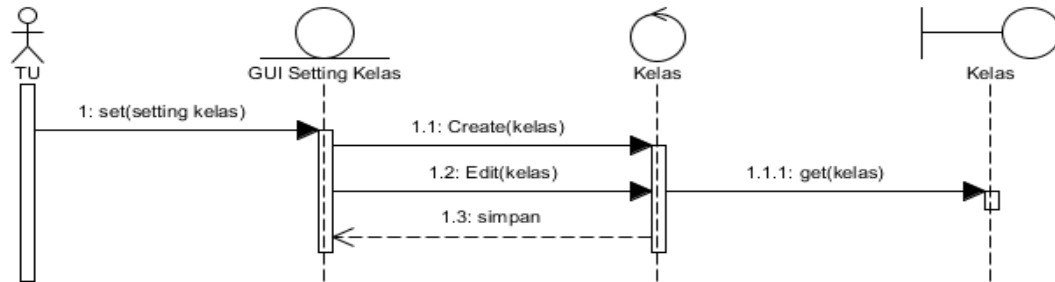
3.4.3 Sequence Diagram Menginput Beban Maksimal dan Minimal Berdasarkan Jabatan



Gambar 3. 5 Sequence Diagram Input Beban

Pada gambar 3.5 dijelaskan bahwa proses TU mengakses GUI Input beban dengan memasukkan beban minimal, maksimal dosen berdasarkan jabatan. Beban yang diinput akan simpan ke dalam *database*.

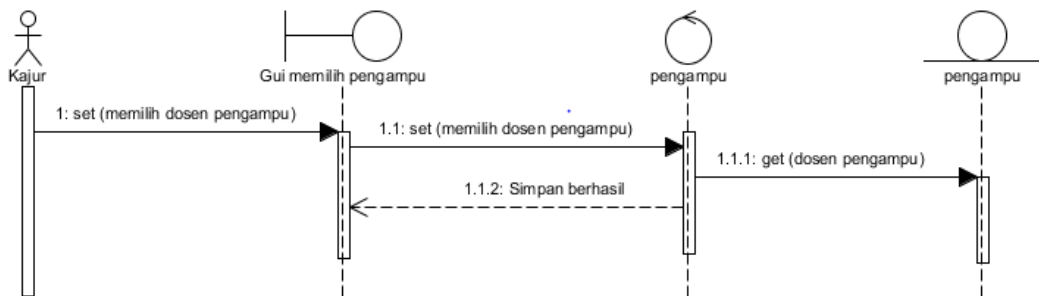
3.4.4 Sequence Diagram Setting Kelas



Gambar 3. 6 Sequence Diagram Setting Kelas

Pada gambar 3.6 dijelaskan TU mengakses GUI Setting kelas untuk melakukan setting kelas dengan membuat kelas berdasarkan mahasiswa yang ada, dan mengedit kelas bila ada pengurangan mahasiswa. Data kelas akan disimpan pada *database* kelas.

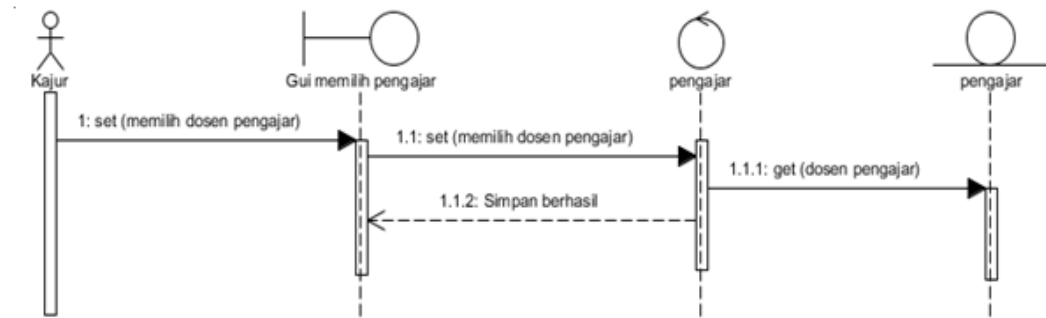
3.4.5 Sequence Diagram Memilih Pengampu



Gambar 3. 7 Sequence Diagram Memilih Pengampu

Pada gambar 3.7 dijelaskan bahwa proses Kajor mengakses GUI memilih pengampu. Kajor akan memilih dosen pengampu sesuai dengan nilai min dan maksimal, dan kelas. Dan akan menampilkan pesan bahwa dosen pengampu yang dipilih Kajor telah melebihi batas beban kerja.

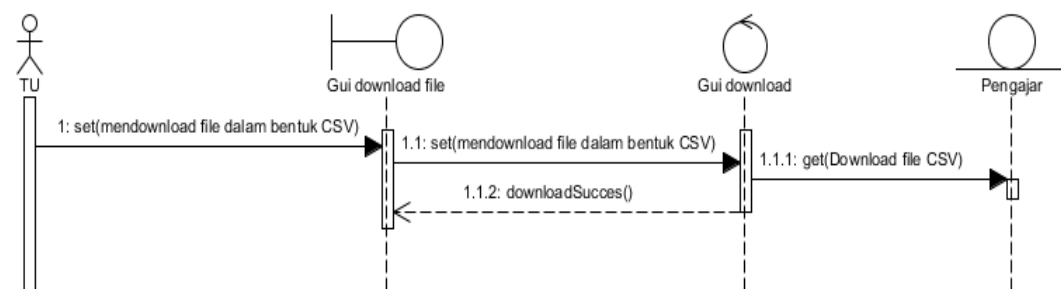
3.4.6 Sequence Diagram Memilih Pengajar



Gambar 3. 8 Sequence Diagram Memilih Pengajar

Pada gambar 3.8 dijelaskan bahwa proses Kajur mengakses GUI memilih pengajar. Kajur akan memilih dosen pengajar sesuai dengan nilai minimal,maksimal mengajar, dan kelas. dan akan menampilkan pesan bila batas pengajar yang dipilih Kajur telah melebihi batas beban dari dosen pengajar tersebut.

3.4.7 Sequence Diagram Mendownload File



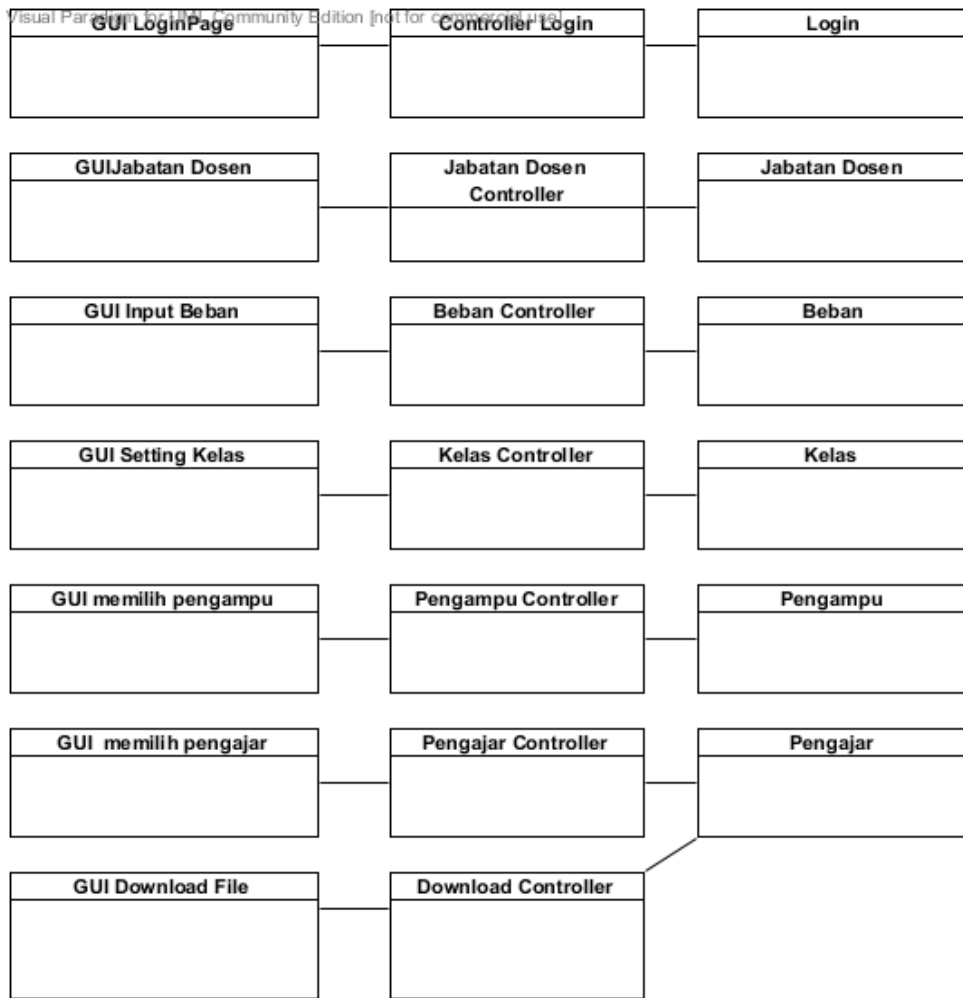
Gambar 3. 9 Diagram Sequence Mendownload file

Pada gambar 3.9 menjelaskan bahwa TU mengakses GUI *Download file* setelah kajur telah memilih dosen pengajar sehingga TU *mendownload* dapat *mendownload file* tersebut dalam format csv.

3.5 Class Diagram

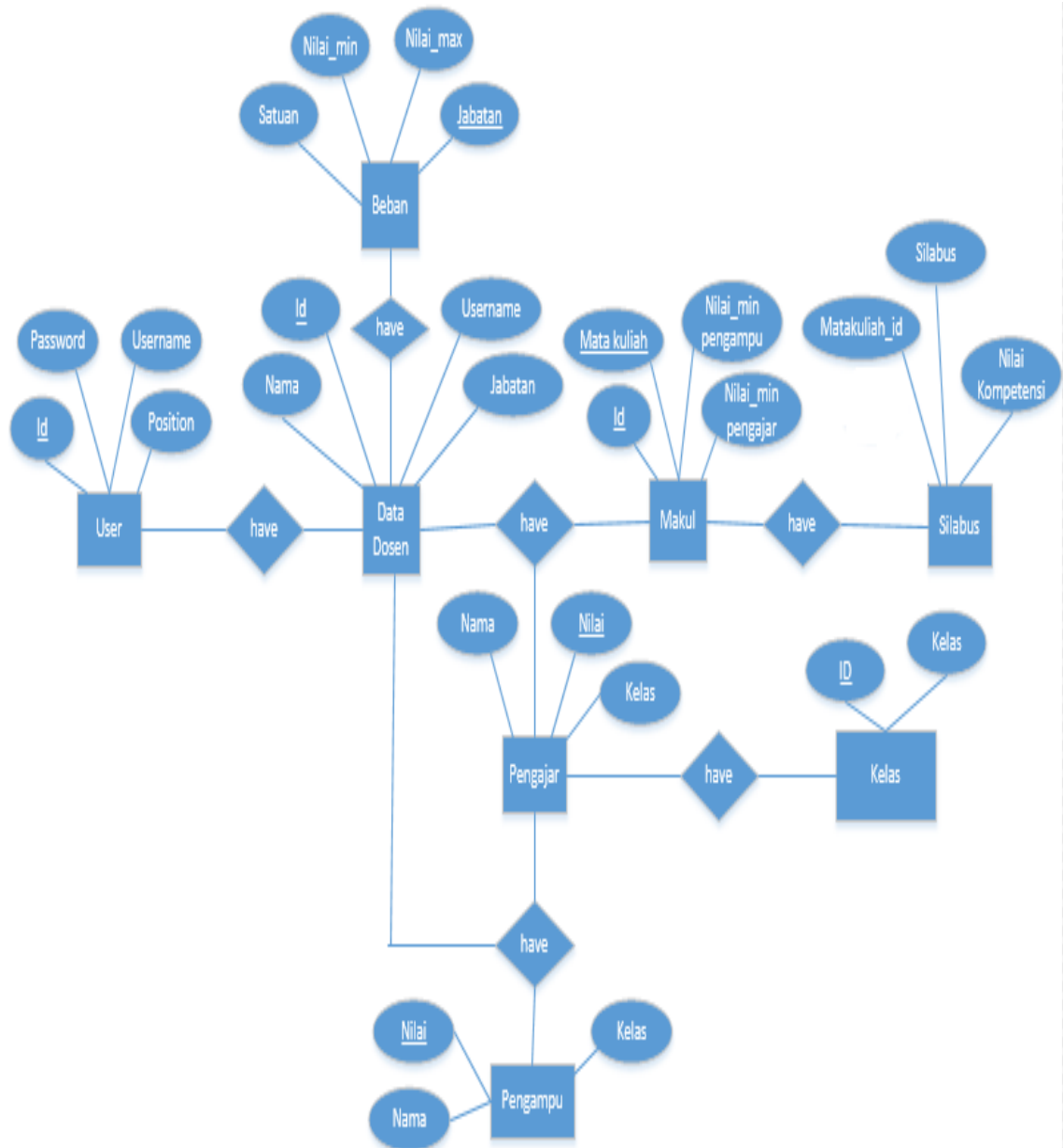
Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas- kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode. Atribut merupakan variable-variable yang dimiliki oleh suatu kelas.

Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Dalam sistem ini ada beberapa kelas yang dimiliki keterkaitan dengan satu sama lain. Adapun struktur kelas dari aplikasi penentuan beban kerja dosen ditunjukkan pada gambar 3.10.



Gambar 3. 10 Class Diagram

3.6 ER Diagram



Gambar 3. 11 ER Diagram

3.7 Spesifikasi Tabel

3.7.1 Spesifikasi Tabel *User*

Nama Tabel : T_ *User*

Deskripsi Isi : Data *User*

Tabel 3. 2 Spesifikasi *User*

Nama Field	Deskripsi Isi	Type & Length	Atribut	Keterangan
<i>Id_ User</i>	Berisi id <i>user</i>	Int (10)	Not null	Primary key
<i>Username</i>	Berisi <i>Username</i>	Varchar(20)	Not null	-
Password	Berisi Password	Varchar(20)	Not null	-
Position	Berisi Posisi <i>User</i>	Varchar(20)	Not null	-

3.7.2 Spesifikasi Tabel Data Dosen

Nama Tabel : T_ Data dosen

Deskripsi : Berisi data dosen berupa id, nama, *username*, dan jabatan.

Tabel 3. 3 Spesifikasi Data Dosen

Nama Field	Deskripsi Isi	Type & Length	Atribut	Keterangan
Id_dosen	Berisi id dosen	Int(10)	Not null	Primary key
nama_lengkap	Berisi nama dosen	Varchar(50)	Not null	-
<i>username</i>	Berisi <i>Username</i>	Varchar(20)	Not null	-
jabatan	Berisi Jabatan Dosen	Varchar(20)	Not null	-

3.7.3 Spesifikasi Tabel Beban

Nama Tabel : T_ beban

Deskripsi : Berisi beban kerja minimal, maksimal dan jabatan.

Tabel 3. 4 Spesifikasi Beban

Nama Field	Deskripsi Isi	Type & Length	Atribut	Keterangan
Beban min	Berisi beban min	Int (10)	Not null	-
Beban max	Berisi beban max	Int(10)	Not null	-
jabatan	Berisi jabatan dosen	Varchar(20)	Not null	Primary key
jenisTugas	Berisi Jenis Tugas Dosen	Varchar(60)	Not null	-

3.7.4 Spesifikasi Tabel Mata kuliah

Nama Tabel : T_mata kuliah

Deskripsi : Berisi tentang mata kuliah, id, nilai min pengampu, nilai min pengajar.

Tabel 3. 5 Spesifikasi Matakuliah

Nama Field	Deskripsi Isi	Type & Length	Atribut	Keterangan
Id	Id	Int(10)	Not null	Primary Key
Mata kuliah	Berisi mata kuliah	Varchar(20)	Not null	Foreign Key
Nilai min pengampu	Nilai_min pengampu	Int(10)	Not null	-
Nilai min pengajar	Nilai_min pengajar	Int(10)	Not null	-
Sesi	Sesi	Int(11)	Not null	-

3.7.5 Spesifikasi Tabel Silabus

Nama Tabel : T_silabus

Deskripsi : Berisi tentang data silabus yang berupa id, matakuliah_id, silabus.

Tabel 3. 6 Spesifikasi Silabus

Nama Field	Deskripsi Isi	Type & Length	Atribut	Keterangan
Id	Id	Int(10)	Not null	Primary Key
Matakuliah_id	Matakuliah_id	Int(10)	Not null	-
silabus	silabus	Varchar(20)	Not null	-
Nilai kompetensi	Nilai kompetensi	Varchar(20)	Not null	-

3.7.6 Spesifikasi Tabel Pengajar

Nama Tabel : T_Pengajar

Deskripsi : berisi tentang pemilihan pengajar

Tabel 3. 7 Spesifikasi Pengajar

Nama Field	Deskripsi Isi	Type & Length	Atribut	Keterangan
Nama	Nama_lengkap	Varchar(20)	Not null	-
Nilai	Nilai	Int(10)	Not null	-
Kelas (A,B,C)	Kelas	Varchar(10)	Not null	-

3.7.7 Spesifikasi Tabel Pengampu

Nama Tabel : T_Pengampu

Deskripsi : berisi tentang pemilihan pengampu

Tabel 3. 8 Spesifikasi Pengampu

Nama Field	Deskripsi Isi	Type & Length	Atribut	Keterangan
Nama	Nama_lengkap	Varchar(20)	Not null	-
Nilai	Nilai	Int(10)	Not null	-

3.7.8 Spesifikasi Tabel Kelas

Nama Tabel : T_Kelas

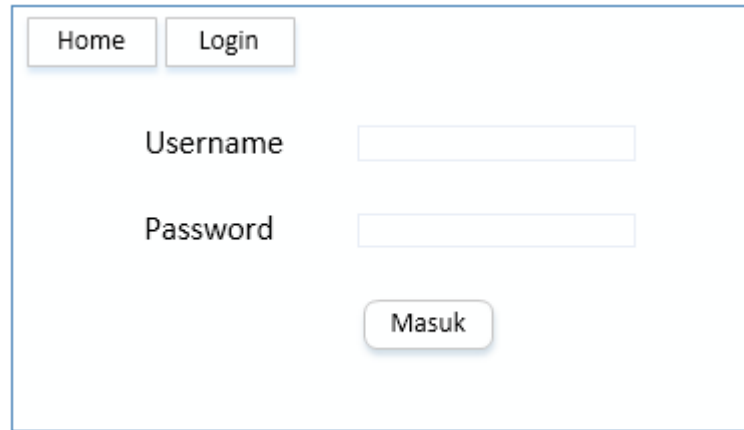
Deskripsi : berisi tentang setting kelas

Tabel 3. 9 Spesifikasi Kelas

Nama Field	Deskripsi Isi	Type & Length	Atribut	Keterangan
Id	Id_kelas	Int(10)	Not null	Primary Key
Kelas	Kelas	Varchar(10)	Not null	-

3.8 Perancangan Antarmuka

1. Rancangan Antarmuka GUI *Login*



Gambar 3. 12 Perancangan tampilan *login*

Deskripsi tampilan *login* dapat ditunjukkan pada tabel 3.10

Tabel 3. 10 Deskripsi Tampilan *login*

Id	Jenis	Keterangan	Fungsi
<i>username</i>	<i>Input text</i>	<i>Username</i>	Diisi dengan <i>username</i>
<i>password</i>	<i>Input password</i>	Password	Diisi dengan <i>password</i> pengguna
<i>masuk</i>	<i>Button</i>	Masuk	Jika pengguna menekan tombol masuk maka akan masuk pada halaman sesuai dengan tampilan akses pengguna.

2. Rancangan Antarmuka GUI Jabatan Dosen

Gambar 3. 13 Perancangan Tampilan *input* jabatan dosen

Deskripsi tampilan *Input* jabatan dosen dapat ditunjukkan pada tabel 3.11

Tabel 3. 11 Deskripsi Tampilan *input* jabatan dosen

Id	Jenis	Keterangan	Fungsi
ID	<i>Input text</i>	ID	Diisi dengan Id dosen
Nama	<i>Input text</i>	Nama dosen	Diisi dengan nama dosen
<i>Username</i>	<i>Input text</i>	<i>Username</i>	Diisi dengan <i>Username</i> dosen
Jabatan	<i>Input text</i>	Jabatan	Diisi dengan jabatan dosen

3. Rancangan Antarmuka GUI beban minimal dan maksimal berdasarkan jabatan

Gambar 3. 14 Perancangan tampilan *input* beban

Deskripsi tampilan *Input* Beban dapat ditunjukkan pada tabel 3.12

Tabel 3. 12 Deskripsi tampilan *input* beban

Id	Jenis	Keterangan	Fungsi
Jabatan	<i>Select menu</i>	Jabatan_dosen	Untuk memilih jabatan dosen
Jenis Tugas	<i>Select menu</i>	Jenis Tugas	Untuk memilih jenis tugas dosen
Beban min	<i>Input text</i>	Beban minimal dosen	Diisi dengan beban minimal dosen
Beban max	<i>Input text</i>	Beban maksimal dosen	Diisi dengan beban maksimal dosen
Satuan	<i>Select menu</i>	Satuan	Untuk memilih satuan
Simpan	<i>Button</i>	Simpan	Klik untuk menyimpan data

4. Rancangan Antarmuka Setting Kelas

ID	Kelas	Pilihan	
1	A	Update	Delete
2	B	Update	Delete
3	C	Update	Delete

Gambar 3. 15 Perancangan tampilan *setting* kelas

Deskripsi tampilan *Setting* kelas dapat ditunjukkan pada tabel 3.13.

Tabel 3. 13 Deskripsi tampilan *setting* kelas

Id	Jenis	Keterangan	Fungsi
Kelas	<i>Input text</i>	Kelas	Diisi dengan Kelas
Simpan	<i>Button</i>	Simpan	Jika diklik akan menyimpan kelas
<i>Update</i>	<i>Button</i>	<i>Update</i>	Pengelola dapat mengupdate kelas dari pengguna terdaftar melalui button ini
<i>Delete</i>	<i>Button</i>	<i>Delete</i>	Jika diklik akan mendelete data

5. Rancangan Antarmuka memilih Pengampu

The image shows a user interface for selecting an instructor. At the top, there are two dropdown menus: 'Mata kuliah' and 'Tugas Akhir'. Below them is a table with two columns: 'Nama' and 'Nilai'. The table contains two rows: 'Dosen1' with a value of 26, and 'Dosen2' with a value of 18. Below the table, there is a text input field labeled 'Nama' and a button labeled 'Simpan'.

Gambar 3. 16 Perancangan Antarmuka memilih pengampu

Deskripsi tampilan memilih pengampu dapat ditunjukkan pada tabel 3.14.

Tabel 3. 14 Deskripsi tampilan memilih pengampu

Id	Jenis	Keterangan	Fungsi
Mata kuliah	<i>Select menu</i>	Mata kuliah	Untuk memilih mata kuliah
Nama	<i>Input text</i>	Nama	Diisi dengan nama dosen yang dipilih sebagai pengampu
simpan	<i>Button</i>	<i>Simpan</i>	Jika diklik akan menyimpan data

6. Rancangan Antarmuka memilih pengajar

The image shows a web form for selecting a lecturer. At the top, there is a label 'Mata kuliah' and a dropdown menu currently showing 'Tugas Akhir'. Below this is a table with three columns: 'Nama', 'Nilai', and 'Kelas'. The table has two data rows: the first row has 'Dosen1' in the 'Nama' column, '26' in the 'Nilai' column, and 'A' in the 'Kelas' column; the second row has 'Dosen2' in the 'Nama' column, '18' in the 'Nilai' column, and '-' in the 'Kelas' column. Below the table, there is a text input field labeled 'Nama'. At the bottom, there are two buttons: 'Simpan' and 'Selesai'.

Gambar 3. 17 Perancangan tampilan memilih pengajar

Deskripsi tampilan memilih pengajar dapat ditunjukkan pada tabel 3.15.

Tabel 3. 15 Deskripsi tampilan memilih pengajar

Id	Jenis	Keterangan	Fungsi
Mata kuliah	<i>Select menu</i>	Mata kuliah	Untuk memilih mata kuliah
Nama	<i>Input text</i>	Nama	Diisi dengan nama dosen yang dipilih sebagai pengampu
Kelas	<i>Select menu</i>	Kelas	Untuk memilih kelas yang akan diajar
simpan	<i>Button</i>	<i>Simpan</i>	Jika diklik akan menyimpan data

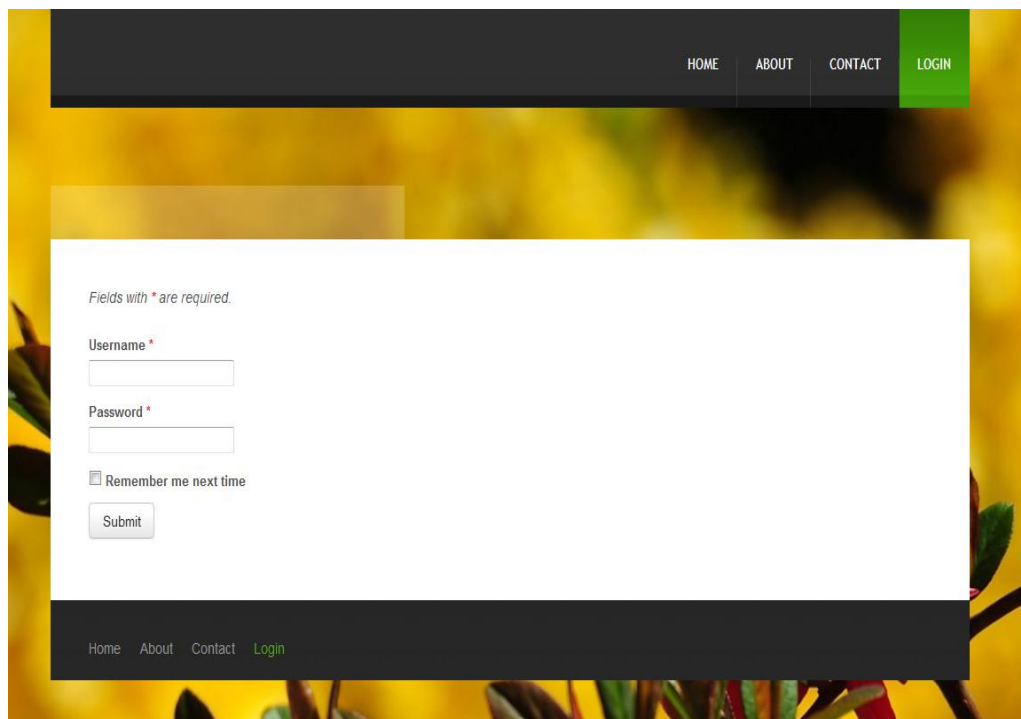
BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Antarmuka

Implementasi antar muka suatu aplikasi sangat dibutuhkan karena dalam hal ini implementasi merupakan tindakan atau pelaksanaan dari sebuah rencana yang telah disusun secara matang dan terperinci. Pada tahap ini akan dijelaskan implementasi antar muka yang telah disusun.

4.1.1 Implementasi Antar Muka *Login*

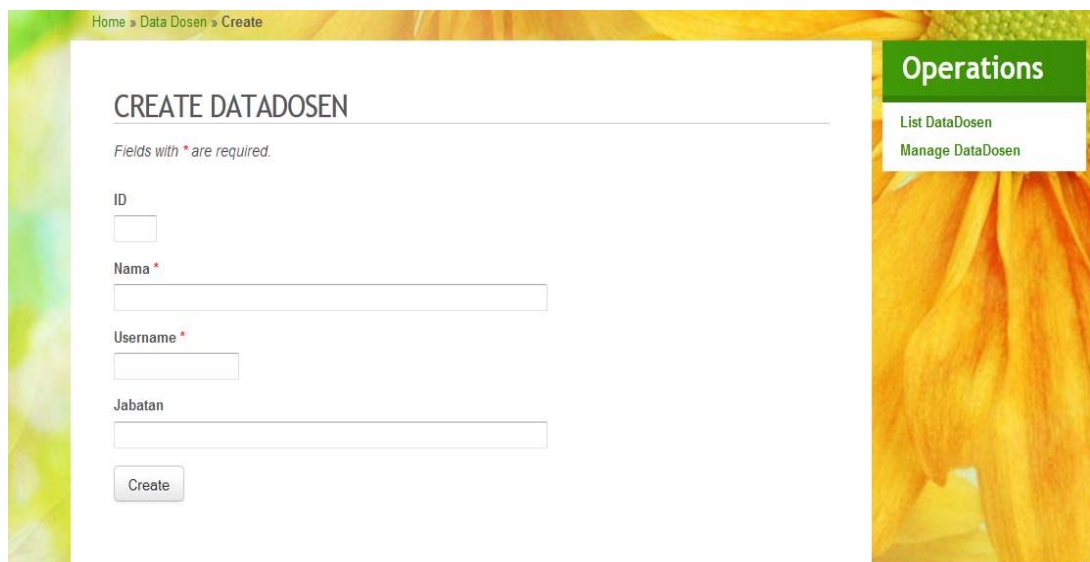
Implementasi fungsi *login* merupakan suatu aktivitas penting dalam aplikasi untuk admin dan *user* masuk dan mengakses akun yang telah divalidasi. Implementasi antar muka *login* dapat dilihat pada gambar 4.1

The image shows a web application interface with a login form. At the top, there is a dark navigation bar with links for HOME, ABOUT, CONTACT, and LOGIN (highlighted in green). The main content area has a white background with a login form. The form includes a message "Fields with * are required.", input fields for "Username *" and "Password *", a checkbox for "Remember me next time", and a "Submit" button. At the bottom, there is a dark footer bar with links for Home, About, Contact, and Login (highlighted in green).

Gambar 4. 1 Antarmuka *Login*

4.1.2 Implementasi Data Jabatan Dosen

Implementasi antar muka *Input* Jabatan pada data dosen merupakan tampilan yang akan terlihat apabila admin telah melakukan *login* dan memilih data dosen. Implementasi antar muka jabatan pada data dosen dapat dilihat pada gambar 4.2.



Home » Data Dosen » Create

CREATE DATADOSEN

Fields with * are required.

ID

Nama *

Username *

Jabatan

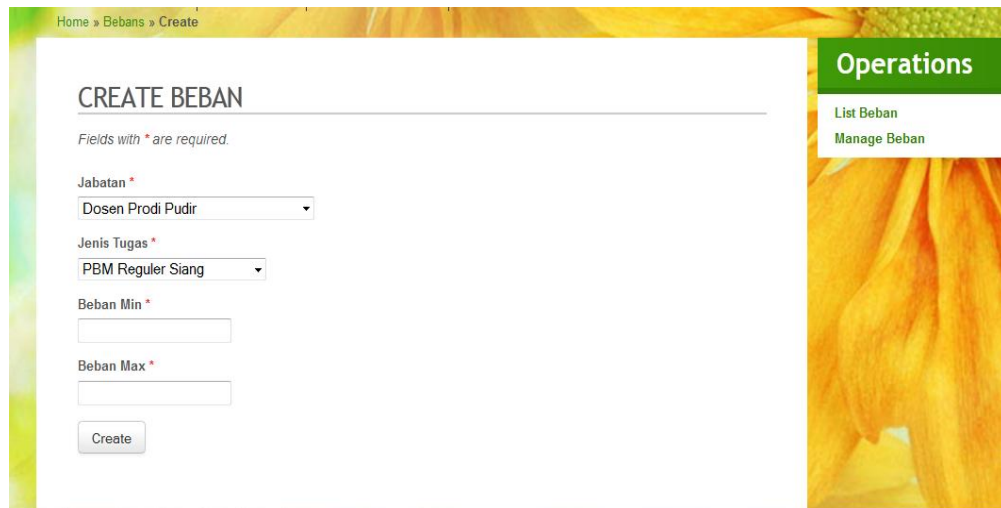
Create

Operations
List DataDosen
Manage DataDosen

Gambar 4. 2 Antarmuka Input Jabatan Dosen

4.1.3 Implementasi Tampilan Beban

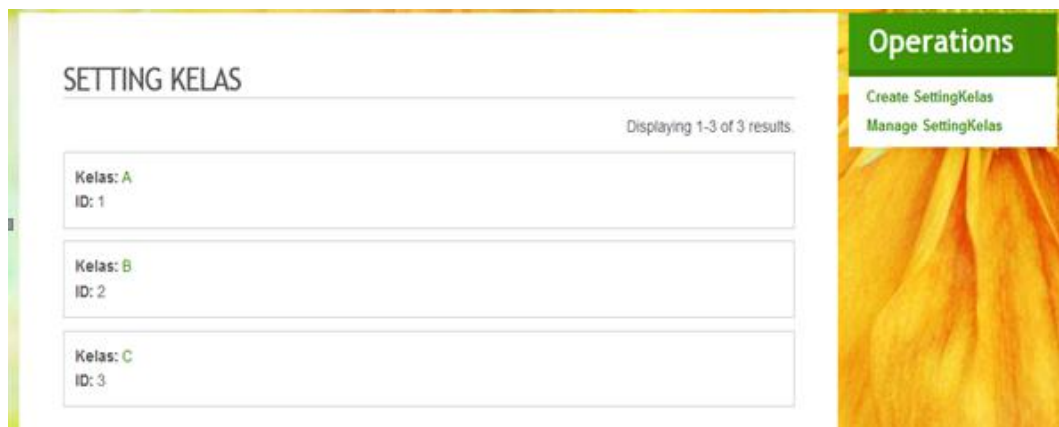
Implementasi antar muka beban berfungsi untuk menginput dan menampilkan data beban dosen. Implementasi antar muka beban dapat dilihat pada gambar 4.3.



Gambar 4. 3 Antarmuka Beban

4.1.4 Implementasi Setting Kelas

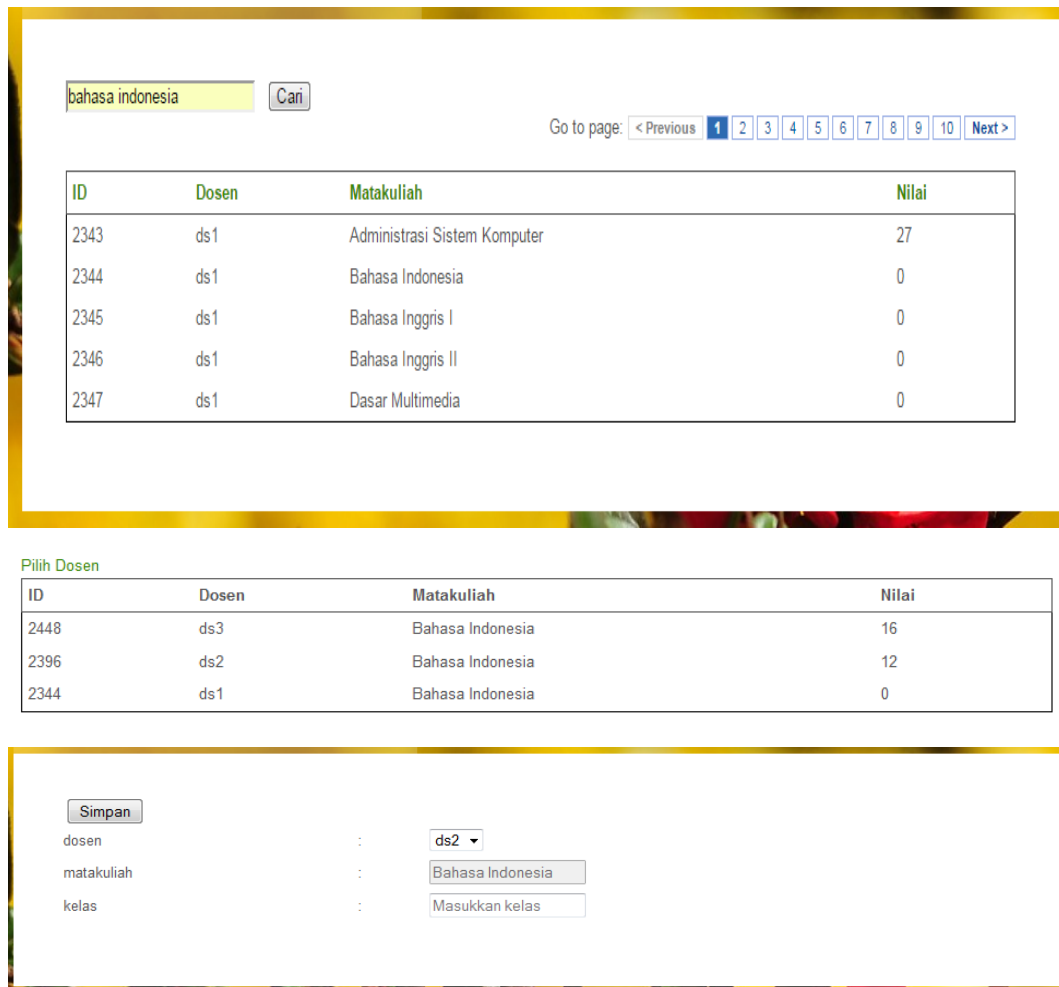
Implementasi antar muka *Setting* Kelas berfungsi untuk menginput dan menampilkan data kelas yang ada. Implementasi antar muka *Setting* Kelas dapat dilihat pada gambar 4.4.



Gambar 4. 4 Antarmuka *Setting* Kelas

4.1.5 Implementasi Memilih Pengajar

Implementasi antar muka memilih pengajar berfungsi untuk memilih dosen berdasarkan matakuliah dan nilai minimal dosen untuk menjadi pengajar dan menyimpan data dosen pengajar yang sudah dipilih Kajor. Implementasi antarmuka memilih pengajar dapat dilihat pada gambar 4.5.



Gambar 4. 5 Antarmuka Memilih Pengajar

4.1.6 Implementasi Memilih Pengampu

Implementasi antarmuka memilih pengampu berfungsi untuk memilih dosen berdasarkan matakuliah dan nilai minimal dosen untuk menjadi pengampu dan menyimpan data dosen pengampu yang sudah dipilih Kajar. Implementasi antarmuka memilih pengampu dapat dilihat pada gambar 4.6.

Go to page: < Previous **1** 2 3 4 5 6 7 8 9 10 Next >

ID	Dosen	Matakuliah	Nilai
2343	ds1	Administrasi Sistem Komputer	27
2344	ds1	Bahasa Indonesia	0
2345	ds1	Bahasa Inggris I	0
2346	ds1	Bahasa Inggris II	0
2347	ds1	Dasar Multimedia	0

Pilih Dosen

ID	Dosen	Matakuliah	Nilai
2448	ds3	Bahasa Indonesia	16
2396	ds2	Bahasa Indonesia	12
2344	ds1	Bahasa Indonesia	0

dosen :

matakuliah :

Gambar 4. 6 Antarmuka Memilih Pengajar

4.1.7 Implementasi *Download*

Implementasi antarmuka *Download file* dari tabel dosen pengajar dalam format Csv agar dapat digunakan langsung pada aplikasi FET (Aplikasi Penjadwalan). Implementasi antarmuka *download file* dapat dilihat pada gambar 4.7.

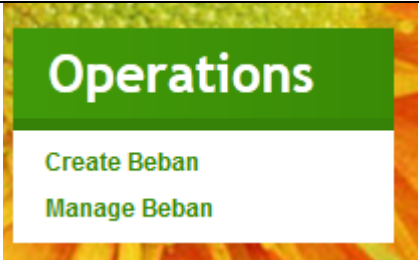
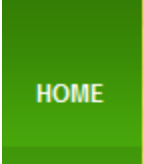
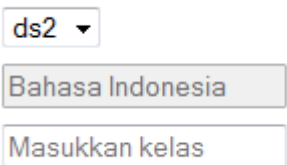
DAFTAR DOSEN PENGAJAR

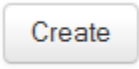
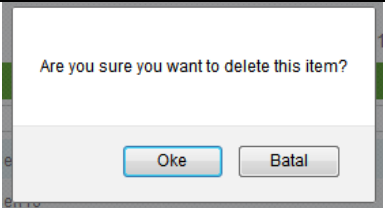
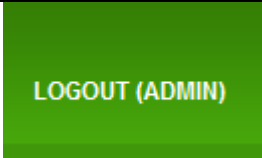
Nama Dosen	Matakuliah	Kelas
ds2	Pendidikan Agama	C
ds2	Matematika	A
ds2	Bahasa Indonesia	A
ds1	Administrasi Sistem Komputer	B
ds1	Internetworking	B
ds1	Pengantar Teknologi Informasi	A
ds1	Organisasi Komputer	A
ds1	Organisasi Komputer	B

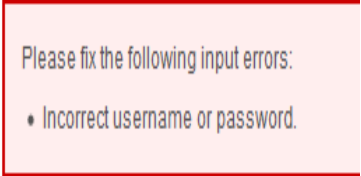
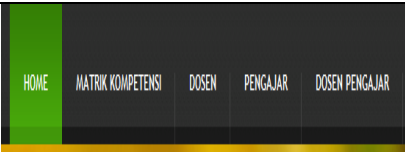
Gambar 4. 7 Antarmuka *Download*

4.2 Implementasi Konsep *User Interface* yang digunakan pada Aplikasi Penentuan Pengajar dan Pengampu Berdasarkan 9 Daya Guna *Heuristik*.

Tabel 4. 1 Implementasi *User Interface* pada aplikasi

No	10 daya guna heuristik	Penerapan <i>user interface</i>	Implementasi pada aplikasi
1	Dialog yang sederhana dan alami	Setiap dialog seharusnya menghindari perintah-perintah yang tidak perlu dan tidak ada hubungannya dengan <i>interface</i>	
2	Berbicara dengan bahasa <i>user</i>	Dialog seharusnya menggunakan bahasa yang dipahami oleh <i>user</i> .	
3	Mengurangi beban ingatan <i>user</i>	<i>User</i> seharusnya tidak dibebani untuk mengingat atau menghafal pada saat berinteraksi dengan sistem. Pada saat kajur menentukan dosen pengajar dan pengampu, kajur tidak lagi melihat nilai dosen dan menginput nama dosen karena nama dosen dan	

		matakuliah sudah otomatis ada.	
4	Konsisten	Penerapan tombol harus konsisten agar <i>user</i> tidak ragu-ragu di saat menggunakan suatu perintah atau fungsi.	
5	Sistem timbal balik	Sistem seharusnya memberitahu pengguna segala aktifitas yang sedang berlaku atau status dari system.	
6	Jalan keluar yang jelas	Apabila <i>user</i> melakukan kesalahan dalam memilih perintah maka ia dapat keluar dari kesalahan tanpa ada masalah, misalnya perintah <i>Undo</i> .	
7	Jalan pintas	Demi kemudahan dan kecepatan interaksi di dalam menggunakan suatu sistem maka sudah seharusnya bila tersedia <i>shortcut</i> yang berguna untuk membantu <i>user</i> agar dapat menggunakan	Tidak ada

		berbagai fungsi dengan mudah.	
8	Pesan-pesan kesalahan yang baik	Menyediakan mekanisme pemberitahuan kesalahan Dengan jelas dan mudah dipahami. Dan penyampaian kesalahan dilakukan secara sopan.	
9	Mencegah kesalahan	Rekayasa <i>interface</i> yang baik seharusnya mampu membuat <i>user</i> menghindari kesalahan, misalnya interaksi dengan menggunakan menu.	
10	Help dan Dokumentasi	Bantuan dan dokumentasi merupakan kemudahan yang diberikan sistem untuk menjelaskan cara penggunaan.	Tidak ada

4.3 Hasil Pengujian

Pengujian aplikasi telah dilaksanakan dengan baik sesuai peraturan akademik yang berlaku dan hasilnya dapat dilihat pada tabel 4.2.

Nama penguji : Artika Hartin

Nim : 3311201001

Tabel 4. 2 Hasil Pengujian

No	Use Case	Skenario	Data Uji	Target	Hasil Uji
1.	<i>Login</i>	1. Memasukkan Username dan Password 2. Menekan tombol Login	Data benar : Username : Admin Password : Admin	Login berhasil masuk ke sistem sesuai dengan level pengguna, tidak muncul pesan kesalahan dan masuk ke halaman utama pengguna.	OK
			Data salah : Username : Admin000 Password : Admin000	Login tidak berhasil, muncul pesan kesalahan kemudian kembali ke <i>login</i> .	OK

2.	<p><i>Input</i> Jabatan pada Data Dosen (<i>User Level</i> :Admin)</p>	<ol style="list-style-type: none"> 1. Menekan tombol “Create” 2. Mengisi data jabatan pada data dosen 3. Menekan tombol “Save” 	<p>Nama: ds1 Username: dosen1 Jabatan: Dosen Prodi Pudir</p>	<p>Data jabatan dosen berhasil ditambah</p>	<p>OK</p>
3	<p><i>Input</i> beban minimal, maksimal dosen berdasarkan jabatan</p>	<ol style="list-style-type: none"> 1. Menekan tombol “Create” 2. Mengisi data beban 3. Menekan tombol “save” 	<p>Jabatan : Dosen Prodi Pudir Jenis Tugas: PBM Reguler Siang Beban Min :6 Beban Max :8</p>	<p>Data beban berhasil disimpan</p>	<p>OK</p>

4.	<i>Update</i> beban minimal, maksimal dosen berdasarkan jabatan	<ol style="list-style-type: none"> 1. Mencari data beban yang akan diubah 2. Menekan tombol “<i>edit</i>” 3. Mengganti data beban 4. Menekan tombol “<i>save</i>” 	<p>Data Lama : Jabatan : Dosen Prodi Pudir Jenis Tugas: PBM Reguler Siang Beban Min :6 Beban Max :8</p> <hr/> <p>Data Baru : Jabatan : Dosen Prodi Pudir Jenis Tugas: PBM Reguler Siang Beban Min :8 Beban Max :10</p>	Data Beban berhasil diupdate	OK
5.	<i>Delete</i> beban minimal, maksimal dosen berdasarkan	<ol style="list-style-type: none"> 1. Mencari data beban yang akan di <i>delete</i> 2. Menekan tombol “<i>delete</i>” 3. Keluar pesan konfirmasi <i>delete</i> 	Jabatan : Dosen Prodi Pudir Jenis Tugas: PBM Reguler Siang Beban Min :8 Beban Max :10	Mengeluarkan pesan berhasil dihapus	OK

	jabatan	4.Menekan tombol “yes”			
6	<i>Input Setting</i> Kelas	1. Menekan tombol “Create” 2. Mengisi data kelas 3.Menekan tombol “save”	Kelas :IF1A	Data Kelas berhasil disimpan	OK
7	<i>Update Setting</i> Kelas	1. Mencari data beban yang akan di <i>update</i> 2. Menekan tombol “ <i>edit</i> ” 3. Mengganti data kelas 4.Menekan tombol “ <i>save</i> ”	Data Lama : Kelas:IF1A	Data Kelas berhasil diubah	OK
			Data Baru : Kelas:IF3A		

8	<i>Delete Setting</i> Kelas jabatan	<ol style="list-style-type: none"> 1. Mencari data Kelas yang akan di <i>delete</i> 2. Menekan tombol “<i>delete</i>” 3. Keluar pesan konfirmasi <i>delete</i> 4. Menekan tombol “<i>yes</i>” 	Kelas: IF2A	Mengeluarkan pesan berhasil dihapus	OK
9	Memilih Pengampu (<i>User Level</i> : Kajur)	<ol style="list-style-type: none"> 1. Masuk ke dalam halaman pengajar 2. Mencari pengajar berdasarkan matakuliah 3. Memilih dosen pengampu sesuai dengan nilai kompetensi dan beban maksimal dosen 4. Menekan tombol “<i>save</i>” 	Cari berdasarkan matakuliah : Administrasi Sistem Komputer Dosen : ds1 Matakuliah : Administrasi Sistem Komputer	Data dosen pengampu berhasil disimpan	OK
				Jika dosen pengampu telah melebihi beban akan muncul pesan error	

10	Memilih Pengajar (<i>User Level : Kajor</i>)	<ol style="list-style-type: none"> Masuk ke dalam halaman pengajar Mencari pengajar berdasarkan matakuliah Memilih dosen pengampu sesuai dengan nilai kompetensi dan beban maksimal dosen Menekan tombol “<i>save</i>” 	Cari berdasarkan matakuliah : Administrasi Sistem Komputer Dosen : ds1 Matakuliah : Administrasi Sistem Komputer Kelas : IF1A	Data dosen pengampu berhasil disimpan	OK
				Jika dosen pengajar telah melebihi beban akan muncul pesan <i>error</i> .	
11	Menghasilkan <i>Output file Csv</i> (<i>User Level :Admin</i>)	<ol style="list-style-type: none"> Masuk ke halaman dosen pengajar Klik <i>Button Export</i> 	Nama dosen : ds1 Matakuliah : Administrasi Sistem Komputer Kelas : IF3B	Menghasilkan data dosen pengajar dalam bentuk file Csv	OK

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan dari aplikasi Penentuan Pengajar dan pengampu adalah sebagai berikut :

1. Aplikasi penentuan pengajar dan pengampu memberikan informasi dengan aspek *user interface* sehingga memudahkan pengguna dalam memahami dan menggunakan navigasi yang ada pada aplikasi.
2. Aplikasi penentuan pengajar dan pengampu dapat mengecek alokasi dosen apakah sudah sesuai dengan aturan beban kerja yang ditetapkan oleh pegawai.
3. Tugas akhir ini sudah menerapkan 80 % konsep daya guna *heuristik* untuk membuat *user interface* interaktif

5.2 Saran

Sebagai langkah pengembangan aplikasi ini pada masa yang akan datang, maka saran yang dapat diberikan adalah sebagai berikut :

1. Aplikasi dapat menangani kelas persemester genap dan ganjil.
2. Aplikasi dapat menangani nilai minimal pengampu dan pengajar otomatis

DAFTAR PUSTAKA

1. B. Shneiderman, C. Plaisant., 2010. *Designing The User Interface*. Addison Wesley.
2. Deborah J. Mayhew, “Principles and Guidelines in *Software User Interface Design*”, PrenticeHall, USA, 1992.
3. Hendrayanti. E. (2010). Analisis Beban Kerja sebagai Dasar Perencanaan Kebutuhan SDM.
4. <http://www.dikti.go.id/files/atur/UU14-2005GuruDosen.pdf> diakses pada 12 september 2014.
5. Komarudin, M. dan dkk., 2010, Sistem Informasi Manajemen Beban Kerja Dosen Berbasis Web.
6. Mandel, Theo., 1997 *The Elements of User Interface Design*.
7. Prasetyo, D.D. 2002. *Administrasi Database Server MySQL*. Jakarta: Elex Media Komputindo.
8. Thomson, Laura dan Welling, Luke., *PHP and MySQL Web Development* SAMS, Indiana-USA, 2001.
9. Winesett, Jeffrey. *Web Application Development with Yii and PHP* Second Edition. Birmingham: Packt Publishing, 2012.
10. Welling, Luke. Thomson, Laura. *PHP and MySQL Web Development* (4th Edition). Boston: Addison-Wesley Professional, 2008.
11. Yogi, A. dan Siregar A., 2013, Tugas Akhir Sistem Informasi Identifikasi Kompetensi Dosen Prodi Teknik Informatika Politeknik Negeri Batam.

Nama penguji : Artika Hartin

Nim : 3311201001

hasil Pengujian Penentuan Pengajar dan Pengampu dalam bentuk tabel dengan menggunakan php

No	Use Case	Skenario	Data Uji	Target	Hasil Uji
1.	Memilih Pengajar dan pengampu	1. Masuk ke dalam halaman penentuan pengajar dan pengampu 2. Mencari matakuliah 3. Memilih dosen pengampu dan pengajar pada tabel yang sesuai dengan nilai kompetensi dan beban maksimal dosen 4. Menekan tombol simpan	Cari berdasarkan matakuliah : Administrasi Sistem Komputer pilih dosen pengampu dan pilih dosen pengajar sesuai dengan kelas dan nilai kompetensi dosen Jika tidak ada dosen untuk mengajar pada suatu matakuliah, dosen akan <i>upgrade</i> nilai dengan menekan link <i>upgrade</i>	Data dosen pengampu dan pengajar berhasil disimpan	OK