

BAB I PENDAHULUAN

1.1 Latar Belakang

Kegiatan Kuliah Kerja Nyata Mahasiswa (yang selanjutnya disingkat dengan KKM) Universitas Islam Negeri Sunan Gunung Djati Bandung merupakan suatu kegiatan akademik mahasiswa yang berlangsung melalui tahapan pembelajaran, penelitian dan pengabdian masyarakat. Lebih jauh, KKM merupakan bagian pembelajaran dengan masyarakat (*learning with community*) sebagai bentuk pengamalan IPTEKS yang telah dipelajari oleh para mahasiswa selama perkuliahan. Karena itu, KKM harus berorientasi pada Visi UIN Sunan Gunung Djati Bandung, yaitu “Menjadi Universitas yang Unggul dan Kompetitif dengan bingkai akhlak karimah di ASEAN berbasis wahyu memandu ilmu”. KKM adalah kegiatan intrakurikuler terpadu Tri Dharma Perguruan Tinggi dengan menempatkan mahasiswa diluar kampus dalam kurun waktu tertentu, untuk melakukan edukasi dan mendampingi masyarakat secara profesional.

Metode pemberian pengalaman belajar dan bekerja kepada mahasiswa dalam kegiatan pemberdayaan masyarakat. Penerapan Kuliah Kerja Nyata Mahasiswa dilakukan untuk mengembangkan ilmu dan teknologi yang diperoleh mahasiswa selama perkuliahan yang berkaitan dengan masyarakat dan membantu masyarakat meningkatkan kemampuan berpikir, bersikap dan bertindak agar sesuai dengan program pembangunan yang telah direncanakan oleh daerahnya masing-masing[1].

Proses pendaftarannya masih dilakukan dengan proses manual yaitu mahasiswa calon peserta Kuliah Kerja Nyata Mahasiswa mendaftarkan diri ke Lembaga Penelitian dan Pengabdian Masyarakat (LP2M) dan mahasiswa

menunggu pengelompokkannya dengan diacak secara manual oleh panitia. Dokumen yang dikumpulkan masih berupa bentuk cetak (*paper based*). Bentuk seperti ini sangat rawan mengalami kerusakan jika tidak disediakan tempat dan perawatan khusus setiap periode waktu tertentu.

Kuliah Kerja Nyata Mahasiswa Universitas Islam Negeri Sunan Gunung Djati Bandung dilaksanakan dalam dua periode setiap tahunnya. Prosedur pelaksanaan Kuliah Kerja Nyata Mahasiswa ini dilakukan oleh setiap mahasiswa dengan mendaftarkan diri sesuai dengan syarat yang telah ditentukan oleh pihak Universitas Islam Negeri Sunan Gunung Djati Bandung. Setiap anggota kelompok terdiri atas beberapa mahasiswa dari berbagai jurusan. Kelompok-kelompok tersebut ditempatkan di lokasi desa yang berbeda di setiap kabupaten.

Proses pengelompokan aplikasi tersebut dilakukan secara acak (*random*) dengan menggunakan kriteria jenis kelamin, jurusan, dan fakultas. Oleh karena itu sistem secara manual ini memiliki kelemahan yaitu komposisi peserta dalam satu kelompok berdasarkan penyebaran fakultas tidak merata dan lolosnya yang tidak membayar SPP, dalam hal ini pengelompokan dilakukan secara *online (website)* yang akan mendeteksi jenis kelamin, jurusan, fakultas, sks yang telah ditempuh. Maka dari itu dilakukanlah penelitian yang menghasilkan perancangan sebuah aplikasi yang dimana terdapat penentuan dan pengelompokan peserta dengan menggunakan algoritma *greedy*, yang dimana gambaran umum dari algoritma *greedy* tersebut merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Persoalan optimasi (*optimization problems*). Berdasarkan hal tersebut, penelitian ini dilakukan dengan merancang Aplikasi KKM yang berfokus pada proses pengelompokan peserta Kuliah Kerja Nyata Mahasiswa

menggunakan algoritma *greedy* dengan komposisi pembagian peserta dalam satu kelompok secara merata berdasarkan jurusan, fakultas, jenis kelamin, dan juga untuk menyimpan asset dokumen data peserta agar bisa tersimpan dalam lokasi penyimpanan berupa bentuk digital, maka dibuatlah **“Perancangan Aplikasi Kuliah Kerja Nyata Mahasiswa Dengan Menggunakan Algoritma *Greedy* Untuk Menentukan Peserta Kuliah Kerja Nyata Mahasiswa”**.

1.2 Perumusan Masalah

Berdasarkan pemaparan dilatar belakang, permasalahan yang akan muncul pada pembuatan aplikasi ini dapat dijelaskan sebagai berikut :

- a. Bagaimana menerapkan algoritma *greedy* untuk pengelompokkan peserta secara otomatis?
- b. Bagaimana kinerja algoritma *greedy* dalam proses penentuan dan pengelompokkan peserta?

1.3 Tujuan dan Manfaat

Dengan permasalahan yang dipaparkan di dalam rumusan masalah maka diharapkan berbagai tujuan dan manfaat terselesaikannya masalah tersebut dapat dicapai, dimana *point* tujuan dan manfaat pembangunan aplikasi ini adalah sebagai berikut :

- a. Untuk menerapkan algoritma *greedy* dalam pengelompokkan peserta secara otomatis.
- b. Untuk mengetahui kinerja algoritma *greedy* dalam proses penentuan dan pengelompokkan peserta.

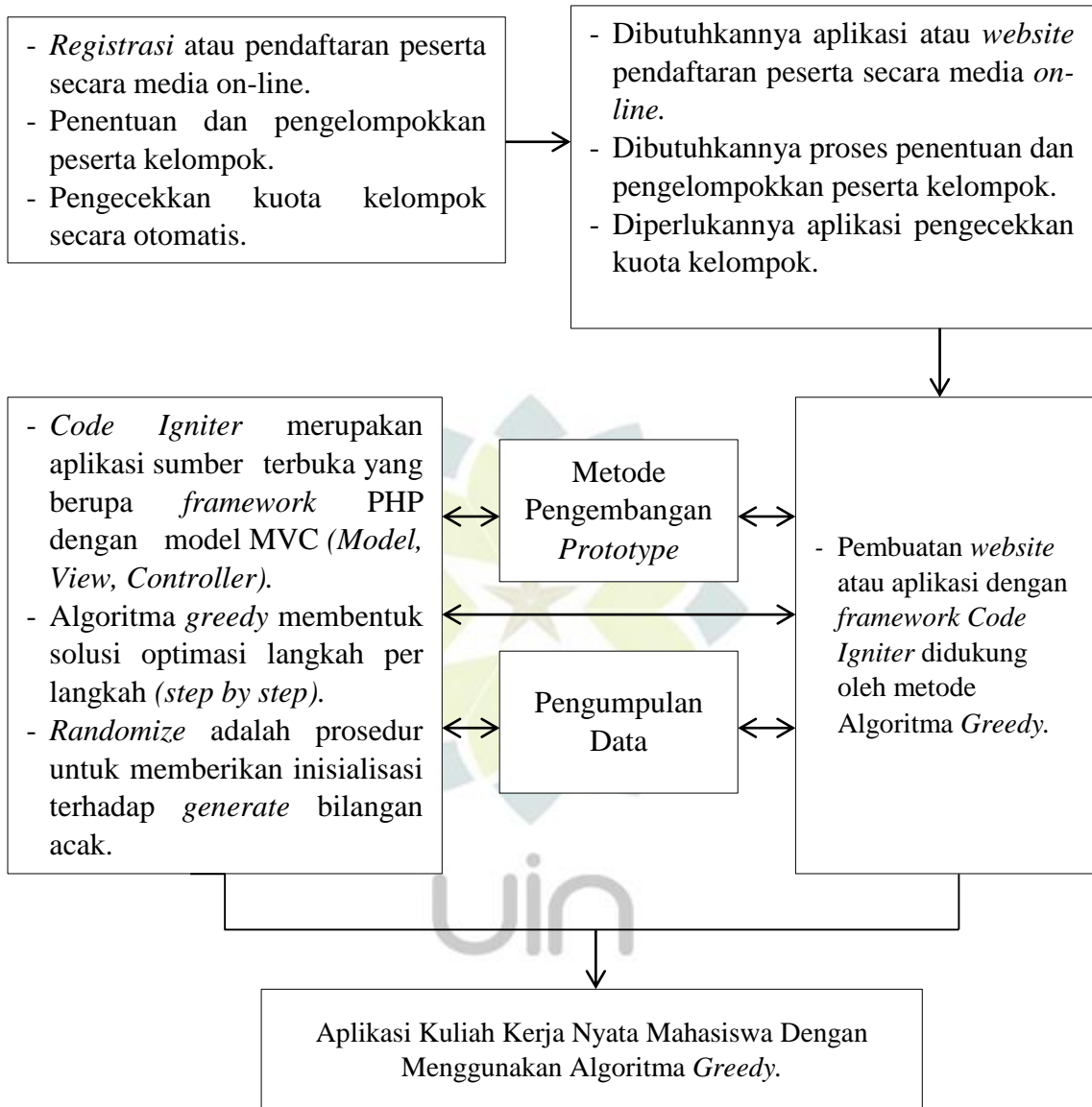
1.4 Batasan Masalah

Untuk meminimalisir berbagai pengembangan masalah yang akan keluar dan bahkan membuat fokus pembuatan aplikasi ini menjadi tidak tercapai tujuannya, maka pada pembuatan aplikasi ini akan difokuskan terhadap beberapa hal berikut :

- a. Penggunaan metode Algoriitma *Greedy* dalam proses penentuan dan pengelompokan pserta kelompok Kuliah Kerja Nyata Mahasiswa.
- b. Penelitian ini untuk mengolah proses pendaftaran sampai penentuan kelompok.
- c. Komposisi peserta dalam satu kelompok berdasarkan kriteria yang telah ditentukan oleh lembaga, sedangkan data yang akan diproses berasal dari data diri peserta yang dimasukkan dalam formulir pendaftaran.
- d. Pengecekan atau penentuan dilakukan untuk mengecek ketersediaan kuota kelompok.
- e. Aplikasi yang dibangun dengan menggunakan *framework CodeIgniter* dengan menggunakan *tools Notepad++* dan *database MySQL*.
- f. *Output* yang dihasilkan oleh aplikasi ini adalah *report* berupa bentuk *Excell (xls)* pengelompokkan peserta yang telah ditempatkan pada lokasi yang telah ditentukan.

1.5 Kerangka Pemikiran

Kerangka pemikiran pembuatan aplikasi ini dapat dilihat pada gambar 1.1 di bawah ini :



Gambar 1.1 Kerangka Pemikiran

Pada gambar 1.1 dijelaskan bahwa aspek yang mendasari penelitian ini adalah penentuan terhadap pengelompokkan peserta, yang tentunya sangat berkaitan erat dengan pengacakan. Atas hal yang mendasari tersebut, tujuan akhir produk dari penelitian ini adalah dibuatnya aplikasi dari berbagai aspek permasalahan tersebut yang dapat penentuan dan pengelompokkan peserta Kuliah Kerja Nyata Mahasiswa berbasis *on-line*.

1.6 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini yaitu meliputi pengumpulan data dan pengembangan sistem.

1.6.1 Tahap Pengumpulan Data

Untuk mengumpulkan berbagai data yang diperlukan oleh kelancaran pembuatan aplikasi ini ada beberapa tahap yang akan dilakukan sehingga hasil yang didapatkan menjadi maksimal.

1. Studi Lapangan

Kegunaan penelitian lapangan ini dalam penyusunan suatu penelitian tiada lain untuk menilai sejauh mana penelitian ini telah tersampaikan bahkan telah terealisasi pada target yang akan kita capai. Pada penelitian ini Lembaga Penelitian dan Pengabdian Masyarakat (LP2M UIN SGD Bandung) sebagai objek lapangan yang akan dikunjungi sebagai bahan pertimbangan serta memperoleh data dan sample untuk penelitian tersebut. Cara yang akan dilakukan adalah dengan meninjau bagaimana cara pendaftaran hingga penentuan kelompok peserta secara teknis.

2. Studi Pustaka

Dengan mendeskripsikan data yang diperoleh dari berbagai sumber referensi yang berhubungan dengan permasalahan yang sedang diteliti, untuk kemudian dianalisa, dan diuraikan dalam bentuk deskripsi yang dilatar belakangi oleh adanya konsep-konsep dan teori yang dikemukakan dalam landasan teori.

3. Wawancara dan Observasi

Pada metode ini, kita lebih melakukan pendekatan kepada pihak-pihak terkait yang berhubungan dengan pembuatan aplikasi ini, seperti pegawai Tata

Usaha atau bagian administrasi dan juga ketua pusat pengabdian masyarakat yaitu Dr. H Ramdani Wahyu S.,M.Ag.,M.Si di Lembaga Penelitian dan Pengabdian Masyarakat UIN Sunan Gunung Djati Bandung, yang tentunya pernah melakukan pendaftaran dan pengelompokkan peserta tetapi masih menggunakan manual.

1.6.2 Tahap Pengembangan Sistem

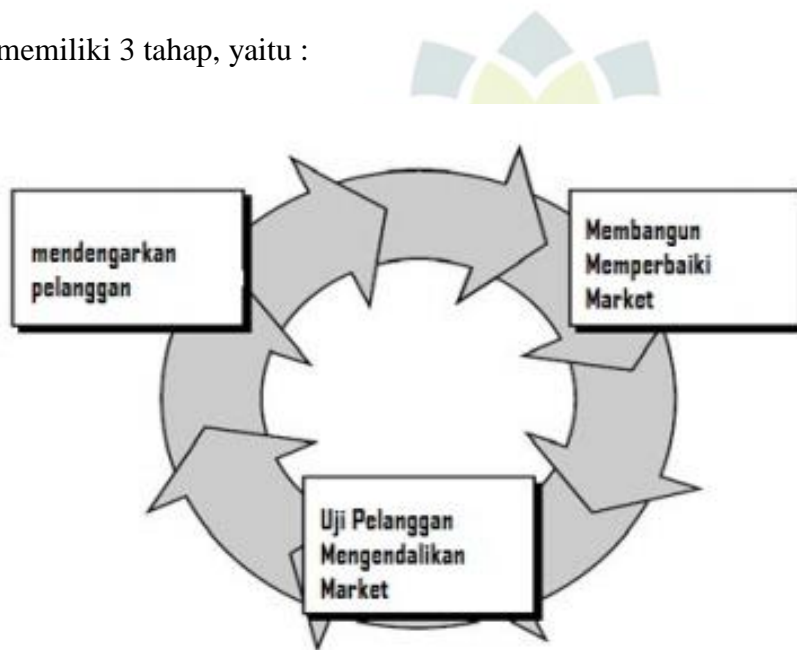
Pada tahap pengembangan sistem aplikasi ini mengembangkan sistem dengan metode *Prototyping Model* adalah metode proses pembuatan sistem yang dibuat secara terperinci dan memiliki beberapa tahap-tahap yang harus dilalui pada pembuatannya, namun jika tahap final dinyatakan bahwa sistem yang telah dibuat belum sempurna atau masih memiliki kekurangan, maka sistem akan dievaluasi kembali dan akan melalui proses dari awal. Dalam hal ini dikatakan sempurna, pada sebelumnya kita telah menyepakati dengan secara tertulis yaitu kerangka acuan (TOR) aplikasi tersebut, jadi apabila aplikasi tersebut telah mencapai hasil, maka aplikasi tersebut sudah mencapai tujuan yang ditentukan atau disepakati.

Prototype lebih memudahkan proses dalam membangun sebuah perancangan aplikasi. Keuntungan dari *prototype* :

1. Menghasilkan syarat yang lebih baik dari produksi yang dihasilkan oleh metode spesifikasi tulisan.
2. User dapat mempertimbangkan sedikit perubahan selama masih bentuk *prototype*.
3. Memberikan hasil yang lebih akurat dari pada perkiraan sebelumnya, karena fungsi yang diinginkan dan kerumitannya sudah dapat diketahui dengan baik.

4. User merasa puas. Pertama, user dapat mengenal melalui komputer. Dengan melakukan *prototype* (dengan analisis yang sudah ada), user belajar mengenai komputer dan aplikasi yang akan dibuatkan untuknya. Kedua, user terlibat langsung dari awal dan memotivasi semangat untuk mendukung analisis selama proyek berlangsung.

Prototyping Model juga dapat diartikan sebagai pembuatan sistem atau *software* dengan metode siklus. Dalam metode *prototyping model* sendiri memiliki 3 tahap, yaitu :



Gambar 1.2 Metode Pengembangan *Prototype*. [3]

Berdasarkan gambar 1.2 proses model *prototyping* dapat dijelaskan sebagai berikut[3] :

1. Mendengarkan pelanggan

Pada tahap ini dilakukan pengumpulan kebutuhan dari *system* dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu *system* yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana *system* yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

2. Merancang dan Membuat Prototype

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

3. Uji Coba

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

1.7 Sistematika Penulisan

Sistematika penulisan pembuatan perangkat lunak ini dibagi menjadi 5 (lima) bab yang masing-masing bab telah dirancang dengan suatu tujuan tertentu. Berikut penjelasan tentang masing-masing bab :

BAB I : PENDAHULUAN

Bab ini berisi mengenai latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian, metodologi pengembangan, kerangka pemikiran, serta sistematika penulisan yang menguraikan urutan penyajian yang digunakan dalam penyusunan skripsi.

BAB II : LANDASAN TEORI

Bab ini berisi tentang uraian teori-teori yang digunakan dalam analisa permasalahan yang ada dan juga teori-teori yang digunakan dalam perancangan dan implementasi.

BAB III : ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai analisis dari permasalahan yang ada saat ini dan analisis kebutuhan yang diperlukan untuk mengatasi permasalahan tersebut. Pembuatan desain dari sistem dengan mengacu pada analisis yang telah dibahas. Desain sistem yang akan dijelaskan terbagi menjadi tiga bagian, meliputi desain *user interface*, desain data dan desain proses.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Pada bab ini dijelaskan tentang spesifikasi aplikasi, kebutuhan aplikasi, implementasi aplikasi, dan pengujian yang dilakukan terhadap aplikasi yang dibangun.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran untuk pengembangan aplikasi lebih lanjut dalam upaya memperbaiki kelemahan pada aplikasi guna untuk mendapatkan hasil kinerja aplikasi yang lebih baik dan pengembangan program selanjutnya.



BAB II LANDASAN TEORI

2.1 *State Of The Art*

Beberapa penelitian yang telah dilakukan dan memiliki korelasi dengan penelitian ini diantaranya :

Dalam penelitian yang dilakukan oleh Adiwazsha Bramianha menjelaskan tentang penerapan algoritma *greedy* untuk memecahkan masalah pohon merentang minimum. Permasalahan membentuk pohon merentang minimum dari sebuah graf dapat diselesaikan dengan menggunakan algoritma *greedy* (algoritma *Prim* dan algoritma *Kruskal*) yang mengupayakan pengambilan pilihan optimum pada setiap langkah dengan harapan akan mendapatkan hasil optimum global pada akhirnya. Algoritma *Prim* dan *Kruskal* memiliki perbedaan dalam jumlah langkah yang harus diambil, cara penentuan langkah dan aturan dalam pembentukan pohon merentang minimum. Algoritma *Kruskal* menitikberatkan pemilihan sisi berdasarkan urutan bobotnya. Dan karena terlebih dahulu harus mengurutkan sisi berdasarkan bobotnya, algoritma ini lebih cocok untuk pohon dengan jumlah simpul sedikit kurang disarankan untuk pohon dengan jumlah simpul yang banyak.[5]

Dalam penelitian yang dilakukan oleh Permana, Nanda A dan Rahmawan P, Elkaf menjelaskan penerapan algoritma *greedy* untuk menentukan jalur terpendek menuju lokasi kebakaran. Aplikasi tersebut adalah berupa GIS (*Geographic Information System*) untuk menentukan lintasan terpendek. Dalam pencarian lintasan terpendek, ada beberapa algoritma untuk menentukan mana yang sesuai dengan kebutuhan dalam pencarian lintasan terpendek yaitu algoritma dinamis dan algoritma *greedy*. Hal yang membedakan pencarian solusi

menggunakan pemrograman dinamis dengan algoritma *greedy* adalah bahwa keputusan yang diambil pada tiap tahap pada algoritma *greedy* hanya berdasarkan pada informasi yang terbatas sehingga nilai optimum saja yang diperoleh pada saat itu. Penyelesaian masalah penentuan lintasan terpendek yang menggunakan algoritma *greedy* sebagai algoritma utama dalam algoritma tersebut. Dengan penggunaan algoritma *greedy*, maka bisa diambil potongan-potongan solusi sementara yang akan memberikan solusi optimal.[6]

Dalam penelitian yang dilakukan oleh Singgih, Martaleli Bettiza, S.Si., M.Sc dan Eka Suswaini, ST, MT Implementasi algoritma *greedy* dalam menentukan rute trayek angkutan kota (angkot) Tanjung Pinang. Rute trayek angkot kota Tanjung Pinang saat ini bisa dikatakan belum cukup memadai. Mulai dari angkotnya itu sendiri hingga rute trayek nya yang masih dipermasalahkan. Oleh karena itu, aplikasi ini digunakan untuk menentukan rute yang nantinya akan digunakan oleh angkot kota Tanjung Pinang. Aplikasi ini dibangun berbasis *android*, metode yang diterapkan dalam aplikasi ini yaitu algoritma *greedy*.

Proses algoritma *greedy* mempunyai 2 tahap, tahap pertama mengambil semua kemungkinan yang memungkinkan menjadi solusi lokal tanpa melihat pengaruh untuk kedepannya, tahap kedua adalah berharap semua solusi lokal bisa menjadi solusi global atau keseluruhan. Aplikasi ini juga menggunakan *google map* untuk tampilan hasil. Pada aplikasi ini juga mempunyai kebutuhan data berupa titik keramaian dan jarak setiap titik keramaian tersebut.[7]

Dalam penelitian yang dilakukan oleh Harisa Eka Septiarani menjelaskan tentang Pengembangan Sistem Informasi Kuliah Kerja Nyata (KKN) dengan algoritma *greedy* untuk menentukan pengelompokan peserta KKN. Pada

penelitian ini, dikembangkan sebuah Sistem Informasi Kuliah Kerja Nyata (KKN) Universitas Lampung dengan algoritma *greedy* untuk menentukan pengelompokan peserta KKN. Konsep utama algoritma *greedy* adalah mengambil pilihan terbaik yang dapat diperoleh saat itu tanpa memperhatikan konsekuensi ke depan. Penerapan algoritma *greedy* dilakukan dengan komposisi pembagian peserta dalam satu kelompok secara merata berdasarkan fakultas, jenis kelamin, dan jurusan.[8]

Rangkuman *state of the art* dapat dilihat pada Tabel 2.1 :

Tabel 2.1 *State of The Art*

No	Judul	Metode	Hasil
1	Penerapan algoritma <i>greedy</i> untuk memecahkan masalah pohon merentang minimum (Adiwazsha Bramianha 2009).[5]	Algoritma <i>Prim</i> dan Algoritma <i>Kruskal</i>	Algoritma <i>Prim</i> dan <i>Kruskal</i> memiliki perbedaan dalam jumlah langkah yang harus diambil, cara penentuan langkah dan aturan dalam pembentukan pohon merentang minimum.
2	Penerapan Algoritma <i>Greedy</i> untuk menentukan jalur terpendek menuju lokasi kebakaran (Permana, Nanda Adi dan Rahmawan P, Elkaf 2013).[6]	Algoritma <i>Greedy</i>	Aplikasi tersebut adalah berupa GIS (<i>Geographic Information System</i>) untuk menentukan lintasan terpendek yaitu algoritma dinamis dan algoritma <i>greedy</i> .
3	Implementasi algoritma <i>greedy</i> dalam menentukan rute trayek angkutan kota (angkot) tanjung pinang (Singgih, Martaleli Bettiza, S.Si., M.Sc dan Eka Suswaini, ST, MT 2014) [7]	Algoritma <i>Greedy</i>	Aplikasi ini digunakan untuk menentukan rute yang nantinya akan digunakan oleh angkot kota Tanjung Pinang. Aplikasi ini dibangun berbasis android.

Tabel 2.1 *State of The Art* (lanjutan)

No	Judul	Metode	Hasil
4	Pengembangan sistem informasi Kuliah Kerja Nyata (KKN) dengan algoritma <i>greedy</i> untuk menentukan pengelompokan peserta KKN (Harisa Eka Septiarani 2015). [8]	Algoritma <i>Greedy</i>	Penerapan algoritma <i>greedy</i> dilakukan dengan komposisi pembagian peserta dalam satu kelompok secara merata berdasarkan fakultas, jenis kelamin, dan jurusan.
5	Sistem pendukung keputusan penentuan peserta jamkesmas dengan metode simple additive weighting (Studi Kasus : Desa Dalu Sepuluh(Zulkifli (0911001))	Simple Additive Weighting (SAW)	Aplikasi sistem pendukung keputusan untuk membantu dalam penentuan peserta jamkesmas yang berbasis komputerisasi sehingga dapat menyeleksi setiap calon peserta

Berdasarkan Tabel 2.1, maka perbedaan penelitian ini dengan penelitian sebelumnya yaitu terletak pada fitur-fitur yang ada dalam aplikasi, jika pada penelitian sebelumnya mengelola sebuah proses yang lebih menitikberatkan kepada hasil pengelompokkan dan setiap langkah penentuan hasil.

2.2 Perancangan

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem.[9]

2.2.1 Konsep Dasar Perancangan

Perancangan adalah suatu proses penterjemahan kebutuhan pemakai informasi kedalam suatu alternatif rancangan yang diinginkan kepada pemakai informasi untuk dapat dipertimbangkan”.

Perancangan merupakan pengaflikasian berbagai macam teknik dan prinsip untuk tujuan pendefinisian secara rinci suatu perangkat, proses atau *system* sehingga dapat direalisasikan dalam suatu bentuk fisik.[10]

2.3 Konsep Dasar Aplikasi

2.3.1 Pengertian Aplikasi

Aplikasi bersasal dari kata *application* yang artinya penerapan, lamaran, dan penggunaan. Secara istilah aplikasi adalah program yang siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.[9]

2.3.2 Pengertian Aplikasi Web

Pada awalnya aplikasi web dibangun dengan hanya menggunakan bahasa yang disebut HTML (*HyperText Markup Language*). Pada perkembangan berikutnya, sejumlah skrip dan obyek yang dikembangkan untuk memperluas kemampuan HTML, seperti PHP dan ASP pada skrip dan *Applet* pada objek.

Aplikasi *Web* dapat dibagi menjadi dua jenis yaitu aplikasi web statis dan dinamis. *Web Statis* dibentuk dengan menggunakan HTML. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi. Kelemahan ini diatasi oleh model aplikasi *web* dinamis. Pada aplikasi web dinamis, perubahan informasi dalam halaman *web* dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi *web* dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung jawab dari *webmaster*. Arsitektur aplikasi web meliputi klien, *web server*, *middleware* dan basis data. Klien berinteraksi dengan web server.

Secara internal, *web server* berkomunikasi dengan *middleware* dan *middleware* yang berkomunikasi dengan basis data. Contoh *middleware* adalah PHP dan ASP. Pada mekanisme aplikasi *web* dinamis, terjadi tambahan proses yaitu *server* menterjemahkan kode PHP menjadi HTML. Kode PHP yang diterjemahkan oleh mesin PHP yang akan diterima oleh klien.[11]

2.4 Kuliah Kerja Nyata Mahasiswa

2.4.1 Pengertian Kuliah Kerja Nyata Mahasiswa

Kegiatan Kuliah Kerja Nyata Mahasiswa (yang selanjutnya disingkat dengan KKM) Universitas Islam Negeri Sunan Gunung Djati Bandung merupakan suatu kegiatan akademik mahasiswa yang berlangsung melalui tahapan pembelajaran, penelitian dan pengabdian masyarakat. Lebih jauh, KKM merupakan bagian pembelajaran dengan masyarakat (*learning with community*) sebagai bentuk pengamalan IPTEKS yang telah dipelajari oleh para mahasiswa selama perkuliahan dikampus. Karena itu, KKM harus berorientasi pada VISI UIN Bandung, yaitu “Menjadi Universitas yang Unggul dan Kompetitif dengan bingkai akhlak karimah di ASEAN berbasis wahyu memandu ilmu”. KKM adalah kegiatan intrakurikuler terpadu Tri Dharma Perguruan Tinggi dengan menempatkan mahasiswa diluar kampus dalam kurun waktu tertentu, untuk melakukan edukasi dan mendampingi masyarakat secara professional.[1]

2.4.2 Penjadwalan Pelaksanaan

Kuliah Kerja Nyata Mahasiswa Universitas Islam Negeri Sunan Gunung Djati Bandung dilaksanakan Dalam dua periode setiap tahunnya yang diikuti oleh ribuan mahasiswa setiap periode. Prosedur pelaksanaan Kuliah Kerja Nyata Mahasiswa ini dilakukan oleh setiap mahasiswa dengan mendaftarkan diri sesuai dengan syarat yang telah ditentukan oleh pihak Universitas Islam Negeri Sunan

Gunung Djati Bandung. Setiap anggota kelompok terdiri atas beberapa mahasiswa dari berbagai jurusan. Kelompok - kelompok tersebut ditempatkan di lokasi desa yang berbeda di setiap kabupaten.[1]

2.4.3 Prosedur Pelaksanaan

Universitas Islam Negeri Sunan Gunung Djati Bandung yang telah diimplementasikan sejak pelaksanaan Kuliah Kerja Nyata Mahasiswa proses pengelompokan aplikasi tersebut dilakukan secara acak (*random*) dengan menggunakan kriteria jenis kelamin, jurusan, dan fakultas.[1]

2.5 Algoritma Greedy

Algoritma greedy merupakan algoritma yang lazim digunakan untuk memecahkan masalah optimasi. Konsep utama algoritma ini adalah mengambil kesempatan yang ada saat itu juga tanpa memperhatikan konsekuensi ke depan, atau biasa disebut dengan prinsip “*take what you can get now!*”.

Setiap keputusan yang diambil diharapkan merupakan langkah optimum pada langkah tersebut, dikenali sebagai solusi optimum lokal, kemudian dengan setiap langkah yang ditempuh diharapkan dapat memperoleh solusi optimum diakhir proses, yaitu solusi optimum global. Algoritma greedy ini akan membentuk solusi langkah per langkah (*step by step*). Pada setiap langkah banyak pilihan yang perlu dieksplorasi oleh karena itu pada setiap langkah perlu dibuat keputusan yang terbaik dalam menentukan pilihan. Pada algoritma ini juga terdapat elemen yang diperlukan, yaitu[2]:

- a) Himpunan Kandidat (C)

Himpunan ini berisi elemen-elemen pembentuk solusi.

- b) Himpunan Solusi (S)

Himpunan ini berisikan kandidat-kandidat yang terpilih sebagai solusi persoalan.

c) Fungsi Seleksi

Ini merupakan fungsi yang dipakai untuk memilih kandidat yang paling memungkinkan mencapai solusi optimal.

d) Fungsi Kelayakan (*Feasible*)

Fungsi ini digunakan untuk memeriksa suatu kandidat yang telah dipilih dapat memberikan solusi yang layak atau tidak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada.

e) Fungsi Obyektif

Fungsi ini merupakan fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Dalam menentukan solusi optimum dari persoalan optimasi dengan algoritma *greedy*, pembuatan prosedur *greedy* dilakukan dengan masukan berupa himpunan kandidat C , dan keluaran berupa himpunan solusi S . Skema umum algoritma *greedy* adalah sebagai berikut[2].

```
Procedure greedy (input C : himpunan_kandidat, Output
S: himpunan solusi)
{Mengembalikan solusi optimum dari persoalan optimasi
dengan algoritma greedy
Masukan : himpunan kandidat C
```

```

Keluaran : himpunan kandidat solusi S}
Deklarasi x : kandidat
Algoritma :
  S ← { }
  while (belum Solusi (S) ) and (C ≠ { } ) do
    x ← SELEKSI (C);
    C ← C - {x}
    if LAYAK (S U {X} ) then
      S ← S U {x}
    endif
  endwhile
  {SOLUSI (S) sudah diperoleh or C ={ }}

```

Gambar 2.1 Skema umum algoritma *greedy* [2]

2.6 Metode Pengembangan *Prototype*

2.6.1 Konsep Dasar *Prototype*

Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode *prototype*. *Prototype* paradigma (Gambar 2.2) dimulai dengan mengumpulkan kebutuhan. Pengembangan dan pelanggan bertemu dan mendefinisikan keseluruhan sistem yang akan dibuat, mengidentifikasi segala kebutuhan yang diketahui, kemudian melakukan perancangan.



Gambar 2.2 Metode Pengembangan *Prototype* [12]

Prototype bisa berfungsi sebagai sistem yang pertama, memang benar bahwa baik pelanggan maupun pengembang menyukai paradigma *prototype*. Para

pemakai merasa enak dengan sistem actual. Berikut tahapan-tahapan yang dilakukan model pengembangan *prototype* :

2. Mendengarkan pelanggan

Pada tahap ini dilakukan pengumpulan kebutuhan dari *system* dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu *system* yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana *system* yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

3. Merancang dan Membuat Prototype

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

4. Uji Coba

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

Secara ideal prototipe berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. *Prototype* bisa menjadi paradigma yang efektif bagi rekayasa perangkat lunak. Kuncinya adalah mendefinisikan aturan-aturan main pada saat awal, yaitu pemakai dan pengembang keduanya harus setuju bahwa *prototype* dibangun untuk berfungsi sebagai mekanisme pendefinisian kebutuhan.

2.6.2 Keunggulan dan Kelemahan *Prototyping*

Keunggulan *prototyping* adalah:

1. Adanya komunikasi yang baik antara pengembang dan pelanggan.
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
3. Pelanggan berperan aktif dalam pengembangan sistem.
4. Lebih menghemat waktu dalam pengembangan sistem.
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Kelemahan *prototyping* adalah :

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek, sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat *prototyping* lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan cetak biru sistem.
3. Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik.

Prototyping bekerja dengan baik pada penerapan-penerapan yang mempunyai ciri-ciri sebagai berikut :

1. Resiko tinggi yaitu untuk masalah-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu dan adanya syarat data yang tidak menentu.
2. Interaksi pemakai penting

Sistem harus menyediakan dialog on-line antara pelanggan dan komputer.

3. Perlunya penyelesaian yang cepat.

Perilaku pemakai yang sulit ditebak.

4. Sistem yang inovatif.

Sistem tersebut membutuhkan cara penyelesaian masalah dan penggunaan perangkat keras yang mutakhir

5. Perkiraan tahap penggunaan sistem yang pendek

2.7 *Framework CodeIgniter*

CodeIgniter adalah sebuah *framework PHP* yang dapat membantu mempercepat *develover* dalam pengembangan aplikasi *web* berbasis *php* dibanding jika menulis semua kode darai awal.

CodeIgniter menyediakan banyak *library* untuk mengerjakan tugas-tugas yang umumnya ada pada sebuah aplikasi berbasis *web*. Selain itu, struktur dan susunan logis dari *CodeIgniter* membuat aplikasi yang dibuat menjadi lebih teratur dan rapi. Dengan demikian, ketika mengerjakan dapat difokuskan pada fitur-fitur apa yang dibutuhkan oleh aplikasi anda dengan membuat kode program seminimal mungkin.

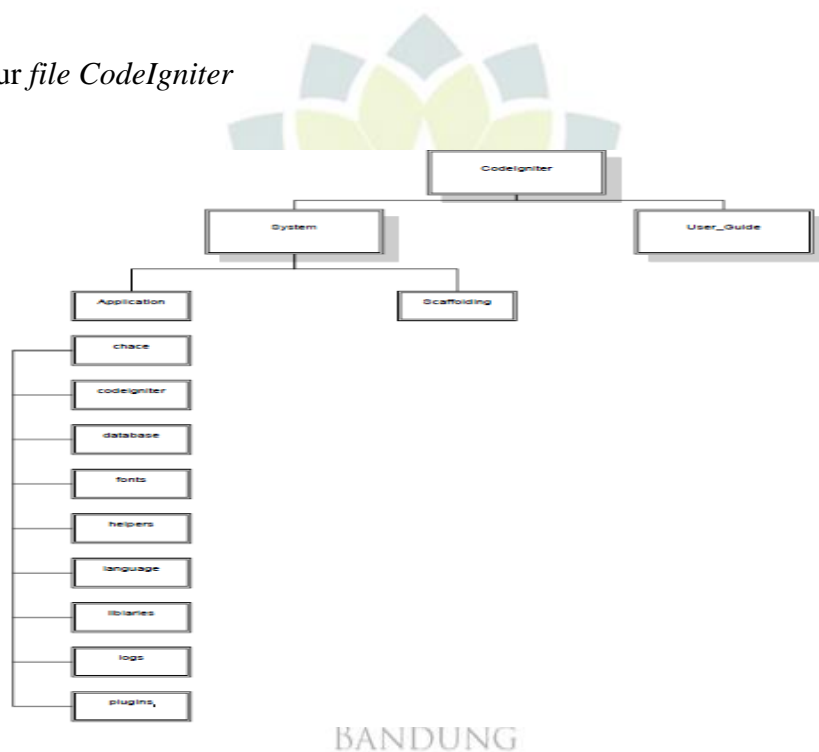
CodeIgniter menggunakan konsep *MVC (model view controller)*, yaitu sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi.

- a. *Model* mempresentasikan struktur data yang dibangun. Umumnya kelas model berisi fungsi-fungsi yang membantu *developer* untuk mengelola, memasukan, dan mengupdate informasi dalam *database*.

b. *View* informasi yang disajikan untuk user berupa tampilan atau *user interface*. *View* umumnya adalah tampilan sebuah halaman *web* itu sendiri, tetapi dalam *CodeIgniter*, *view* dapat juga menjadi bagian-bagian atau penggalan-penggalan halaman seperti *header* atau *footer*.

c. *Controller* bertugas sebagai penghubung antara *model*, *view* dan beberapa *resource* lainnya yang dibutuhkan untuk memproses *HTTP request* dan untuk meng-*generate* sebuah halaman web. [13]

d. Struktur file *CodeIgniter*



Gambar 2.3 Struktur file *CodeIgniter* [14]

a. Keuntungan menggunakan *CodeIgniter*[15]

Ada beberapa keuntungan dalam menggunakan *framework CodeIgniter* :

1. Gratis (*free*)

CodeIgniter berlisensi dibawah *Apache/BSD open source*, jadi bisa digunakan secara bebas dan gratis.

2. Ditulis menggunakan *php4*

Meskipun *CodeIgniter* dapat berjalan pada *php 5*, namun pada saat ini kode program *CodeIgniter* masih dibuat dengan *php 4*. Hal ini dilakukan agar *CodeIgniter* dapat berkembang lebih luas di komunitas *php*. Karena hingga saat ini, *web hosting* masih menggunakan *php 4*. Jika *CodeIgniter* dibuat dengan *php 5*, tentu saja hasilnya juga akan jauh lebih canggih, karena bisa memanfaatkan teknologi *php 5* yang saat ini masih belum dapat dilakukan oleh *php 4*, misalnya untuk menerapkan konsep *OOP multiple inheritance*.

3. Berukuran kecil

Ukuran *CodeIgniter* yang kecil merupakan keunggulan tersendiri. Dibanding *framework* lain yang berukuran besar, serta membutuhkan *resource* yang besar pula untuk berjalan. Pada *CodeIgniter*, bisa diatur agar *system meload library* yang dibutuhkan saja, sehingga sistem dapat berjalan ringan dan cepat.

4. Menggunakan konsep MVC

CodeIgniter menggunakan konsep *M-V-C (model-view-controler)* yang memungkinkan pemisahan antara *layer* aplikasi (*application-logic*) dan *presentation*.

5. URL yang sederhana

Secara *default*, *URL* yang dihasilkan *CodeIgniter* sangat bersih dan *search engine friendly*.

6. Memiliki paket *Library* yang lengkap

CodeIgniter memiliki *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis *web*, misal mengakses *database*, mengirim *email*, memvalidasi form, menangani *session*.

7. Extensible

Sistem dapat dikembangkan dengan mudah dengan menggunakan plugin dan helper, atau dengan menggunakan hooks.

8. Tidak Memerlukan *Template Engine*

Meskipun *template* dilengkapi dengan *template parser* sederhana yang dapat digunakan tetapi hal ini tidak mengharuskan untuk menggunakannya. Penggunaan *template engine* dapat mengurangi *performance* dari sistem.

9. Dokumentasi Lengkap Dan Jelas

Dari sekian *Framework*, *CodeIgniter* adalah satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas. Dari tim pengembang *CodeIgniter* berkomitmen bahwa dokumentasi juga sama pentingnya dengan kode program *CodeIgniter* itu sendiri. *Source Code CodeIgniter* juga dilengkapi komentar didalamnya, sehingga memperjelas fungsi sebuah kode program.

10. Komunitas

Komunitas pengguna *CodeIgniter* saat ini berkembang pesat, salah satu forum terbesar *CodeIgniter* <http://codeigniter.com/forums/>.

2.8 UML (*Unified Modeling Language*)

Bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. Definisi lain juga menyatakan *Unified Modeling Language (UML)* adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (*OOP*) serta aplikasinya.[16]

UML menyediakan beberapa macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu diantaranya :[16]

1. Use Case Diagram

2. *Class Diagram*
3. *Behaviour Diagram*
 - a. *Activity Diagram*
 - b. *Statechart Diagram*
 - c. *Interaction Diagram*
 - a. *Sequence Diagram*
 - b. *Collaboration Diagram*
5. *Implementation Diagram*
 - i. *Component Diagram*
 - ii. *Deployment Diagram*

Berikut akan dijelaskan 3 macam diagram yang penulis gunakan dalam pembangunan aplikasi berorientasi object, yaitu *use case diagram*, *class diagram*, *sequence diagram* dan *activity diagram*.



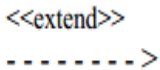
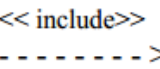
2.8.1 *Diagram Use Case (Use Case Diagram)*

Use Case diagram adalah model fungsional sebuah sistem yang menggunakan *actor* dan *case* (proses). *Use Case* adalah layanan (*services*) atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunanya. *Use Case* adalah suatu pola atau gambaran yang menunjukkan kelakuan atau kebiasaan sistem. Setiap *Use Case* adalah suatu urutan (*sequence*) transaksi yang saling berhubungan dan dilakukan oleh sebuah actor dan sistem dalam bentuk sebuah *dialog*. *Use Case diagram* mempresentasikan kegunaan atau fungsi-fungsi sistem dari perspektif pengguna.[17]

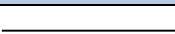
Use case diagram menggambarkan apa yang dilakukan aktor terhadap sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari


sebuah sistem. Sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses dalam *use case* dirinya. Sebuah *use case* juga dapat melakukan *extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan *generalisasi* antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Tabel 2.2 di bawah ini memuat notasi simbol-simbol yang digunakan dalam diagram *use case*.

Tabel 2.2 Simbol dalam Diagram *Use Case* [17]

No	Lambang	Nama	Keterangan
1		<i>Actor</i>	Aktor (<i>actor</i>) merupakan sebuah entitas yang berinteraksi dengan <i>use case</i> . Nama aktor dituliskan di bawah gambar tersebut. Aktor dapat berupa orang atau sistem lain diluar sistem yang tengah dianalisis.
2		<i>Use Case</i>	<i>Use case</i> menggambarkan sebuah fungsi tertentu yang disediakan oleh sistem, sebuah subsistem atau urutan pertukaran pesan antar anggota sistem dan satu atau lebih aktor melakukan aksi yang dikerjakan oleh sistem.
3		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
4		<i>Include</i>	Relasi <i>use case</i> yang dimana proses bersangkutan akan dilanjutkan ke proses yang dituju.

Tabel 2.2 Simbol dalam Diagram *Use Case* (lanjutan) [17]

No	Lambang	Nama	Keterangan
5		<i>Asosiasi</i>	Komunikasi antara <i>actor</i> dan


			<i>use case</i> yang berpartisipasi pada <i>usecase</i> atau <i>usecase</i> memiliki interaksi dengan actor
6		<i>Generalisasi</i>	Dimana fungsi yang satu adalah fungsi yang umum dari yang lainnya.

2.8.2 Activity Diagram

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *Statechart*. Diagram *Statechart* berfokus pada objek dalam suatu proses (atau proses menjadi suatu objek), diagram *Activity* berfokus pada aktivitas-aktivitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktivitas-aktivitas tersebut bergantung satu sama lain.

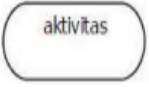



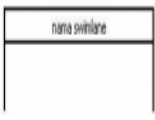
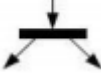

Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir di pusat lingkaran *stop* hitam atau putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi. Berikut adalah simbol yang digunakan dalam *activity diagram* :

Tabel 2.3 Simbol Activity Diagram [17]

No	Simbol	Nama	Keterangan
1		Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal

Tabel 2.3 Simbol Activity Diagram (lanjutan) [17]

No	Simbol	Nama	Keterangan
----	--------	------	------------

2		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja
3		Pecabangan atau <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		Penggabungan atau <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
7		<i>Fork</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara <i>parallel</i>
8		<i>Join</i>	Digunakan untuk menunjukkan kegiatan yang digabungkan

2.8.3 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok : nama, atribut, dan metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut[18] :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.

2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*.

Dengan demikian *interface* mendukung resolusi metode pada saat *run time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Hubungan antar *Class* terdiri dari sebagai berikut[18] :

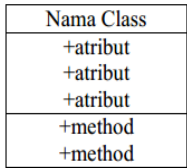





1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain atau *class* yang harus mengetahui eksistensi *class* lain.

2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).

3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga bisa disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah *generalisasi*.

4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang dipassing dari satu *class* kepada *class* lain. Berikut adalah simbol yang dimiliki *class* diagram :

Tabel 2.4 Simbol *Class Diagram*[17]

No	Simbol	Nama	Keterangan
1		Kelas	Kelas adalah bentuk umum dari objek-objek yang memiliki struktur dan tingkah laku (atribut, operasi, relasi dan arti/definisi) yang sama. Kelas merupakan bentuk abstrak dari objek di dunia nyata.
2		Antar muka atau <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
3		Asosiasi atau <i>association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>
4		Asosiasi berarah atau <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai <i>multiplicity</i> .
5		<i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
6		Agregasi atau <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole part</i>).



2.8.4 Sequence Diagram

Interaction diagram menunjukkan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa *behavior*. UML memiliki beberapa bentuk *interaction diagram* dan yang paling umum digunakan adalah *sequence diagram*. Sebuah *sequence diagram* adalah diagram yang secara khusus menjabarkan urutan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang mewakili objek ini di dalam *use case*. [17]



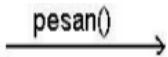
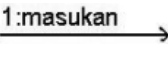
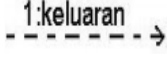
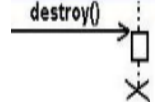
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metode dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. [17]

Adapun komponen-komponen yang digunakan dalam *sequence diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Simbol *Sequence Diagram*[17]

No	Simbol	Nama	Keterangan
1		<i>Lifeline</i>	a. Menandakan kehidupan objek selama urutan. b. Diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi.
2		Objek	a. Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan. b. Ditempatkan di bagian atas diagram.

Tabel 2.5 Simbol *Sequence Diagram* (lanjutan) [17]

No	Simbol	Nama	Keterangan
3		Aktor	<p>a. Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem.</p> <p>b. Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan.</p> <p>c. Ditempatkan dibagian atas diagram.</p>
4		Objek sedang aktif berinteraksi	<p>Fokus control :</p> <p>a. Adalah persegi panjang yang sempit panjag ditempatkan di atas sebuah garis hidup.</p> <p>b. Menandakan ketika suatu objek mengirim atau menerima pesan.</p>
5		Pesan	Objek mengirim satu pesan ke objek lainnya.
6		1:masukan	Menyatakan bahwa suatu objek mengirimkan masukan ke objek lainnya. Arah panah mengarah pada objek yang dikirim.
7		1:keluaran	Objek atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
8		<i>Destroy()</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

2.9 Basis Data

2.9.1 Pengertian Basis Data

Data base atau basis data, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system, DBMS*). Sistem basis data dipelajari dalam ilmu informasi. Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya : penjelasan ini disebut skema. Skema menggambarkan objek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data. Ini dikenal sebagai model basis data atau model data.[18]

2.9.2 Model Data

Model data adalah sekumpulan cara / peralatan / tool untuk mendeskripsikan data-data, hubungannya satu sama lain, semantiknya, serta batasan konsistensi. Ada dua model data, yaitu : *Entity Relationship Diagram (ERD)* dan model relasional. Keduanya menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika :

1. *Conceptual Data Model (CDM)*

Model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas-entitas itu.

2. *Physical Data Model (PDM)*

Model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik.

Memiliki 4 jenis objek, yaitu [19] :

a. Entitas

Segala sesuatu yang dapat digambarkan oleh data. Entitas juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain.

Ada dua macam entitas yaitu entitas kuat dan entitas lemah. Entitas kuat merupakan entitas yang tidak memiliki ketergantungan dengan entitas lainnya. Contohnya entitas anggota. Sedangkan entitas lemah merupakan entitas yang kemunculannya tergantung pada keberadaan entitas lain dalam suatu relasi.

b. *Strong entity* (entitas kuat)

Entitas yang mandiri, yang keberadaannya tidak bergantung pada keberadaan entitas yang lainnya. Instansiasi entitas kuat selalu memiliki karakteristik yang unik disebut identifier (sebuah atribut tunggal atau gabungan atribut-atribut yang secara unik dapat digunakan untuk membedakannya dari entitas kuat yang lain).

c. *Weak entity* (entitas lemah)

Entitas yang keberadaannya sangat bergantung pada keberadaan entitas yang lainnya. Entitas lemah tidak memiliki arti apa-apa dan tidak dikehendaki kehadirannya dalam diagram ER tanpa kehadiran entitas di mana mereka bergantung. Entitas di mana entitas lemah bergantung dinamakan *identifying*

owner. Entitas lemah tidak memiliki *identifier* sendiri. Secara umum, dalam diagram ER entitas lemah memiliki atribut yang berperan sebagai partial *identifier* (*identifier* yang berfungsi secara sebagian).

d. Atribut

Atribut merupakan pendeskripsian karakteristik dari entitas. Atribut digambarkan dalam bentuk lingkaran atau elips. Atribut yang menjadi kunci entitas atau key diberi garis bawah.

e. *Relasi* atau Hubungan

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Derajat *relasi* atau *kardinalitas* menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Macam-macam *kardinalitas* adalah :

1. Satu ke satu (*one to one*), Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.
2. Satu ke banyak (*one to many*), Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.
3. Banyak ke banyak (*many to many*), Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

f. *Agregasi*

Agregasi adalah suatu keadaan di mana suatu relasi hanya dapat direalisasikan setelah relasi yang lain ada terlebih dahulu. Relasi yang bertipe *agregasi* tidak dimungkinkan ada jika relasi yang menjadi prasyaratnya tidak terealisasi. Sesungguhnya *agregasi* dapat dipandang sebagaimana relasi pada umumnya (yang menghubungkan 2 entitas). Karena relasi ini dibentuk dari relasi

lain (relasi prasyarat) yang secara kronologis lebih dulu terbentuk, maka pengimplementasiannya juga harus dilakukan setelah relasi prasyarat tersebut terimplementasikan. Selanjutnya kita tinggal meninjau derajat relasi dari relasi agregasinya.

1. Total constraint adalah constraint yang mana data dalam entitas yang memiliki constraint tersebut terhubung secara penuh ke dalam entitas dari relasinya.
2. *Constraint partial* adalah constraint yang mana data dalam entitas yang memiliki *constraint* tersebut terhubung ke dalam entitas dari relasinya.

Fungsi dari pembuatan model data diantaranya sebagai berikut :

1. Untuk menghindari *Redudancy*
2. Menghemat penyimpanan (*storage*) data
3. Menambah efektifitas dan kecepatan akses
4. Untuk menghindari terjadinya asinkronisasi data pada saat *diupdate*. [19]

2.10 MYSQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (bahasa Inggris : *database management system*) atau *DBMS* yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. (Anonim)

a. Kelebihan MYSQL

Selain karena *Open Source* program, *MySQL* juga memiliki kelebihan-kelebihan yang tak kalah bagusnya dengan *Database Server* lainnya, seperti *SQL server*, *Sybase* bahkan *Oracle*. Kelebihan-kelebihan itu antara lain [20] :

1. Dapat bekerja di beberapa *platform* yang berbeda, seperti *LINUX*, *Windows*, *MacOS* dll.
2. Dapat dikoneksikan pada bahasa *C*, *C++*, *Java*, *Perl*, *PHP* dan *Python*.
3. Memiliki lebih banyak tipe data seperti : *signed/unsigned integer* yang memiliki panjang data sebesar 1,2,3,4 dan 8 *byte*, *FLOAT*, *DOUBLE*, *CHAR*, *VARCHAR*, *TEXT*, *BLOB*, *DATE*, *TIME*, *DATETIME*, *TIMESTAMP*, *YEAR*, *SET* dan tipe *ENUM*.
4. Mendukung penuh terhadap kalimat SQL *GROUP BY* dan *ORDER BY*.
Mendukung terhadap fungsi penuh (*COUNT ()*, *COUNT (DISTINCT)*, *AVG ()*, *STD ()*, *SUM ()*, *MAX ()* AND *MIN ()*).
5. Mendukung terhadap *LEFT OUTER JOIN* dengan *ANSI SQL* dan sintak *ODBC*.
6. Mendukung *ODBC for Windows 95* (dengan source program). Semua fungsi *ODBC 2.5* dan sebagainya. Sebagai contoh kita dapat menggunakan *Access* untuk connect ke *MySQL server*.
7. Menggunakan *GNU automake*, *autoconf*, dan *LIBTOOL* untuk portabilitas.
8. Dapat menggabungkan beberapa tabel dari *database* yang berbeda dalam *query* yang sama.
9. Ditulis dengan menggunakan bahasa *C* dan *C++*. Diuji oleh *compiler* yang sangat jauh berbeda.
10. *Privilege* (hak) dan password sangat fleksibel dan aman serta mengujikan '*Host-Based*' Verifikasi.[20]

2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah sebuah elemen topik yang memiliki

cakupan luas dan sering dikaitkan dengan verifikasi dan validasi. Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik.

Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai kebutuhan pelanggan.[21]

2.11.1 Teknik Pengujian *Black-Box*

Teknik pengujian yang digunakan adalah pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada *white-box* [21].

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,

Dan selanjutnya akan dilakukan pengujian untuk mengetahui presentase jumlah pengujian yang benar dari hasil analisa.

2.11.2 Pengujian Kinerja Algoritma *Greedy*

Pengujian hasil pengelompokan peserta KKM menggunakan data peserta periode tahun 2016, dimana periode tahun 2016 sejumlah 2400 peserta. Penentuan batas maksimal jumlah anggota setiap kelompok adalah 12 anggota yang diinginkan untuk memulai proses pengelompokan.

Menurut pihak LP2M UIN SGD Bandung, pengujian kinerja algoritma *greedy* berusaha menemukan ideal atau tidak ideal dalam kategori kriteria ideal dalam satu kelompok KKM yakni apabila :

1. Setiap kelompok terdiri dari maksimal 12 peserta.
2. Setiap anggota kelompok berasal dari fakultas (3 peserta) dan jurusan yang sama (3 peserta) dan dengan perbedaan jenis kelamin yaitu laki-laki 7 peserta dan wanita 5 peserta untuk batas maksimal.

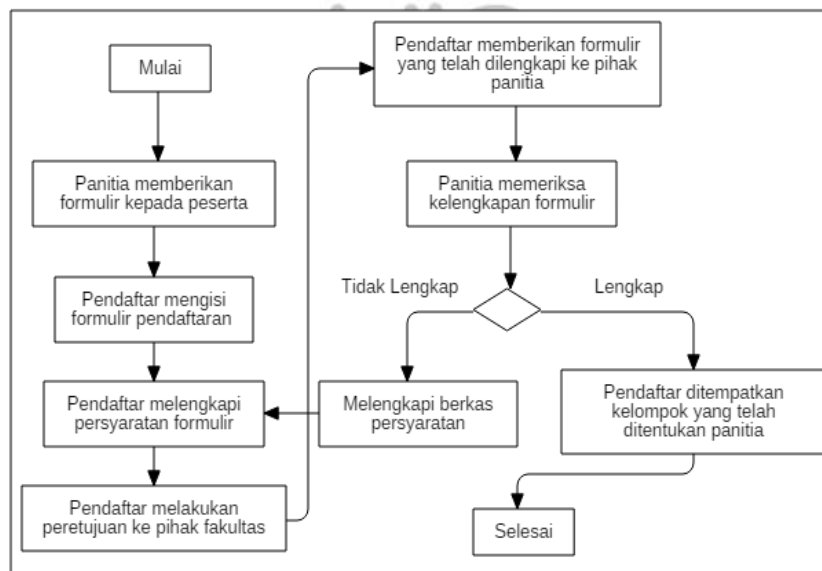
Dan selanjutnya akan dilakukan pengujian untuk mengetahui presentase jumlah pengujian yang benar dari hasil analisa.[8]

BAB III ANALISIS DAN PERANCANGAN

Berdasarkan hasil pengumpulan data yang telah dilakukan dengan berbagai metode yang telah ditentukan dari mulai wawancara, penelitian kepustakaan (landasan teori), serta penelitian langsung terhadap bagian administrasi atau Instansi/Lembaga Peneliti dan Pengembangan Masyarakat (LP2M) Universitas Islam Negeri Sunan Gunung Djati Bandung yang mempunyai wewenang dalam permasalahan yang telah dirumuskan, menghasilkan berbagai perancangan yang sebaiknya dibuat untuk memenuhi berbagai kebutuhan.

3.1 Analisis *Current* Sistem (*Quick Plan Phases*)

Proses pendaftaran calon peserta KKM di UIN Sunan Gunung Djati Bandung saat ini masih menggunakan sistem konvensional. Berikut di bawah urutan proses yang terjadi dalam pendaftaran calon peserta KKM sistem konvensional.



Gambar 3.1 *flowchart current system* pendaftaran peserta KKM konvensional

Keterangan gambar 3.1 :

1. Panitia memberikan formulir pendaftaran beserta persyaratan yang harus dilengkapi pihak pendaftar.
2. Calon peserta melakukan pengisian formulir pendaftaran.
3. Calon peserta melengkapi persyaratan yang dicantumkan dan mengisi formulir pendaftaran dengan lengkap.
4. Calon peserta melakukan persetujuan ke pihak fakultas masing-masing.
5. Calon peserta mengembalikan ke petugas pendaftaran atau panitia, formulir yang sudah dilengkapi.
6. Pihak panitia melakukan pemeriksaan terhadap formulir peserta dan melakukan *sorting* per jurusan dengan formulasi seperti yang telah dijelaskan di BAB II.
7. Pihak panitia melakukan pengumuman hasil seleksi di papan informasi dan di *website official* UIN Sunan Gunung Djati Bandung.

Dari gambaran sistem di atas, setelah dianalisis ke dalam bentuk sistem *website* maka alur prosesnya menjadi sebagai berikut :



Gambar 3.2. Alur *system* pendaftaran peserta KKM berbasis *website*

Keterangan gambar 3.2 :

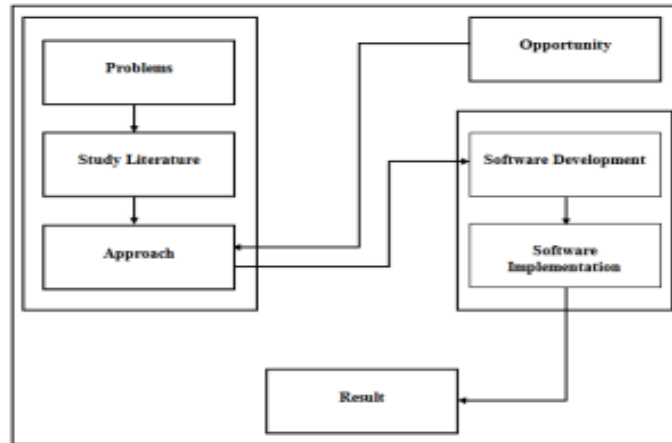
1. Calon peserta *login* dengan *username* dan *password* sesuai dengan akun calon peserta dan apabila belum memiliki akun mengisi *form* data diri akun terlebih dahulu.
2. Calon peserta melakukan pengisian formulir pendaftaran secara lengkap dan mengirimkan berkas persyaratan yang telah ditentukan yaitu : Pas foto, surat persetujuan mengikuti KKM dari pihak fakultas masing-masing dan resi pembayaran spp.
3. Calon peserta mencetak formulir pendaftaran yang telah diisi oleh calon peserta
4. Calon peserta melakukan persetujuan formulir kepada pihak fakultas.
5. Sistem menampilkan hasil seleksi di *web* Aplikasi KKM dan peserta melakukan pengecekan kelompok.
6. Calon peserta melakukan *upload* formulir yang telah disetujui pihak fakultas agar tervalidasi sebagai peserta KKM.

Dari proses di atas setelah dianalisis maka ada tiga aktor yang berinteraksi terhadap sistem yaitu admin, dosen pembimbing lapangan dan pendaftar. Setiap aktor mempunyai hak akses yang berbeda terhadap sistem.

Untuk penjelasan lebih *detail* mengenai sistem secara keseluruhan dan aktor yang berinteraksi terhadap sistem, berikut di bawah akan dijelaskan deskripsi secara keseluruhan.

3.2 Kerangka Kerja Konseptual

Gambar 3.3 merupakan gambar kerangka pemikiran atau kerangka kerja konseptual dari aplikasi yang akan dibangun.



Gambar 3.3 Kerangka Kerja Konseptual

Penjelasan kerangka kerja pada Gambar 3.3 diatas yaitu sebagai berikut :

1. *Problems*

Masalah yang terjadi dalam penelitian ini yaitu untuk menentukan dan pengelompokkan anggota kelompok pada KKM dengan acak (*random*) secara *on-line* atau terkomputerisasi . Sehingga membuat bingung pihak lembaga yang belum menerapkan sistem secara *on-line* dan pengacakan dan penentuan pengelompokkan manual yang sangat rumit.

2. *Study Literature*

Setelah adanya permasalahan itu, dilakukan studi literatur yang dapat membantu pemecahan masalah yang terjadi. Studi literatur yang dilakukan yaitu dengan mencari data-data yang dibutuhkan. Data yang dibutuhkan dalam penelitian ini dicari melalui wawancara, *website*, artikel-artikel serta buku yang berkaitan dengan masalah penelitian.

3. *Opportunity*

Selain melakukan studi literatur, adanya peluang dalam penelitian ini. Peluang tersebut diantaranya belum adanya aplikasi yang dapat melakukan

penentuan dan pengelompokkan anggota kelompok dengan menggunakan metode pengacakan. Selain itu, aplikasi yang akan dibuat juga berbasis *website* karena penggunaan sistem *on-line* yang sedang berkembang dan sangat mudah dijangkau oleh setiap pengguna dengan menggunakan koneksi internet.

4. *Approach*

Setelah melakukan studi literature dan adanya peluang untuk membuat aplikasi berbasis *website* untuk memecahkan masalah di atas maka didapatkan metode yang cocok untuk digunakan dalam penelitian, yaitu metode (*Greedy Algorithm*).

5. *Software Development*

Metode pengembangan perangkat lunak yang digunakan yaitu *Prototype*.

6. *Software Implementation*

Implementasi sistem yang akan dibuat yaitu proses sistem terjadi dengan mengambil data dari peserta yang akan ditentukan oleh sistem dan *output* dari sistem yaitu *report* atau hasil pengelompokkan anggota kelompok yang telah tersusun.

7. *Result*

Hasil akhir dari proses-proses di atas yaitu sebuah aplikasi Kuliah Kerja Nyata Mahasiswa untuk menentukan dan mengelompokkan peserta KKM dengan menggunakan algoritma *greedy* yang berbasis *website* dengan menggunakan *framework codeigniter*.

3.3 Analisis Kebutuhan Hardware dan Software

Setelah dilakukan analisis sistem yang sedang berjalan, kemudian dilakukan persiapan untuk pembuatan sistem aplikasi penentuan dan pengelompokan peserta KKM dengan menggunakan algoritma *greedy* di LP2M UIN SGD Bandung. Pada tahapan ini dilakukan analisis kebutuhan *hardware* dan *software* yang digunakan untuk membuat sistem secara *minimum requirement*. Kebutuhan *hardware* dan *software* yang digunakan dalam pembuatan sistem yang akan digunakan yaitu sebagai berikut :

1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam perancangan aplikasi ini (*minimum requirement*) adalah

a. 1 (Satu) unit *Netbook* dengan spesifikasi :

- *Processor* : Pentium 4
- *Installed memory* (RAM) : 2,00 GB
- *System type* : 32 bit *Operating System*

b. *Printer* Epson TX121X

Mencetak *report* atau laporan pengelompokan peserta.

2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam pengembangan sistem ini adalah :

- a. Sistem Operasi Windows XP
- b. *Notepad ++* versi 1.0
- c. MySQL
- d. XAMPP versi 3.2.1
- e. *Web Browser* (*Google Chrome*) versi 5.0.375

f. *Star UML* versi 2.2.23

g. *Balsamiq Mockups* versi 3.3.11

3.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik. Analisis yang dilakukan serta dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain deskripsi global dari aplikasi, arsitektual sistem yang digunakan.

3.4.1 Deskripsi Global Aplikasi

Perangkat lunak penentuan dan pengelompokan peserta KKM dengan menggunakan algoritma *greedy* merupakan perangkat lunak yang diterapkan di *website* yang berisi tentang dari mulai suatu proses pendaftaran sampai dengan proses pengelompokan peserta, yang akan dijadikan sebagai sebuah kelompok suatu peserta dalam agenda pelaksanaan Kuliah Kerja Nyata Mahasiswa.

1. Analisis Penentuan Dengan Algoritma *Greedy*

Algoritma *greedy* membentuk solusi langkah per langkah (*step by step*).

Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat mengubah lagi pada langkah selanjutnya.

Proses pengelompokan sistem informasi tersebut dilakukan secara acak (*random*) dengan menggunakan kriteria :

1. Jenis kelamin,

2. Jurusan, dan

3. Fakultas

Menurut pihak LP2M UIN SGD Bandung, dengan ketentuan sebagai berikut *system* akan menemukan solusi optimal dalam kategori kriteria ideal dalam satu kelompok KKM yakni apabila :

1. Setiap kelompok terdiri dari maksimal 12 peserta.

2. Setiap anggota kelompok berasal dari fakultas (3 peserta) dan jurusan yang sama (3 peserta) dan dengan perbedaan jenis kelamin yaitu laki-laki 7 peserta dan wanita 5 peserta untuk batas maksimal.

Berdasarkan hal tersebut, penelitian ini dilakukan dengan mengembangkan Aplikasi Kuliah Kerja Nyata Mahasiswa yang berfokus pada proses pengelompokan peserta KKM menggunakan algoritma *greedy* dengan komposisi pembagian peserta dalam satu kelompok secara merata berdasarkan hal diatas.

Pada pengelompokkan peserta KKM:

- a. Himpunan kandidat : himpunan peserta yang merepresentasikan nilai jurusan, fakultas dan jenis kelamin.
- b. Himpunan solusi : menentukan solusi dari ketiga hal yang menjadi acuan.
- c. Fungsi seleksi : memilih dan menyeleksi dalam setiap kelompok, apakah kuota kelompok tersebut masih bisa dimasuki oleh peserta baru.
- d. Fungsi layak : memeriksa apakah nilai total kuota dari himpunan kelompok yang dipilih tidak melebihi jumlah peserta.
- e. Fungsi obyektif : jumlah total peserta dalam kelompok minimum

Skema umum algoritma *greedy*:


```

Procedure greedy (input C : himpunan_kandidat, Output
S: himpunan solusi)
{Mengembalikan solusi optimum dari persoalan optimasi
dengan algoritma greedy
Masukan : himpunan kandidat C
Keluaran : himpunan kandidat solusi S}
Deklarasi x : kandidat
Algoritma :
  S ← { }
  while (belum Solusi (S) ) and (C ≠ { }) do
    x ← SELEKSI (C);
    C ← C - {x}
    if LAYAK (S U {X} ) then
      S ← S U {x}
    endif
  endwhile
  {SOLUSI (S) sudah diperoleh or C = { }}

```

Gambar 3.4 Skema Algoritma Greedy [2]

2. Perhitungan Manual Proses Pengelompokkan (Algoritma Greedy)

Terdapat proses pengacakan (*randomize*) yaitu untuk mengacak nomor kelompok dari batas minimal hingga maksimal kelompok yang disediakan, lalu selanjutnya penerapan *greedy* dengan fungsi selektif yang dimana percabangan bersarang dan pemeriksaan kuota, memilih layak atau tidak layak masuk ke dalam kelompok tersebut dengan beberapa pertimbangan yaitu jenis kelamin, jurusan dan fakultas.

Fungsi *query select* dengan menampilkan data kode kelompok yang kuota masih tersedia (jenis kelamin, jurusan dan fakultas). Didapat kelompok secara ascending, maka kelompok tersebut masuk ke dalam inputan kelompok peserta.

Penerapan Greedy pada proses pengelompokkan peserta KKM :

Diketahui : Tersedia 210 Kelompok

Pendaftar : Fakultas : Sains dan Teknologi

Jurusan : Teknik Informatika

Jenis Kelamin : Laki-laki

Strategi *greedy* : Pada setiap langkah, pilihlah kelompok dengan nilai terbesar dari himpunan yang tersisa.

(Pengelompokkan secara bertahap)

Langkah 1 : Sistem menampilkan data kelompok berdasarkan fakultas dengan syarat menampilkan terlebih dahulu yang bukan fakultas pendaftar dengan secara terurut menurun dari yang masing kosong.

Fakultas : Sains dan Teknologi

Terdapat Kelompok : 6, 28, 77, 109, 164

Langkah 2 : Setelah terurut berdasarkan fakultas, mulai mengerucut dan sistem menampilkan data kelompok berdasarkan jurusan dengan syarat menampilkan terlebih dahulu yang bukan jurusan pendaftar dengan secara terurut menurun dari yang masing kosong.

Jurusan : Teknik Informatika

Terdapat Kelompok : 28, 77, 6, 109, 164

Langkah 3 : Setelah terurut berdasarkan jurusan dan fakultas, mulai lebih mengerucut dan sistem menampilkan data kelompok berdasarkan jenis kelamin dengan syarat menampilkan terlebih dahulu yang bukan jenis kelamin pendaftar dengan secara terurut menurun dari yang masing kosong.

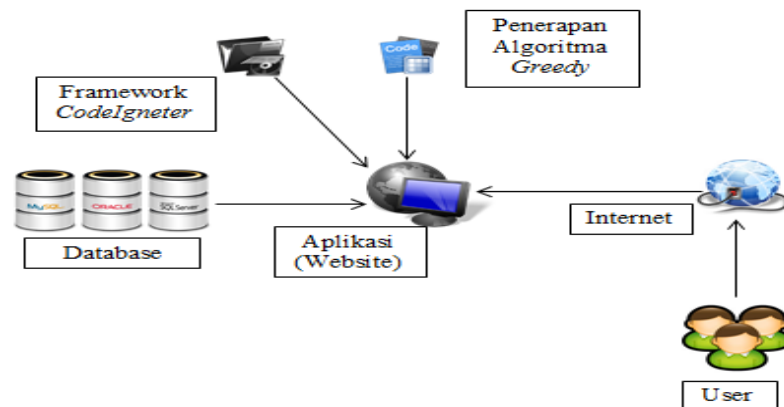
Jenis Kelamin : Laki-laki

Terdapat Kelompok : 28, 6, 77, 109, 164

Solusi : Sistem menampilkan kelompok 28 untuk pendaftar tersebut (Solusi Optimal)

3.4.2 Arsitektur Sistem

Arsitektur system aplikasi yang akan dibuat terdapat beberapa komponen yang digunakan atau terhubung satu sama lain sehingga menjadikan aplikasi yang dapat menerapkan penerapan pengelompokkan Algoritma *Greedy* berbasis *website* dengan menggunakan *Framework Codeigneter*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.5 arsitektur *system*.



Gambar 3.5 Arsitektur Sistem

Pada Gambar 3.5 merupakan arsitektur *system* terhadap aplikasi yang akan dibangun, terdapat beberapa 5 komponen. Berikut penjelasan masing-masing komponen sebagai berikut :

1. *User*

User disini diartikan sebagai yang berhubungan dengan aplikasi tersebut, terdapat 3 pengguna yaitu : *Admin* (Panitia Pelaksana-KKM), Mahasiswa dan Dosen Pembimbing Lapangan.

2. *Internet*

Suatu jaringan komputer yang satu dengan yang lain saling terhubung untuk keperluan komunikasi, yang akan mengakses aplikasi tersebut, dikarenakan harus tersambung dengan koneksi *internet*.

3. Penerapan Algoritma *Greedy*

Komponen yang berfungsi untuk menentukan pengelompokan peserta KKM yang dibutuhkan oleh aplikasi

4. *Framework CodeIgneter*

Sebuah framework PHP yang dipakai dalam pembuatan aplikasi KKM dan dikarenakan dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis PHP dibandingkan jika menulis semua kode program dari awal.

5. *Database*

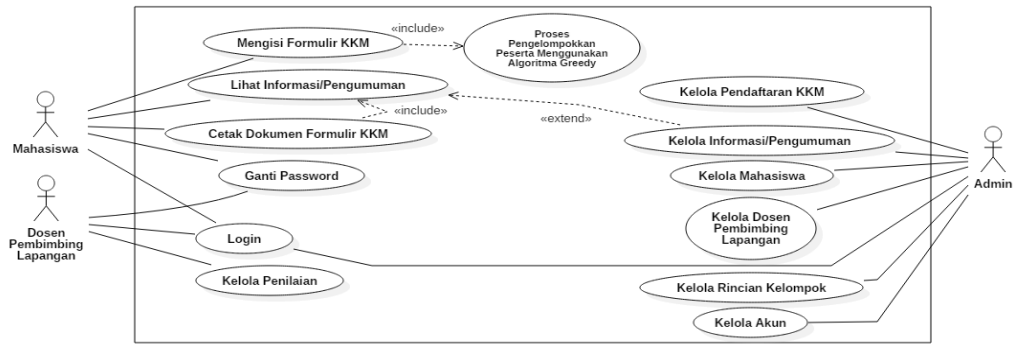
Database yaitu kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*), aplikasi tersebut menggunakan MYSQL.

3.5 Perancangan Aplikasi (*Modeling Quick Design Phases*)

Perancangan aplikasi ini menggunakan pendekatan berorientasi objek dengan menggunakan metode *Unified Modeling Language* (UML), dan berikut ini gambarannya.

3.5.1 *Use case Diagram*

Use case diagram digunakan untuk memodelkan dan menyatakan unit fungsi atau layanan yang disediakan oleh sistem ke pemakai. *Use case diagram* juga mendeskripsikan kelakuan sistem dari sudut pandang pengguna, berguna untuk membantu memahami kebutuhan.



Gambar 3.6 Use Case Diagram

Aplikasi Berdasarkan Gambar 3.4 tentang *use case* yang dirancang untuk aplikasi ini, maka akan ada beberapa *actor* dan *use case* yang dapat dijelaskan.

Tabel 3.1 Tabel Aktor

No	Aktor	Keterangan
1	Admin (PP-KKM)	Mempunyai hak akses penuh terhadap sistem, mengolah data yang berhubungan dengan KKM.
2	Dosen Pembimbing Lapangan	Melakukan pengelolaan penilaian terhadap mahasiswa yang dibimbingnya.
3	Mahasiswa	Melakukan proses pendaftaran yang bertujuan untuk menjadi peserta Kuliah Kerja Nyata Mahasiswa.

3.5.2 Skenario Use Case Diagram

Skenario *use case* digunakan untuk menjelaskan dari gambaran *use case* yang ada dengan menggambarkan identifikasinya dan *scenario per use case*.

a. Use Case Scenario Login

Tabel 3.2 Tabel Use Case Login

IDENTIFIKASI	
No	UCS01
Nama	Login
Tujuan	Untuk dapat masuk dan atau mengakses aplikasi

Tabel 3.2 Tabel Use Case Login (lanjutan)

Deskripsi	Proses pada <i>use case</i> ini sistem akan mendeteksi dan verifikasi akun <i>user</i> yang telah terdaftar dalam <i>database</i> atau tidak.
IDENTIFIKASI	
Aktor	<i>Admin</i> , Mahasiswa dan DPL
SKENARIO UTAMA	
Kondisi Awal	Tampilan <i>Login</i> Aplikasi
Aksi Aktor	Reaksi Sistem
1. Memasukkan <i>username</i> dan <i>password</i>	2. Memverifikasi <i>username</i> dan <i>password</i> pada <i>database</i> . 3. Jika <i>password</i> salah sistem memunculkan pesan kesalahan tetapi Jika <i>password</i> dan <i>username</i> benar maka akan otomatis masuk ke halaman utama untuk mengakses menu aplikasi.

b. *Use Case Scenario* Mengisi Formulir KKM

Tabel 3.3 Tabel *Use Case* Mengisi Formulir KKM

IDENTIFIKASI	
No	UCS02
Nama	Mengisi Formulir Pendaftaran KKM
Tujuan	Untuk dapat melakukan proses pendaftaran dan mengisi formulir.
Deskripsi	Proses pada <i>use case</i> ini sistem akan menampilkan formulir pendaftaran yang akan diisi oleh mahasiswa.
Aktor	Mahasiswa
SKENARIO UTAMA	
Kondisi Awal	Tampilan Formulir Pendaftaran
Aksi Aktor	Reaksi Sistem
1. Klik tombol menu formulir pendaftaran 3. Mahasiswa melakukan pengisian formulir pendaftaran peserta dan identitas lengkap.	2. Aplikasi menampilkan formulir pendaftaran. 4. Menverifikasi dan menyimpan data yang telah diinputkan oleh <i>actor</i> . 5. Sistem melakukan pengacakan dan penentuan kelompok.

c. *Use Case Scenario* Proses Pengelompokkan Peserta (*Algoritma Greedy*)

Tabel 3.4 Tabel *Use Case* Proses Pengelompokkan Peserta (Algoritma *Greedy*)

IDENTIFIKASI	
No	UCS03
Nama	Proses Pengelompokkan Peserta (Algoritma <i>Greedy</i>)
Tujuan	Untuk dapat melakukan proses nelompokkan dan penentuan peserta secara <i>random</i>
Deskripsi	Proses pada <i>use case</i> ini sistem akan melakukan pengelompokkan <i>random</i> menggunakan Algoritma <i>Greedy</i>
IDENTIFIKASI	
Aktor	Mahasiswa
SKENARIO UTAMA	
Kondisi Awal	Tampilan Formulir Pendaftaran
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik tombol menu formulir pendaftaran 3. Mahasiswa melakukan pengisian formulir pendaftaran peserta dan identitas lengkap. 	<ol style="list-style-type: none"> 2. Aplikasi menampilkan formulir pendaftaran. 4. Menverifikasi dan menyimpan data yang telah diinputkan oleh <i>actor</i>. 5. Sistem melakukan pengacakkan dan penentuan kelompok.

d. *Use Case Scenario* Melihat Informasi/Pengumuman

Tabel 3.5 Tabel *Use Case* Melihat Informasi/Pengumuman

IDENTIFIKASI	
No	UCS05
Nama	Melihat Informasi/pengumuman
Tujuan	Untuk dapat menampilkan informasi atau pengumuman
Deskripsi	Proses pada <i>use case</i> ini sistem akan menampilkan secara detail informasi atau pengumuman
IDENTIFIKASI	
Aktor	Mahasiswa
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Informasi Kelompok
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik tombol menu informasi 	<ol style="list-style-type: none"> 2. Menampilkan informasi atau pengumuman.

e. *Use Case Scenario* Cetak Dokumen Formulir KKM

Tabel 3.6 Tabel *Use Case* Cetak Dokumen Formulir KKM

IDENTIFIKASI	
No	UCS06
Nama	Cetak Dokumen Formulir KKM
Tujuan	Untuk dapat melihat hasil cetakan formulir
Deskripsi	Proses pada <i>use case</i> ini sistem akan menampilkan dokumen formulir peserta yang telah di proses pengelompokkan oleh sistem
IDENTIFIKASI	
Aktor	Mahasiswa
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Cetak Dokumen Formulir
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik tombol menu cetak dokumen formulir 3. Klik tombol cetak 	<ol style="list-style-type: none"> 2. Menampilkan report informasi pengelompokkan 4. Melakukan proses pencetakan, dan export dokumen berupa file <i>berextension</i> (pdf)

f. *Use Case Scenario* Kelola Penilaian

Tabel 3.7 Tabel *Use Case* Kelola Penilaian

IDENTIFIKASI	
No	UCS07
Nama	Mengelola Penilaian
Tujuan	Untuk dapat melakukan proses pengelolaan penilaian terhadap peserta
Deskripsi	Proses pada <i>use case</i> ini sistem akan mengolah penilaian
IDENTIFIKASI	
Aktor	Dosen Pembimbing Lapangan
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Kelola Penilaian
Aksi Aktor	Reaksi Sistem
<ol style="list-style-type: none"> 1. Klik tombol menu kelola penilaian 2. Memberikan penilaian terhadap peserta dan menyimpannya 	<ol style="list-style-type: none"> 3. Menampilkan data peserta per kelompok 4. Sistem merespon dengan menyimpan ke <i>database</i>.

g. *Use Case Scenario* Ganti Password

Tabel 3.8 Tabel *Use Case* Ganti *Password*

IDENTIFIKASI	
No	UCS08
Nama	Ganti <i>Password</i>
Tujuan	Untuk dapat melakukan pergantian password akun
Deskripsi	Proses pada <i>use case</i> ini sistem akan mengganti <i>password</i> akun aktor
IDENTIFIKASI	
Aktor	Mahasiswa dan Dosen Pembimbing Lapangan
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Ganti <i>Password</i>
Aksi Aktor	Reaksi Sistem
1. Klik tombol menu ganti <i>password</i> 2. Memasukkan <i>password</i> lama dan <i>password</i> baru	3. Menampilkan kolom penggantian <i>password</i> 4. Sistem merespon dengan menyimpan ke <i>database</i> .

h. *Use Case Scenario* Kelola Pendaftaran KKM

Tabel 3.9 Tabel *Use Case* Kelola Pendaftaran KKM

IDENTIFIKASI	
No	UCS09
Nama	Kelola Pendaftaran KKM
Tujuan	Untuk dapat mengelola dokumen pendaftaran dari peserta
Deskripsi	Proses pada <i>use case</i> ini sistem akan mengolah data dari peserta yang melakukan pendaftaran
IDENTIFIKASI	
Aktor	Admin
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Kelola Pendaftaran
Aksi Aktor	Reaksi Sistem

Tabel 3.9 Tabel *Use Case* Kelola Pendaftaran KKM (lanjutan)

1. Klik tombol menu pendaftaran	2. Menampilkan data pendaftar secara lengkap 3. Sistem menyediakan menu untuk <i>create, read, update</i> dan <i>delete</i> data.
---------------------------------	--

i. *Use Case Scenario* Kelola Informasi/Pengumuman

Tabel 3.10 Tabel *Use Case* Kelola Informasi/Pengumuman

IDENTIFIKASI	
No	UCS10
Nama	Kelola Informasi Kelompok
Tujuan	Untuk dapat mengelola data informasi/pengumuman
Deskripsi	Proses pada <i>use case</i> ini sistem akan mengolah data informasi
IDENTIFIKASI	
Aktor	<i>Admin</i>
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Kelola Informasi Kelompok
Aksi Aktor	Reaksi Sistem
1. Klik tombol menu informasi	2. Menampilkan data pengumuman 3. Sistem menyediakan menu untuk <i>create, read, update</i> dan <i>delete</i> data.

j. *Use Case Scenario* Kelola Mahasiswa

Tabel 3.11 Tabel *Use Case* Kelola Mahasiswa

IDENTIFIKASI	
No	UCS11
Nama	Mengelola Data Mahasiswa
Tujuan	Untuk dapat mengelola data mahasiswa yang terhubung kepada aplikasi
Deskripsi	Proses pada <i>use case</i> ini sistem akan mengolah data mahasiswa
IDENTIFIKASI	
Aktor	<i>Admin</i>
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Kelola Data Mahasiswa
Aksi Aktor	Reaksi Sistem

Tabel 3.11 Tabel *Use Case* Kelola Mahasiswa (lanjutan)

1. Klik tombol menu kelola mahasiswa	2. Menampilkan data mahasiswa
4. <i>admin</i> mengelola data mahasiswa yang berhubungan dengan aplikasi.	3. Sistem menyediakan menu untuk <i>create, read, update</i> dan <i>delete</i> data.
	5. Sistem menyimpan segala perubahan ke dalam <i>database</i>

k. *Use Case Scenario* Kelola Dosen Pembimbing Lapangan

Tabel 3.12 Tabel *Use Case* Kelola Dosen Pembimbing Lapangan

IDENTIFIKASI	
No	UCS12
Nama	Mengelola Dosen Pembimbing Lapangan
Tujuan	Untuk dapat mengelola data Dosen Pembimbing Lapangan yang terhubung kepada aplikasi
Deskripsi	Proses pada <i>use case</i> ini sistem akan mengolah data Dosen Pembimbing Lapangan
IDENTIFIKASI	
Aktor	<i>Admin</i>
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Kelola Dosen Pembimbing Lapangan
Aksi Aktor	Reaksi Sistem
1. Klik tombol menu kelola Dosen Pembimbing Lapangan	2. Menampilkan data Dosen Pembimbing Lapangan
4. <i>Admin</i> mengelola data Dosen Pembimbing Lapangan yang berhubungan dengan aplikasi.	3. Sistem menyediakan menu untuk <i>create, read, update</i> dan <i>delete</i> data.
	5. Sistem menyimpan segala perubahan ke dalam <i>database</i>

1. *Use Case Scenario* Kelola Rincian Kelompok

Tabel 3.13 Tabel *Use Case* Kelola Rincian Kelompok

IDENTIFIKASI	
No	UCS13
Nama	Mengelola Rincian Kelompok
Tujuan	Untuk dapat mengelola data Rincian Kelompok yang terhubung kepada aplikasi
Deskripsi	Proses pada <i>use case</i> ini sistem akan mengolah data Rincian Kelompok
IDENTIFIKASI	

Tabel 3.13 Tabel *Use Case* Kelola Rincian Kelompok (lanjutan)

Aktor	<i>Admin</i>
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Kelola Rincian Kelompok
Aksi Aktor	Reaksi Sistem
1. Klik tombol menu kelola Rincian Kelompok 4. <i>Admin</i> mengelola data Rincian Kelompok yang berhubungan dengan aplikasi.	2. Menampilkan data Rincian Kelompok 3. Sistem menyediakan menu untuk <i>create, read, update</i> dan <i>delete</i> data. 5. Sistem menyimpan segala perubahan ke dalam <i>database</i>

m. *Use Case Scenario* Kelola Akun

Tabel 3.14 Tabel *Use Case* Kelola Akun

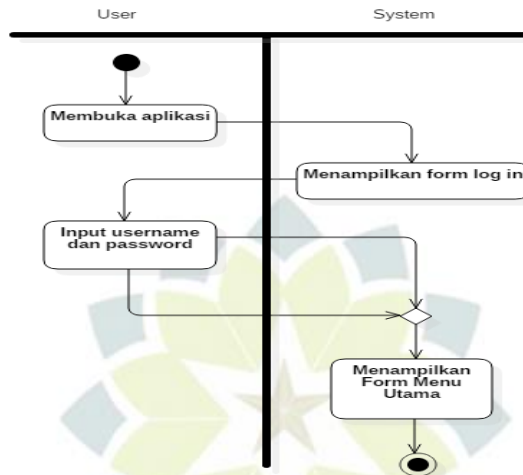
IDENTIFIKASI	
No	UCS14
Nama	Mengelola Akun/ <i>User</i>
Tujuan	Untuk dapat mengelola data <i>user</i> yang terhubung kepada aplikasi
Deskripsi	Proses pada <i>use case</i> ini sistem akan mengolah data <i>user</i>
IDENTIFIKASI	
Aktor	<i>Admin</i>
SKENARIO UTAMA	
Kondisi Awal	Tampilan Menu Kelola <i>User</i>
Aksi Aktor	Reaksi Sistem
1. Klik tombol menu kelola user 3. <i>Admin</i> mengelola data <i>user</i> yang berhubungan dengan aplikasi.	2. Menampilkan data user 4. Sistem menyediakan menu untuk <i>create, read, update</i> dan <i>delete</i> data. 5. Sistem menyimpan segala perubahan

3.5.3 Analisis Activity Diagram

Activity diagram merupakan alur aktifitas yang dilakukan aktor dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan *login*, pendaftaran sampai pengolahan penilaiannya.

a. *Activity diagram* login

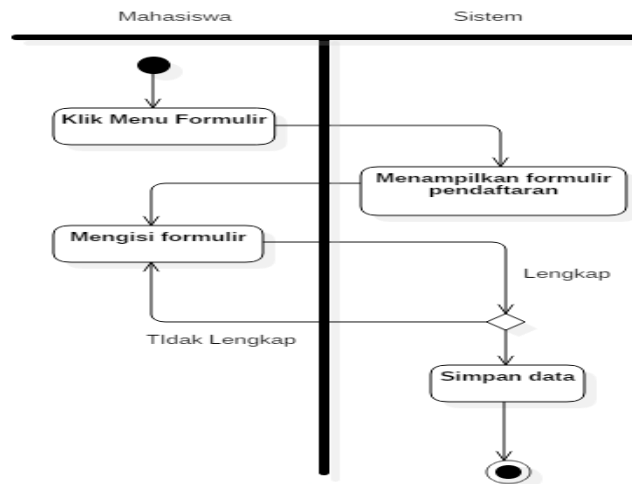
Gambar 3.7 merupakan alur aktifitas yang dilakukan aktor dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan *login*. Interaksi aktor dan *user* digambarkan bahwa user memasukan *username* dan *password*, dan akan masuk ke menu utama jika yang dimasukkan *valid*.



Gambar 3.7 Activity diagram login

b. Activity Diagram Mengisi Formulir KKM

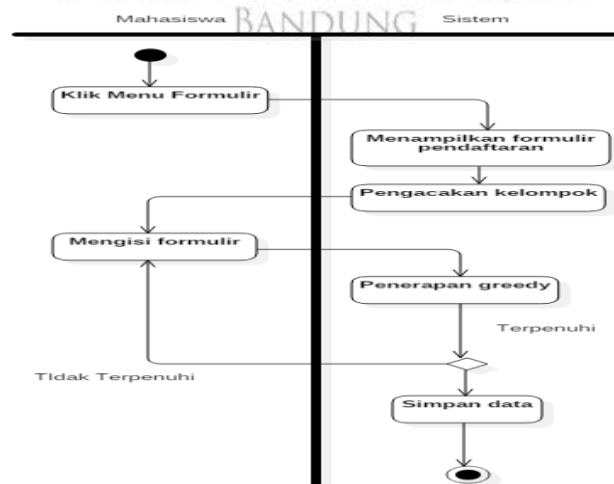
Gambar 3.8 merupakan alur aktifitas yang dilakukan aktor dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan pengisian formulir pendaftaran KKM. Interaksi *actor* (mahasiswa) digambarkan bahwa aktor melakukan pengisian identitas peserta secara lengkap dan mengirim dokumen persyaratan yang telah tertera pada prosedur pendaftaran dan aplikasi akan menyimpan data tersebut kedalam *database*.



Gambar 3.8 Activity diagram mengisi formulir

c. Activity diagram Proses Pengelompokan Peserta (Algoritma Greedy)

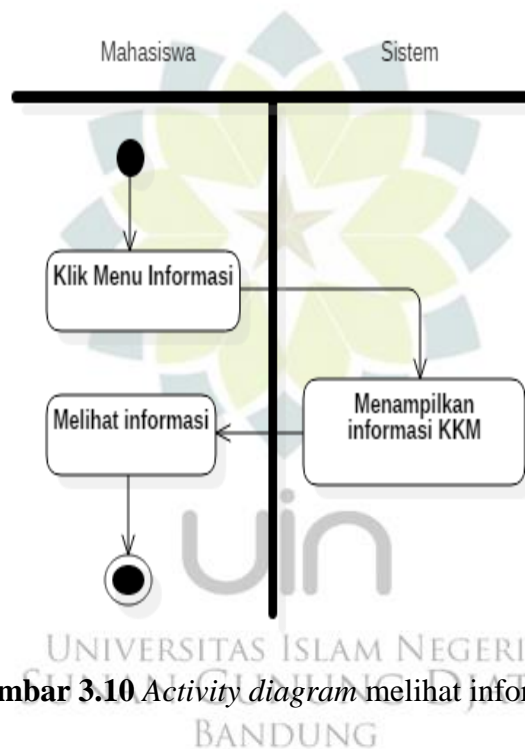
Gambar 3.9 merupakan alur aktifitas yang dilakukan aktor dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan pengisian proses pengelompokan peserta. Interaksi aktor dan sistem digambarkan bahwa mahasiswa melakukan pengisian formulir dan dilakukan pengecekan menggunakan algoritma *greedy*, memeriksa kelompok tersebut maksimum dalam hal kuota, diantaranya jenis kelamin, jurusan, fakultas dan batas maksimum kelompok tersebut.



Gambar 3.9 Activity diagram proses pengelompokan peserta

d. *Activity diagram* Melihat Informasi/Pengumuman

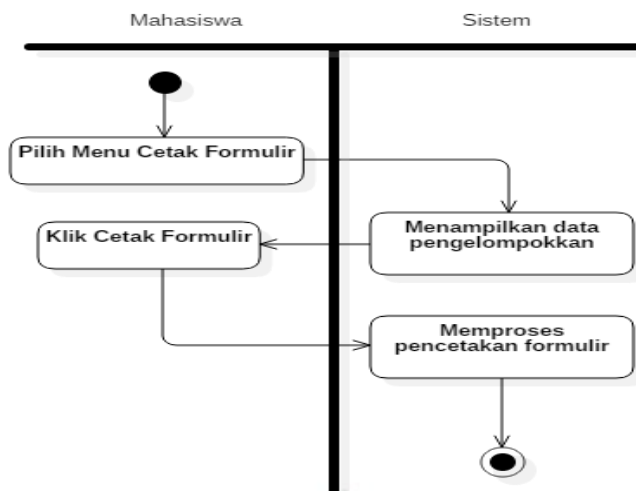
Gambar 3.10 merupakan alur aktifitas yang dilakukan aktor dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melihat informasi/pengumuman perihal yang berkenaan dengan kegiatan KKM. Interaksi *actor* (mahasiswa) dan *sistem* digambarkan bahwa *actor* memilih menu informasi, sistem akan merespon dengan menampilkan semua informasi yang berkenaan dengan kegiatan dan dibutuhkan oleh *actor*.



Gambar 3.10 *Activity diagram* melihat informasi

e. *Activity diagram* Cetak Dokumen Formulir

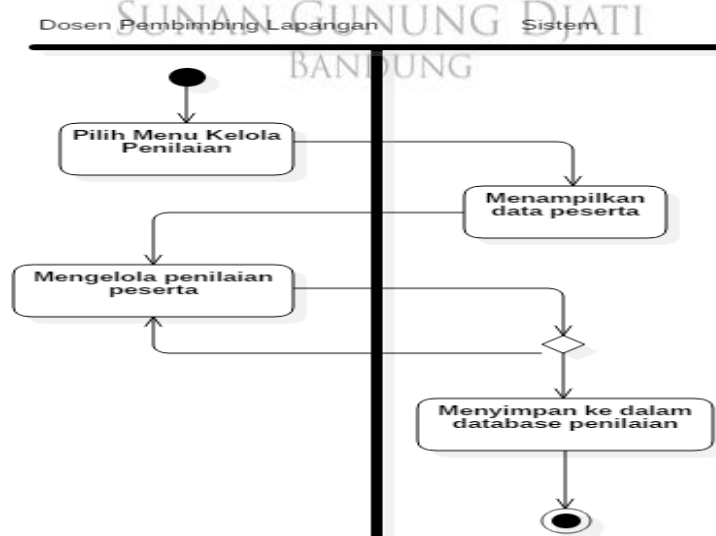
Gambar 3.11 merupakan alur aktifitas yang dilakukan aktor dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan proses pencetakan formulir pendaftaran KKM. Interaksi aktor (mahasiswa) dan sistem digambarkan bahwa sistem memproses perintah dari aktor sebagai peserta KKM mencetak hasil pendaftaran dan identitas kelompok yang telah ditentukan oleh sistem.



Gambar 3.11 Activity diagram cetak dokumen formulir

f. Activity diagram Kelola Penilaian

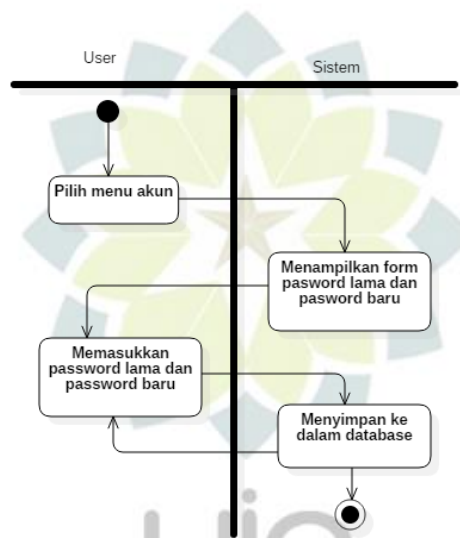
Gambar 3.12 merupakan alur aktifitas yang dilakukan aktor (Dosen Pembimbing Lapangan) dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan dan pengolahan penilaian kepada peserta berdasarkan kelompok yang masing-masing pembimbing. Interaksi aktor dan sistem digambarkan bahwa aktor melakukan pengolahan data penilaian peserta KKM dan selanjutnya aplikasi akan menyimpan data tersebut ke dalam *database*.



Gambar 3.12 Activity diagram kelola penilaian

g. *Activity diagram Ganti Password*

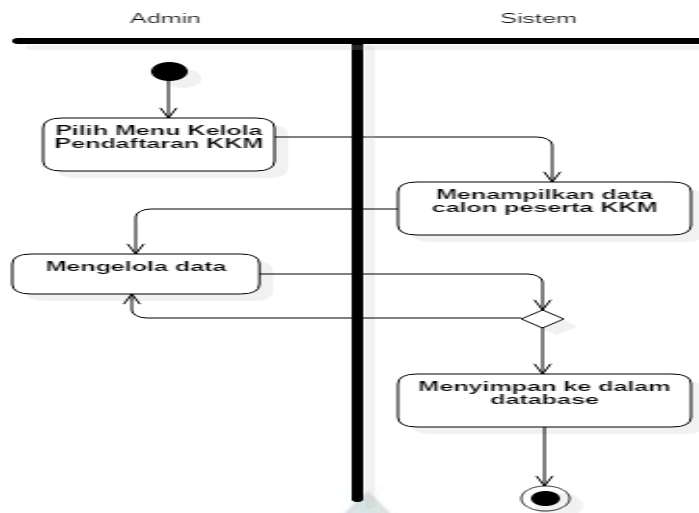
Gambar 3.13 merupakan alur aktifitas yang dilakukan aktor (Mahasiswa dan Dosen Pembimbing Lapangan) dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan pergantian *password*. Interaksi aktor dan sistem digambarkan bahwa aktor memasukkan *password* lama dan *password* baru dan selanjutnya aplikasi akan menyimpan data tersebut ke dalam *database*.



Gambar 3.13 *Activity diagram ganti password*

h. *Activity diagram Kelola Pendaftaran KKM*

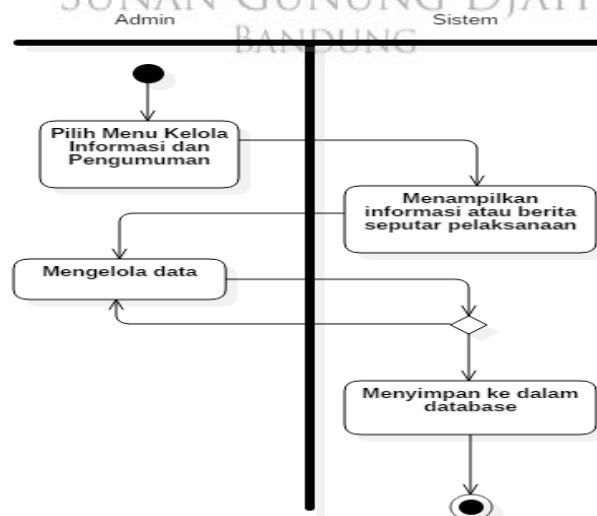
Gambar 3.15 merupakan alur aktifitas yang dilakukan *actor* (*Admin*) dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan pengelolaan pendaftaran KKM oleh *admin* atau bagian Panitia Pelaksana Kuliah Kerja Nyata Mahasiswa (PP-KKM) dari lembaga. Interaksi aktor dan sitem digambarkan bahwa *user* melakukan pengecekan sampai ke pengolahan data dari peserta, dalam hal pendaftaran.



Gambar 3.15 Activity diagram kelola pendaftaran KKM

i. Activity diagram Kelola Informasi/Pengumuman

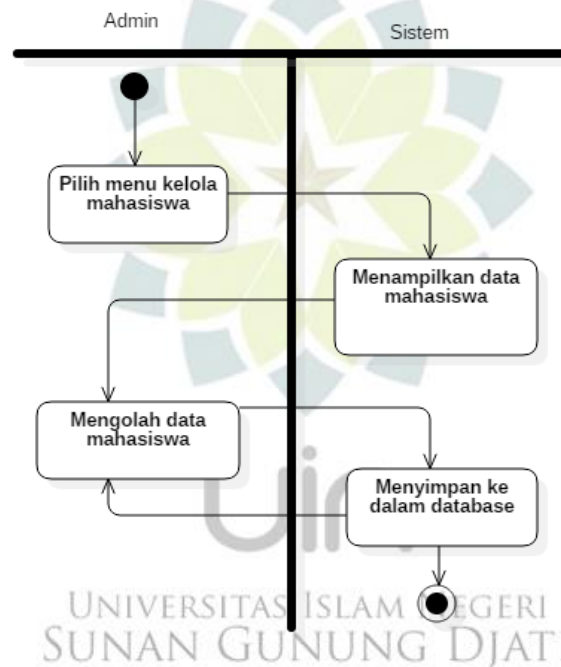
Gambar 3.16 merupakan alur aktifitas yang dilakukan aktor (*Admin/PP-KKM*) dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan mengelola pengumuman atau informasi. Interaksi aktor dan sistem digambarkan bahwa *actor* melakukan mengolah pengumuman dan informasi seputar pelaksanaan KKM selanjutnya aplikasi akan menyimpan data tersebut ke dalam *database*.



Gambar 3.16 Activity diagram kelola informasi/pengumuman

j. *Activity diagram* Kelola Mahasiswa

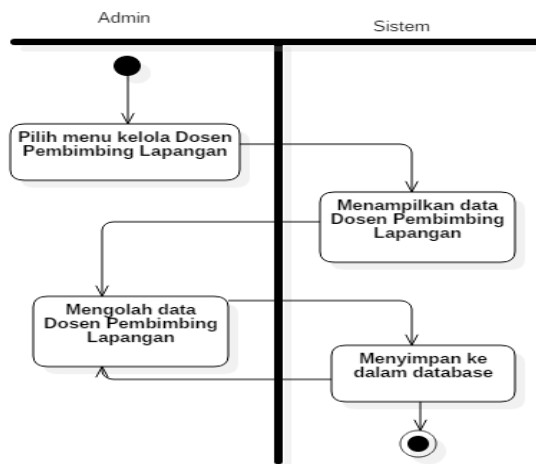
Gambar 3.17 merupakan alur aktifitas yang dilakukan aktor (*Admin/PP-KKM*) dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan pengolahan data mahasiswa. Interaksi *actor* (*Admin/PP-KKM*) dan sistem digambarkan bahwa aktor melakukan pengolahan data mahasiswa. dan selanjutnya aplikasi akan menyimpan segala perubahan data tersebut ke dalam *database*.



Gambar 3.17 *Activity diagram* kelola mahasiswa

k. *Activity diagram* Kelola Dosen Pembimbing Lapangan

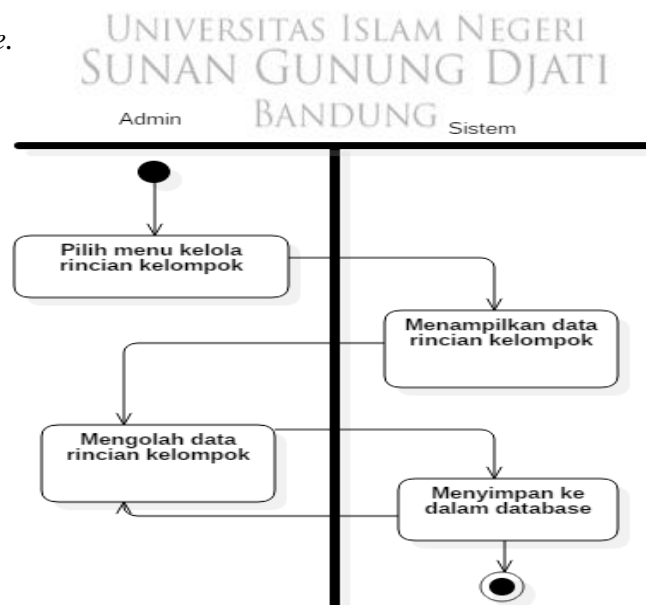
Gambar 3.18 merupakan alur aktifitas yang dilakukan aktor (*Admin/PP-KKM*) dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan pengolahan data Dosen Pembimbing Lapangan. Interaksi aktor dan sistem digambarkan bahwa aktor melakukan pengolahan Dosen Pembimbing Lapangan dan selanjutnya aplikasi akan menyimpan segala perubahan data tersebut ke dalam *database*.



Gambar 3.18 Activity diagram kelola Dosen Pembimbing Lapangan

1. Activity diagram Kelola Rincian Kelompok

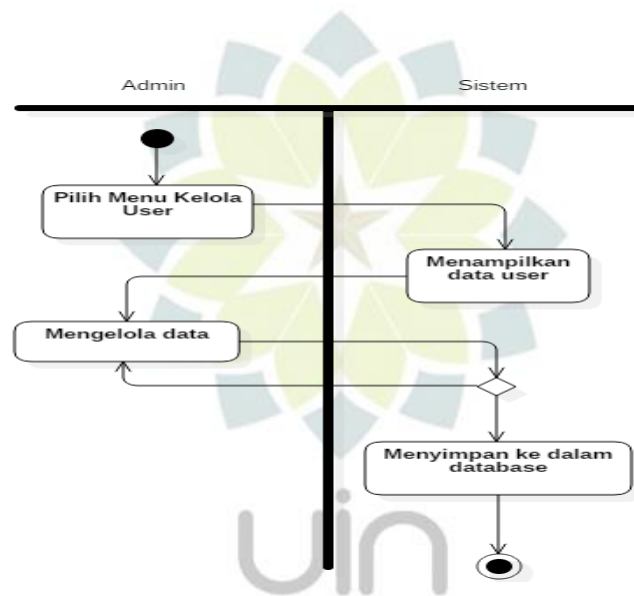
Gambar 3.19 merupakan alur aktifitas yang dilakukan aktor (*Admin/PP-KKM*) dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan pengolahan data rincian kelompok. Interaksi aktor dan sistem digambarkan bahwa aktor melakukan pengolahan data rincian kelompok yaitu mengolah Dosen Pembimbing Lapangan, kuota dalam kelompok tersebut dan selanjutnya aplikasi akan menyimpan segala perubahan data tersebut ke dalam *database*.



Gambar 3.19 Activity diagram rincian kelompok

m. *Activity diagram* Kelola Akun

Gambar 3.20 merupakan alur aktifitas yang dilakukan aktor (*Admin/PP-KKM*) dengan sistem, dimana dalam gambar tersebut dijelaskan bagaimana tahap saat akan melakukan pengolahan data *user* atau akun. Interaksi aktor dan sistem digambarkan bahwa aktor melakukan pengolahan data *user* yaitu Mahasiswa, Dosen Pembimbing Lapangan dan selanjutnya aplikasi akan menyimpan segala perubahan data tersebut ke dalam *database*.

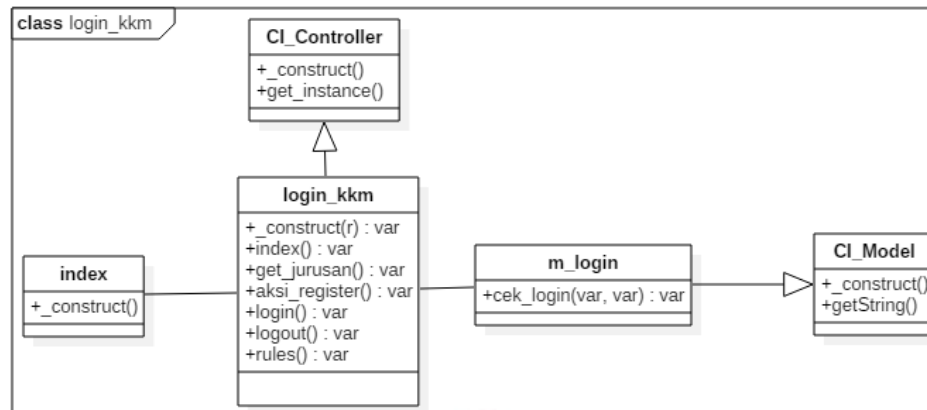


Gambar 3.20 *Activity diagram* kelola user

3.5.4 *Class Diagram*

Aktivitas-aktivitas yang dilakukan pada tahap perancangan kelas merupakan kelanjutan dari tahap analisis kelas. Pada tahap perancangan kelas ini akan dilakukan identifikasi operasi, identifikasi atribut, identifikasi asosiasi dan generalisasi. Tahap perencanaan kelas merupakan tahapan yang memperhalus hasil yang didapatkan pada tahap analisis kelas. Adapun kelas-kelas yang terbentuk dari hasil analisis kelas antara lain seperti ditunjukkan pada gambar di bawah :

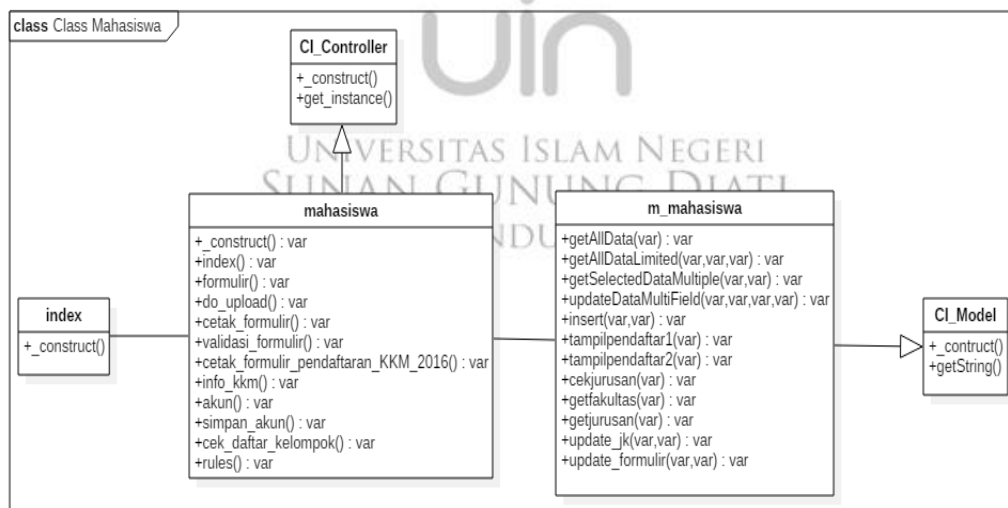
a. Diagram Kelas *Login*



Gambar 3.21 Diagram Kelas *login (login_kkm)*

Keterangan gambar 3.21 : Gambar di atas menjelaskan diagram kelas pada *use case login* dimana diagram di atas direalisasi dari diagram interaksi antar objek *Tbl_login* yang terdiri dari *index*, *login_kkm*, *m_login*, *CI_Controller* dan *CI_Model*.

b. Diagram Kelas Mahasiswa

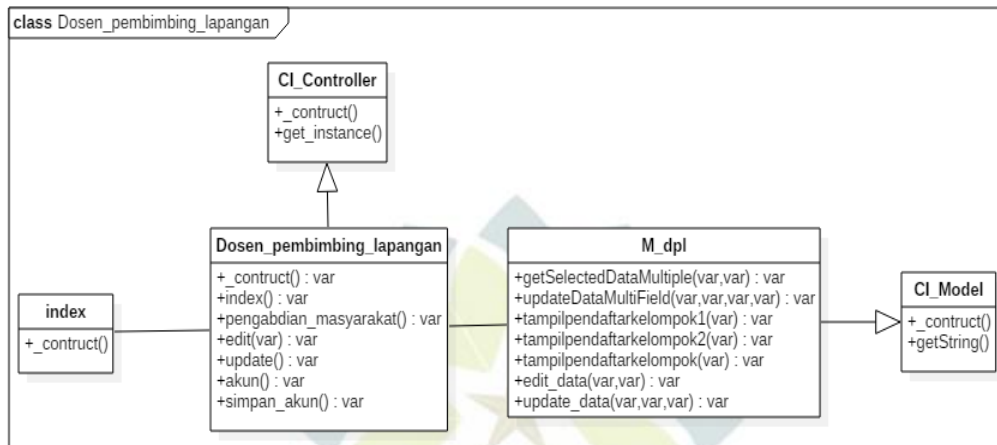


Gambar 3.22 Diagram Kelas Mahasiswa

Keterangan gambar 3.22 : Gambar diatas menjelaskan diagram kelas pada *use case* mengisi formulir KKM, melihat informasi/pengumuman, cetak formulir

pendaftaran KKM, ganti password akun. Dimana diagram di atas direalisasi dari diagram interaksi antar objek Mahasiswa yang terdiri dari index, mahasiswa, m_mahasiswa, CI_Controller dan CI_Model.

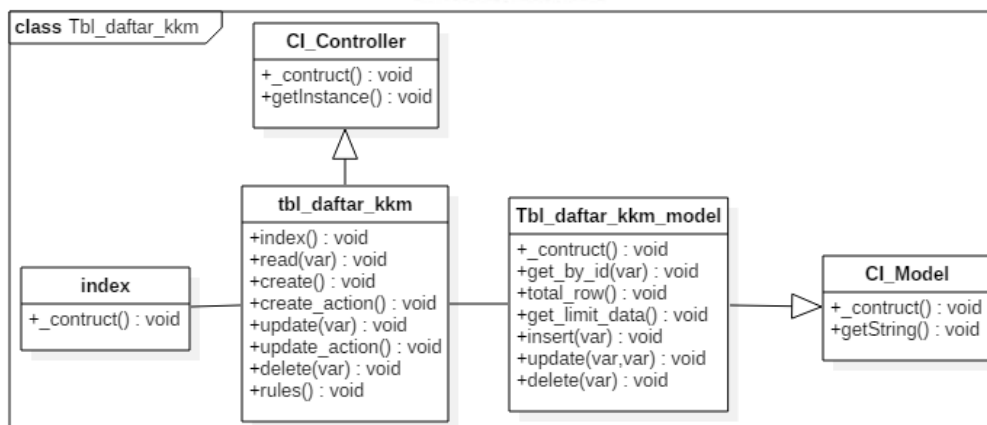
c. Diagram Kelas Dosen Pembimbing Lapangan



Gambar 3.23 Diagram Kelas Dosen Pembimbing Lapangan

Keterangan gambar 3.23 : Gambar di atas menjelaskan diagram kelas pada *use case login* kelola penilaian dang anti password, dimana diagram di atas direalisasi dari diagram interaksi antar objek Dosen Pembimbing Lapangan yang terdiri dari index, Dosen_pembimbing_lapangan, M_dpl, CI_Controller dan CI_Model.

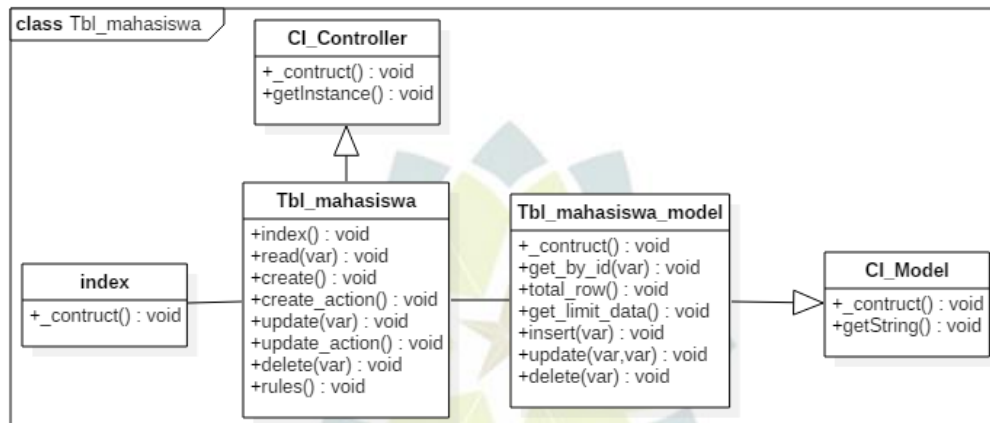
d. Diagram Kelas Kelola Pendaftaran KKM



Gambar 3.24 Diagram Kelas Kelola Pendaftaran KKM

Keterangan gambar 3.24 : Gambar di atas menjelaskan diagram kelas pada *use case* kelola pendaftaran KKM dimana diagram di atas direalisasi dari diagram interaksi antar objek *Tbl_daftar_kkm* yang terdiri dari *index*, *Tbl_dafta_kkm*, *m_tbl_daftar_kkm_model*, *CI_Controller* dan *CI_Model*.

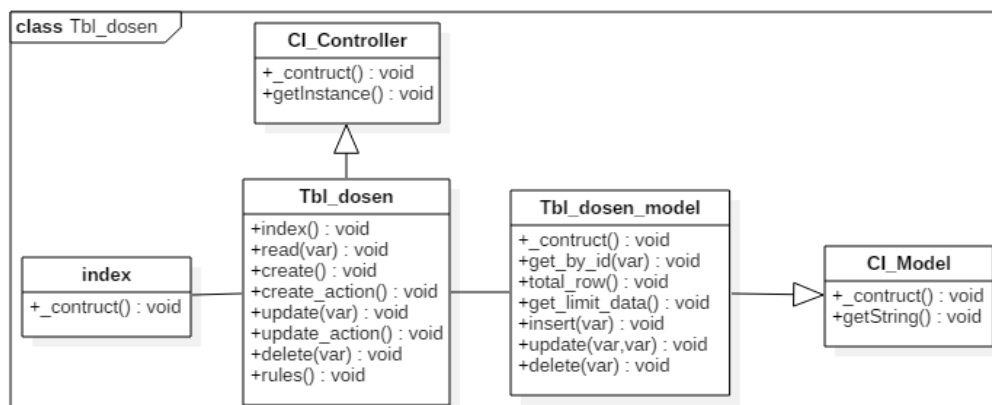
e. Diagram Kelas Kelola Mahasiswa



Gambar 3.25 Diagram Kelas Kelola Mahasiswa

Keterangan gambar 3.25 : Gambar di atas menjelaskan diagram kelas pada *use case* kelola mahasiswa dimana diagram di atas direalisasi dari diagram interaksi antar objek *Tbl_mahasiswa* yang terdiri dari *index*, *Tbl_mahasiswa*, *m_tbl_mahasiswa_model*, *CI_Controller* dan *CI_Model*.

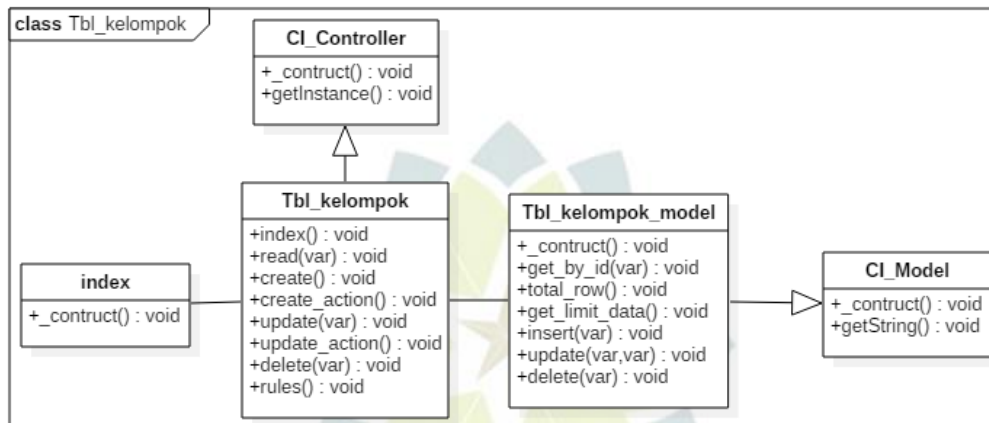
f. Diagram Kelas Kelola Dosen Pembimbing Lapangan



Gambar 3.26 Diagram Kelas Kelola Dosen Pembimbing Lapangan

Keterangan gambar 3.26 : Gambar di atas menjelaskan diagram kelas pada *use case* kelola Dosen Pembimbing Lapangan dimana diagram di atas direalisasi dari diagram interaksi antar objek *Tbl_dosen* yang terdiri dari *index*, *Tbl_dosen*, *tbl_dosen_model*, *CI_Controller* dan *CI_Model*.

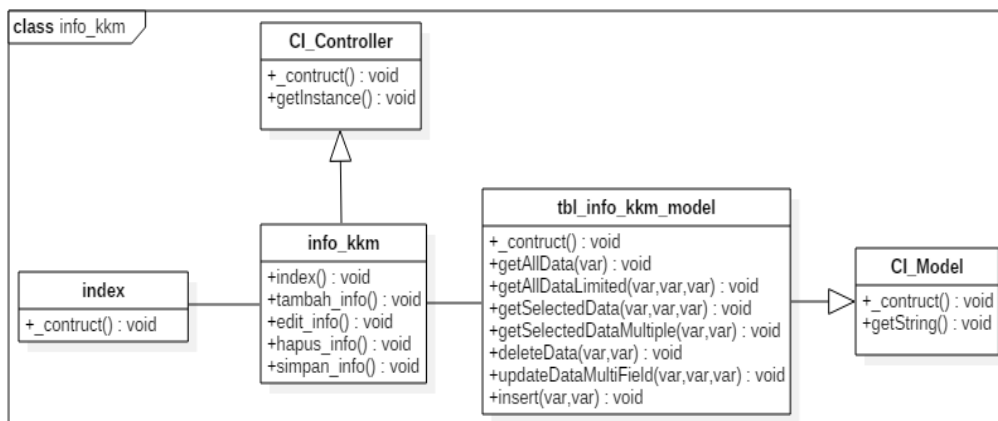
g. Diagram Kelas Kelola Rincian Kelompok



Gambar 3.27 Diagram Kelas Kelola Rincian Kelompok

Keterangan gambar 3.27 : Gambar di atas menjelaskan diagram kelas pada *use case* kelola mahasiswa dimana diagram di atas direalisasi dari diagram interaksi antar objek *Tbl_kelompok* yang terdiri dari *index*, *Tbl_kelompok*, *tbl_kelompok_model*, *CI_Controller* dan *CI_Model*.

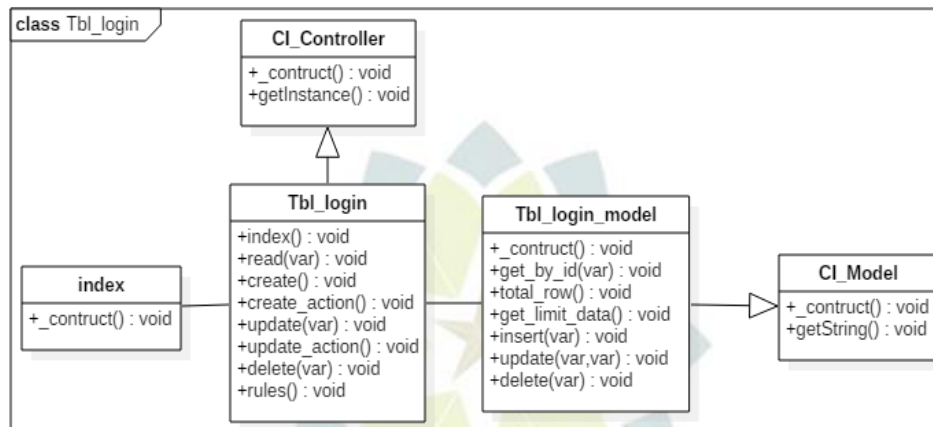
h. Diagram Kelas Kelola Informasi/Pengumuman



Gambar 3.28 Diagram Kelas Kelola Informasi/Pengumuman

Keterangan gambar 3.28 : Gambar di atas menjelaskan diagram kelas pada *use case* kelola Informasi/Pengumuman dimana diagram di atas direalisasi dari diagram interaksi antar objek *Tbl_mahasiswa* yang terdiri dari *index*, *info_kkm*, *tbl_info_kkm_model*, *CI_Controller* dan *CI_Model*.

i. Diagram Kelas Kelola Akun/*User*

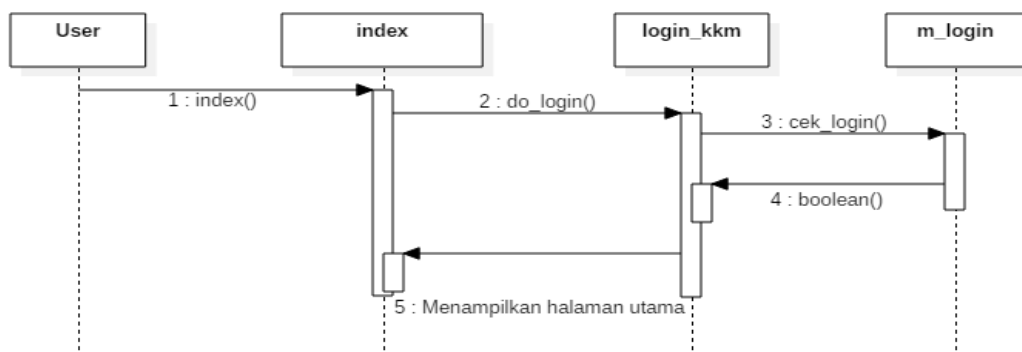


Gambar 3.29 Diagram Kelas Kelola Akun/*User*

Keterangan gambar 3.29 : Gambar di atas menjelaskan diagram kelas pada *use case* kelola Informasi/Pengumuman dimana diagram di atas direalisasi dari diagram interaksi antar objek *Tbl_login* yang terdiri dari *index*, *tbl_login*, *tbl_login_model*, *CI_Controller* dan *CI_Model*.

3.5.5 *Sequence Diagram*

A. *Sequence Diagram Login*



Gambar 3.30 *Sequence diagram login*

Gambar 3.30 merupakan *Sequence Diagram Login* yang menggambarkan komunikasi antar objek pada *use case* proses login. Dimulai dengan *actor/user* (Mahasiswa, Dosen Pembimbing Lapangan dan *Admin*) masuk ke dalam *index* terlebih dahulu, kemudian selanjutnya *user* dapat melakukan login dengan memasukkan *Username* dan *Password* untuk dapat masuk ke dalam sistem. Proses *Login* itu sendiri diikuti juga validasi yang dilakukan oleh sistem dimana terdapat proses verifikasi *Username* dan *Password* apakah sesuai atau tidak dengan menggunakan objek *cek_login*. Jika berhasil *user* tersebut dapat masuk dan menggunakan sistem.

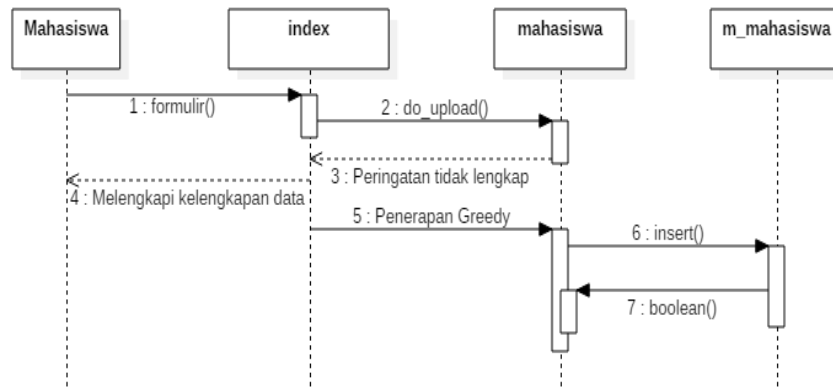
b. *Sequence Diagram* Mengisi Formulir/Pendaftaran



Gambar 3.31 *Sequence diagram* mengisi formulir/pendaftaran

Gambar 3.31 merupakan *Sequence Diagram* mengisi formulir/pendaftaran yang menggambarkan komunikasi antar objek pada *use case* proses pendaftaran KKM. Dimulai dengan *user* (Mahasiswa) masuk ke dalam *form* pendaftaran terlebih dahulu, kemudian selanjutnya *user* dapat melakukan pengisian data dengan memasukkan identitas pribadi secara lengkap untuk dapat melakukan pendaftaran. Proses pendaftaran itu sendiri diikuti juga validasi yang dilakukan oleh sistem dimana terdapat proses verifikasi data apakah sesuai atau tidak. Jika berhasil *user* tersebut dapat melakukan pendaftaran dengan berhasil.

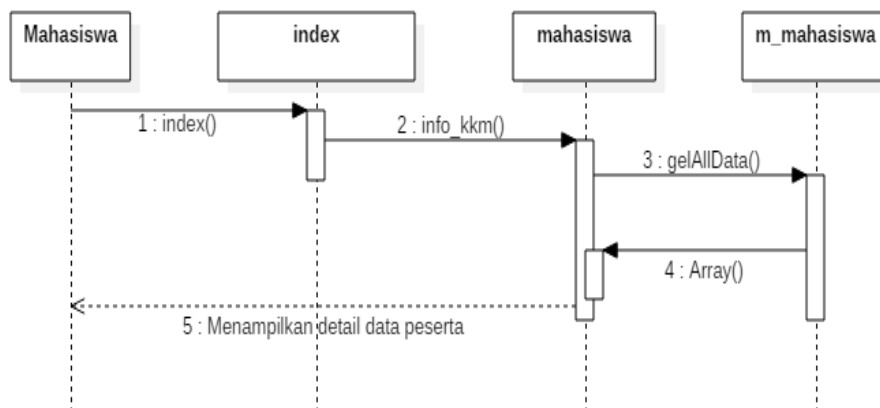
c. *Sequence Diagram* Proses Pengelompokkan Peserta (*Greedy*)



Gambar 3.32 *Sequence diagram* proses pengelompokkan peserta (*Greedy*)

Gambar 3.32 merupakan *Sequence Diagram* mengisi formulir/pendaftaran yang menggambarkan komunikasi antar objek pada *use case* proses pendaftaran KKM. Dimulai dengan *user* (Mahasiswa) masuk ke dalam *form* pendaftaran terlebih dahulu, kemudian selanjutnya *user* dapat melakukan pengisian data dengan memasukkan identitas pribadi secara lengkap untuk dapat melakukan pendaftaran. Proses pendaftaran itu sendiri diikuti juga validasi yang dilakukan oleh sistem dimana terdapat proses verifikasi data apakah sesuai atau tidak. Jika berhasil *user* tersebut dapat melakukan pendaftaran dengan berhasil.

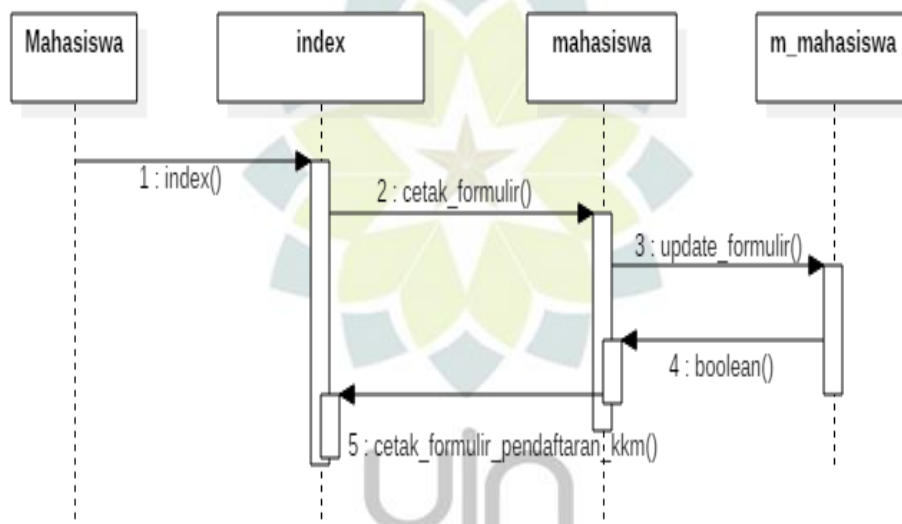
d. *Sequence Diagram* Melihat Informasi/Pengumuman



Gambar 3.33 *Sequence diagram* melihat informasi kelompok

Gambar 3.33 merupakan *Sequence Diagram* melihat informasi/pengumuman yang menggambarkan komunikasi antar objek pada *use case* melihat pengumuman atau informasi. Dimulai dengan *user* (Mahasiswa) masuk ke dalam menu informasi kelompok terlebih dahulu, kemudian selanjutnya sistem menampilkan informasi atau pengumuman, dan *user* dapat melihat secara detail informasi.

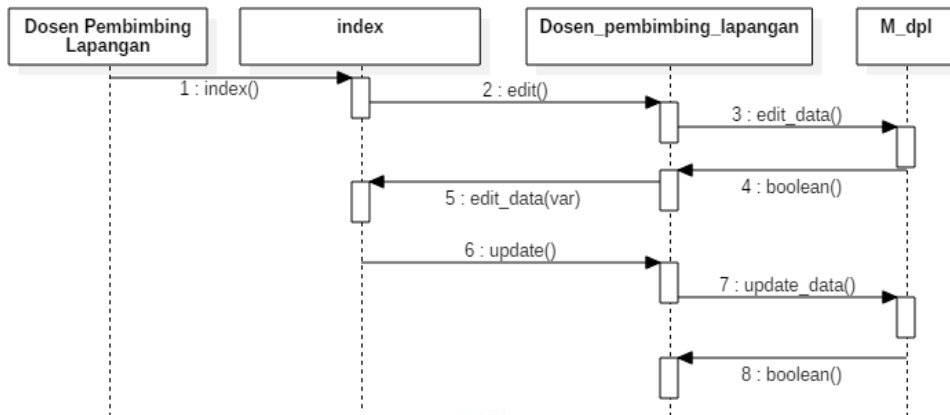
e. *Sequence Diagram* Cetak Dokumen Formulir



Gambar 3.34 *Sequence diagram* cetak dokumen formulir

Gambar 3.34 merupakan *Sequence Diagram* cetak dokumen formulir yang menggambarkan komunikasi antar objek pada *use case* mencetak formulir pendaftaran KKM. Dimulai dengan *user* (Mahasiswa) masuk ke dalam menu index terlebih dahulu, kemudian selanjutnya *user* dapat melakukan pencetakan formulir, dan sistem langsung meresponnya dengan mencetak berdasarkan data peserta yang bersangkutan.

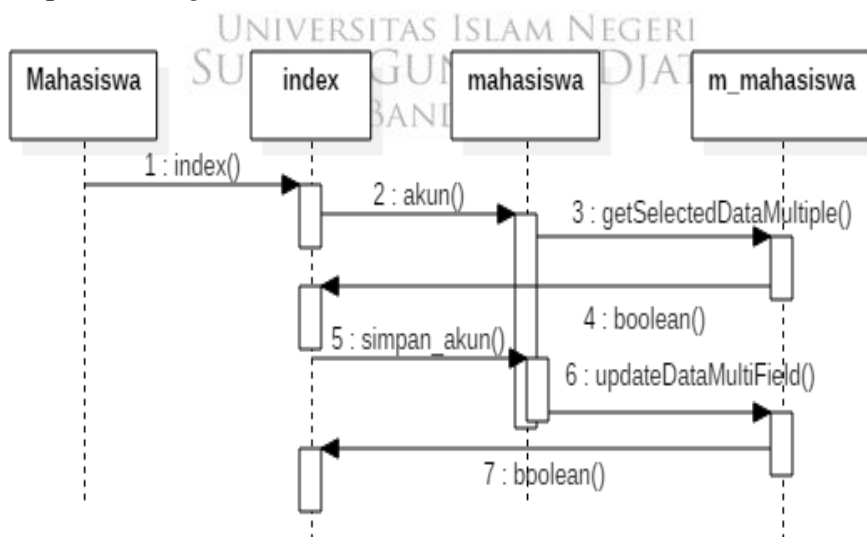
f. *Sequence Diagram* Kelola Penilaian



Gambar 3.35 *Sequence diagram* kelola penilaian

Gambar 3.35 merupakan *Sequence Diagram* kelola penilaian yang menggambarkan komunikasi antar objek pada *use case* proses kelola penilaian peserta KKM. Dimulai dengan *user* (Dosen Pembimbing Lapangan) masuk ke dalam menu kelola penilaian terlebih dahulu, kemudian selanjutnya *user* dapat pencarian data penilaian peserta yang telah tersimpan dalam *database*, dan ubah nilai peserta tersebut berdasarkan kelompok yang dibimbing.

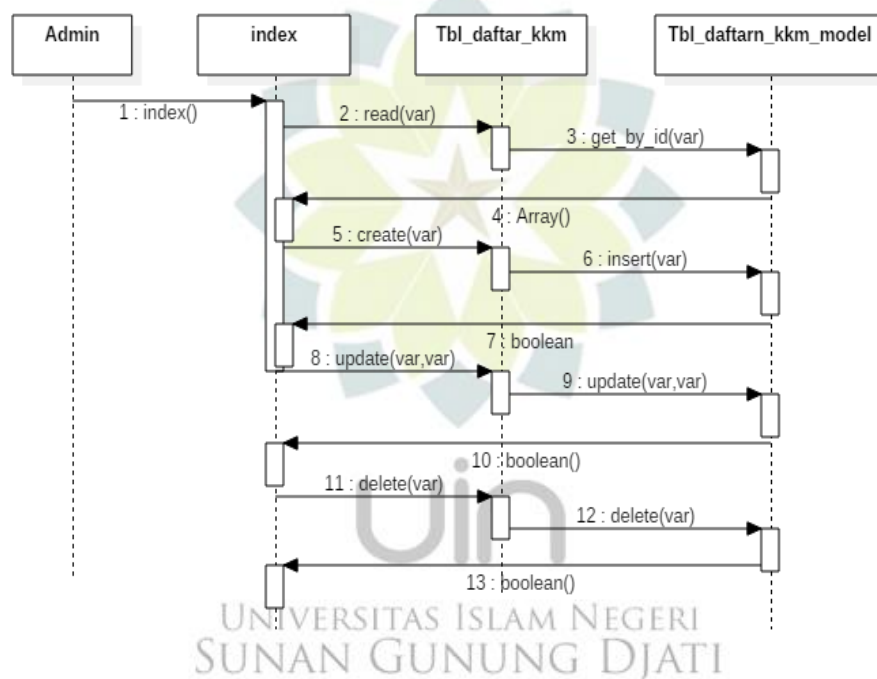
g. *Sequence Diagram* Ganti Password



Gambar 3.36 *Sequence diagram* ganti password

Gambar 3.36 merupakan *Sequence Diagram* ganti *password* yang menggambarkan komunikasi antar objek pada *use case* proses penggantian *password*. Dimulai dengan *user* (Mahasiswa dan Dosen Pembimbing Lapangan) masuk ke dalam menu kelola akun, kemudian selanjutnya *user* dapat memasukkan *password* lama dan *password* baru dan sistem langsung meresponnya dengan menyimpan, mengubah, data berdasarkan data olahan yang ditentukan.

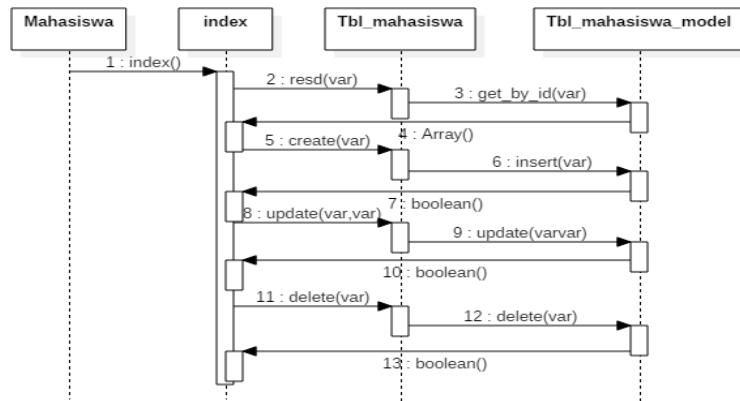
h. Sequence Diagram Kelola Pendaftaran KKM



Gambar 3.37 *Sequence diagram* kelola pendaftaran KKM

Gambar 3.37 merupakan *Sequence Diagram* kelola pendaftaran KKM yang menggambarkan komunikasi antar objek pada *use case* Kelola Pendaftaran KKM. Dimulai dengan *user* (*Admin*) masuk ke dalam menu kelola pendaftaran KKM terlebih dahulu, kemudian selanjutnya *user* dapat melakukan pengolahan data pendaftaran pelaksanaan KKM. Aktor di atas adalah *admin* dan objek yang saling berkomunikasi adalah *index*, *Tbl_daftar_kkm* dan *tbl_daftar_kkm_model*. Tanda panah menjelaskan pesan yang disampaikan antar objek.

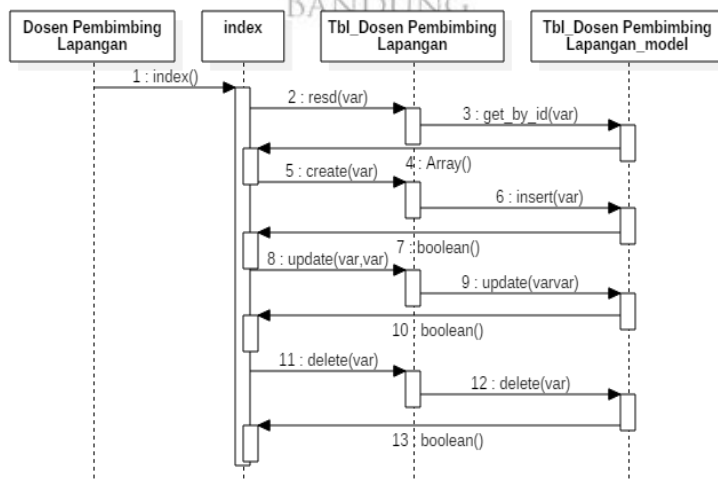
i. *Sequence Diagram* Kelola Mahasiswa



Gambar 3.38 *Sequence diagram* kelola mahasiswa

Gambar 3.38 merupakan *Sequence Diagram* kelola pendaftaran KKM yang menggambarkan komunikasi antar objek pada *use case* Kelola Pendaftaran KKM. Dimulai dengan *user (Admin)* masuk ke dalam menu kelola mahasiswa terlebih dahulu, kemudian selanjutnya *user* dapat melakukan pengolahan data mahasiswa. Aktor di atas adalah *admin* dan objek yang saling berkomunikasi adalah *index*, *Tbl_mahasiswa* dan *Tbl_mahasiswa_model*. Tanda panah menjelaskan pesan yang disampaikan antar objek.

j. *Sequence Diagram* Kelola Dosen Pembimbing Lapangan



Gambar 3.39 *Sequence diagram* kelola Dosen Pembimbing Lapangan

Gambar 3.39 merupakan *Sequence Diagram* kelola pendaftaran KKM yang menggambarkan komunikasi antar objek pada *use case* Kelola Dosen Pembimbing Lapangan. Dimulai dengan *user (Admin)* masuk ke dalam menu Dosen Pembimbing Lapangan terlebih dahulu, kemudian selanjutnya *user* dapat melakukan pengolahan data Dosen Pembimbing Lapangan. Aktor di atas adalah *admin* dan objek yang saling berkomunikasi adalah *index*, *Tbl_Dosen Pembimbing Lapangan* dan *tbl_Dosen_Pembimbing Lapangan_model*. Tanda panah menjelaskan pesan yang disampaikan antar objek.

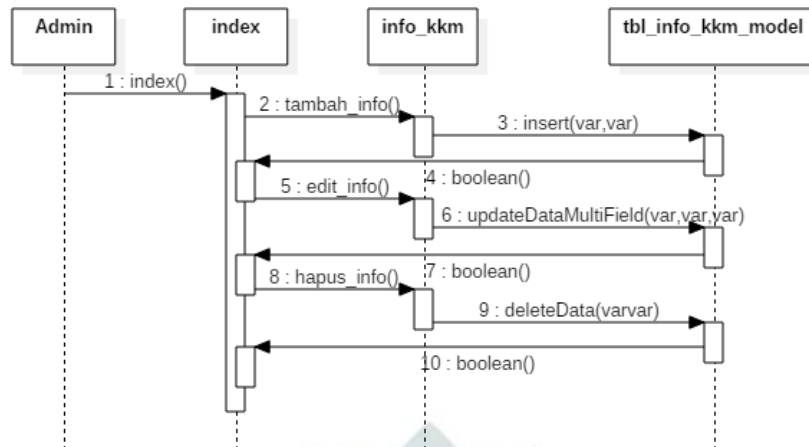
k. *Sequence Diagram* Kelola Rincian Kelompok



Gambar 3.40 *Sequence diagram* kelola rincian kelompok

Gambar 3.40 merupakan *Sequence Diagram* kelola pendaftaran KKM yang menggambarkan komunikasi antar objek pada *use case* rincian kelompok. Dimulai dengan *user (Admin)* masuk ke dalam menu kelola rincian kelompok, kemudian selanjutnya *user* dapat melakukan pengolahan data rincian kelompok. Aktor di atas adalah *admin* dan objek yang saling berkomunikasi adalah *index*, *Tbl_kelompok* dan *Tbl_kelompok_model*. Tanda panah menjelaskan pesan yang disampaikan antar objek.

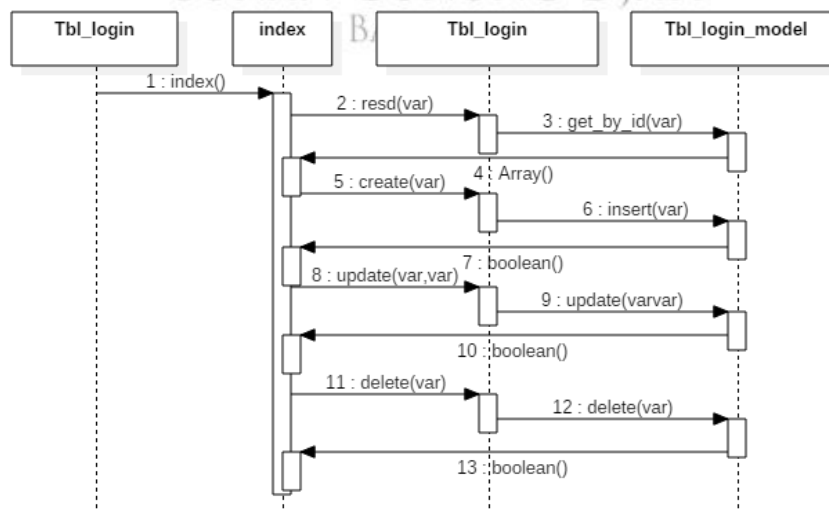
l. *Sequence Diagram* Kelola Informasi/Pengumuman



Gambar 3.41 *Sequence diagram* kelola informasi/pengumuman

Gambar 3.41 merupakan *Sequence Diagram* kelola informasi/pengumuman yang menggambarkan proses penginformasian pelaksanaan KKM. Dimulai dengan *user* (*Admin*) masuk ke dalam menu kelola informasi/pengumuman terlebih dahulu, kemudian selanjutnya *user* dapat melakukan pengolahan informasi yang dibutuhkan oleh mahasiswa dan dosen pembimbing lapangan dalam pelaksanaan KKM.

m. *Sequence Diagram* Kelola Akun/User



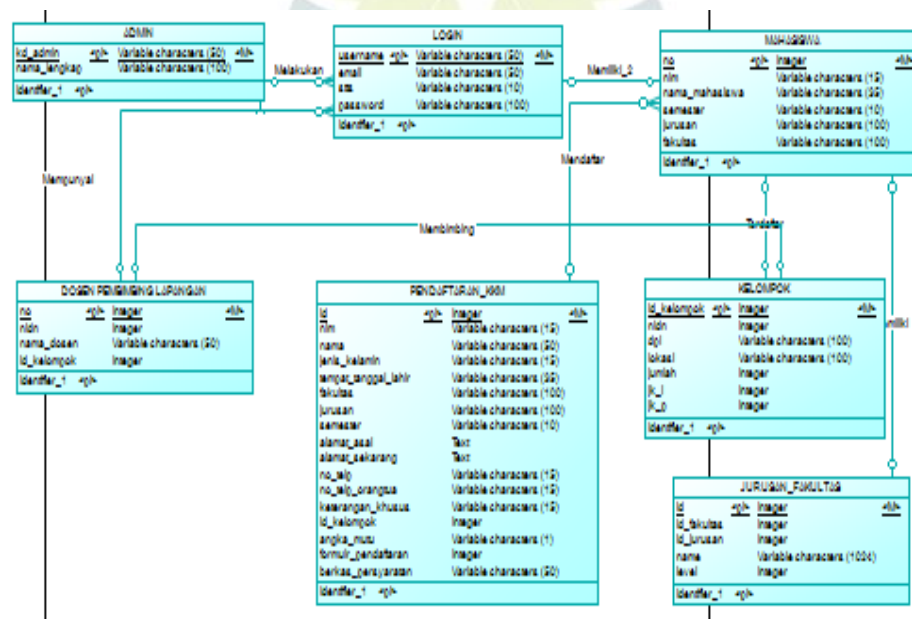
Gambar 3.42 *Sequence diagram* kelola akun/user

Gambar 3.42 merupakan *sequence diagram* kelola *user* yang menggambarkan proses mengelola dan mengolah data *user* pelaksanaan KKM. Dimulai dengan *user Admin/PP-KKM* masuk ke dalam menu kelola *user* terlebih dahulu, kemudian selanjutnya *user* dapat melakukan pengolahan data *user* dan sistem langsung meresponnya dengan menyimpan, mengubah, menghapus data berdasarkan data *user* yang ditentukan.

3.6 Perancangan Database

3.6.1 Conceptual Data Model (CDM)

Conceptual Data Model (CDM) dari aplikasi Kuliah Kerja Nyata Mahasiswa terdapat 7 tabel. Masing-masing tabel mempunyai relasi ke table-table yang lain seperti pada Gambar 3.43.



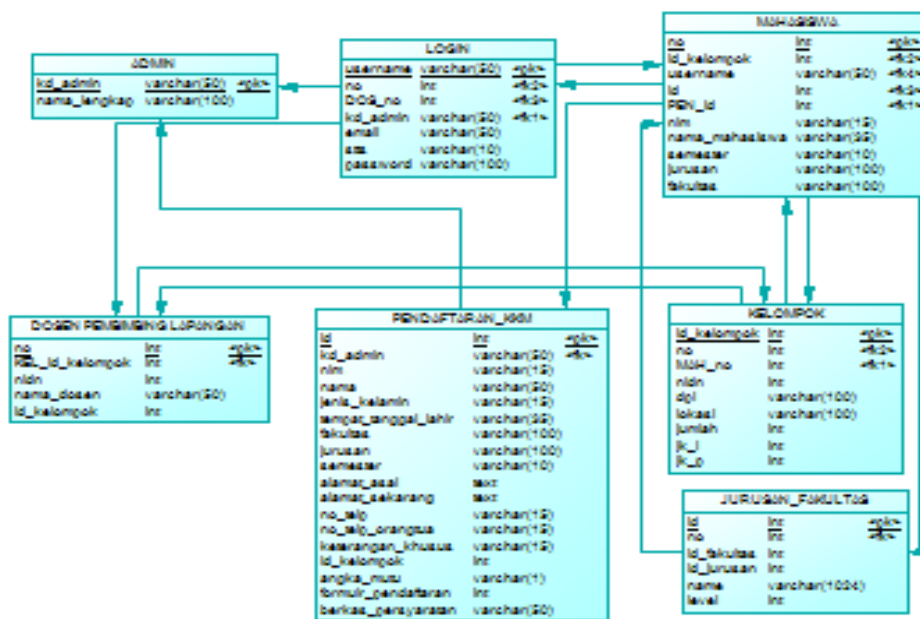
Gambar 3.43 *Conceptual Data Model (CDM)*

Keterangan gambar 3.43 :

Gambar di atas menjelaskan diagram tabel dari hasil analisis tabel yang ada di *database* sistem. Tanda panah menunjukan hubungan *foreign key* antar tabel.

3.6.2 Physical Data Model

Physical Data Model (PDM) adalah hasil dari *generate* dari CDM. Data tabel pada PDM inilah yang akan digunakan pada saat membuat aplikasi. Pada PDM terdapat 7 (tujuh) tabel dengan tipe data dan panjangnya yang masing-masing tabel memiliki satu *primary key* dan banyak *foreign key*. PDM dari aplikasi Kuliah Kerja Nyata Mahasiswa UIN Sunan Gunung Djati Bandung dilihat pada Gambar 3.44.



Gambar 3.44 Physical Data Model (PDM)

3.6.3 Identifikasi Tabel

1. Tabel Login

Tabel *login* merupakan tempat disimpnannya berbagai informasi tentang *login*. Tabel 3.15 ini menjelaskan tentang perancangan tabel *login*.

Tabel 3.15 Tabel tbl_login

Kolom	Tipe Data	Ukuran Field	Keterangan
username	varchar	50	(primary key) Untuk menyimpan data <i>username</i> (user)

Tabel 3.15 Tabel tbl_login (lanjutan)

Kolom	Tipe Data	Ukuran Field	Keterangan
password	varchar	50	untuk menyimpan data <i>password (user)</i>
email	varchar	50	untuk menyimpan data <i>email user</i>
stts	varchar	50	untuk menyimpan data status <i>user</i>

2. Tabel Admin

Tabel *admin* merupakan tempat disimpannya berbagai informasi tentang *admin* Tabel 3.16 menjelaskan tentang perancangan tabel *admin*.

Tabel 3.16 Tabel *tbl_admin*

Kolom	Tipe Data	Ukuran Field	Keterangan
username	varchar	50	(primary key) Untuk menyimpan data <i>username (user)</i>
nama_lengkap	varchar	100	Untuk menyimpan data nama <i>admin</i>

3. Tabel Dosen

Tabel dosen merupakan tempat disimpannya berbagai informasi tentang DPL (Dosen Pembimbing Lapangan). Tabel 3.17 menjelaskan tentang perancangan tabel dosen.

Tabel 3.17 Tabel *tbl_dosen*.

Kolom	Tipe Data	Ukuran Field	Keterangan
no	varchar	15	(primary key) untuk menyimpan data nomor
nidn	int	50	(foreign key) untuk menyimpan data <i>username (user)</i>
nama_dosen	varchar	50	untuk menyimpan data nama lengkap dosen
id_kelompok	int	10	untuk menyimpan data nomor kelompok

4. Tabel Mahasiswa

Tabel mahasiswa merupakan tempat disimpannya berbagai informasi tentang mahasiswa Tabel 3.18 menjelaskan tentang perancangan tabel mahasiswa.

Tabel 3.18 Tabel tbl_mahasiswa

Kolom	Tipe Data	Ukuran Field	Keterangan
no	int	15	(<i>primary key</i>) untuk menyimpan data nomor
nim	int	15	(<i>foreign key</i>) untuk menyimpan data nim mahasiswa
nama_mahasiswa	varchar	50	untuk menyimpan data nama lengkap mahasiswa
semester	varchar	10	untuk menyimpan data semester
jurusan	varchar	100	untuk menyimpan data jurusan mahasiswa
fakultas	varchar	100	untuk menyimpan data fakultas mahasiswa

5. Tabel Pendaftaran KKM

Tabel pendaftaran KKM merupakan tempat disimpannya berbagai informasi tentang pendaftaran KKM. Tabel 3.19 menjelaskan tentang perancangan tabel pendaftaran KKM.

Tabel 3.19 Tabel tbl_daftar_kkm.

Kolom	Tipe Data	Ukuran Field	Keterangan
id	int	15	(<i>primary key</i>) untuk menyimpan data id pendaftar
nim	int	15	(<i>foreign key</i>) untuk menyimpan data nim mahasiswa
nama	varchar	50	untuk menyimpan data nama lengkap mahasiswa

Tabel 3.19 Tabel tbl_daftar_kkm (lanjutan)

Kolom	Tipe Data	Ukuran Field	Keterangan
semester	varchar	10	untuk menyimpan data semester
jurusan	varchar	100	untuk menyimpan data jurusan mahasiswa
fakultas	varchar	100	untuk menyimpan data fakultas mahasiswa
jenis_kelamin	varchar	15	untuk menyimpan data jenis kelamin
alamat_asal	text		untuk menyimpan data alamat asal
alamat_sekarang	text		untuk menyimpan data alamat sekarang
no_telp	varchar	15	untuk menyimpan data nomor telepon
no_telp_orangtua	varchar	15	untuk menyimpan data nomor telepon orang tua
keterangan_khusus	varchar	15	untuk menyimpan data keterangan khusus
id_kelompok	int	15	untuk menyimpan nomor kelompok
angka_mutu	varchar	1	untuk menyimpan data angka mutu penilaian
formulir_pendaftaran	varchar	50	untuk menyimpan data dokumen formulir pendaftaran validasi
berkas_persyaratan	varchar	50	untuk menyimpan data berkas persyaratan

6. Tabel Jurusan Fakultas

Tabel jurusan fakultas merupakan tempat disimpannya berbagai informasi tentang jurusan dan fakultas di UIN Sunan Gunung Djati Bandung. Tabel 3.20 menjelaskan tentang perancangan tabel jurusan fakultas.

Tabel 3.20 Tabel tbl_jurusan_fakultas

Kolom	Tipe Data	Ukuran Field	Keterangan
id	int	10	untuk menyimpan data nomor tabel
id_fakultas	int	10	untuk menyimpan data kode fakultas
id_jurusan	Int	10	untuk menyimpan data kode jurusan
name	Varchar	100	untuk menyimpan data nama jurusan dan fakultas
level	Int	1	untuk menyimpan data lever menentukan jurusan atau fakultas

7. Tabel Kelompok

Tabel kelompok merupakan tempat disimpannya berbagai informasi tentang kelompok. Tabel 3.21 menjelaskan tentang perancangan tabel kelompok.

Tabel 3.21 Tabel tbl_kelompok

Kolom	Tipe Data	Ukuran Field	Keterangan
id_kelompok	int	10	untuk menyimpan data nomor kelompok
nidn	int	30	untuk menyimpan data nomor induk dosen
dpl	varchar	100	untuk menyimpan data nama dosen
lokasi	varchar	100	untuk menyimpan data lokasi kkm
jumlah	int	2	untuk menyimpan data kuota kelompok
jk_l	int	2	untuk menyimpan data kuota kelompok laki-laki
jk_p	int	2	untuk menyimpan data kuota kelompok perempuan

8. Tabel Informasi/Pengumuman

Tabel informasi/pengumuman merupakan tempat disimpannya berbagai

informasi tentang informasi/pengumuman. Tabel 3.22 menjelaskan tentang perancangan tabel informasi/pengumuman.

Tabel 3.22 Tabel tbl_info

Kolom	Tipe Data	Ukuran Field	Keterangan
kd_info	int	10	(<i>primary key</i>) untuk menyimpan data kode informasi
judul	varchar	200	untuk menyimpan data judul
waktu_post	varchar	30	untuk menyimpan data waktu posting
isi	text		untuk menyimpan data isi posting

9. Tabel Periode Pendaftaran

Tabel periode pendaftaran merupakan tempat disimpannya berbagai informasi tentang periode pendaftaran. Tabel 3.23 menjelaskan tentang perancangan tabel periode pendaftaran.

Tabel 3.23 Tabel periode pendaftaran.

Kolom	Tipe Data	Ukuran Field	Keterangan
kd_periode	int	5	untuk menyimpan data kode periode
waktu	varchar	20	untuk menyimpan data waktu periode
keterangan	text		untuk menyimpan data keterangan

3.7 Perancangan Proses Pendaftaran

3.7.1 Aspek Input

Aspek input pada proses seleksi penentuan kelompok adalah :

1. Nomor Induk Mahasiswa.
2. Jenis kelamin.

3. Pilihan jurusan.

4. Pilihan fakultas

3.7.2 Aspek proses

Aspek proses pada seleksi penentuan kelompok adalah sebagai berikut :

1. Iterasi ke satu

- Sistem menyeleksi pendaftar berdasarkan Nomor Induk Mahasiswa, jika ada pada kelompok tersebut maka kelompok akan diacak kembali.

2. Iterasi ke dua

- Sistem menyeleksi berdasarkan pilihan jenis kelamin, jika pada kelompok tersebut sudah penuh yaitu kuota laki-laki dan perempuan. Maka kelompok akan diacak kembali.

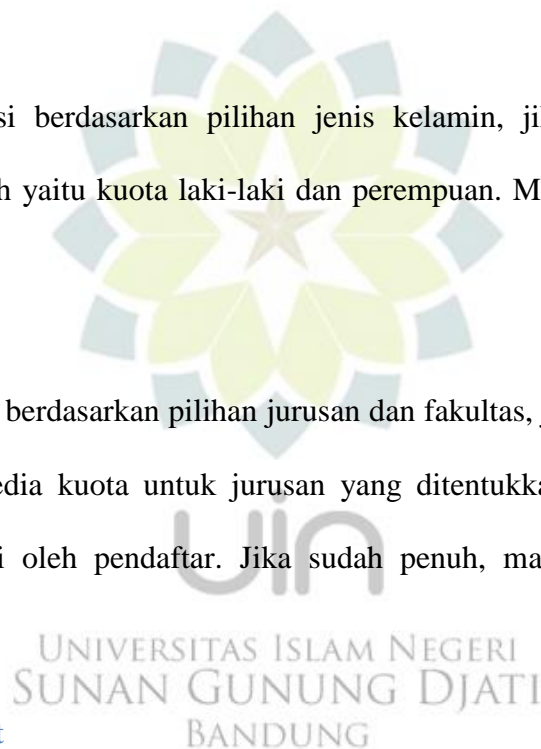
3. Iterasi ke tiga

- Sistem menyeleksi berdasarkan pilihan jurusan dan fakultas, jika pada kelompok tersebut masih tersedia kuota untuk jurusan yang ditentukan pendaftar, maka kelompok bisa diisi oleh pendaftar. Jika sudah penuh, maka kelompok akan diacak kembali.

3.7.3 Aspek output

Aspek output dari proses seleksi di sistem adalah :

- Pendaftar terseleksi berdasarkan NIM, jenis kelamin, jurusan dan fakultasnya.
- Kelompok terisi berdasarkan kuota dan jumlah pendaftar yang memilih ke kelompok tersebut dan tentunya yang telah melewati proses seleksi.
- Sistem menampilkan *report* hasil seleksi.

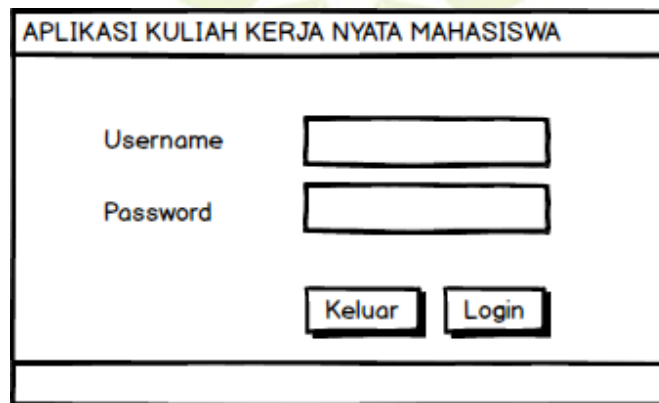


3.8 Perancangan Antarmuka (*Construction of Prototype*)

Pada tahap ini akan dilakukan pembangunan perancangan *interface* terhadap aplikasi yang akan dibuat. Berikut perancangan yang ditunjukkan pada gambar di bawah ini.

a. Rancangan Antarmuka *Form Login*

Antarmuka tampilan login ini merupakan tampilan pertama yang akan muncul saat aplikasi dijalankan. Dalam tampilan awal ini terdapat berbagai fungsi *interface* didalamnya salah satunya sebagai tampilan *login* untuk masuk pada jendela tampilan menu utama. Dimana didalam tampilan login terdapat beberapa isian yaitu *username*, *password* dan tombol *button login* yang dapat diklik sebagai tombol masuk aplikasi ini, untuk lebih jelasnya dapat dilihat pada Gambar 3.45.

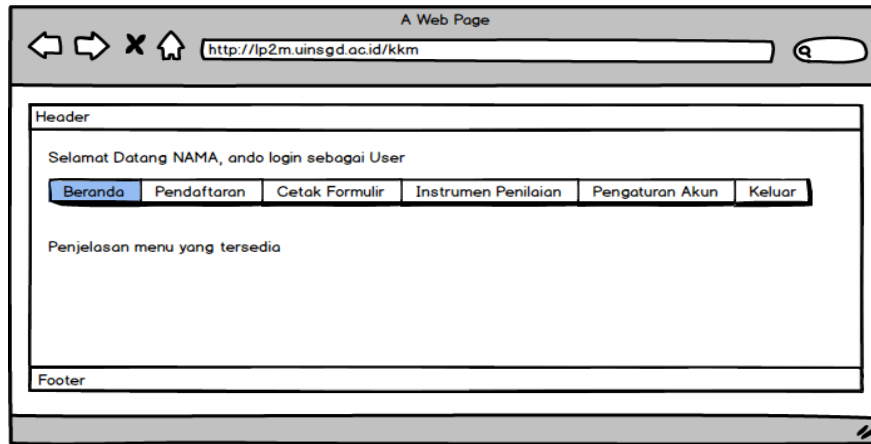


The image shows a mockup of a login form titled "APLIKASI KULIAH KERJA NYATA MAHASISWA". The form contains two input fields: "Username" and "Password". Below the input fields are two buttons: "Keluar" and "Login". The form is enclosed in a rectangular border.

Gambar 3.45 *Mockup* tampilan form *Login*

b. Rancangan Antarmuka Tampilan Awal

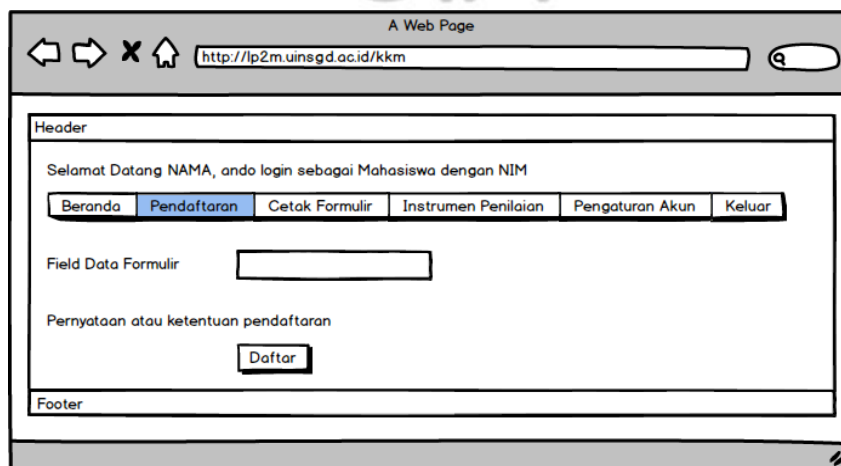
Antarmuka tampilan awal ini merupakan tampilan pertama yang akan muncul saat aplikasi telah berhasil *login*. Dalam tampilan awal ini terdapat berbagai fungsi *interface* didalamnya salah satunya sebagai tampilan menu untuk masuk pada jendela tampilan yang lain, yaitu menu pendaftaran, cetak formulir, instrument penilaian, pengaturan akun dan keluar. Untuk lebih jelasnya dapat dilihat pada Gambar 3.46.



Gambar 3.46 *Mockup* tampilan halaman utama

c. Rancangan Antarmuka Tampilan Pendaftaran

Antarmuka tampilan pendaftaran ini merupakan tampilan yang akan muncul saat akan mengisi formulir pendaftaran KKM. Dalam tampilan ini terdapat berbagai fungsi *interface* didalamnya salah satunya sebagai tampilan beberapa *field* yang harus diisi oleh mahasiswa atau calon peserta KKM, dan apabila sudah lengkap, terdapat *button* daftar yang dima tombol tersebut untuk mendaftarkan formulir. Untuk lebih jelasnya dapat dilihat pada gambar 3.47.



Gambar 3.47 *Mockup* tampilan form formulir pendaftaran

Pada proses pengelompokkan peserta terdapat penerapan algoritma *greedy* untuk menentukan pengelompokkan peserta, berikut *pseudo code* dari algoritma *greedy* pada Gambar 3.48.

```

Procedure greedy (input C: himpunan_kandidat, Output
S: himpunan solusi)
{Mengembalikan solusi optimum dari persoalan optimasi
dengan algoritma greedy
Masukan : himpunan kandidat C
Keluaran : himpunan kandidat solusi S}
Deklarasi x : kandidat
Algoritma :
  S ← { }
  while (belum Solusi (S) ) and (C ≠ { }) do
    x ← SELEKSI (C);
    C ← C - {x}
    if LAYAK (S U {x} ) then
      S ← S U {x}
    endif
  endwhile
  {SOLUSI (S) sudah diperoleh or C = { }}

```

Gambar 3.48 *Pseudo code* penerapan algoritma *greedy*

Pada *pseudo code* berdasarkan gambar merupakan proses pengelompokkan peserta KKM. Terdapat proses pengacakan (*randomize*) yaitu untuk mengacak nomor kelompok dari batas minimal hingga maksimal kelompok yang disediakan, lalu selanjutnya penerapan *greedy* dengan fungsi selektif yang dimana percabangan bersarang dan pemeriksaan kuota, memilih layak atau tidak layak masuk ke dalam kelompok tersebut dengan beberapa pertimbangan yaitu jenis kelamin, jurusan dan fakultas.

```

$jenis_kelamin=$bc['jenis_kelamin'];
$jurusan=$bc['jurusan'];
$fakultas=$bc['fakultas'];
$bc['cek_kelompok'] = $this->m_mahasiswa->get_pengacakan($jurusan,$jenis_kelamin,$fakultas);

```

Keterangan : Pada *controller* memanggil fungsi `get_pengacakan()` dengan parameter `jenis_kelamin`, `jurusan` dan `fakultas`.

```

public function get_pengacakan($jurusan_id,$jenis_kelamin,$fakultas_id)
{
    .....
    $kelompok=270;
    if ($jenis_kelamin="LAKI-LAKI") {
        return $this->db->query("SELECT c.id_kelompok FROM tbl_mahasiswa a, tbl_jurusan_fakultas b,
tbl_kelompok c WHERE a.jurusan='".$jurusan_id.'" AND c.jk_l>'1'");
    }
    else if ($jenis_kelamin="PEREMPUAN") {
        return $this->db->query("SELECT c.id_kelompok FROM tbl_mahasiswa a, tbl_jurusan_fakultas b,
tbl_kelompok c WHERE a.jurusan='".$jurusan_id.'" AND c.jk_p>'1'");
    }
}
}

```

Keterangan : Pada *Model* memanggil database yang dimana kondisi berdasarkan jenis kelamin.

```

if (mysql_num_rows($prosesceknim)>0) { //proses mengingatkan data sudah ada
    echo "<script>alert('NIM sudah digunakan untuk mendaftar');history.go(-1) </script>";
    $id_kelompok=$id_kelompok-1;
}
}

```

Keterangan : Proses pengecekan Nomor Induk Mahasiswa

```

else if (mysql_num_rows($prosescekjurusan)>1) { //proses mengingatkan data jurusan sudah penuh
    echo "<script>alert('Kuota kelompok sudah terpenuhi untuk Jurusan tersebut, klik daftar kemb:
}
}

```

Keterangan : Proses pengecekan kuota jurusan dalam kelompok tersebut

```

else if (mysql_num_rows($prosescekfakultas)>1) { //proses mengingatkan data fakultas sudah penuh
    echo "<script>alert('Kuota kelompok sudah terpenuhi untuk fakultas tersebut, klik daftar kemb:
}
}

```

Keterangan : Proses pengecekan kuota fakultas dalam kelompok tersebut

```

else if ($jumlah<=0) { //proses mengingatkan data sudah penuh
    echo "<script>alert('Kuota kelompok sudah penuh, klik daftar kembali untuk mendaftar ulang d:
}
}
else if (($_POST['jenis_kelamin']=='LAKI-LAKI')&&($jk_l<=0)) { //proses mengingatkan data sudah :
    echo "<script>alert('Kuota laki-laki kelompok tersebut sudah penuh, klik daftar kembali untu:
}
}
else if (($_POST['jenis_kelamin']=='PEREMPUAN')&&($jk_p<=0)) { //proses mengingatkan data sudah :
    echo "<script>alert('Kuota perempuan kelompok tersebut sudah penuh, klik daftar kembali untu:
}
}

```

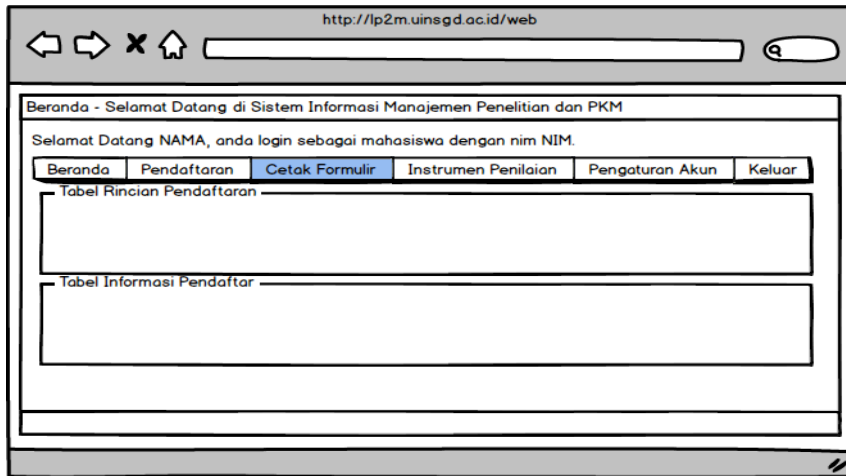
Keterangan : Proses pengecekan kuota berdasarkan jenis kelamin

Gambar 3.49 Potongan *script* penerapan algoritma *greedy*

d. Rancangan Antarmuka Tampilan Cetak Formulir

Antarmuka tampilan cetak formulir ini merupakan tampilan yang akan muncul saat akan proses mencetak formulir pendaftaran KKM. Dalam tampilan ini terdapat berbagai fungsi *interface* didalamnya salah satunya sebagai tampilan beberapa yang telah berhasil terdaftar oleh mahasiswa atau calon peserta KKM, terdapat *button* cetak yang dima tombol tersebut untuk mencetak formulir dan

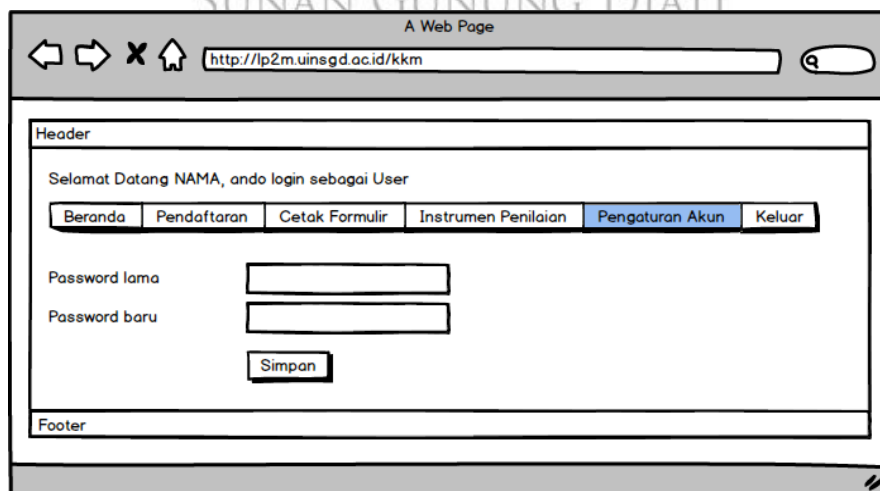
akan di *export* ke dalam format (*pdf*). Untuk lebih jelasnya dapat dilihat pada Gambar 3.50.



Gambar 3.50 *Mockup* tampilan *report* hasil cetak formulir pendaftaran

e. Rancangan Antarmuka Pengaturan Akun

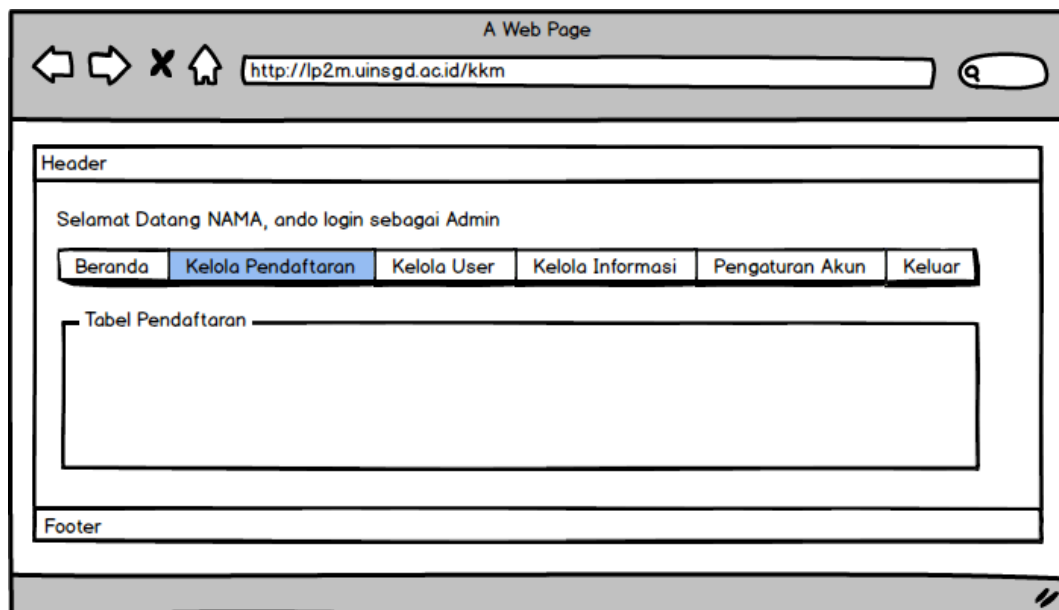
Antarmuka tampilan pengaturan akun ini merupakan tampilan yang akan muncul saat akan proses penggantian *password user*. Dalam tampilan ini terdapat berbagai fungsi *interface* didalamnya salah satunya *field* yang harus diisi oleh *user* yaitu *password* lama dan *password* baru dan *button* simpan berfungsi menyimpan perubahan yang terjadi. Untuk lebih jelasnya dapat dilihat pada gambar 3.51.



Gambar 3.51 *Mockup* tampilan pengaturan akun

f. Rancangan Antarmuka kelola pendaftaran, *user* dan informasi

Antarmuka tampilan kelola pendaftaran, *user* dan informasi ini merupakan tampilan yang akan muncul saat akan proses pengolahan data-data pelaksanaan KKM. Dalam tampilan ini terdapat berbagai fungsi *interface* didalamnya salah satunya sebagai tampilan beberapa data yang akan diolah oleh *Admin/PP-KKM*. Untuk lebih jelasnya dapat dilihat pada gambar 3.52.



Gambar 3.52 *Mockup* tampilan kelola pendaftaran
UNIVERSITAS ISLAM NEGERI
SUNAN GUNUNG DJATI
BANDUNG

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada tahap ini dilakukan implementasi dan pengujian terhadap aplikasi atau sistem, tahap ini dilakukan setelah perancangan selesai kemudian diimplementasikan terhadap bahasa pemrograman. Kemudian dilakukan pengujian terhadap sistem yang telah dibangun untuk mengetahui kelayakan dan kekurangan-kekurangan yang terdapat dalam sistem ini.

4.1 Implementasi Sistem

Langkah-langkah pada implementasi sistem diuraikan pada beberapa tahap diantaranya persiapan perangkat keras (*Hardware*), persiapan perangkat lunak (*Software*), implementasi *database* dan *user interface*.

4.1.1 Persiapan Sistem

Tahapan awal yang harus dilakukan yaitu mempersiapkan sistem agar dapat dioperasikan. Berikut hal-hal yang harus dipersiapkan yaitu :

1. Persiapan Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam perancangan aplikasi ini adalah

- c. 1 (Satu) unit *Netbook* dengan spesifikasi :
 - *Processor* : Intel (R) Core (TM) i3 CPU @2.00 GHz 2.50 GHz
 - *Installed memory* (RAM) : 4,00 GB
 - *System type* : 64 bit *Operating System*
- d. *Printer* Epson TX121X

Mencetak *report* atau laporan pengelompokkan peserta.

2. Persiapan Perangkat Lunak (*Software*)

Software yang digunakan dalam perancangan aplikasi ini adalah

- a. Sistem Operasi Windows 7 Ultimate

- b. *Adobe Dreamweaver* versi CS6
- c. MySQL
- d. Framework *CodeIgniter version 2*
- e. XAMPP Versi 5.5.35
- f. *Web Browser* (Mozilla Firefox dan Google Chrome)

4.1.2 Konversi Sistem

Tahap konversi sistem dilakukan setelah kode program selesai dibuat dan sarana sistem sudah disiapkan dengan baik yaitu dengan menggunakan data-data yang diperlukan pada sistem yang sudah dibuat. Sehingga sistem yang sudah dibuat yaitu aplikasi Kuliah Kerja Nyata Mahasiswa (KKM) dengan menggunakan algoritma *greedy* dan framework *CodeIgniter* berbasis website harus dilakukan pengujian sebelum siap dioperasikan.

4.1.3 Pengujian Sistem

Tahap pengujian sistem merupakan tahapan dilakukannya pengujian pada sistem yang telah dibuat agar sistem dapat bekerja secara optimal sesuai dengan tujuan awal yang direncanakan dan tidak terjadi kesalahan atau *error*. Tujuan utamanya adalah untuk memastikan bahwa komponen-komponen pada sistem sudah berfungsi sesuai yang diharapkan. Pengujian sistem ini sangat penting untuk dilakukan karena hal ini dilakukan untuk mengetahui apakah program dapat menerima *input*, memproses dan menghasilkan *output* sesuai dengan yang diharapkan.

4.2 Implementasi Basis Data

a. Implementasi Tabel User

Tabel 4.1 Tabel *User/Login*

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>username</u>	varchar(50)	latin1_swedish_ci		No	None	
2	email	varchar(50)	latin1_swedish_ci		No	None	
3	stts	varchar(10)	latin1_swedish_ci		No	None	
4	password	varchar(100)	latin1_swedish_ci		No	None	

b. Implementasi Tabel *Admin*

Tabel 4.2 Tabel *Admin*

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>username</u>	varchar(50)	latin1_swedish_ci		No	None	
2	nama_lengkap	varchar(100)	latin1_swedish_ci		No	None	

c. Implementasi Tabel Mahasiswa

Tabel 4.3 Tabel Mahasiswa

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>no</u>	int(30)			No	None	AUTO_INCREMENT
2	nim	int(15)			No	None	
3	nama_mahasiswa	varchar(35)	latin1_swedish_ci		No	None	
4	semester	varchar(10)	latin1_swedish_ci		No	None	
5	jurusan	varchar(100)	latin1_swedish_ci		No	None	
6	fakultas	varchar(100)	latin1_swedish_ci		No	None	

d. Implementasi Tabel Dosen Pembimbing Lapangan

Tabel 4.4 Tabel Dosen Pembimbing Lapangan

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>no</u>	int(15)			No	None	AUTO_INCREMENT
2	nidn	int(15)			No	None	
3	nama_dosen	varchar(50)	latin1_swedish_ci		No	None	
4	id_kelompok	int(10)			No	None	

e. Implementasi Tabel Kelompok

Tabel 4.5 Tabel Kelompok

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>id_kelompok</u>	int(20)			No	None	AUTO_INCREMENT
2	<u>nidn</u>	int(30)			No	None	
3	<u>dpl</u>	varchar(100)	latin1_swedish_ci		No	None	
4	<u>lokasi</u>	varchar(100)	latin1_swedish_ci		No	None	
5	<u>jumlah</u>	int(2)			No	None	
6	<u>jk_l</u>	int(2)			No	None	
7	<u>jk_p</u>	int(2)			No	None	

f. Implementasi Tabel Informasi KKM

Tabel 4.6 Tabel Informasi KKM

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>kd_info</u>	int(10)			No	None	AUTO_INCREMENT
2	<u>judul</u>	varchar(200)	latin1_swedish_ci		No	None	
3	<u>waktu_post</u>	varchar(30)	latin1_swedish_ci		No	None	
4	<u>isi</u>	text	latin1_swedish_ci		No	None	

g. Implementasi Tabel Pendaftaran

Tabel 4.7 Tabel Pendaftaran

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>id</u>	int(11)			No	None	AUTO_INCREMENT
2	<u>nim</u>	int(15)			No	None	
3	<u>nama</u>	varchar(50)	latin1_swedish_ci		No	None	
4	<u>jenis_kelamin</u>	varchar(15)	latin1_swedish_ci		No	None	
5	<u>tempat_tanggal_lahir</u>	varchar(35)	latin1_swedish_ci		No	None	
6	<u>fakultas</u>	varchar(100)	latin1_swedish_ci		No	None	
7	<u>jurusan</u>	varchar(100)	latin1_swedish_ci		No	None	
8	<u>semester</u>	int(2)			No	None	
9	<u>alamat_asal</u>	varchar(100)	latin1_swedish_ci		No	None	
10	<u>alamat_sekarang</u>	varchar(100)	latin1_swedish_ci		No	None	
11	<u>no_telp</u>	varchar(15)	latin1_swedish_ci		No	None	
12	<u>no_telp_orangtua</u>	varchar(15)	latin1_swedish_ci		No	None	
13	<u>keterangan_khusus</u>	varchar(15)	latin1_swedish_ci		No	None	
14	<u>id_kelompok</u>	int(10)			No	None	
15	<u>angka_mutu</u>	varchar(1)	latin1_swedish_ci		No	None	
16	<u>formulir_pendaftaran</u>	varchar(50)	latin1_swedish_ci		No	None	
17	<u>berkas_persyaratan</u>	varchar(50)	latin1_swedish_ci		No	None	

h. Implementasi Tabel Informasi KKM

Tabel 4.8 Tabel Informasi KKM

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	kd_info	int(10)			No	None	AUTO_INCREMENT
2	judul	varchar(200)	latin1_swedish_ci		No	None	
3	waktu_post	varchar(30)	latin1_swedish_ci		No	None	
4	isi	text	latin1_swedish_ci		No	None	

i. Implementasi Tabel Informasi KKM

Tabel 4.9 Tabel Informasi KKM

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(10)			No	None	AUTO_INCREMENT
2	id_fakultas	int(10)			No	None	
3	id_jurusan	int(15)			No	None	
4	name	varchar(100)	latin1_swedish_ci		No	None	
5	level	int(1)			No	None	

4.3 Implementasi *User Interface*

a. **Implementasi Antarmuka Tampilan *Login***

Tampilan *login* ini merupakan tampilan pertama yang akan muncul saat aplikasi dijalankan. Dalam tampilan awal ini terdapat berbagai fungsi *interface* didalamnya salah satunya sebagai tampilan *login* untuk masuk pada jendela tampilan menu utama. Dimana didalam tampilan logim terdapat beberapa field yaitu *username*, *password* dan tombol *button login* yang dapat diklik sebagai tombol masuk aplikasi ini, untuk lebih jelasnya dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Form Login

g. Implementasi Antarmuka Tampilan Halaman Utama

Antarmuka tampilan awal ini merupakan tampilan pertama yang akan muncul saat aplikasi telah berhasil login. Dalam tampilan awal ini terdapat berbagai fungsi *interface* didalamnya salah satunya sebagai tampilan menu untuk masuk pada jendela tampilan yang lain, yaitu menu pendaftaran, cetak formulir, instrument penilaian, pengaturan akun dan keluar. Untuk lebih jelasnya dapat dilihat pada gambar 4.2.



Gambar 4.2 Tampilan halaman utama

h. Implementasi Antarmuka Tampilan Halaman Pendaftaran

Antarmuka tampilan pendaftaran ini merupakan tampilan yang akan muncul saat akan mengisi formulir pendaftaran KKM. Dalam tampilan ini terdapat berbagai fungsi *interface* didalamnya salah satunya sebagai tampilan beberapa field yang harus diisi oleh mahasiswa atau calon peserta KKM, dan apabila sudah lengkap, terdapat *button* daftar yang dima tombol tersebut untuk mendaftarkan formulir. Untuk lebih jelasnya dapat dilihat pada gambar 4.3.

Formulir Pendaftaran KKM - Lembaga Penelitian dan Pengabdian Masyarakat

Selamat Datang **FURIANSYAH DIPRAJA**, anda login sebagai mahasiswa dengan nim **1127050062**.

[Beranda](#) | [Formulir Pendaftaran](#) | [Cetak Formulir](#) | [Informasi](#) | [Pengisian Akun](#) | [Kelas](#)

**PERMOHONAN PENDAFTARAN
 PESERTA KULIAH KERJA NYATA MAHASISWA (KKM)
 UIN SUNAN GUNUNG DJATI BANDUNG
 TAHUN AKADEMIK 2015/2016**

CEK KUOTA KELOMPOK

KELOMPOK:
 NAMA:
 NIM:
 JENIS KELAMIN: L P
 TEMPAT/TANGGAL LAHIR:
 FAKULTAS:
 JURUSAN:
 SEMESTER:
 ALAMAT ASAL (LENGKAP):
 ALAMAT SEKARANG (LENGKAP):
 NO. TELP/HP:
 NO. TELP/HP (ORANG TUA):
 KETERANGAN KHUSUS** Menakut Harus Sakit Kronis Bekerja Tidak ada
 UPLOAD: No file chosen

Petunjuk Pendaftaran Online:

- Formulir ini diisi oleh ybs, dengan huruf kapital.
- Melampirkan surat keterangan telah menyelesaikan kuliah Semester VII atau 75% jumlah SKS dari fakultas masing-masing.
- Melampirkan bukti telah melunasi SPP semester genap Tahun Akademik 2015/2016.
- Melampirkan foto berukuran 2 x 3 (format).
- Ketiga lampiran tersebut dimasukkan ke dalam format berkas (RAR/ZIP)
 Format penulisan nama pada file(RAR) lampiran sh: NIM-Jurusan-Fakultas-Lampiran
 Penyusunan KKM 2016 dan diuploadkan berukuran Maks 2 MB

Dengan ini :

- Mengajukan permohonan untuk menjadi peserta Kuliah Kerja Nyata Mahasiswa.
- Bersedia ditempatkan di salah wilayah Kuliah Kerja Nyata Mahasiswa.
- Bersedia memenuh dan mematuhi ketentuan yang telah ditetapkan PP-KKM, serta bersedia menerima sanksi jika melanggar ketentuan tersebut.
- Bertanggung jawab terhadap isian formulir pendaftaran KKM ini.

Catatan : Pastikan data diri dan lampiran telah lengkap dan benar.

LEMBAGA PENELITIAN DAN PENGABDIAN MASYARAKAT
 UIN SUNAN GUNUNG DJATI BANDUNG
 © 2016. All rights reserved | Design by Dipraja
 Halaman ini dibuat selama 0.3459 detik

Gambar 4.3 Tampilan *form* formulir pendaftaran

Pada proses pengisian form formulir pendaftaran terdapat penerapan algoritma *greedy* untuk menentukan pengelompokkan peserta, berikut potongan *script* dari algoritma *greedy* pada gambar 4.4.


```

$jenis_kelamin=$bc['jenis_kelamin'];
$jurusan=$bc['jurusan'];
$fakultas=$bc['fakultas'];
$bc['cek_kelompok'] = $this->m_mahasiswa->get_pengacakan($jurusan,$jenis_kelamin,$fakultas);

```

Keterangan : Pada *controller* memanggil fungsi `get_pengacakan()` dengan parameter `jenis_kelamin`, `jurusan` dan `fakultas`.

```

public function get_pengacakan($jurusan_id,$jenis_kelamin,$fakultas_id)
{
    $kelompok=270;
    if ($jenis_kelamin="LAKI-LAKI") {
        return $this->db->query("SELECT c.id_kelompok FROM tbl_mahasiswa a, tbl_jurusan_fakultas b,
tbl_kelompok c WHERE a.jurusan='".$jurusan_id.'" AND c.jk_l>'1'");
    }
    else if ($jenis_kelamin="PEREMPUAN") {
        return $this->db->query("SELECT c.id_kelompok FROM tbl_mahasiswa a, tbl_jurusan_fakultas b,
tbl_kelompok c WHERE a.jurusan='".$jurusan_id.'" AND c.jk_p>'1'");
    }
}

```

Keterangan : Pada *Model* memanggil database yang dimana kondisi berdasarkan jenis kelamin.

```

if (mysql_num_rows($prosesceknim)>0) { //proses mengingatkan data sudah ada
    echo "<script>alert('NIM sudah digunakan untuk mendaftar');history.go(-1) </script>";
    $id_kelompok=$id_kelompok-1;
}

```

Keterangan : Proses pengecekan Nomor Induk Mahasiswa

```

else if (mysql_num_rows($prosescekjurusan)>1) { //proses mengingatkan data jurusan sudah penuh
    echo "<script>alert('Kuota kelompok sudah terpenuhi untuk Jurusan tersebut, klik daftar kemb
}

```

Keterangan : Proses pengecekan kuota jurusan dalam kelompok tersebut

```

else if (mysql_num_rows($prosescekfakultas)>1) { //proses mengingatkan data fakultas sudah penuh
    echo "<script>alert('Kuota kelompok sudah terpenuhi untuk fakultas tersebut, klik daftar kemb
}

```

Keterangan : Proses pengecekan kuota fakultas dalam kelompok tersebut

```

else if ($jumlah<=0) { //proses mengingatkan data sudah penuh
    echo "<script>alert('Kuota kelompok sudah penuh, klik daftar kembali untuk mendaftar ulang d
}
else if (($_POST['jenis_kelamin']=="LAKI-LAKI")&&($jk_l<=0)) { //proses mengingatkan data sudah
    echo "<script>alert('Kuota laki-laki kelompok tersebut sudah penuh, klik daftar kembali untu
}
else if (($_POST['jenis_kelamin']=="PEREMPUAN")&&($jk_p<=0)) { //proses mengingatkan data sudah
    echo "<script>alert('Kuota perempuan kelompok tersebut sudah penuh, klik daftar kembali untu
}

```

Keterangan : Proses pengecekan kuota berdasarkan jenis kelamin

Gambar 4.4 Potongan *Script* penerapan algoritma *greedy*

Kode program yang berdasarkan gambar 4.4 merupakan proses pengelompokkan peserta KKM. Terdapat proses pengacakan (*randomize*) yaitu untuk mengacak nomor kelompok dari batas minimal hingga maksimal kelompok yang disediakan, lalu selanjutnya penerapan *greedy* dengan fungsi selektif yang dimana percabangan bersarang dan pemeriksaan kuota, memilih layak atau tidak layak masuk ke dalam kelompok tersebut dengan beberapa pertimbangan yaitu jenis kelamin, jurusan dan fakultas.

Keterangan : *controller* (mengecek langkah per langkah tentang optimal dari sebuah kelompok, yang dimana satu per satu dicek apakah kuota tersebut masih layak atau tidak calon peserta masuk pada kelompok

4.4 Pengujian Sistem

Proses pengujian yaitu mencoba aplikasi apakah sudah bisa digunakan oleh pengguna. Pada tahap ini merupakan kelanjutan dari tahap implementasi yaitu melakukan pengujian terhadap aplikasi yang dibangun. Pengujian yang akan dilakukan yaitu dengan pengujian *black box* yang berfokus pada persyaratan fungsional perangkat lunak.

a. Uji halaman *login*

Pengujian halaman login bisa dilihat pada Tabel 4.10.

Tabel 4.10 Pengujian Halaman *Login*

No	Kode	Skenario	Hasil		Keterangan
			Sukses	Gagal	
1	P03	Isi field <i>username</i>	√		
2	P04	Isi field <i>password</i>	√		Pakai Enkripsi MD5
3	P05	Klik tombol <i>login</i>	√		Berhasil memproses aksi login dan memverifikasi
4	P06	Klik tombol keluar	√		Berhasil keluar dari aplikasi

b. Uji halaman utama

Pengujian halaman menu bisa dilihat pada Tabel 4.11.

Tabel 4.11 Pengujian Halaman Utama

No	Kode	Skenario	Hasil		Keterangan
			Sukses	Gagal	
1	P07	Menklik menu formulir pendaftaran	√		Berhasil menampilkan tampilan menu formulir pendaftaran
2	P08	Klik menu cetak formulir	√		Berhasil menampilkan tampilan menu cetak formulir
3	P09	Klik menu instrumen penilaian	√		Berhasil menampilkan tampilan menu instrumen penilaian
4	P10	Klik menu pengaturan akun	√		Berhasil menampilkan tampilan menu pengaturan akun
5	P11	Klik keluar	√		Berhasil <i>logout</i> dari aplikasi

. Pengujian Use Case

Dalam pengujian sistem perlu juga pengujian aplikasi terhadap skenario *use case*. Pengujian ini di fungsikan untuk mengetahui apakah *use case* yang dapat dilakukan *actor (user)* dapat berjalan dengan baik atau tidak.

Tabel 4.12 Pengujian Skenario Use Case

Use Case	Kode Uji	Kasus Uji	Parameter Pengujian	Hasil yang diharapkan	Reaksi Sistem	Validasi
UCS-01 (Login)	TBI-01	Isi field <i>username</i> dan <i>password</i>	<i>User</i> memasukkan isi <i>field</i>	Menampilkan peringatan kesalahan <i>input</i> jika memasukkan simbol	Menampilkan peringatan kesalahan <i>input</i> jika memasukkan simbol	Valid

Tabel 4.12 Pengujian Skenario *Use Case* (lanjutan)

Use Case	Kode Uji	Kasus Uji	Parameter Pengujian	Hasil yang diharapkan	Reaksi Sistem	Validasi
	TBI-02	Melakukan <i>login</i>	User mengklik tombol <i>login</i>	Aplikasi menampilkan halaman utama ketika berhasil <i>login</i> dan menampilkan peringatan gagal login ketika terjadi kesalahan	Aplikasi menampilkan halaman utama ketika berhasil <i>login</i> dan menampilkan peringatan gagal login ketika terjadi kesalahan	Valid
UCS-02 (Mengisi Formulir Pendaftaran KKM)	TBI-03	Mengisi <i>field</i> formulir	User memasukkan data	Menampilkan peringatan kesalahan input jika memasukkan inputan yang dilarang	Menampilkan peringatan kesalahan input jika memasukkan inputan yang dilarang	Valid
	TBI-04	Melakukan pendaftaran	User mengklik tombol daftar	Aplikasi akan memproses pendaftaran	Aplikasi akan memproses pendaftaran	Valid
UCS-03 (Proses Pengelompokan Peserta (Algoritma <i>Greedy</i>))	TBI-05	Melakukan pengacakan dan pengelompokan	User mengklik tombol daftar	Aplikasi mengacak nomor kelompok dan menentukan/ menyeleksi kelompok	Aplikasi mengacak nomor kelompok dan menentukan/ menyeleksi kelompok	Valid
UCS-04 (Melihat Informasi Kelompok)	TBI-06	Mengklik tombol menu informasi	User mengklik menu informasi	Aplikasi menampilkan tampilan informasi	Aplikasi menampilkan tampilan informasi	Valid
UCS-05 (Cetak Dokumen Formulir)	TBI-07	Masuk menu cetak formulir	User mengklik menu cetak formulir	Aplikasi menampilkan tampilan cetak formulir	Aplikasi menampilkan tampilan cetak formulir	Valid
	TBI-08	Melakukan proses cetak	User mengklik tombol cetak	Aplikasi memproses dan menampilkan <i>print</i> formulir	Aplikasi memproses dan menampilkan <i>print</i> formulir	Valid
UCS-06 (Kelola Pendaftaran)	TBI-09	Mengolah data	User mengklik tombol <i>CRUD</i>	Aplikasi merespon dan melakukan proses <i>CRUD</i>	Aplikasi merespon dan melakukan proses <i>CRUD</i>	Valid
UCS-07 (Kelola Informasi Kelompok)	TBI-10	Mengelola Informasi	User mengklik tombol <i>CRUD</i>	Aplikasi merespon dan melakukan proses <i>CRUD</i>	Aplikasi merespon dan melakukan proses <i>CRUD</i>	Valid
UCS-08 (Kelola <i>User</i>)	TBI-11	Mengelola Informasi	User mengklik tombol <i>CRUD</i>	Aplikasi merespon dan melakukan proses <i>CRUD</i>	Aplikasi merespon dan melakukan proses <i>CRUD</i>	Valid

Tabel 4.12 Pengujian Skenario *Use Case* (lanjutan)

Use Case	Kode Uji	Kasus Uji	Parameter Pengujian	Hasil yang diharapkan	Reaksi Sistem	Validasi
UCS-09 (Verifikasi Dokumen)	TBI-12	Mengecek dokumen persyaratan	User mengklik download daftar peserta	Aplikasi melakukan proses <i>download</i>	Aplikasi melakukan proses <i>download</i>	Valid
UCS-10(Kelola Penilaian)	TBI-13	Mengolah data	User mengklik tombol <i>CRUD</i>	Aplikasi merespon dan melakukan proses <i>CRUD</i>	Aplikasi merespon dan melakukan proses <i>CRUD</i>	Valid

d. Pengujian Penerapan Algoritma *Greedy*

1. randomized kelompok
2. if waktu daftar != periode pendaftaran
3. randomized kelompok
4. endif
5. else if nim > 0
6. randomized kelompok
7. endif
8. else if jurusan > 1
9. randomized kelompok
10. endif
11. else if fakultas > 1)
12. randomized kelompok
13. endif
14. else if semester == 2|| semester == 4
15. randomized kelompok
16. endif
17. else if jumlah kuota <= 0
18. randomized kelompok
19. endif
20. else if kuota jenis_kelamin == LAKI-LAKI && jk_l <= 0
21. randomized kelompok
22. endif
23. else if kuota jenis_kelamin == PEREMPUAN && jk_p <= 0
24. randomized kelompok
25. endif
26. else if persyaratan != lengkap
27. randomized kelompok
28. endif
29. else
30. inputkan formulir pendaftaran

Tabel 4.13 Pengujian Penerapan Algoritma *Greedy*

No	Kasus Uji	Hasil Yang Diharapkan	Hasil Uji Kasus	Keterangan
1	if waktu daftar != periode pendaftaran randomized kelompok endif	Muncul pemberitahuan belum masa periode pendaftaran	Muncul pemberitahuan belum masa periode pendaftaran	[√] Alur Terlewati [] Alur Tidak Terlewat
2	else if nim > 0 endif	Mengecek nim, muncul pemberitahuan nim sudah digunakan	Mengecek nim, muncul pemberitahuan nim sudah digunakan	[√] Alur Terlewati [] Alur Tidak Terlewat
3	else if jurusan > 1 endif	Mengecek jurusan, muncul pemberitahuan kuota jurusan pada kelompok tersebut penuh	Mengecek jurusan, muncul pemberitahuan kuota jurusan pada kelompok tersebut penuh	[√] Alur Terlewati [] Alur Tidak Terlewat
4	else if fakultas > 1) endif	Mengecek fakultas, muncul pemberitahuan kuota fakultas pada kelompok tersebut penuh	Mengecek fakultas, muncul pemberitahuan kuota fakultas pada kelompok tersebut penuh	[√] Alur Terlewati [] Alur Tidak Terlewat
5	else if semester == 2 semester == 4 endif	Mengecek semester, muncul pemberitahuan semester tidak bisa daftar pada semester 2 dan 4 karena tidak memenuhi persyaratan	Mengecek semester, muncul pemberitahuan semester tidak bisa daftar pada semester 2 dan 4 karena tidak memenuhi persyaratan	[√] Alur Terlewati [] Alur Tidak Terlewat
6	else if jumlah kuota <= 0 endif	Mengecek kuota, muncul pemberitahuan kuota sudah penuh	Mengecek kuota, muncul pemberitahuan kuota sudah penuh	[√] Alur Terlewati [] Alur Tidak Terlewat
7	else if kuota jenis_kelamin == LAKI-LAKI && jk_1 <= 0 endif	Mengecek kuota dengan berjenis kelamin laki-laki, muncul pemberitahuan kuota sudah penuh	Mengecek kuota dengan berjenis kelamin laki-laki, muncul pemberitahuan kuota sudah penuh	[√] Alur Terlewati [] Alur Tidak Terlewat
8	else if kuota jenis_kelamin == PEREMPUAN && jk_p <= 0	Mengecek kuota dengan berjenis kelamin perempuan, muncul pemberitahuan kuota sudah penuh	Mengecek kuota dengan berjenis kelamin perempuan, muncul pemberitahuan kuota sudah penuh	[√] Alur Terlewati [] Alur Tidak Terlewat
9	else if persyaratan != lengkap endif	Mengecek kelengkapan persyaratan, jika lengkap dilanjutkan	Mengecek kelengkapan persyaratan, jika lengkap dilanjutkan	[√] Alur Terlewati [] Alur Tidak Terlewat
10	else inputkan formulir pendaftaran	Menyimpan kedalam database, jika semua tahap seleksi sudah terpenuhi	Menyimpan kedalam database, jika semua tahap seleksi sudah terpenuhi	[√] Alur Terlewati [] Alur Tidak Terlewat

e. Analisis Pengujian Kinerja Algoritma *Greedy*

Pengujian hasil pengelompokan peserta KKM menggunakan data peserta periode tahun 2016, dimana periode tahun 2016 sejumlah 2400 peserta. Penentuan jumlah anggota setiap kelompok dilakukan secara manual, dimana admin memasukkan jumlah anggota yang diinginkan untuk memulai proses pengelompokan. Menurut pihak LP2M UIN SGD Bandung, kriteria ideal dalam satu kelompok KKM yakni apabila:

1. Setiap kelompok terdiri dari maksimal 12 peserta.
2. Setiap anggota kelompok berasal dari fakultas dan jurusan yang berbeda dengan perbedaan jenis kelamin yang merata yaitu maksimal 7 perempuan dan 5 laki-laki.

Oleh karena itu, pengujian hasil pengelompokan dilakukan dengan jumlah anggota setiap kelompok 12 orang dan memprioritaskan pengelompokan berdasarkan fakultas. Perbedaan jenis kelamin dan jurusan juga diperhatikan untuk memperoleh anggota kelompok yang lebih heterogen. Gambar 4.5 merupakan contoh hasil pengelompokan yang dilakukan dengan prioritas perbedaan fakultas menggunakan data periode tahun 2016.

id	nama	nim	jenis_kelamin	tempat_tanggal_lahir	fakultas	jurusan	semester	alamat_asal	alamat_sekarang	no_telp	no_telp_orangtua	keterangan_khusus	persyaratan	id_kelompok
109	TARA SANNI	1125030271	PEREMPUAN	PEMALANG, 05/05/1995	ADAB DAN HUMANIORA	BAHASA DAN SAstra INGGRIS	8	JL. SAMOSIR TMUR I/237 RT03 RW14 KEL. BOJONGBA...	JL. RIUNG HEGAR III NO. 17 A RT. 10 RW10 KEL. CIS...	085786090001	083837148770	TIDAK ADA		258
178	FITRIAH	1124030029	PEREMPUAN	SUBANG, 23/02/1994	DAKWAH DAN KOMUNIKASI	MANAJEMEN DAKWAH	8	KP. OLEAT DESA GARDUSAYANG KECAMATAN CICALAK KABUPATEN...	JL. TANJUNGSARI NO.24 OLEUNYI WETAN KABUPATEN BA...	089858785858	089510728371	MENIKAH.		258
211	Novela Anggraeni	1124080058	PEREMPUAN	CIAMIS, 13/11/1994	DAKWAH DAN KOMUNIKASI	Hubungan Masyarakat	8	JL. PANGANDARAN KEC. KALUPUCANG RT01/RW02 KAB. PAN...	JL. MANISI KP. JATI KEL. PASIRBIRU KEC. CIBIRU BAN...	089607181659	085289211858	TIDAK ADA		258
334	NENENG KHOCERUNISA	1124040020	PEREMPUAN	SUBANG, 19/04/1995	DAKWAH DAN KOMUNIKASI	PMI	8	DUSUN RANCABANGO KEC. PATOK/BEUSI KAS. SUBANG RW 08 ...	JL. DESA CIPADUNG	089529182388	082214822081	TIDAK ADA		258
512	FURIANSYAH DIPRAJA	1127050082	LAKI-LAKI	BANDUNG, 12/09/1994	SAINS DAN TEKNOLOGI	TEKNIK INFORMATIKA	8	KP. BUNUT 02/05 DS MARGAHURIP KEC BANJARAN KAB BAND...	KP. BUNUT 02/05 DS MARGAHURIP KEC BANJARAN KAB BAND...	08988841197	082318092810	TIDAK ADA		258
589	AHMAD SOHEH	1127010005	LAKI-LAKI	CIANJUR, 27/07/1994	SAINS DAN TEKNOLOGI	MATEMATIKA	8	KP. BABAKAN KARAWAT DS. GIRMULYA KEC. CIBEBER KAB...	UJUNG BERUNG KOTA BANDUNG	085704043727	085720876743	TIDAK ADA		258
693	HASIB ABDULLAH	1127010029	LAKI-LAKI	BANDUNG, 20/10/1994	SAINS DAN TEKNOLOGI	MATEMATIKA	8	Jl. Cilutra RT 05/02 Kel Naglasari Kec. Cibeunying...	Jl. Cilutra RT 05/02 Kel Naglasari Kec. Cibeunying...	085784079080	081321722132	TIDAK ADA		258
1045	IQBALUL MUNAJAT	1127050078	LAKI-LAKI	BOGOR, 21/09/1994	SAINS DAN TEKNOLOGI	TEKNIK INFORMATIKA	8	KOMPLEK AMANAH ASRI LEUWILIANG BOGOR RT001/013	KOMPLEK VIJAYA KUSUMA BLOK A3K NO 13	085885835272	085721871219	TIDAK ADA		258
1188	Hilmi Zuraida	1211103082	PEREMPUAN	JAKARTA, 01/02/1993	USHULUDDIN	ILMU AL-QURAN DAN ILMU HADITS	10	Jl. ANTASARI 2 No. 12 RT/RW 02/17 KEL. ANTAPANI K...	OBANGKONOL RT 01 RW05	085722728023	081584888801	MENIKAH.		258
1173	AHMAD TALUAN DZUL FARHAN	1123010088	LAKI-LAKI	BANDUNG, 28/12/1993	SYARIAH DAN HUKUM	AHWAL SYAKHSIYAH	8	JALAN HOLIS GIG. IBU ITI 3 NO 28/82 RT002 RW001 K...	JALAN HOLIS GIG. IBU ITI 3 NO 28/82 RT002 RW001 K...	081220940285	081345888888	TIDAK ADA		258
1305	SANTY PURNAMASARI	1123010109	PEREMPUAN	GARUT, 28/05/1994	SYARIAH DAN HUKUM	AHWAL SYAKHSIYAH	8	KP. JATIWANGI RT 01 RW02 DESA CIBAHUR KECAMATAN S...	KP. JATIWANGI RT 01 RW02 DESA CIBAHUR KECAMATAN S...	082240581185	085320848425	MENIKAH.		258
1645	YULIANA RACHMAH	1124050158	PEREMPUAN	GARUT, 13/01/1994	DAKWAH DAN KOMUNIKASI	JURNALISTIK	VIII	JL. SUMBANVA I NO.128 RT.04/RW 02 KELURAHAN MERPETA	JL. SUMBANVA I NO.128 RT.04/RW 02 KELURAHAN MERPETA	087822733450	085722388534	TIDAK ADA		258

Gambar 4.5 Hasil Pengelompokan periode tahun 2016

Setelah perbedaan fakultas, selanjutnya proses pengelompokan dilakukan berdasarkan dua asumsi, yaitu :

1. Setiap kelompok terdiri dari dua belas peserta KKM dengan komposisi maksimal tiga peserta dari jurusan yang sama dan maksimal tiga peserta dari fakultas yang sama.

2. Setiap kelompok terdiri dari dua belas peserta KKM dengan komposisi lima peserta pria dan tujuh peserta wanita. Hasil pengelompokan berdasarkan asumsi pengelompokan pertama, yakni perbedaan jurusan untuk periode tahun 2016, ditunjukkan pada Gambar 4.5.

f. Analisis Perhitungan Persentase Pengelompokan

Pengujian hasil pengelompokan dilakukan dengan menghitung persentase ideal seluruh kelompok berdasarkan kedua asumsi di atas sekaligus. Analisis hasil

pengelompokan ditentukan berdasarkan kriteria buruk, baik, dan sangat baik untuk banyaknya kelompok yang sesuai dengan kedua asumsi. Interval penilaian setiap kriteria dapat dicari menggunakan rumus sebagai berikut :

$$\text{Interval kriteria} = \frac{100 \%}{\text{Banyaknya item interval}}$$

Dari hasil perhitungan, diperoleh interval penilaian kriteria hasil pengelompokan 33,33 %. Sehingga kriteria penilaian hasil pengelompokan dapat ditunjukkan pada Tabel 4.14.

Tabel 4.14 Kriteria penilaian hasil

Kriteria	Interval
Buruk	0 – 33,33%
Baik	33,34% – 66,66%
Sangat baik	66,67% – 100%

Setelah dilakukan perhitungan interval penilaian hasil pengelompokan, selanjutnya dilakukan analisa dari setiap kelompok berdasarkan kategori ideal dan tidak ideal. Hasil kelompok dikatakan ideal apabila kedua asumsi pengelompokan terpenuhi, sedangkan tidak ideal apabila kedua asumsi pengelompokan tidak terpenuhi. Persentase ideal dan tidak ideal dari hasil pengelompokan menggunakan data periode tahun 2016 ditunjukkan pada Tabel 2.

Tabel 4.14 Persentase hasil pengelompokkan peserta KKM menggunakan data periode tahun 2016

Kategori	Periode Tahun 2016	
	Jumlah Kelompok	Persentase
Ideal	218	90%
Tidak Ideal	33	10%
Total Kelompok	251	100%

Tabel 4.14 menunjukkan bahwa persentase jumlah kelompok yang masuk ke dalam kategori ideal pada periode tahun 2016 sebesar 90%. Dengan demikian, hasil pengelompokan peserta yang dilakukan menggunakan data periode tahun 2016 pada aplikasi KKM masuk ke dalam interval pertama (66,66 %-100 %) yakni kriteria sangat baik.

Periode tahun 2016 terdapat 33 yang termasuk tidak ideal dengan persentase 10%. Dengan demikian, hal ini membuktikan bahwa perbedaan jumlah data dari masing-masing fakultas juga mempengaruhi besar kecilnya persentase ideal kelompok. Bukti tersebut sesuai dengan strategi pengelompokan menggunakan algoritma *greedy*, dimana himpunan kandidat merupakan seluruh peserta dari fakultas(i), dengan jumlah peserta fakultas (i) tidak bernilai nol.

g. Kesimpulan Pengujian

Berdasarkan hasil pengujian terhadap aplikasi Kuliah Kerja Nyata Mahasiswa, dapat ditarik kesimpulan aplikasi ini dapat berfungsi sesuai dengan yang diharapkan serta aplikasi ini yang dibangun telah cukup memenuhi tujuan awal pembangunan.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari skripsi yang berjudul “Perancangan Aplikasi Kuliah Kerja Nyata Mahasiswa Dengan Menggunakan Algoritma Greedy Untuk Menentukan Pengelompokan Peserta Kuliah Kerja Nyata Mahasiswa” ini sebagai berikut

1. Telah berhasil diterapkan algoritma *greedy* pada aplikasi Kuliah Kerja Nyata Mahasiswa (KKM) yang mampu menentukan pengelompokan peserta dengan komposisi pembagian peserta dalam satu kelompok secara merata berdasarkan fakultas.
2. Berdasarkan hasil pengujian kinerja pengelompokan menggunakan algoritma *greedy*, proses pengelompokan peserta KKM menggunakan aplikasi informasi Kuliah Kerja Nyata Mahasiswa (KKM) masuk dalam interval sangat baik dengan persentase kelompok yang masuk ke dalam kategori ideal sebesar 90%.
3. Implementasi algoritma *greedy* pada penentuan pengelompokan peserta itu kinerjanya, lebih baik daripada pengelompokan secara manual yang sebelumnya diterapkan oleh lembaga tersebut didalam penentuan pengelompokan peserta.

5.2 Saran

Pada aplikasi ini masih terdapat banyak sekali kekurangan, maka dari itu diperlukan beberapa pembaharuan sebagai penyempurnaan dari aplikasi ini diantaranya :

1. Dalam proses pengumpulan data dan kebutuhan pengguna dalam hal ini PP-KKM, diharapkan menjalin komunikasi lebih baik lagi, agar terciptanya aplikasi yang lebih baik.
2. Diharapkan membuat aplikasi terintegrasi ke Sistem Informasi Akademik Universitas, agar terciptanya sistem yang saling mendukung dalam hal data yang dibutuhkan.
3. Agar terciptanya aplikasi yang lebih baik lagi, supaya dikembangkan lagi dan pakai metode atau algoritma yang lain, untuk membandingkan dan bertujuan supaya membuat aplikasi bekerja lebih baik lagi.





uin

UNIVERSITAS ISLAM NEGERI
SUNAN GUNUNG DJATI
BANDUNG

DAFTAR ISI

HALAMAN JUDUL	
LEMBAR PERSETUJUAN	
LEMBAR PENGESAHAN	
LEMBAR PERNYATAAN BEBAS PLAGIAT	
LEMBAR PERSEMBAHAN	
HALAMAN MOTTO	
RIWAYAT HIDUP	
KATA PENGANTAR	i
ABSTRAK	iv
ABSTRACT	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan dan Manfaat.....	3
1.4 Batasan Masalah.....	4
1.5 Kerangka Pemikiran.....	5
1.6 Metodologi Penelitian.....	6
1.6.1 Tahap Pengumpulan Data.....	6
1.6.2 Tahap Pengembangan Sistem.....	7
1.7 Sistematika Penulisan.....	9
BAB II LANDASAN TEORI	11
2.1 <i>State Of The Art</i>	11
2.2 Perancangan.....	14
2.2.1 Konsep Dasar Perancangan.....	14
2.3 Konsep Dasar Aplikasi.....	15
2.3.1 Pengertian Aplikasi.....	15
2.3.2 Pengertian Aplikasi <i>Web</i>	15

2.4	Kuliah Kerja Nyata Mahasiswa.....	16
2.4.1	Pengertian Kuliah Kerja Nyata Mahasiswa	16
2.4.2	Penjadwalan Pelaksanaan.....	16
2.4.3	Prosedur Pelaksanaan.....	17
2.5	Algoritma <i>Greedy</i>	17
2.6	Metode Pengembangan <i>Prototype</i>	19
2.6.1	Konsep Dasar <i>Prototype</i>	19
2.6.2	Keunggulan dan Kelemahan <i>Prototyping</i>	20
2.7	<i>Framework CodeIgniter</i>	22
2.8	UML (<i>Unified Modeling Language</i>).....	25
2.8.1	Diagram <i>Use Case (Use Case Diagram)</i>	26
2.8.2	<i>Activity Diagram</i>	28
2.8.3	<i>Class Diagram</i>	29
2.8.4	<i>Sequence Diagram</i>	31
2.9	Basis Data.....	34
2.9.1	Pengertian Basis Data	34
2.9.2	Model Data.....	34
2.10	MYSQL	37
2.11	Pengujian Perangkat Lunak	38
2.11.1	Teknik Pengujian <i>Black-Box</i>	39
2.11.2	Pengujian Kinerja Algoritma <i>Greedy</i>	40
BAB III ANALISIS DAN PERANCANGAN		41
3.1	Analisis <i>Current Sistem (Quick Plan Phases)</i>	41
3.2	Kerangka Kerja Konseptual	43
3.3	Analisis Kebutuhan Hardware dan Software.....	46
3.4	Analisis Kebutuhan Fungsional.....	47
3.4.1	Deskripsi Global Aplikasi	47
3.4.2	Arsitektur Sistem.....	51
3.5	Perancangan Aplikasi (<i>Modeling Quick Design Phases</i>).....	52
3.5.1	<i>Use case Diagram</i>	52
3.5.2	Skenario <i>Use Case Diagram</i>	53
3.5.3	Analisis <i>Activity Diagram</i>	60
3.5.4	<i>Class Diagram</i>	69

3.5.5	<i>Sequence Diagram</i>	74
3.6	Perancangan <i>Database</i>	83
3.6.1	<i>Conceptual Data Model (CDM)</i>	83
3.6.2	Physical Data Model	84
3.6.3	Identifikasi Tabel	84
3.7	Perancangan Proses Pendaftaran	89
3.7.1	Aspek Input	89
3.7.2	Aspek proses	90
3.7.3	Aspek output	90
3.8	Perancangan Antarmuka (<i>Construction of Protoype</i>)	91
BAB IV IMPLEMENTASI DAN PENGUJIAN		97
4.1	Implementasi Sistem.....	97
4.1.1	Persiapan Sistem	97
4.1.2	Konversi Sistem	98
4.1.3	Pengujian Sistem.....	98
4.2	Implementasi Basis Data	98
4.3	Implementasi <i>User Interface</i>	101
4.4	Pengujian Sistem	105
BAB V KESIMPULAN DAN SARAN		114
5.1	Kesimpulan	114
5.2	Saran.....	114
DAFTAR PUSTAKA		116
LAMPIRAN		119