RANCANG BANGUN KARTU RENCANA STUDI (KRS) DAN KARTU HASIL STUDI (KHS) *ONLINE* MENGGUNAKAN *FRAMEWORK CODEIGNITER* STUDI KASUS: STIE YBPK PALANGKARAYA

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Penulisan Tugas Akhir Pada

Sekolah Tinggi Manajemen Informatika Dan Komputer

(STMIK) Palangkaraya

OLEH

MARIAH RISKA MAHBUBAH

NIM C1455201040

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2021

RANCANG BANGUN KARTU RENCANA STUDI (KRS) DAN KARTU HASIL STUDI (KHS) *ONLINE* MENGGUNAKAN *FRAMEWORK CODEIGNITER* STUDI KASUS: STIE YBPK PALANGKARAYA

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Penulisan Tugas Akhir Pada

Sekolah Tinggi Manajemen Informatika Dan Komputer

(STMIK) Palangkaraya

OLEH

MARIAH RISKA MAHBUBAH

NIM C1455201040

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

(STMIK) PALANGKARAYA

2021

ABSTRAK

Mariah Riska Mahbubah, C1455201040, 2021. Rancang Bangun Kartu Rencana Studi (Krs) Dan Kartu Hasil Studi (Khs) *Online* Menggunakan *Framework Codeigniter* Studi Kasus: Stie Ybpk Palangkaraya, Pembimbing I Maura Widyaningsih, S.Kom., M.Cs., Pembimbing II H. Suratno S.kom., M.Si.

STIE YBPK merupakan lembaga pendidikan tinggi dengan jurusan akuntansi dengan jenjang D3 dan S1. Pada saat ini STIE YBPK Palangkaraya belum memanfaatkan teknologi informasi pada penyusunan kartu rencana studi(KRS) dan kartu hasil studi(KHS). Oleh karena itu, *website* ini bertujuan untuk memudahkan dalam mengolah data akademik yang ada di STIE YBPK Palangkaraya seperti KRS dan KHS.

Aplikasi ini menggunakan bahasa pemrograman *HTML* dan *PHP*. Sedangkan sistem penyimpanan menggunakan *Mysql.* Aplikasi ini diharapkan dapat membantu dalam menyimpan dan mengolah data akademik, sehingga bisa mengoptimalkan pemrosesan data.

Kata kunci : Aplikasi, KRS, KHS

Mariah Riska Mahbubah, C1455201040, 2021. Design Of Study Plan Card (Krs) And Study Result Card (Khs) Online To Use Framework Codeigniter Case Study: STIE YBPK Palangkaraya Pembimbing I Maura Widyaningsih, S.Kom., M.Cs., Pembimbing II H. Suratno S.kom., M.Si.

STIE YBPK is a higher education institution with a major in accounting with D3 and S1 levels. At this time STIE YBPK Palangkaraya has not utilized information technology in the preparation of study plan cards (KRS) and study result cards (KHS). Therefore, this website aims to facilitate the processing of academic data at STIE YBPK Palangkaraya such as KRS and KHS.

This application uses the HTML and PHP programming languages. While the storage system uses Mysql. This application is expected to assist in storing and processing academic data, so that it can optimize data processing.

Keywords : Application, KRS, KHS

DAFTAR ISI

BAB I PENDAHULUAN

A. Latar Belakang Masalah 1

B. Perumusan Masalah 2

C. Batasan Masalah 2

D. Tujuan dan Manfaat 4

E. Metode Penelitian 4

F. Sistematika Penulisan 7

BAB II KAJIAN TEORI

A. Tinjauan Pustaka 9

B. Kajian Teori 16

BAB III ANALISA DAN DESAIN SISTEM

A. Tinjauan Umum 18

B. Analisis 19

1. Analisis Kelemahan Sistem 19

2.Analisis Kebutuhan 19

3.Analisis Kelayakan Sistem 21

C. Desain Sistem 22

1.Desain Proses 22

2.Desain Interface 50

BAB IV IMPLEMENTASI DAN PEMBAHASAN

A. Pengujian 62

B. Implementasi Tampilan *Web* 19

BAB V KESIMPULAN DAN SARAN

A. Kesimpulan 74

B. Saran 19

Daftar Tabel

Tabel 1. Perbandingan Kajian Penelitian Relevan (Lanjutan) 2
Tabel 2. Simbol–Simbol *Use Case Diagram* 10
Tabel 3. Simbol–Simbol *Sequence Diagram* 13
Tabel 4. Simbol–Simbol *Activity Diagram* 15
Tabel 5. Simbol–Simbol *Class Diagram* 16
Tabel 6. Deskripsi *User* 24
Tabel 7. Deskripsi *Login* mahasiswa 24
Tabel 8. Deksripsi mengisi KRS 25
Tabel 9. Deskripsi Kartu Hasil Studi 25
Tabel 10. Deskripsi *Login* Admin 26
Tabel 11. Deskripsi *Login* Akademik 26
Tabel 12. Hasil Pengujian Otentifikasi 54
Tabel 13. Hasil Pengujian Data oleh pengguna mahasiswa 56
Tabel 8. Hasil Pengujian Data oleh Pengguna Akademik 56
Tabel 9. Hasil Pengujian Data oleh Pengguna *Admin* 57

DAFTAR GAMBAR

Gambar 1. Cara Kerja *MVC* 3
Gambar 2. *Waterfall* 5
Gambar 3. *Dreamweaver CS 5* 7
Gambar 4. *Dynamic Draw5* 8
Gambar 5. *Use Case Diagram* 23
Gambar 6. *Activity Diagram Login* (Admin,Akademik,dan Mahasiswa) 28
Gambar 7. *Activty Diagram* Kelola Data Prodi (*Admin*) 29
Gambar 8. *Activty Diagram* Kelola Data Tahun Ajaran (*Admin*) 30
Gambar 9. *Activity Diagram* Kelola Data Kelas (*Admin*) 31
Gambar 10. *Activity Diagram* Kelola Data *User* (*Admin*) 32
Gambar 11. *Activity Diagram* Kelola Data Mata Kuliah (Akademik) 33
Gambar 10. *Activity Diagram* Kelola Data Dosen (Akademik) 34
Gambar 12. *Activity Diagram* Data Mahasiswa (Akademik) 34
Gambar 13. *Activity Diagram* Data KRS (Akademik) 35
Gambar 14. *Activity Diagram* Kelola Nilai (Akademik) 36
Gambar 15. *Activity Diagram* Input KRS (Mahasiswa) 37
Gambar 16. *Activity Diagram* Lihat KHS 38
Gambar 17. *Activity Diagram* Kelola Data Diri (Mahasiswa) 39
Gambar 18. *Sequence Diagram Login* 40
Gambar 19. *Sequence Diagram* Kelola Data User (*Admin*) 41
Gambar 20. *Sequence Diagram* Lihat Nilai (*Admin*) 42
Gambar 21. *Sequence Diagram* Lihat KRS (Akademik) 43
Gambar 22. *Sequence Diagram* Kelola Data Prodi (*Admin*) 44
Gambar 23. *Sequence Diagram* Kelola Data Mata Kuliah (Akademik) 44
Gambar 24. *Sequence Diagram* Kelola Data Dosen (Akademik) 45
Gambar 25. *Sequence Diagram* Kelola Data Mahasiswa (Akademik) 46
Gambar 26. *Sequence Diagram* Kelola Nilai (Akademik) 47
Gambar 27. *Sequence Diagram* Kelola KRS (Mahasiswa) 48
Gambar 28. *Sequence Diagram* Melihat Data Diri (Mahasiswa) 49

Gambar 29. *Sequence Diagram* Lihat KHS (Mahasiswa) 50
Gambar 30. *Interface* Sebelum *Login* 50
Gambar 31. *Halaman* Utama *Login* 57
Gambar 32. *Interface* sesudah *Login* 58
Gambar 33. *Interface* Program Studi 58
Gambar 34. *Interface* Semester 58
Gambar 35. *Interface* Data *User* 59
Gambar 36. *Interface* Data Mata Kuliah 59
Gambar 37. *Interface* mahasiswa 59
Gambar 38. *Interface* Data KRS 60
Gambar 39. *Interface* Lihat KHS Mahasiswa 60

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Sekolah Tinggi Ilmu Ekonomi (STIE) YBPK adalah Perguruan Tinggi Swasta yang terletak di Provinsi Kalimantan Tengah, Jalan H. Ikap No. 17 Palangka Raya, memiliki program pendidikan Sarjana dengan jurusan Akuntansi dan program pendidikan Diploma III Akuntansi. STIE YBPK adalah salah satu Perguruan Tinggi yang belum mempunyai sebuah Sistem untuk mengelola yang data, khususnya pengisian Kartu Rencana Studi (KRS) secara komputerisasi sehingga pengisian KRS dilakukan dengan cara manual. Hal ini mengakibatkan proses pengisian KRS oleh mahasiswa sering mengalami keterlambatan karena jadwal proses pengisian KRS tersebut dilakukan pada saat mahasiswa libur semester, kebanyakan mahasiswa belum mengetahui informasi tersebut.

Dengan adanya KRS dan KHS secara *online*, maka mahasiswa bisa menggunakan komputer atau laptop masing-masing dan sistem ini bisa digunakan tanpa keterbatasan tempat dan waktu asalkan terhubung jaringan *internet* untuk menyusun KRS. Pada proses pengisian KRS *online* mahasiswa harus memilih mata kuliah yang akan diambilnya, berdasarkan daftar nama mata kuliah dengan jadwal dan kelas yang dibuka. Selain proses KRS adapula proses KHS yang berisi hasil di semester sebelumnya, Pada KHS akan terdapat

IP yang didapatkan dan IP tersebut menentukan berapa SKS yang bisa diambil pada semester selanjutnya.

Hasil akhir yang diharapkan adalah dengan adanya proses KRS *online* ini menjadi lebih terstruktur dan untuk meningkatkan pelayanan. Sistem ini memiliki beberapa hal yang harus diperhatikan oleh Mahasiswa agar KRS *online* ini menjadi lebih nyaman dan kemudahan pada mahasiswa. Karena jika Mahasiswa yang mengulur-ngulur waktu untuk menyusun KRS maka dianggap Non-Aktif/Cuti.

Berdasarkan hal tersebut yang telah dikemukakan diatas, maka penulis mengangkat judul "**Rancang Bangun Kartu Rencana Studi (KRS) dan Kartu Hasil Studi (KHS) *Online* Menggunakan *Framework CodeIgniter* Studi Kasus : STIE YBPK Palangkaraya**".

B. Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas maka perumusan masalah yang muncul adalah Bagaimana membuat Aplikasi Kartu Rencana Studi (KRS) dan Kartu Hasil Studi (KHS) *Online* Pada Sekolah Tinggi Ilmu Ekonomi YBPK Palangkaraya Menggunakan *Framework CodeIgniter?*

C. Batasan Masalah

Dalam penelitian ini, penulis memberikan batasan agar tidak menyimpang dari tujuan guna mendapatkan hasil yang optimal. Batasan tersebut antara lain sebagai berikut :

1. Manajemen hak akses untuk aplikasi Kartu Rencana Studi (KRS) *Online* pada Sekolah Tinggi Ilmu Ekonomi YBPK Palangkaraya akan dibagi menjadi 3 (tiga) bagian, yaitu:
    a. **Admin,** berhak mengakses dan mengelola fitur *web* secara menyeluruh. yaitu, data *user*, data tahun ajaran, data semester dan data program studi.
    b. **Akademik**, memiliki hak akses yang terbatas dan hanya dapat mengakses, yaitu: Lihat KRS data mata kuliah, data dosen, data mahasiswa, data kelas, kelola KRS, dan KHS.
    c. **Mahasiswa**, memiliki hak akses yang terbatas dan hanya dapat mengakses, yaitu: Mengisi KRS, lihat KHS, dan data diri.
2. Aplikasi ini dibuat menggunakan *CodeIgniter*, *MySQL* sebagai *database*, dan *apache* sebagai *web server*.
3. Tidak membahas pembuatan jadwal mata perkuliahan yang sedang berjalan.
4. KRS dan KHS diprint out 4 rangkap dengan warna yang berbeda, yaitu :
    a. Kertas warna putih = Mahasiswa
    b. Hijau = Dosen PA
    c. Merah = Jurusan
    d. Biru = BAAK

D. Tujuan dan Manfaat

1. Tujuan

   Adapun tujuan dari penelitian ini adalah membuat Aplikasi KRS dan KHS *Online* agar dapat dipergunakan oleh mahasiswa dan pihak akademik lainnya.

2. Manfaat

   Manfaat yang diharapkan bermanfaat bagi kepentingan berbagai pihak antara lain:

   a. Bagi Sekolah Tinggi Ilmu Ekonomi YBPK Palangkaraya, adalah terbantunya dalam penyusunan KRS dan KHS secara *online* agar lebih terstruktur dan terkomputerisasi. Dengan adanya KRS dan KHS *online* bisa menghemat waktu.
   b. Bagi Penulis, diharapkan bermanfaat untuk mengembangkan ilmu yang didapat pada bangku kuliah dan mengembangkan ilmu dibidang teknologi pemrograman *Web*.
   c. Bagi STMIK Palangkaraya, Sebagai sumber informasi untuk meningkatkan mutu pendidikan dan menambah sumber pengetahuan di kampus STMIK Palangkaraya.

E. Metode Penelitian

Metode yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Tahap Pengumpulan Data

a. Observasi

Metode yang dilakukan secara langsung dengan cara survey ke tempat penelitian yaitu STIE YBPK Palangka Raya mengenai proses-proses yang sedang berjalan, wawancara secara langsung dengan yang bersangkutan, dan menganalisa permasalahan untuk memperoleh data sesuai dengan kebutuhan.

b. Studi Pustaka

Metode yang dilakukan dengan mengumpulkan mempelajari buku-buku atau referensi lainnya yang ada sangkut pautnya tentang pembuatan skripsi ini di perpustakaan STMIK Palangkaraya.

c. Wawancara

Pengumpulan data melalui tatap muka dan tanya jawab langsung dengan pihak-pihak yang berkepentingan yang berhubungan dengan penelitian.

2. Tahap Pengembangan Sistem

Pengembangan perangkat lunak dalam pembuatan aplikasi ini menggunakan *Waterfall* (Air Terjun). *Waterfall* ini dikenal dengan air terjun karena penurunan dari satu tahap ke tahap lainnya, tahap berikutnya tidak boleh dimulai sebelum tahap sebelumnya selesai. Adapun tahap-tahap dari ini seperti:

1) *Requirement Analisis*

Sebelum membuat sebuah penelitian tugas akhir, sangat diperlukan adanya komunikasi dengan pihak STIE YBPK dan diimplementasikan dalam sebuah *web*. Hasil dari komunikasi tersebut, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*.

2) *System Design*

Bentuk desain menggunakan *Unifed Modeling Language* (UML) dan perancangan desain sistem menggunakan aplikasi *Dynamic Draw 5. Use case* ini dibagi menjadi dua, untuk memudahkan dalam membaca *use case* diagramnya.

3) *Implementation*

Pada tahap ini, menyiapkan semua perangkat keras dan perangkat lunak yang diperlukan. Kemudian seluruh data yang didapatkan diimplementasikan ke dalam sebuah *web* untuk menjadikan sebuah informasi menggunakan perangkat lunak dan perangkat lunak yang sudah disediakan. Bahasa pemrograman yang digunakan adalah *CodeIgniter* (*CI)* merupakan *framework PHP* untuk membuat sebuah halaman *web* dan pemeriksaan program sangatlah penting.

4) Integration & Testing

Pada tahap ini seluruh data sudah dimasukkan dalam sebuah aplikasi *web* dan memiliki *database* untuk menyimpan data-data yang didapat.

Adanya pengujian (*Testing*) pada aplikasi *web* menggunakan *blackbox* untuk mengetahui kelebihan dan kelemahan pada aplikasi yang telah dibuat.

5) *Operation & Maintenance*

Tahap akhir dalam *waterfall*. Data yang didapat telah mejadi sebuah aplikasi *web* dan dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang belum ada pada tahap sebelumnya. Pemeliharaan ini hendaknya berhubungan dengan pihak yang terkait (STIE YBPK Palangkaraya).

F. Sistematika Penulisan

Untuk memudahkan dalam memahami laporan tugas akhir, penulis mengemukakan sistematika penulisan yang terdiri dari 5(Lima) bab, yaitu :

BAB I PENDAHULUAN

Pada bab ini berisi tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat, metode penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini berisi tentang definisi, karakteristik sistem, konsep pemodelan sistem dan perangkat lunak yang digunakan.

BAB III ANALISA DAN DESAIN SISTEM

Pada bab ini berisi tentang tinjauan umum objek penelitian, analisis kelemahan sistem, analisis kebutuhan, analisis kelayakan, dan desain proses.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini berisi tentang implementasi Aplikasi, uji aplikasi, manual program, pembahasan listing program, pembahasan interface dan pembahasan respon hasil pengguna.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan yang didapat dari pembahasan pada bab-bab sebelumnya dan saran-saran yang dianggap perlu diperhatikan sehubungan dengan implementasi alat dan pengembangan untuk kedepannya.

BAB II
LANDASAN TEORI

A. Tinjauan Pustaka

Untuk menghindari duplikasi dari pihak lain, penulis melakukan penelusuran terhadap penelitian-penelitian terdahulu. Dari hasil penelusuran penelitian terdahulu, diperoleh beberapa masalah yang berkaitan dengan masalah yang akan diteliti, yaitu :

1. Penelitian yang dilakukan Siti, dkk. (2013) yang berjudul "Pembuatan Sistem Informasi Kartu Rencana Studi (KRS) Dan Kartu Hasil Studi (KHS) Pada Program Studi Teknik Informatika Universitas Surakarta" penelitian ini bertujuan untuk membuat suatu sistem informasi KRS dan KHS yang dibangun dengan menggunakan bahasa pemrograman *PHP* dan menggunakan *database MySQL*. Melihat permasalahan yang ada maka perlu mengembangkan sistem informasi pada pengelolaan data akademik mahasiswa mulai dari pengisian KRS sampai penilaian KHS mahasiswa Surakarta.
2. Penelitian yang relevan yang dilakukan Nugroho (2015) yang berjudul "Rancang Bangun KRS STMIK Palangka Raya menggunakan *Framework Codeigniter*" penelitian ini membahas tentang dalam pengisian KRS tanpa harus mengantri pada loket pengisian KRS, karena sistem KRS telah *online* dan mahasiswa dapat mengisi KRS tanpa harus dibatasi oleh waktu dan tempat. Data mahasiswa yang digunakan adalah angkatan tahun 2014.

3. Penelitian yang relevan yang dilakukan Hasan (2017) yang berjudul "Sistem Informasi Pengisian Kartu Rencana Studi (KRS) Berbasis *Web* Pada Politeknik Sains Dan Teknologi Wiratama Maluku Utara". Dengan adanya aplikasi KRS berbasis *web* untuk memudahkan mahasiswa dalam proses pengisian Kartu Rencana Studi (KRS) tanpa harus datang mengantri di kampus, juga memudahkan bagi pihak kampus dalam mengontrol dan memantau data jumlah mahasiswa yang aktif pada semester berjalan.

Berikut adalah tabel perbandingan kajian penelitian relevan yang penulis gunakan kemudian dengan melakukan perbandingan dari hasil penelitian yang sudah ada seperti pada tabel 1.

Tabel 1. Perbandingan Kajian Penelitian Relevan

No	Nama dan Tahun	Metode, Bahasa Pemograman	Hasil	Perbedaan dan Persamaan
1	2	3	4	5
1	Siti, dkk. (2013)	*Waterfall, Processing Hypertext Programming* (PHP)	Hasil penelitian ini adalah sistem informasi KRS dan KHS Pada Program Studi Teknik Informatika Universitas Surakarta yang dapat memudahkan mahasiswa dalam pengisian KRS dan mempermudah kinerja admin dalam pengolahan data KRS dan KHS.	Perbedaan : Aplikasi KRS dan KHS *Online* pada STIE YBPK Palangkaraya perancangan desain sistem yang digunakan *Unifed Modeling Language* (UML). Sedangkan sistem informasi KRS dan KHS Pada Program Studi Teknik Informatika Universitas Surakarta perancangan desain sistem yang digunakan adalah *Data Flow Diagram (DFD).* Persamaan : Aplikasi ini menggunakan *waterfall.*
2	Nugroho (2015)	*Waterfall, Framework Codeigniter*	Dengan adanya aplikasi KRS online pada STMIK Palangkaraya ini memberikan kemudahan kepada mahasiswa dalam mengisi KRS tanpa harus dibatasi oleh waktu dan tempat.	Perbedaan : Aplikasi KRS dan KHS *Online* pada STIE YBPK Palangkaraya bisa diakses dengan *internet* dan desain sistem yang digunakan *Dynamic Draw 5* sedangkan aplikasi KRS *Online* Pada STMIK Palangka Raya hanya dapat diakses dengan *intranet* dan desain sistem yang digunakan adalah *Dia Diagram Editor.* Persamaan: Aplikasi ini menggunakan *waterfall* dan *framework CodeIgniter.*

Tabel 1. Perbandingan Kajian Penelitian Relevan (Lanjutan)

No	Nama dan Tahun	Metode, Bahasa Pemograman	Hasil	Perbedaan dan Persamaan
3	Hasan (2017)	*Waterfall, Processing Hypertext Programming* (PHP)	Sistem Informasi Pengisian Kartu Rencana Studi (KRS) Berbasis *Web* dapat memudahkan mahasiswa untuk melakukan proses pengisian (KRS) tanpa harus datang langsung mengantri di kampus dan memudahkan pihak kampus dalam pengelolaan data akademik khususnya dapat mengontrol dan memantau data jumlah mahasiswa yang aktif dalam semester berjalan.	Perbedaan : Aplikasi KRS dan KHS *Online* pada STIE YBPK Palangkaraya alur sistem yang digunakan adalah *UML (Unifed Modeling Language)* sedangkan Aplikasi Sistem Informasi Pengisian Kartu Rencana Studi (KRS) menggunakan *Entity Relationship Diagram (ERD).* Persamaan : Aplikasi ini menggunakan *waterfall.*

B. Kajian Teori

1. *Framework*

   Framework secara sederhana adalah kumpulan instruksi-instruksi yang dikumpulkan dalam *class* dan *function-function* untuk tujuan tertentu sehingga memudahkan dan mempercepat pekerjaan seorang programer, dalam pemanggilannya tanpa harus menuliskan *syntax* program yang sama berulang-ulang agar *source code* lebih rapi dan terstuktur.

2. *CodeIgniter*

   CodeIgniter adalah sebuah *Framework PHP* yang bersifat *Open Source* dengan Model *MVC (Model, View, Controller)* untuk membuat *web* dinamis dengan menggunakan *PHP*. *CodeIgniter* memudahkan *Developer* untuk membuat aplikasi *web* dengan cepat dan mudah.

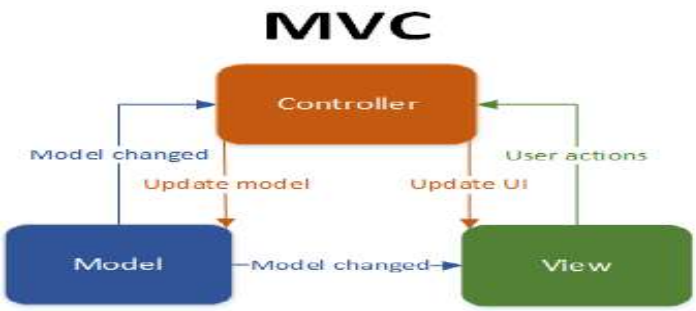
   Model, View, Controller merupakan suatu konsep yang cukup populer dalam pembangunan *web*, berawal pada bahasa pemrograman *Small Talk MVC* memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu aplikasi yaitu :

   a) *View*, merupakan bagian yang menangani halaman *user interface* atau halaman yang muncul pada *user*. Tampilan dari *user* interface dikumpulkan pada *view* untuk memisahkannya dengan *controller* dan *model* sehingga memudahkan *web designer* dalam melakukan

pengembangan tampilan halaman *web*. *View* berfungsi untuk menerima dan merepresentasikan data hasil dari *model* dan *controller* kepada *user*. View tidak memiliki akses langsung terhadap bagian *model*.

b) *Model* merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi *database*, misalnya mengambil data dari *database*, menginput dan pengolahan *database* lainnya. Semua intruksi yang berhubungan dengan pengolahan *database* diletakkan didalam *model*. *Model* juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file *XML* maupun *webservice*. Biasanya didalam *model* akan berisi *class* dan fungsi untuk mengambil, melakukan *update* dan menghapus data *web*. Sebuah aplikasi *web* biasanya menggunakan basis data dalam menyimpan data, maka pada bagian *Model* biasanya akan berhubungan dengan perintah-perintah *query SQL*.

c) *Controller*, merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*. Pada *controller* terdapat *class-class* dan fungsi-fungsi yang memproses permintaan dari *View* ke dalam struktur data di dalam *model*. *Controller* juga tidak boleh berisi kode untuk mengakses basis data Karena tugas megakses data telah diserahkan kepada *model*. Tugas *controller* adalah menyediakan berbagai variable yang akan ditampilkan di *view*, memanggil *model* untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.



Gambar 1. Cara Kerja *MVC*

Dengan menggunakan prinsip *MVC* suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developernya*. *Controller* sebagai penghubung antara *view* dan *model*. Pada aplikasi yang menampilkan data dengan menggunakan metode konsep *MVC*, *controller* memanggil instruksi pada *model* yang mengambil data pada *database*, kemudian *controller* yang melanjutkan pada *view* untuk ditampilkan. Pengembangan aplikasi dengan menggunakan *MVC* sangat mudah karena *web designer* untuk mendesign tampilan aplikasi dan pengembangan aplikasi dapat dilakukan dengan cepat.

Adapun beberapa kelebihan dari *CodeIgniter* (CI) yaitu :

1) Konfigurasi Yang Sangat Minim (*Nearly Zero Configuration*)

   Untuk menyesuaikan dengan *database* dan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan *codeIgniter* dengan setting standard, anda hanya perlu mengubah sedikit saja file pada folder *config*.

2) Performa Tercepat

   Kontruksi yang simpel dan dikelompokan berdasarkan fungsi dan tugas prosesnya masing-masing membuat peforma pemrosesan lebih maksimal.

3) Dokumentasi yang Baik

   Pada pengembangan sebuah aplikasi penting sekali adanya dokumentasi *framework*. Setiap paket instalasi *codeIgniter* sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

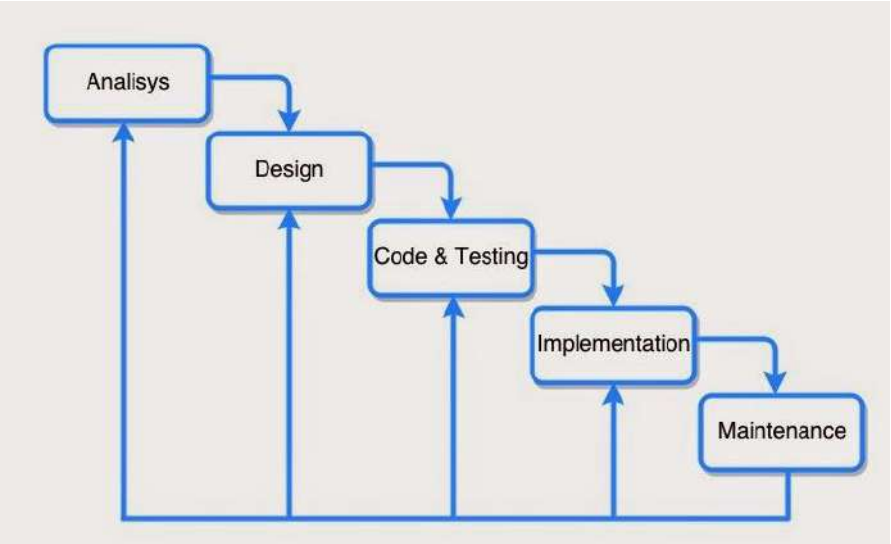
3. *WEB*

Website pertama kali ditemukan oleh Sir Timothy John, Tim Berners-Lee. Pada tahun 1991 *website* terhubung dengan jaringan. Tujuan dari dibuatnya *website* pada saat itu yakni untuk mempermudah tukar menukar dan memperbaharui informasi kepada sesama peneliti di tempat mereka bekerja.

World Wide Web, yang lebih populer disingkat dengan *WWW* merupakan satu buah ruang informasi yg digunakan oleh pengenal global yang dinamakan dengan *URL (Uniform Resource Locator)* untuk mengenal pasti sumber daya berguna. *WWW* merupakan kumpulan situs *server* dari seluruh dunia yang memiliki kegunaan untuk menyediakan data dan info untuk bisa digunakan bersama. *WWW* adalah yang paling menarik dari *Internet*. Dengan adanya *web*, semua pengguna bisa terhubung informasi-informasi yang tidak hanya berupa teks namun mampu pula berupa gambar, nada, video dan animasi.

4. *Waterfall*

Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012).



Gambar 2. *Waterfall*

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu: *requirement* (analisis kebutuhan), design system (desain sistem), *Coding* (pengkodean) & *Testing* (pengujian), Penerapan Program, pemeliharaan.

5. *PHP (Hypertext Prepocessor)*

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. Selain itu, *PHP* juga bisa digunakan sebagai bahasa pemrograman umum. *PHP* di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. *PHP* merupakan bahasa pemrograman paling populer yang digunakan dalam membuat sebuah *website*.

6. *HyperText Markup Language (HTML)*

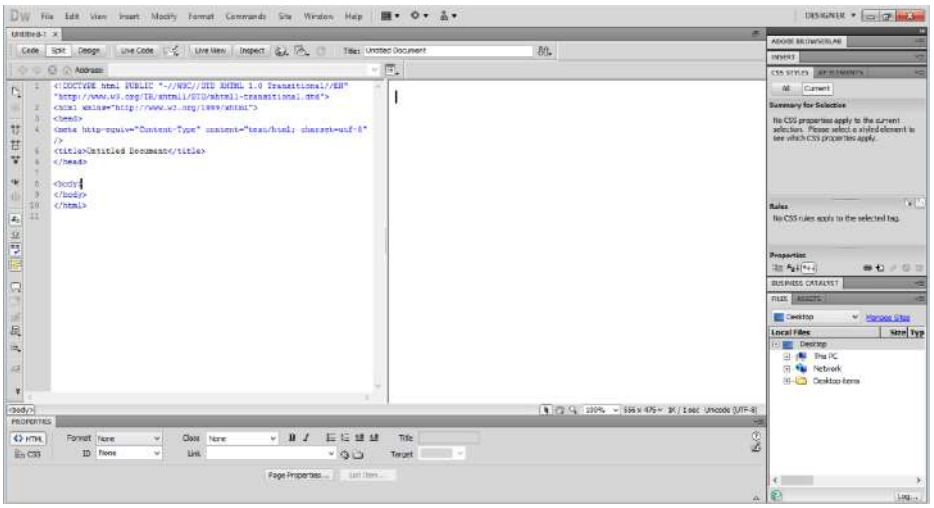
HTML merupakan salah satu bahasa yang paling banyak digunakan dalam mebuat halaman *web*. *Hypertext* mengacu pada cara di mana halaman *web* (dokumen *HTML*) dihubungkan. Link yang tersedia pada halaman *web* disebut *Hypertext*. Seperti namanya, *HTML* adalah bahasa *Markup* yang berarti kamu menggunakan *HTML* hanya untuk "*mark-up*" dokumen teks dengan tag yang akan memberitahukan *browser* struktur untuk menampilkan sebuah desain *layout web.*

7. *Dreamweaver*

Aplikasi ini merupakan editor yang sangat hebat dimana kemampuan *debugging*-nya teruji dan memudahkan untuk membuat

aplikasi dengan *PHP*. *Dreamweaver* memiliki kemampuan untuk mengkoreksi sintaks yang salah dan selalu akan menampilkan *code PHP*. *Dreamweaver* merupakan software utama yang digunakan oleh *Web* Desainer dan *Web Programmer* dalam mengembangkan suatu situs *web*.

Dalam perkembangannya, *Adobe Dreamweaver* telah mencapai versinya yang terbaru atau lebih dikenal dengan *Adobe Dreamweaver CS5*. Fitur-fitur yang dimiliki semakin lengkap dan handal, untuk membuat pengguna *Dreamweaver CS5* semakin dapat berkreasi dan berinovasi dengan bebas dalam mendesain *web*.

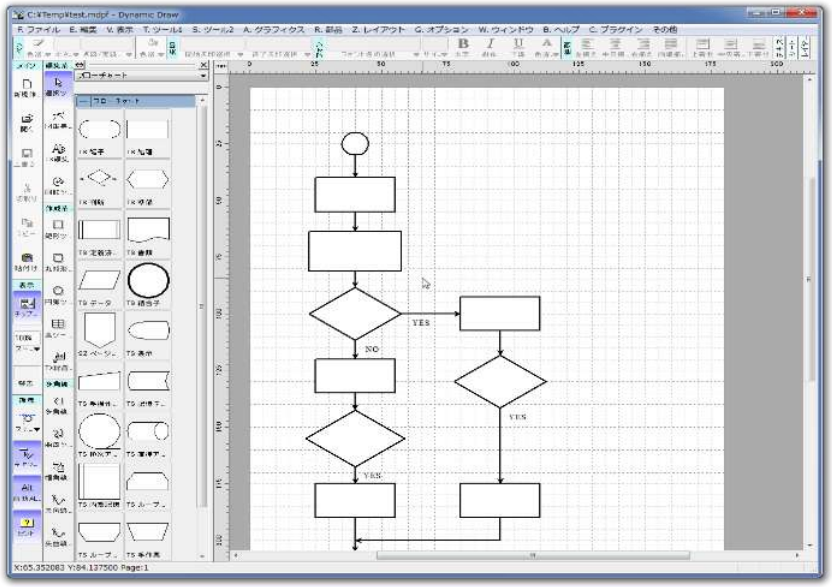


Gambar 3. *Dreamweaver CS 5*

8. *Dynamic Draw*

Dynamic Draw adalah *editor grafis vektor* yang berguna untuk membuat *flowcharts, line drawings, org charts, buttons,* dan lain sebagainya, baik menggambar *object-oriented* dan desain program dengan sistem yang menghubungkan begitu akurat. *Software* ini pun dapat

mengekspor ke format vektor grafis lainnya seperti *SVG* maupun *non-grafis vektor format seperti JPG, TIFF, PNG, EMF, WMF, PSD, BMP,* dan *PSD.*



Gambar 4. *Dynamic Draw5*

9. *Web Server*

Web server adalah sebuah *software* yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari *HTTP* atau *HTTPS* pada klien yang dikenal dengan nama *web browser* (*Mozilla Firefox, Google Chrome)* dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman *web* dan pada umumnya akan berbentuk dokumen *HTML*. Salah satu contoh dari *Web Server* adalah *Apache. Apache (Apache Web Server – The HTTP Web Server)* merupakan *web server* yang paling banyak dipergunakan di *Internet*. *Apache* mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya.

10. *MySQL*

Pada perkembangannya, *MySQL* disebut *SQL* yang merupakan kepanjangan dari *Structured Query Language. SQL* merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. *MySQL* adalah sebuah sistem manajemen database yang bersifat *open source*.

MySQL merupakan sistem manajamen *database* yang bersifat relational. Hal ini berarti semua data yang dikelola dalam *database* akan diletakkan dalam beberapa *table* yang terpisah, sehingga manipulasi data akan menjadi jauh lebih cepat.

11. *Database*

Database adalah struktur penyimpanan data, untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah *database* komputer, diperlukan sistem manajemen *database* seperti *MYSQL Server*. Dengan demikian *Database* dapat disimpulkan kumpulan dari seluruh data yang berbasis komputerisasi pada suatu perusahaan/organisasi, atau kumpulan data yang berada di bawah kendali *software* sistem manajemen *database*.

12. *UML (Unifed Modeling Language)*

UML merupakan sintaks umum untuk membuat model logika dari suatu sistem dan digunakan untuk menggambarkan sistem agar dapat dipahami selama fase analisis dan desain. *UML* biasanya disajikan dalam bentuk diagram atau gambar yang meliputi *class* beserta atribut dan operasinya, serta hunbungan antar class yang meliputi *inheritance*, association dan komposisi. *UML* terdiri dari banyak diagram antara lain sebagai berikut (Sugiarti, 2013).

a. *Use Case Diagram*

Use case diagram adalah gambaran dari beberapa atau seluruh aktor dan *use case* dengan tujuan mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditentukan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interkasi antara aktor dengan sistem.

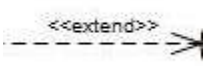
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama *pada use case* yaitu pendefinisian apa yang disebut aktor dan *use case* yaitu:

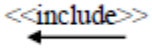
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (Sugiarti, 2013).

Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel 2. Simbol–Simbol *Use Case Diagram*
(sumber : Sugiarti, 2013)

Nama	**Simbol**	**Deskripsi**
1	**2**	**3**
Use Case		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar

Nama	Simbol	Deskripsi
1	**2**	**3**
		pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di frase nama *Use Case.*
Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor hanya memberikan informasi ke sistem, aktor hanya menerima informasi dari sistem, aktor memberikan dan menerima informasi ke sistem dan dari sistem.
Asosiasi		Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. Asosiasi merupakan hubungan statis antar elemen yang menggambarkan elemen yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain.
Ekstensi		Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walaupun tanpa *use case* tambahan itu, mirip dengan

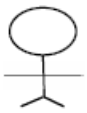
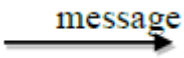
Nama	Simbol	Deskripsi
1	**2**	**3**
		prinsip *inheritance* pada pemrograman berorientasi objek. Biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan. Misalnya arah panah mengarah pada *use case* yang ditambahkan, biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.
Generalisasi		Hubungan *generalisasi* dan spesialisasi (umum-khusus) anatara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya, misalnya : arah panah mengarah pada *use case* yang menjadi generalisasinya (umum). *Generalisasi* merupakan hubungan hirarkis antara elemen. Elemen dapat mewarisi semua atribut dan metode elemen asalnya dan menambah fungsionalitas baru.
Include		Relasi *use case* tambahan ke sebuah use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat.

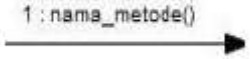
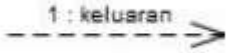
b. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar termasuk pengguna, *display*, dan sebagainya berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antara dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima objek. Oleh karena itu, untuk menggambarkan *sequence diagram* harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta dengan metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use case* (Sugiarti, 2013).

Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel 3. Simbol–Simbol *Sequence Diagram*
(sumber : Sugiarti, 2013)

Nama	**Simbol**	**Deskripsi**
1	**2**	**3**
Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Lifeline		Menyatakan kehidupan suatu objek, untuk menggambarkan kelas dan objek.
Objek		Menyatakan objek yang berinteraksi (pesan).
Waktu Aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif.

Nama	Simbol	Deskripsi
1	2	3
Pesan tipe *call*		Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, sesuai dengan kelas objek yang berinteraksi.
Pesan tipe *return*		Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

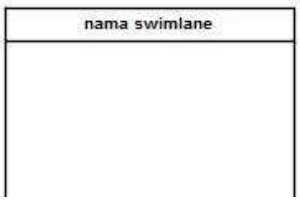
c. *Activity Diagram*

Activity diagram menggambarkan rangkaian alir aktivitas dalam sistem yang dirancang, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi. *Activity diagram* berupa *flowchart* yang digunakan untuk memperlihatkan alir kerja dari sistem. Diagram aktivitas juga banyak digunakan dalam mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak (Sugiarti, 2013).

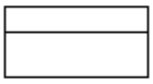
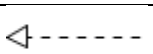
Tabel 4. Simbol–Simbol *Activity Diagram*
(sumber : Sugiarti, 2013)

Nama	**Simbol**	**Deskripsi**
1	**2**	**3**
Status Awal		Status awal aktivitas sistem, sebauh aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Penggabungan / *join*		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

d. Class Diagram

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Diagram kelas mendefinisikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka (Sugiarti, 2013).

Tabel 5. Simbol–Simbol *Class Diagram*
(sumber : Sugiarti, 2013)

Nama	**Simbol**	**Deskripsi**
1	**2**	**3**
Generalization		Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
Nary Association		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
Class		Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
Collaboration		Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
Realization		Operasi yang benar-benar dilakukan oleh suatu objek.
Dependency		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
Association		Apa yang menghubungkan antara objek satu dengan objek lainnya.

13. *Black – box Testing*

Black – box Testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *Black Box* dapat dilakukan untuk memverifikasi apakah perilaku komponen memang seperti yang diharapkan tanpa harus mempertimbangkan bagaimana perilaku tersebut secara internal diimplementasikan. Pengujian pada jenis ini terutama melihat pada apakah komponen itu diberi asupan tertentu serta berjalan dalam *state* tertentu.

Pengujian *Black-Box* dilakukan untuk menentukan beberapa macam kesalahan, yaitu:

1. Fungsi – fungsi yang tidak benar atau hilang,
2. Kesalahan *Interface*.
3. Kesalahan dalam struktur data atau akses *database eksternal*,
4. Kesalahan Kinerja,
5. Inisialisasi dan kesalahan terminasi

BAB III

ANALISIS DAN PERANCANGAN SISTEM

A. Tinjauan Umum

Sekolah Tinggi Ilmu Ekonomi (STIE) YBPK adalah Perguruan Tinggi Swasta yang terletak di Provinsi Kalimantan Tengah, Jalan H. Ikap No. 17 Palangka Raya, memiliki program pendidikan Sarjana dengan jurusan Akuntansi dan program pendidikan Diploma III Akuntansi. STIE YBPK adalah salah satu Perguruan Tinggi yang belum mempunyai sebuah Sistem untuk mengelola yang data, khususnya pengisian Kartu Rencana Studi (KRS) secara komputerisasi sehingga pengisian KRS dilakukan dengan cara manual. Hal ini mengakibatkan proses pengisian KRS oleh mahasiswa sering mengalami keterlambatan karena jadwal proses pengisian KRS tersebut dilakukan pada saat mahasiswa libur semester, kebanyakan mahasiswa belum mengetahui informasi tersebut.

Dengan adanya KRS dan KHS secara *online*, maka mahasiswa bisa menggunakan komputer atau laptop masing-masing dan sistem ini bisa digunakan tanpa keterbatasan tempat dan waktu asalkan terhubung jaringan internet untuk menyusun KRS. Pada proses pengisian KRS *online* mahasiswa harus memilih mata kuliah yang akan diambilnya, berdasarkan daftar nama mata kuliah dengan jadwal dan kelas yang dibuka. Selain proses KRS adapula proses KHS yang berisi hasil di semester sebelumnya, Pada KHS akan terdapat

IP yang didapatkan dan IP tersebut menentukan berapa SKS yang bisa diambil pada semester selanjutnya.

Seiring dengan meningkatnya kemajuan teknologi sekarang maka penulis dengan memanfaatkan teknologi tersebut membuat aplikasi yang memberi informasi kepada masyarakat tentang aplikasi KRS dan KHS pada STIE YBPK Palangkaraya. Maka dengan ini penulis membuat tugas akhir dengan judul Rancang Bangun Kartu Rencana Studi (KRS) dan Kartu Hasil Studi (KHS) *Online* Menggunakan *Framework CodeIgniter* Studi Kasus : STIE YBPK Palangkaraya.

B. Analisis

1. Analisis Kelemahan Sistem

   Pada saat ini STIE YBPK Palangka Raya masih memakai metode manual dengan menggunakan *Microsoft Exel* Dalam melakukan pembuatan Kartu Rancang Studi (KRS) dan Kartu Hasil Studi sehingga proses pengerjaan memerlukan waktu yang lama, kurang akurat, dan tidak efisien.

2. Analisis Kebutuhan

   a. Kebutuhan Perangkat Keras

   Perangkat Keras yang dibutuhkan dalam pembuatan program ini yaitu :

   1) Type : ACER Swift 3 SF314-56G (Core i5-8265U)

   2) Processor : Intel Core i5-8265U

3) RAM : 4GB DDR4

4) LCD : 14 Inch FHD

5) Harddisk : 1TB

b. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam pembuatan program ini yaitu :

1) XAMPP 5.5.38

2) *Adobe Dreamweaver cs5*

3) *Framework Codeigniter*

4) Mozilla Firefox

c. Kebutuhan Informasi

Dalam Program ini ada 2 jenis output/laporan yaitu laporan Kartu Rancang Studi (KRS) dan Kartu Hasil Studi (KHS).

d. Kebutuhan pengguna

Website ini digunakan oleh mahasiswa untuk mengisi KRS dan melihat KHS secara *online*. *Administrator* yang mengelola *website* hanya satu orang saja tapi tidak menutup kemungkinan dikemudian hari dapat menambah *administrator* untuk membantu mengelola *website*nya dan mendaftar serta menyetujui syarat-syarat yang berlaku pada *website*.

3. Analisis Kelayakan Sistem

Tahapan ini menguraikan analisis kelayakan sistem baru jika di implementasikan.

a. Kelayakan Teknologi

Kelayakan teknologi yang digunakan untuk membuat atau merancang aplikasi ini sudah menggunakan berbasis *web* dimana pengguna nantinya yang dapat menggunakan ini tidak perlu khawatir dengan jenis sistem operasi yang berbeda-beda pada PC atau Laptop mereka. Karena dengan mempunyai *browser* yang mendukung *HTML*, pengguna sudah dapat mengisi KHS dan melihat KRS.

b. Kelayakan Hukum

Untuk kelayakan hukum aplikasi ini semua menggunakan aplikasi yang *open source (free)* sehingga tidak ada masalah dengan pelanggaran atau *software* bajakan. Aplikasi yang dimaksud dalam pembuatan aplikasi ini seperti *Adobe Dreamweaver cs5*, *Xampp*, dan *Mozilla Firefox*

c. Kelayakan Ekonomi

Aplikasi *web* KRS dan KHS pada STIE YBPK menggunakan *Framework Codeigniter* ini memberi dampak bagi pada STIE YBPK di kota Palangkaraya karena dengan adanya aplikasi ini pengguna lebih menghemat waktu dan tempat untuk mengisi KRS dan melihat KHS.

C. Desain Sistem

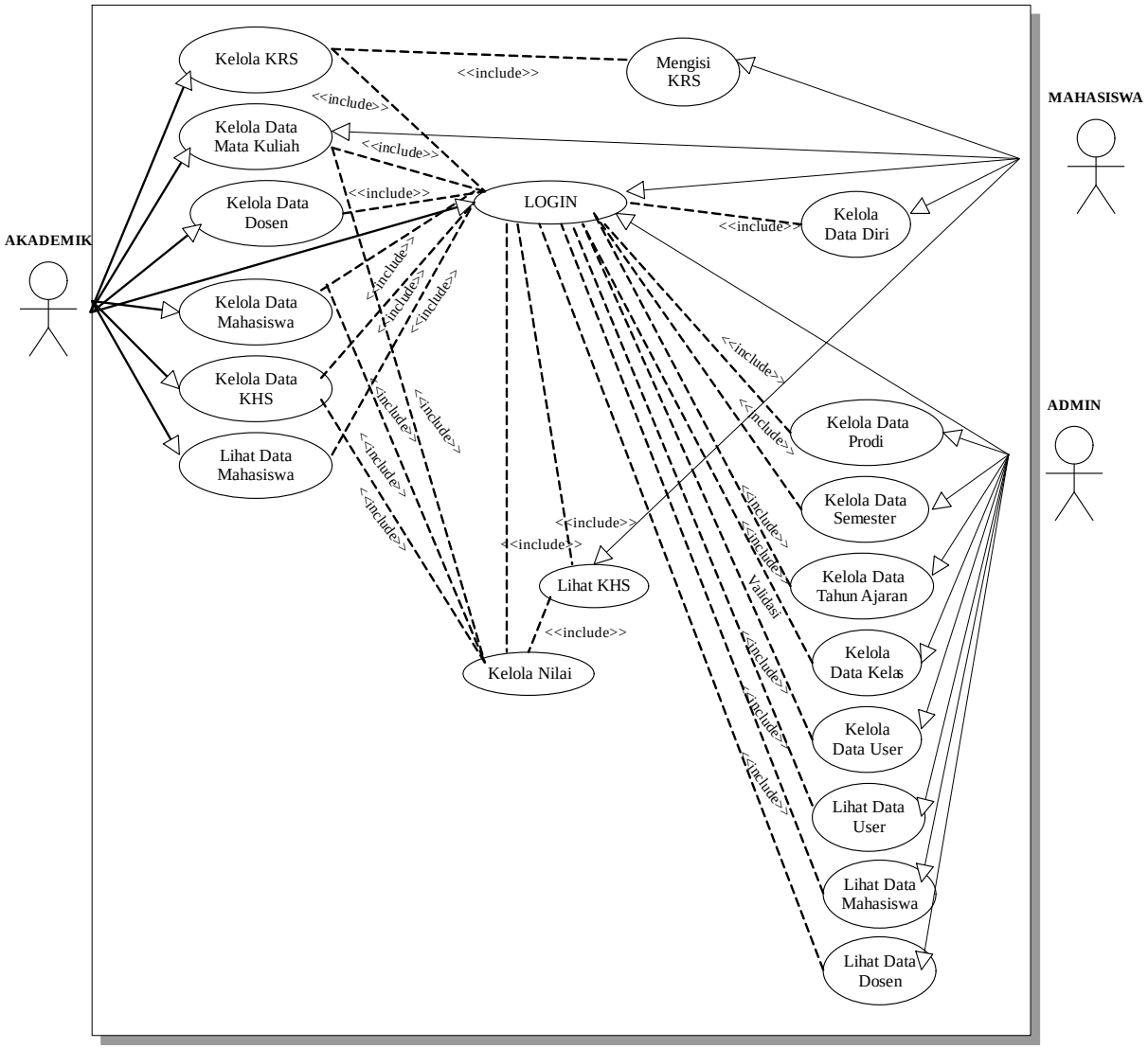
Berdasarkan pemodelan sistem yang penulis gunakan, maka penulis menjelaskan tentang *Unfield Modelling Language (UML).*

1. **Desain Proses**

a. Use Case Diagram

Use Case Diagram adalah salah satu jenis diagram pada *UML* (*Unfield Modelling Language*). *Use Case Diagram* sendiri didalam *UML* termasuk didalam kategori *Behavior Diagram*. *Behavior Diagram* adalah pemodelan yang digunakan untuk menggambarkan kelakuan sistem dari *website* yang dibuat. *Use Case Diagram* dibuat untuk menggambarkan proses sistem atau kebutuhan sistem dari sudut pandang *user* atau pengguna *Use case.*

Berikut adalah *Use Case* dari *website* menggunakan *Framework Codeigniter*.



Gambar 5. *Use Case Diagram*

Tabel 6. Deskripsi *User*

Aktor	**Deskripsi**
Admin	Memiliki hak akses yang terbatas dan hanya dapat mengakses, yaitu: kelola data prodi, kelola data semester, data tahun ajaran, data kelas, data user, melihat laporan user, melihat laporan mahasiswa, dan melihat laporan dosen.
Akademik	Akademik di sini memiliki 2 akun, yaitu akun jurusan akuntasi jenjang D3 dan S1. Memiliki hak akses yang terbatas dan hanya dapat mengakses, yaitu: kelola KRS, kelola mata kuliah, kelola data dosen, kelola data mahasiswa, KHS, dan melihat laporan data mahasiswa.
Mahasiswa	memiliki hak akses yang terbatas dan hanya dapat mengakses, yaitu: mengisi KRS, lihat KHS, dan data diri.

1) *Login* mahasiswa

Use case title : *Login* mahasiswa

Tabel 7. Deskripsi *Login* mahasiswa

Actor Action	*System Response*
1	2
Mengisi *Username* dan *Password*	-

Menekan Tombol *Login*	Proses *username* dan *password* di *database*

2) Mengisi KRS

Use Case Title : Mengisi KRS

Tabel 8. Deksripsi mengisi KRS

Actor Action	*System Response*
1	2
Menekan tombol tambah	Menambah Mata Kuliah yang ingin diambil
Menekan tombol pilih	Menampilkan mata kuliah yang telah dipilih
Menekan tombol hapus	Menghapus mata kuliah yang telah dipilih
Menekan tombol print	Men*download* KRS

3) Kartu Hasil Studi

Use case title : Kartu Hasil Studi

Tabel 9. Deskripsi Kartu Hasil Studi

Actor Action	System Response

1	2
Menekan tombol Kartu Hasil Studi	Menampilkan form Kartu Studi Mahasiswa

4) *Login* Admin

Use case title : *Login* Admin

Tabel 10. Deskripsi *Login* Admin

Actor Action	System Response
1	2
Mengisi *Username* dan *Password*	-
Menekan Tombol *Login*	Proses *username* dan *password* di *database*

5) *Login* Akademik

Use case title : *Login* Akademik

Tabel 11. Deskripsi *Login* Akademik

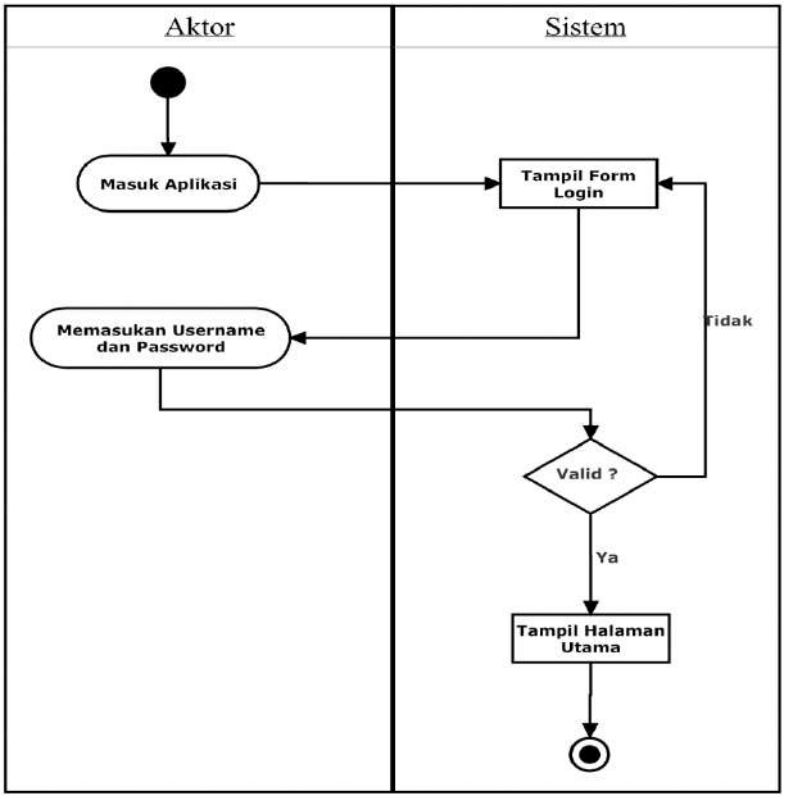
Actor Action	System Response
1	2
Mengisi *Username* dan *Password*	-

Menekan Tombol *Login*	Proses *username* dan *password* di *database*

b. *Activity Diagram*

Activity diagram menggambarkan rangkaian alir aktivitas dalam sistem yang dirancang, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti use case atau interaksi. *Activity diagram* berupa *flowchart* yang digunakan untuk memperlihatkan alir kerja dari sistem.

1. *Activity Diagram Login* (Admin,Akademik,dan Mahasiswa)

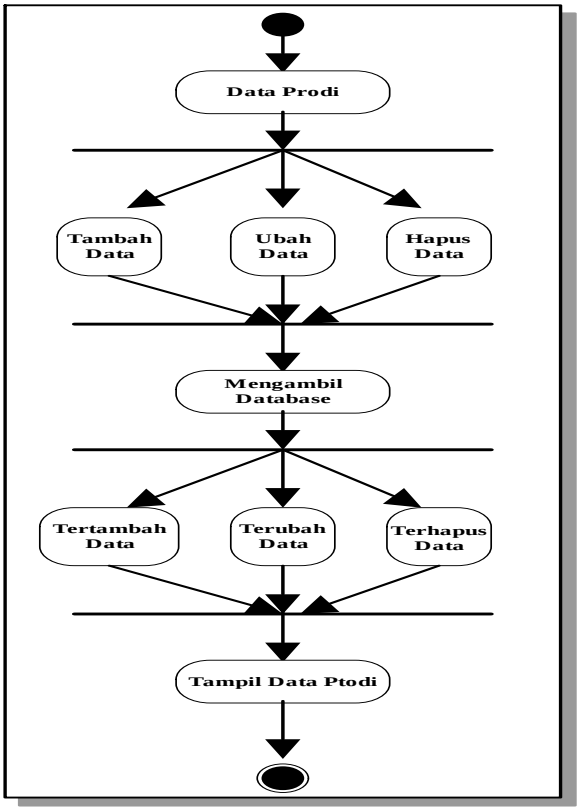


Gambar 6. *Activity Diagram Login* (Admin,Akademik,dan Mahasiswa)

Diagram Activity Login admin, akademik, dan mahasiswa adalah dimana pengguna jika ingin masuk ke aplikasi harus mengisi *username* dan *password.* kemudian menekan tombol masuk selanjutnya sistem akan verifikasi *username* dan *password* di *database,* jika benar maka pengguna akan masuk ke halaman utama *form* akademik atau mahasiswa, jika salah maka akan kembali ke halaman *login*.

2. *Activity Diagram* Kelola Data Prodi (*Admin*)

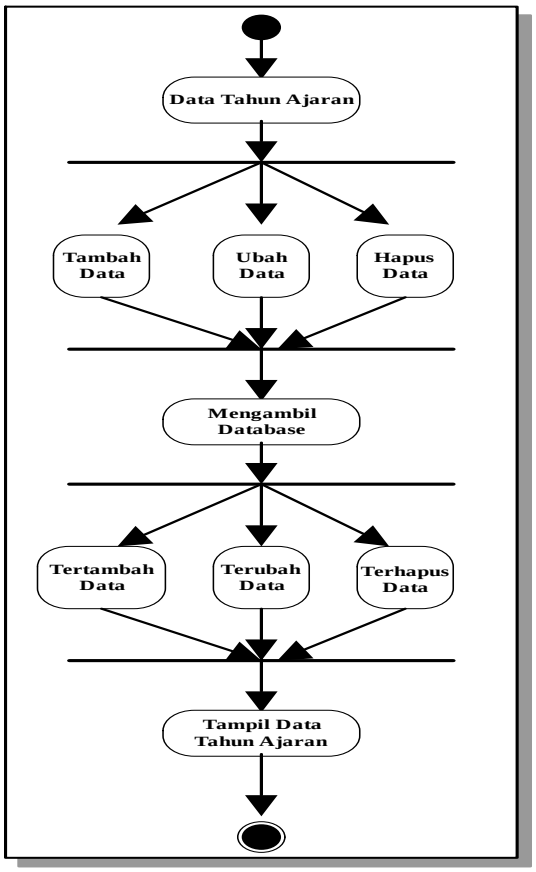
Diagram Activity kelola data prodi *(Admin)* adalah dimana pada menu data program studi ada tombol tambah, ubah, dan hapus data.



Gambar 7. *Activty Diagram* Kelola Data Prodi (*Admin*)

3. *Activity Diagram* Kelola Data Tahun Ajaran (*Admin*)

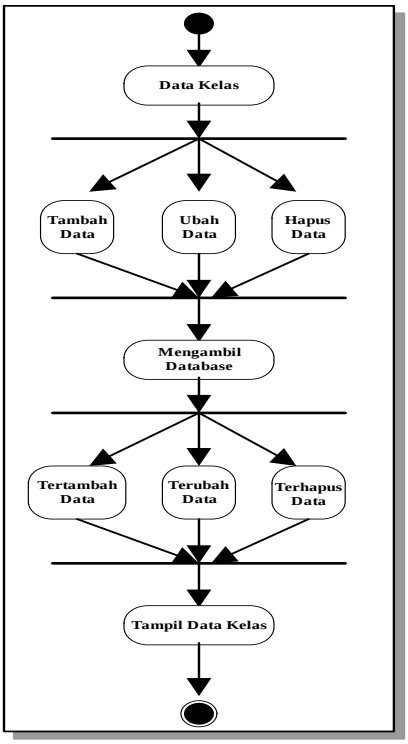
   Diagram Activity kelola data tahun ajaran *(Admin)* adalah dimana pada menu data tahun ajaran ada tombol tambah, ubah, dan hapus data.



Gambar 8. *Activty Diagram* Kelola Data Tahun Ajaran (*Admin*)

4. *Activity Diagram* Kelola Kelas *(Admin)*

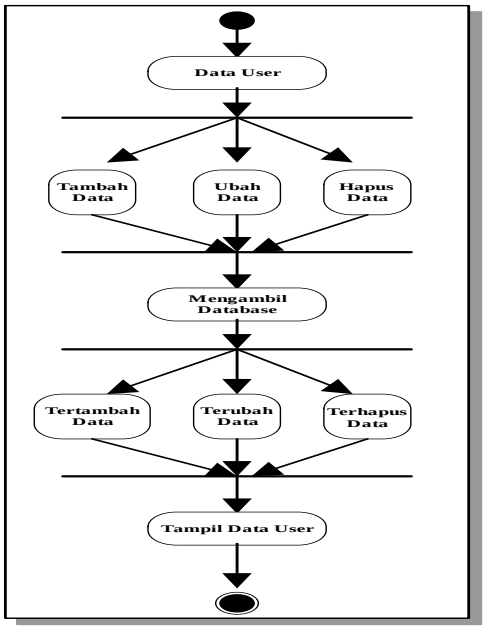
   Diagram Activity kelola data kelas (*Admin*) adalah dimana pada menu data kelas dan data telah ditampilkan akan ada tombol tambah, ubah, dan hapus data.



Gambar 9. *Activity Diagram* Kelola Data Kelas (*Admin*)

5. *Activity Diagram* Kelola Data *User (Admin)*

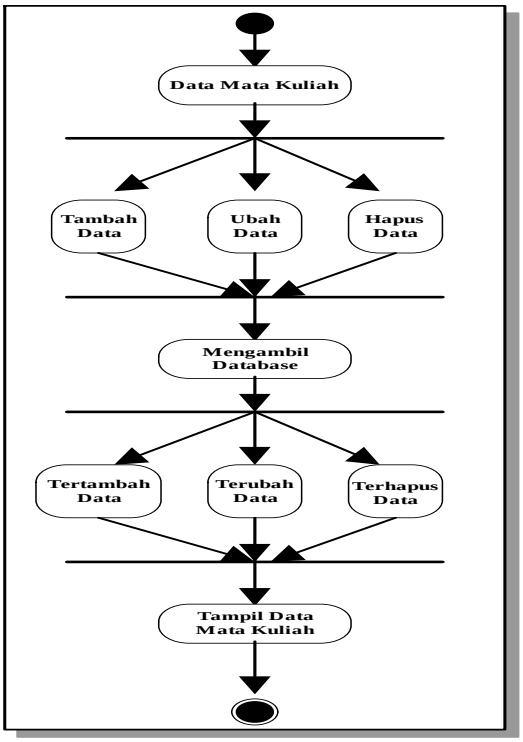
   Diagram activity kelola data *user (Admin)* adalah menu data *user* ada tombol tambah, ubah, dan hapus data.



Gambar 10. *Activity Diagram* Kelola Data *User* (*Admin*)

6. *Activity Diagram* Kelola Mata Kuliah (Akademik)

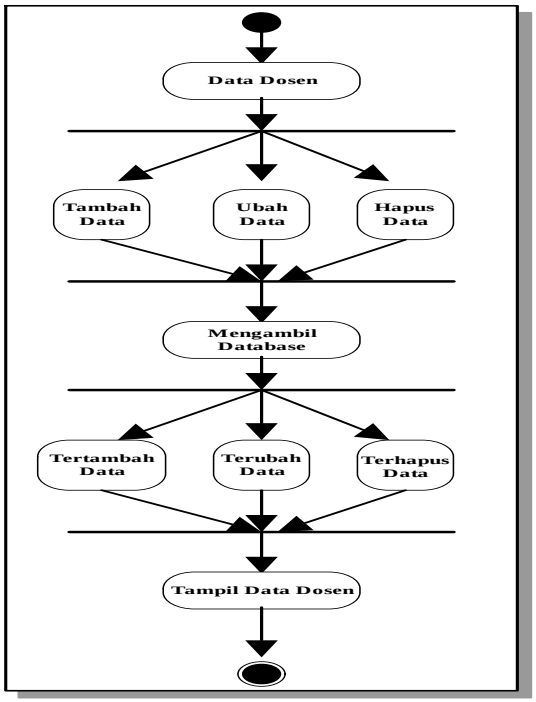
   Diagram activity kelola data mata kuliah *(*akademik*)* adalah menu data mata kuliah ada tombol tambah, ubah, dan hapus data.



Gambar 11. *Activity Diagram* Kelola Data Mata Kuliah (Akademik)

7. *Activity Diagram* Kelola Data Dosen (Akademik)

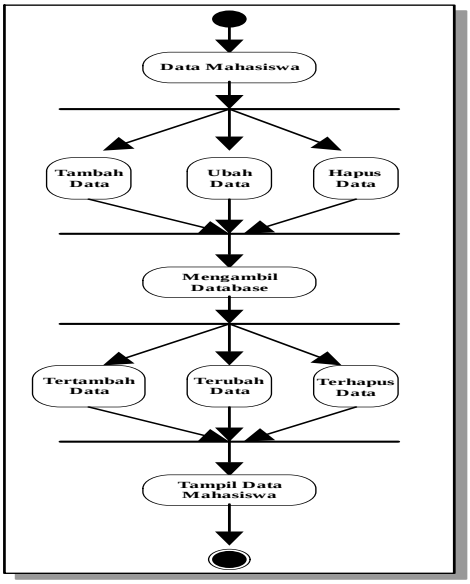
   Diagram Activity kelola data dosen adalah dimana akademik bisa melihat, menambah, mengubah, dan menghapus data dosen.



Gambar 10. *Activity Diagram* Kelola Data Dosen (Akademik)

8. *Activity Diagram* Kelola Data Mahasiswa (akademik)

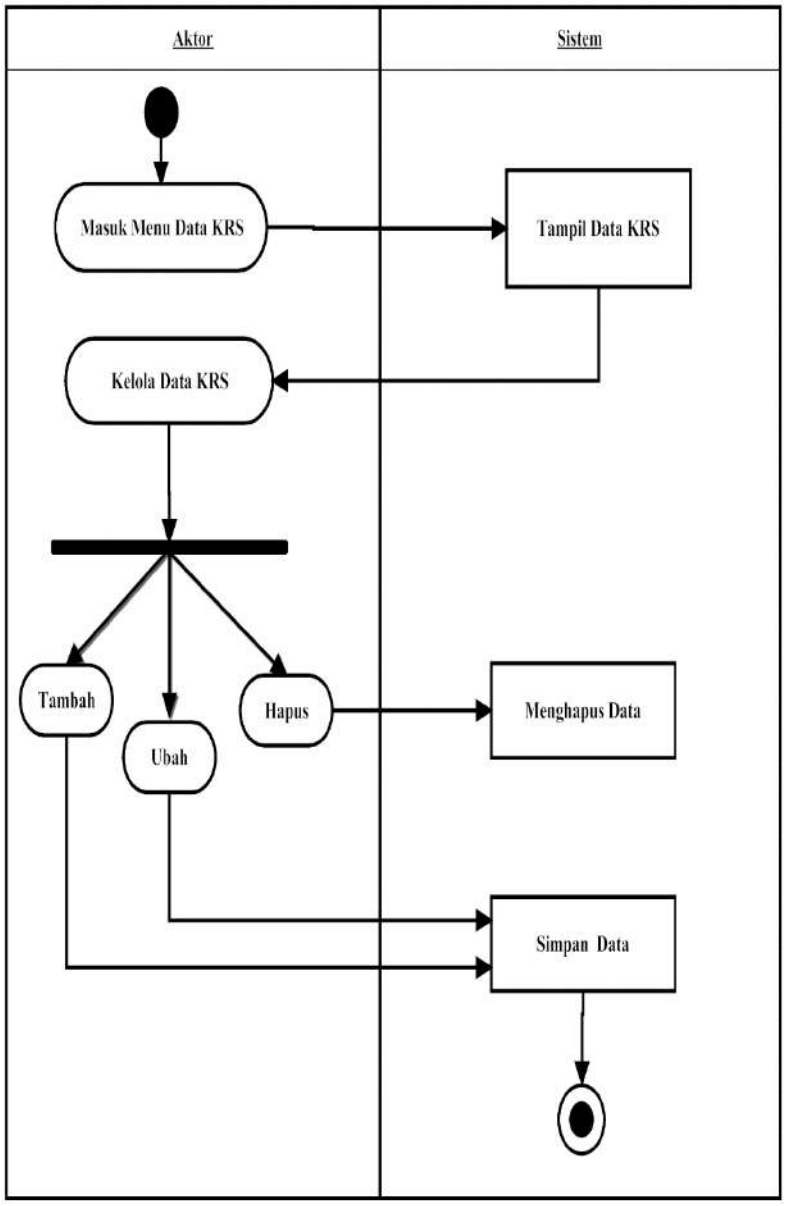
   Diagram Activity Lihat Nilai (akademik) adalah untuk melihat data biodata mahasiswa, menambah, mengubah, dan menghapus data.



Gambar 12. *Activity Diagram* Data Mahasiswa (Akademik)

9. *Activity Diagram* KRS (Akademik)

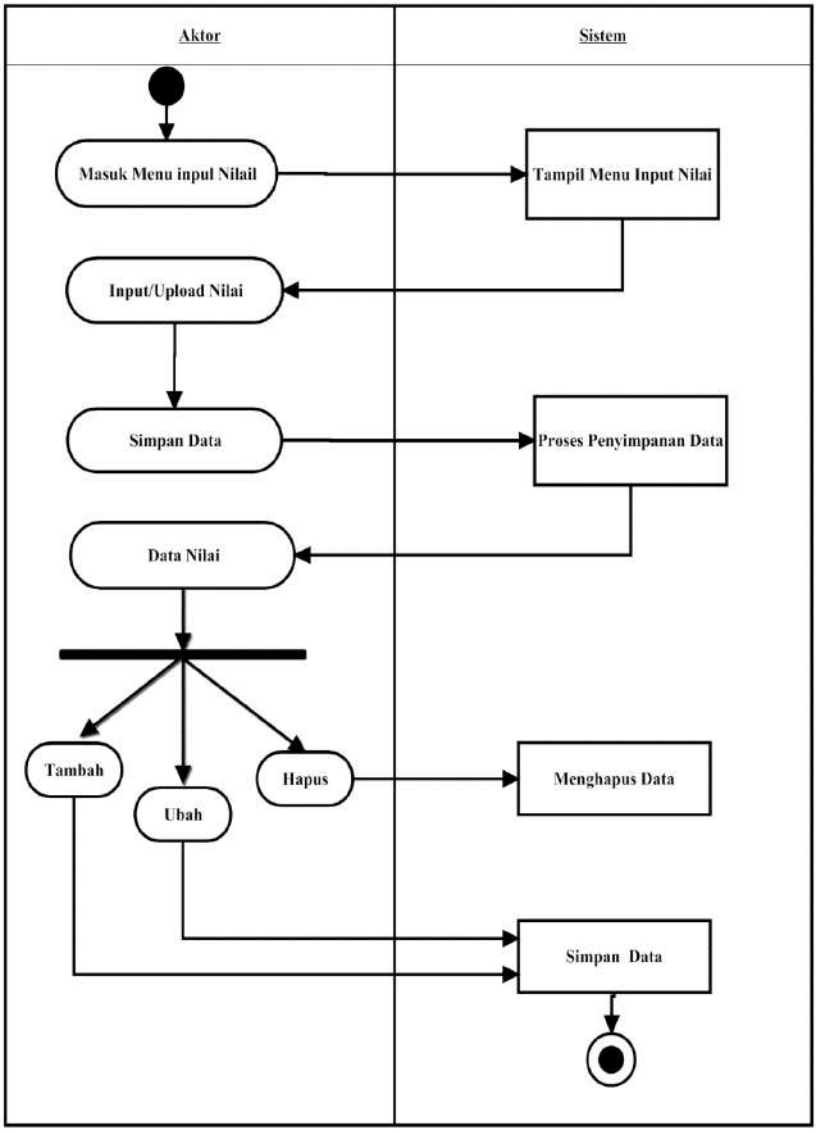
   Diagram Activity KRS *(Akademik)* adalah dimana pengguna (akademik) masuk menu data KRS, mengelola KRS dan memiliki tombol tambah, edit dan hapus data.



Gambar 13. *Activity Diagram* Data KRS (Akademik)

10. *Activity Diagram* KHS (Akademik)

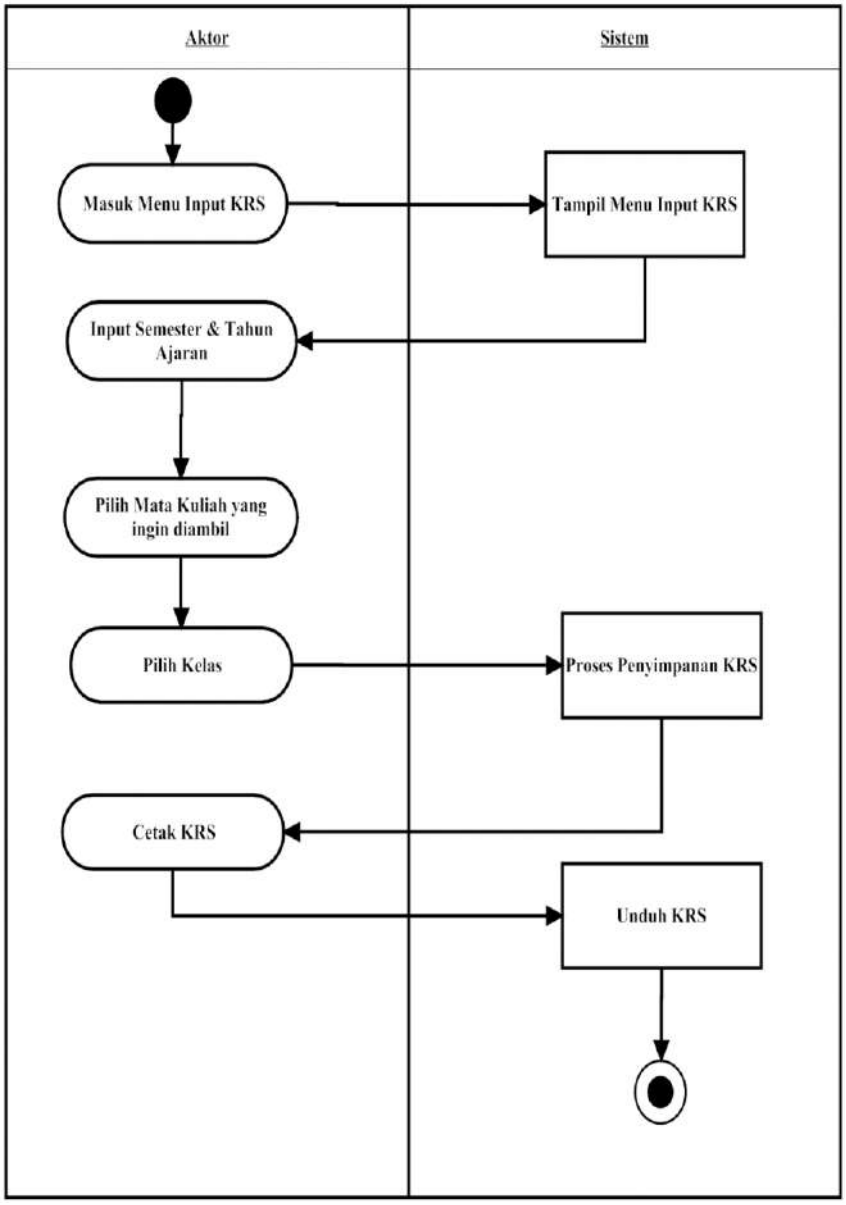
Diagram Activity kelola KHS (akademik) adalah dimana pada menu input nilai maka pengguna (akademik) menginput atau upload nilai dan menyimpannya. Jika proses penyimpanan sudah selesai maka akan menjadi sebuah data nilai. pada menu data nilai akan terdapat tombol tambah, ubah, dan hapus.



Gambar 14. *Activity Diagram* Kelola Nilai (Akademik)

11. *Activity Diagram* mengisi KRS (Mahasiswa)

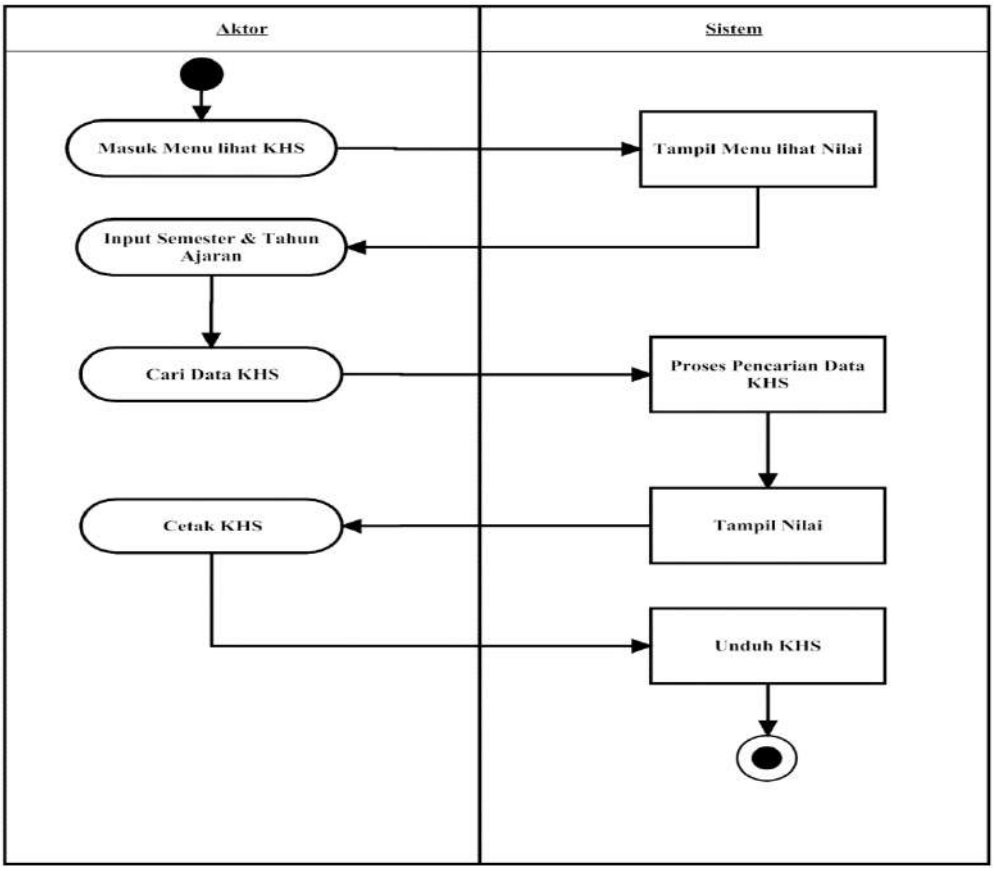
Diagram Activity mengisi KRS (mahasiswa) adalah dimana pada menu input KRS dan masukkan semester dan tahun ajaran. Kemudian, pilih mata kuliah yang ingin diambil, pilih kelas dan simpan KRS yang sudah di susun. Langkah terakhir adalah cetak KRS dengan mengunduh.



Gambar 15. *Activity Diagram* Input KRS (Mahasiswa)

12. *Activity Diagram* Lihat KHS (mahasiswa)

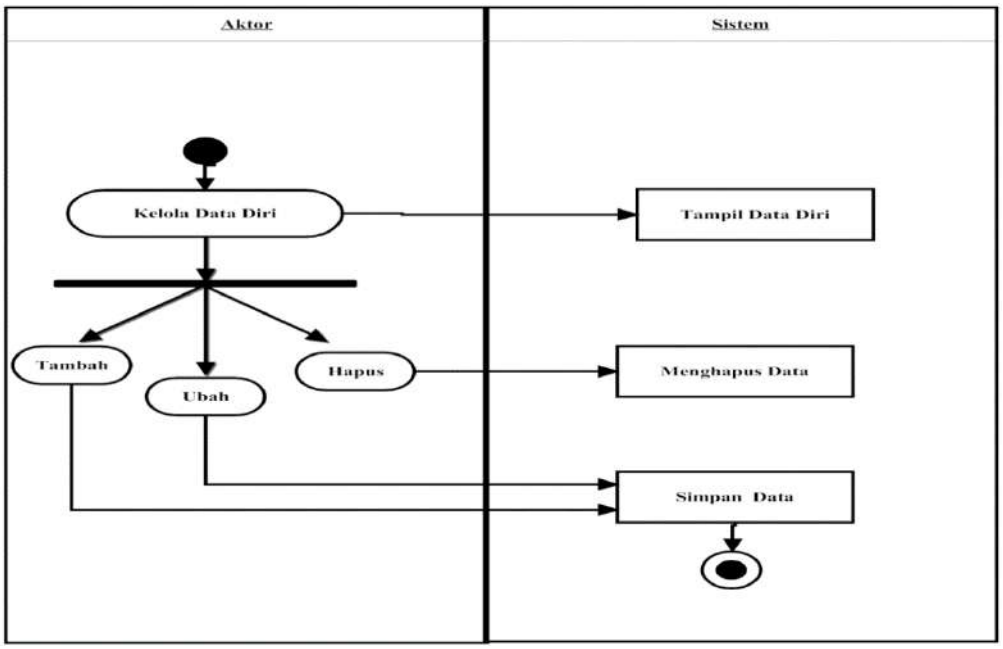
Diagram Activity lihat khs adalah dimana pada menu lihat KHS pengguna menginput semester dan tahun ajaran, kemudian mencari data khs dan jika data yang dicari ada, maka nilai akan tampil dan mengunduh KHS.



Gambar 16. *Activity Diagram* Lihat KHS

13. *Activity Diagram* Kelola Data Diri (Mahasiswa)

Diagram Activity kelola data diri (mahasiswa) adalah dimana pada menu data diri dan data telah ditampilkan akan ada tombol tambah, ubah, dan hapus data.



Gambar 17. *Activity Diagram* Kelola Data Diri (Mahasiswa)

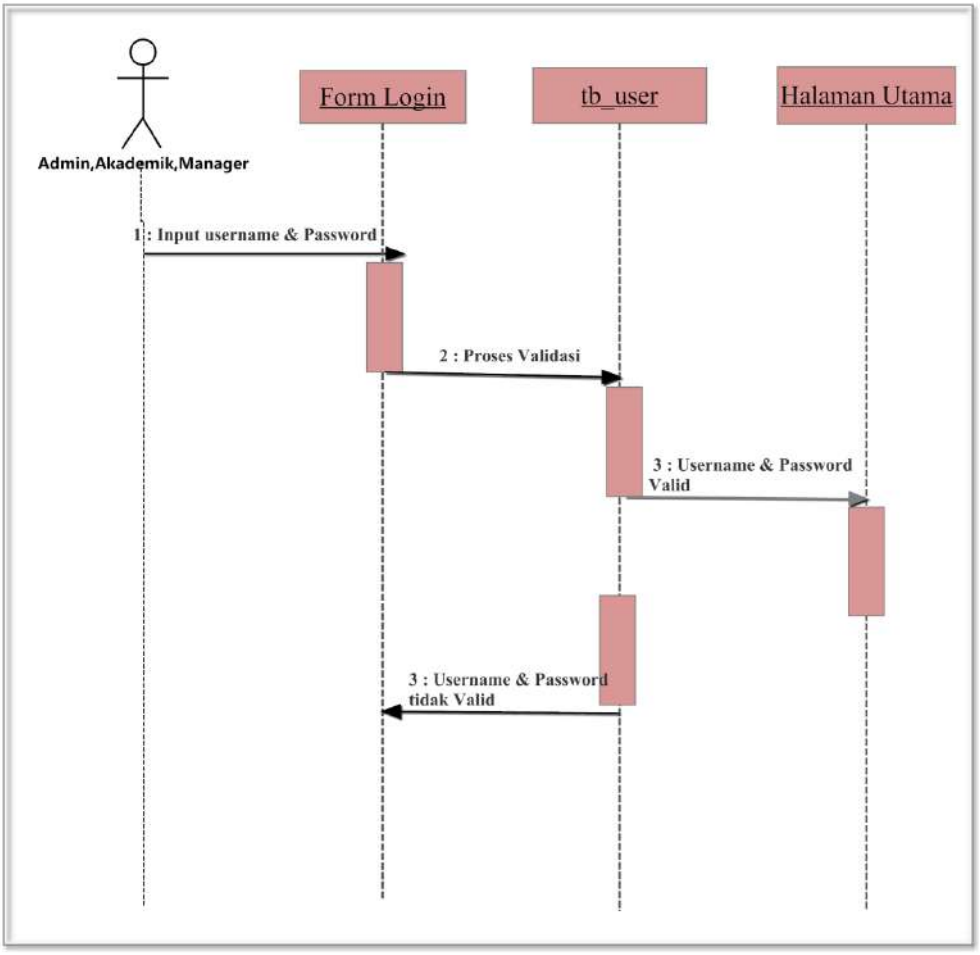
c. Sequence Diagram

Sequence Diagram dibuat untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even pada sebuah urutan atau rangkaian waktu untuk menghasilkan output tertentu. *Sequence Diagram* menjelaskan secara detail urutan proses yang dilakukan dalam sistem. operasi apa saja yang

terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

1. *Sequence Diagram Login*

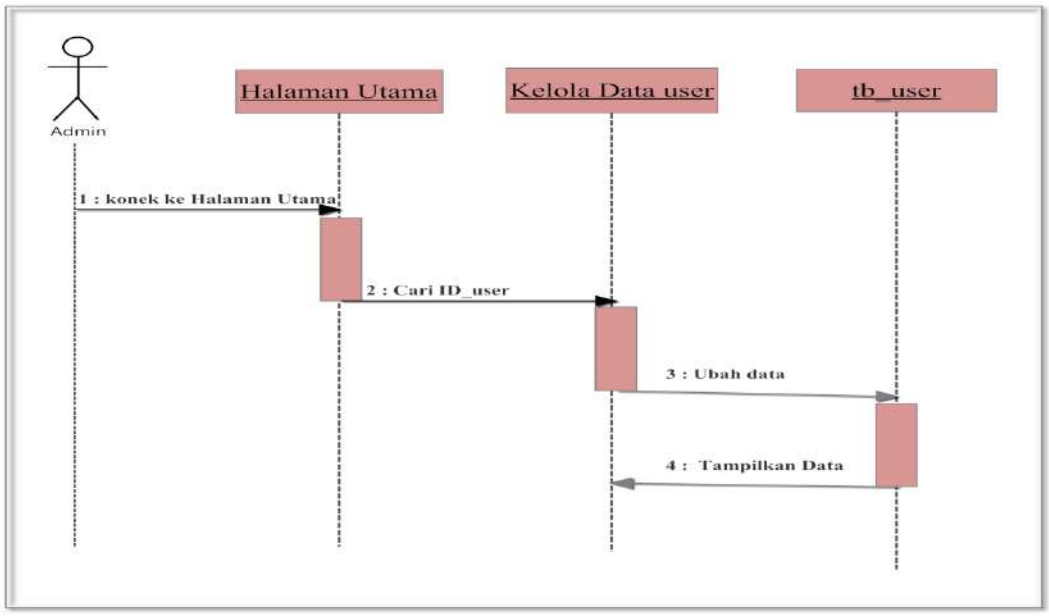
   Sequence Diagram Login adalah ketika pengguna (admin, akademik, mahasiswa) menginputkan *username* dan *password* ke *form login* lalu proses validasi pada *tb_user*, jika *username* dan *password* valid akan ke halaman utama, jika tidak valid maka akan kembali ke *form login.*



Gambar 18. *Sequence Diagram Login*

2. *Sequence Diagram* Kelola Data *User (*Admin)

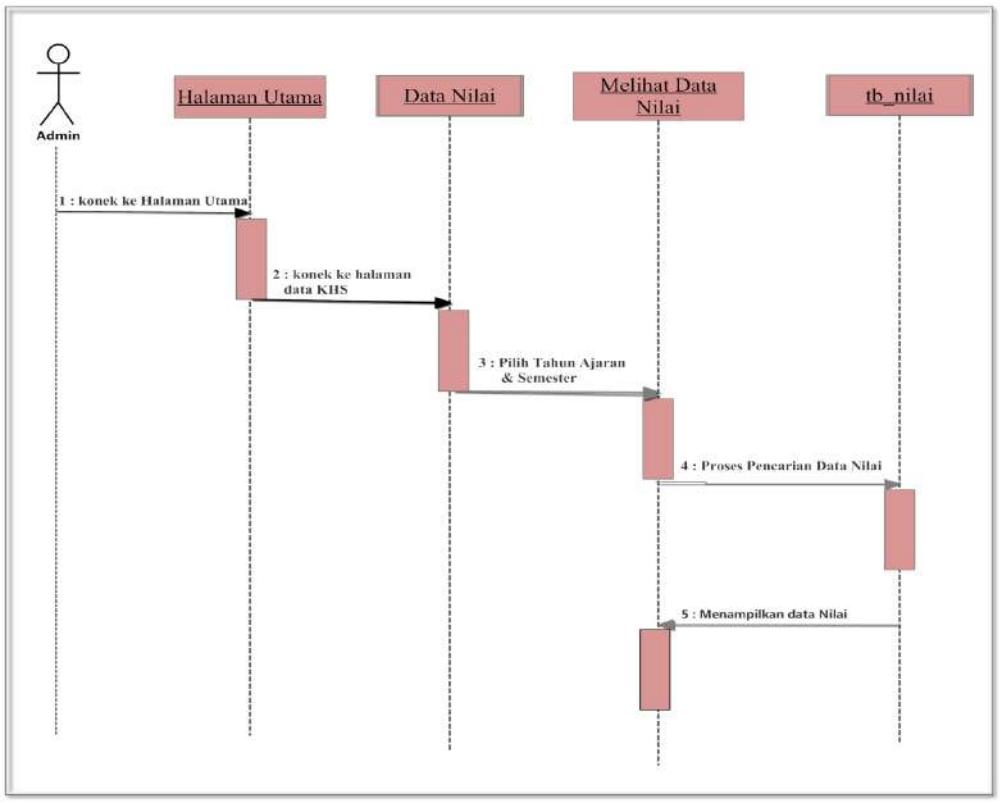
   Sequence Diagram Kelola Data *User* adalah ketika pengguna (admin) sudah login ke halaman utama, kemudian mencari id_user dan mengubah data pada tb_user. Jika ada perubahan maka tampilkan data.



Gambar 19. *Sequence Diagram* Kelola Data User (*Admin*)

3. *Sequence Diagram* Lihat Nilai (*Admin*)

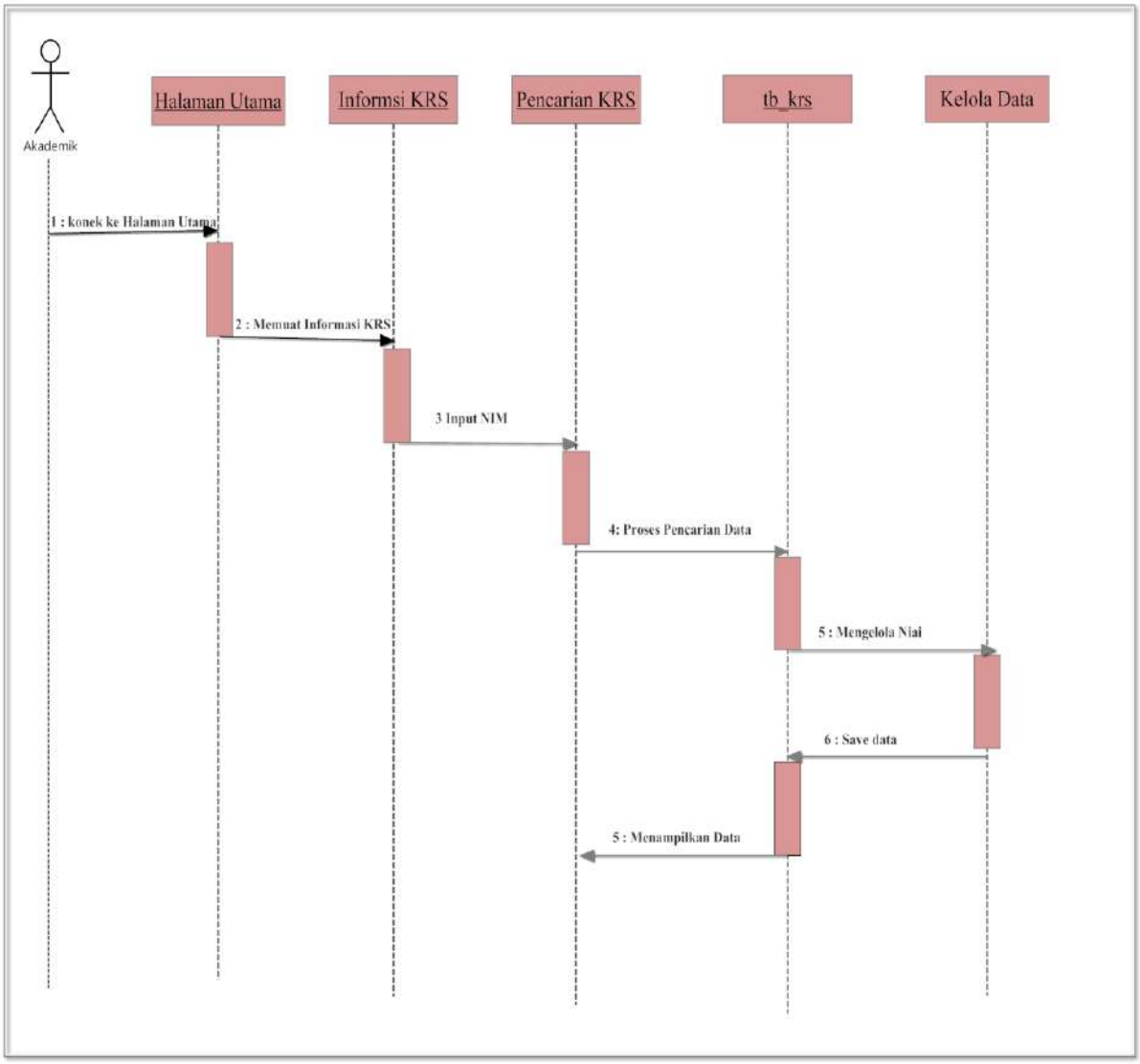
   Sequence Diagram Lihat Nilai (*Admin*) adalah ketika pengguna (admin) sudah login dan berada pada halaman utama dan ingin melihat nilai harus memilih menu data KHS dan pilih tahun ajaran dan semester, jika proses pencarian data nilai pada tb_nilai ada maka nilai akan ditampilkan.



Gambar 20. *Sequence Diagram* Lihat Nilai (*Admin*)

4. *Sequence Diagram* Lihat KRS (Akademik)

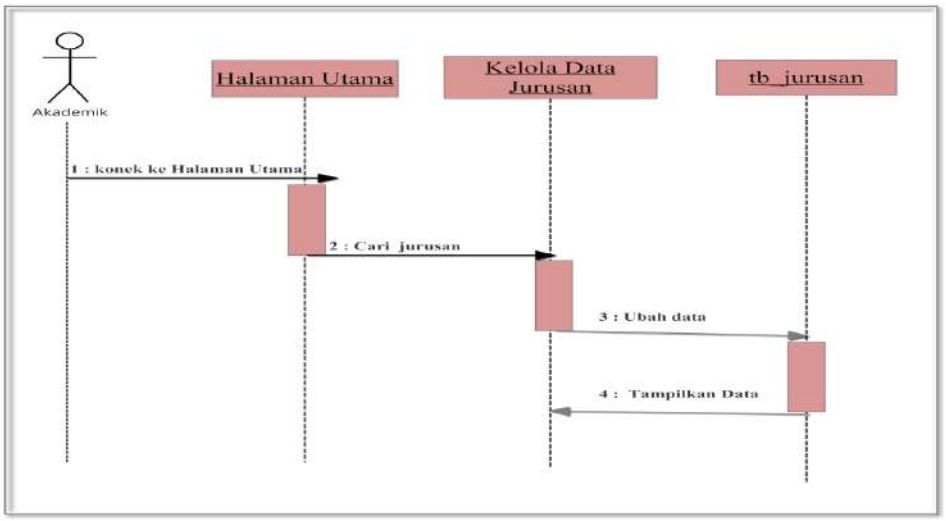
   Sequence Diagram Lihat KRS (Akademik) adalah ketika pengguna (akademik) sudah login dan berada di halaman utama. Jika pengguna (akademik) ingin memuat informasi KRS harus melalui proses pencarian data pada tb_krs. Kemudian pengguna (akademik) mengelola nilai, menyimpan, dan menampilkan data.



Gambar 21. *Sequence Diagram* Lihat KRS (Akademik)

5. *Sequence Diagram* Kelola Data Prodi (*Admin*)

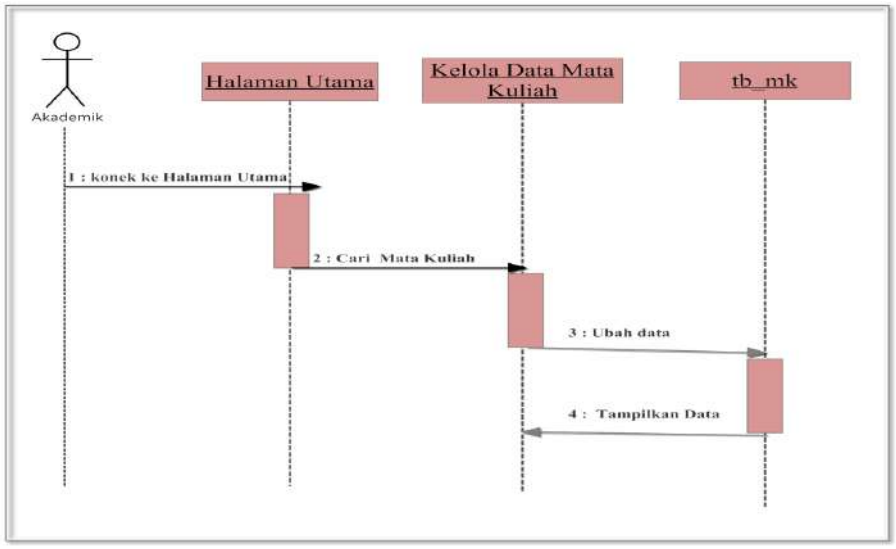
   Sequence Diagram Kelola Data Prodi (*Admin*) adalah ketika pengguna (*Admin*) sudah login dan berada di halaman utama, pengguna mencari jurusan untuk mengelola data jurusan pada tb_prodi, mengubah data dan menampilkan hasil data yang telah diubah.



Gambar 22. *Sequence Diagram* Kelola Data Prodi (*Admin*)

6. *Sequence Diagram* Kelola Data Mata Kuliah (Akademik)

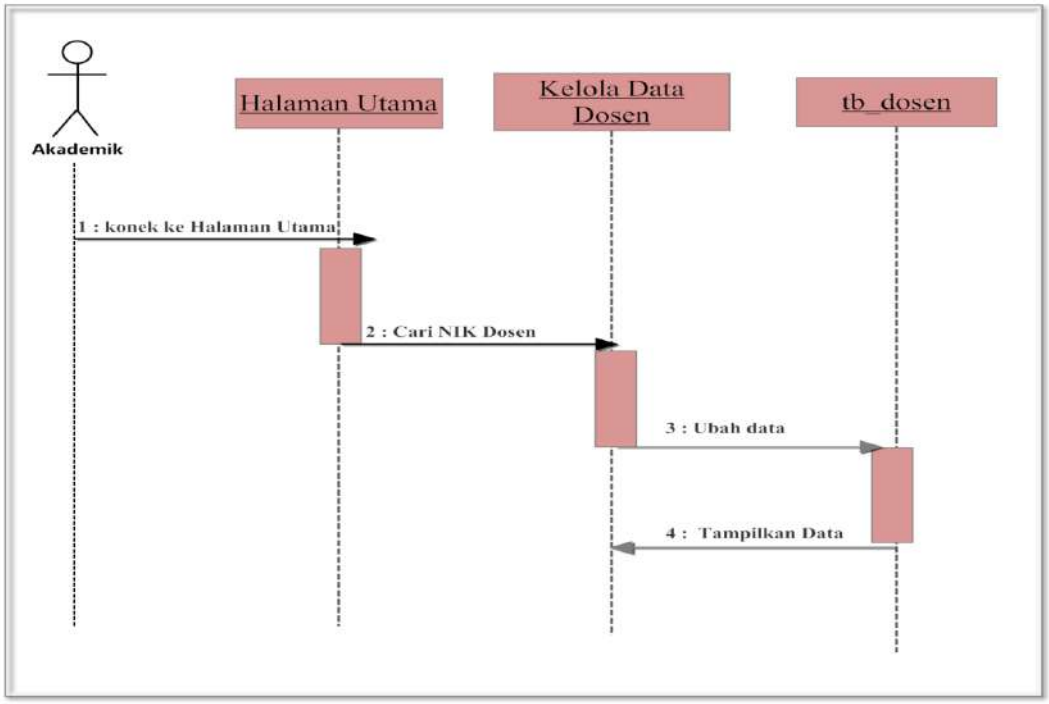
   Sequence Diagram Kelola Data Mata kuliah (Akademik) adalah ketika pengguna (akademik) sudah login dan berada di halaman utama, pengguna mencari mata kuliah dan mengubah data pada tb_mk, kemudian menampilkan hasil data yang telah diubah.



Gambar 23. *Sequence Diagram* Kelola Data Mata Kuliah (Akademik)

7. *Sequence Diagram* Kelola Data Dosen (Akademik)

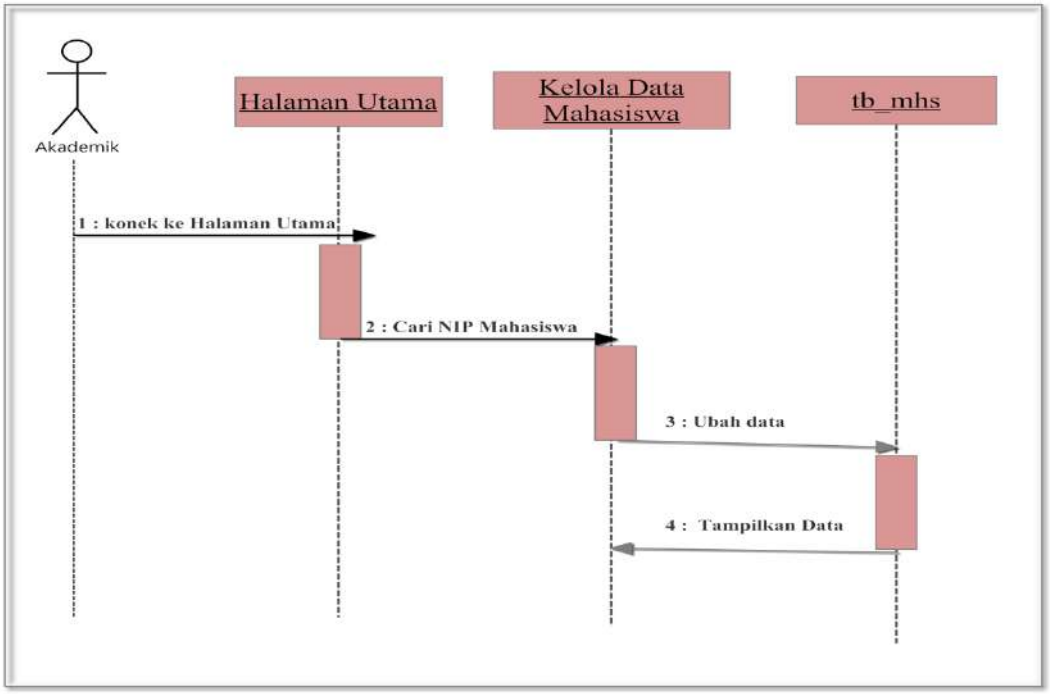
   Sequence Diagram Kelola Data Mata kuliah (Akademik) adalah ketika pengguna (akademik) sudah login dan berada di halaman utama, pengguna mencari NIK Dosen untuk mengelola dan menampilkan data yang telah diubah atau data yang ingin ditampilkan.



Gambar 24. *Sequence Diagram* Kelola Data Dosen (Akademik)

8. *Sequence Diagram* Kelola Data Mahasiswa (Akademik)

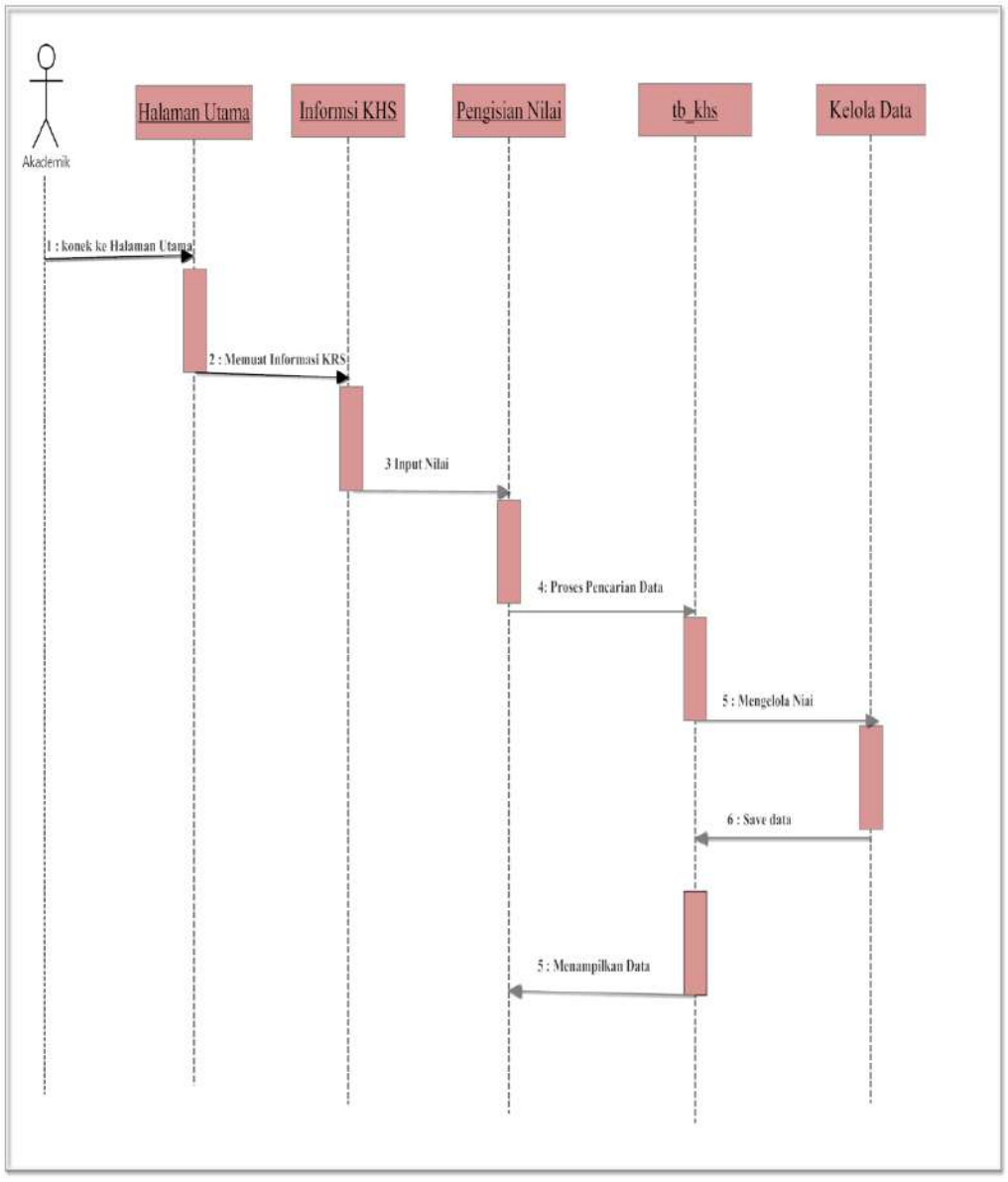
   Sequence Diagram Kelola Data Mata kuliah (Akademik) adalah ketika pengguna (akademik) sudah login dan berada di halaman utama, pengguna mencari NIM untuk mengelola data dan menampilkan data yang ingin ditampilkan.



Gambar 25. *Sequence Diagram* Kelola Data Mahasiswa (Akademik)

9. *Sequence Diagram* Kelola Nilai (Akademik)

   Sequence Diagram Kelola nilai (Akademik) adalah ketika pengguna (akademik) sudah login dan berada di halaman utama, pengguna memuat informasi KRS, kemudian menginputkan nilai. langkah selanjutnya proses pencarian data, mengelola nilai, menyimpan data, dan menampilkan data tersebut.

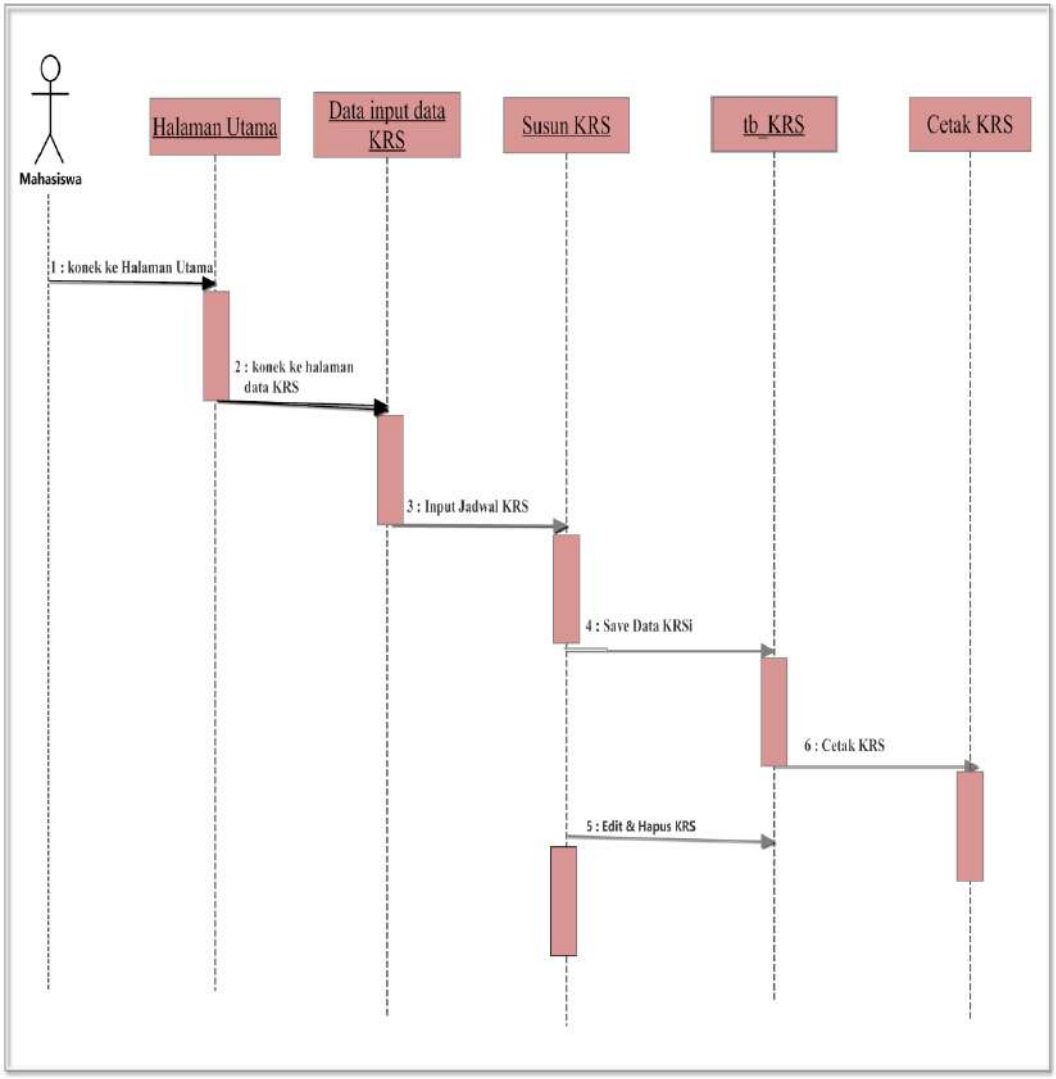


Gambar 26. *Sequence Diagram* Kelola Nilai (Akademik)

10. *Sequence Diagram* Kelola KRS (Mahasiswa)

Sequence Diagram Kelola KRS (KRS) adalah ketika pengguna (mahasiswa) sudah login dan berada di halaman utama, pengguna memilih menu KRS, menyusun KRS sesuai jadwal yang telah ditentukan, kemudian simpan data KRS pada tb_KRS, pada menu

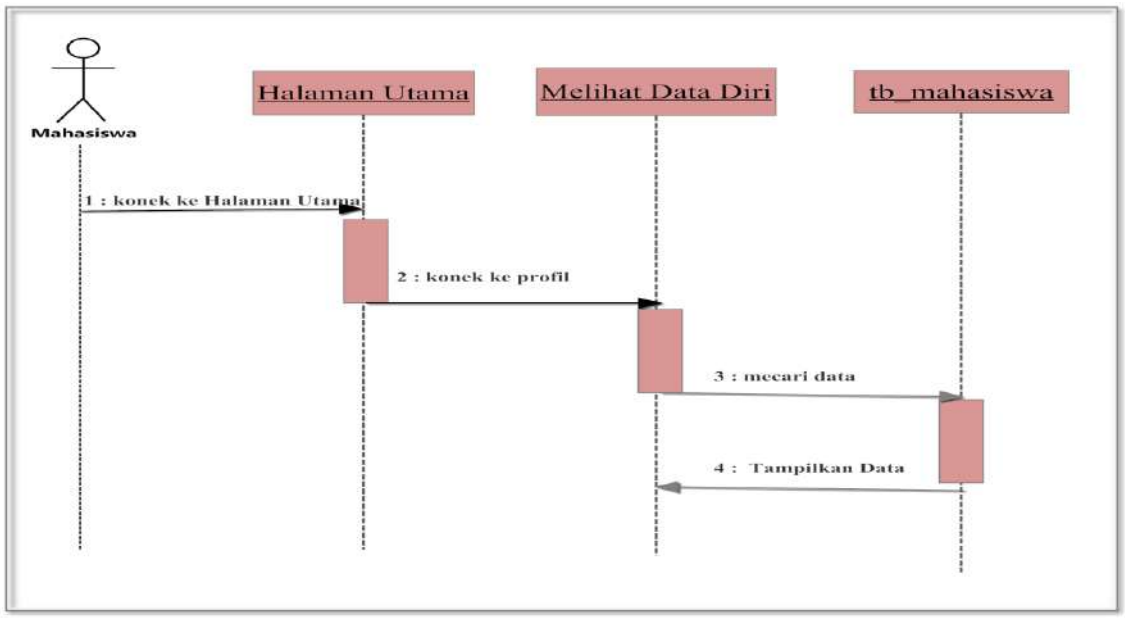
KRS ada tombol edit dan hapus KRS, dan langkah terakhir adalah cetak KRS.



Gambar 27. *Sequence Diagram* Kelola KRS (Mahasiswa)

11. *Sequence Diagram* Melihat Data diri (Mahasiswa)

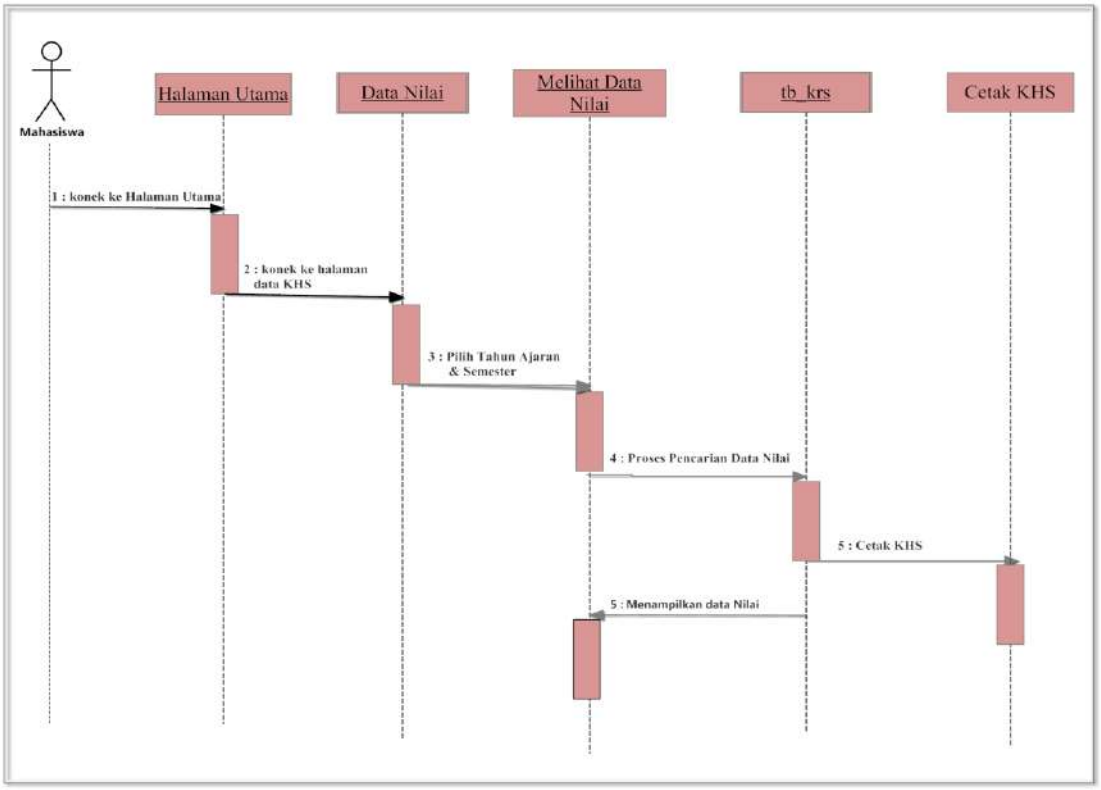
Sequence Diagram melihat data diri (Mahasiswa) adalah ketika pengguna (akademik) sudah login dan berada di halaman utama, pengguna melihat data diri, mencari data dan menampilkan data.



Gambar 28. *Sequence Diagram* Melihat Data Diri (Mahasiswa)

12. *Sequence Diagram* Lihat KHS (Mahasiswa)

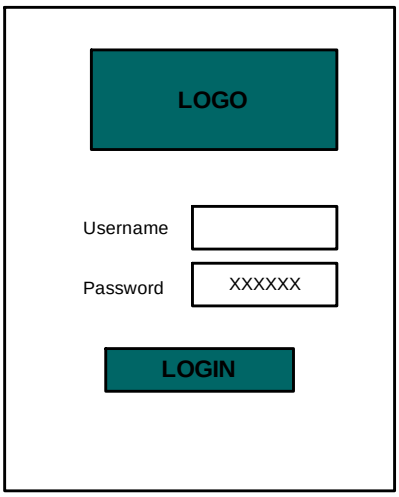
Sequence Diagram lihat KHS (Mahasiswa) adalah ketika pengguna (mahasiswa) sudah login dan berada di halaman utama, pengguna ingin melihat KHS, memilih tahun ajaran dan semester, kemudian proses pencarian data nilai pada tb_krs, tampilkan nilai dan bisa mencetak KHS.



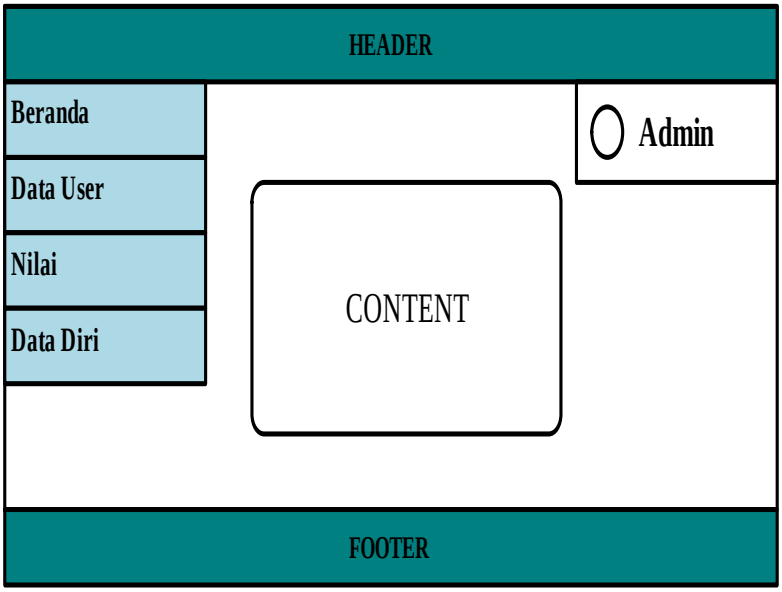
Gambar 29. *Sequence Diagram* Lihat KHS (Mahasiswa)

2. Desain Interface

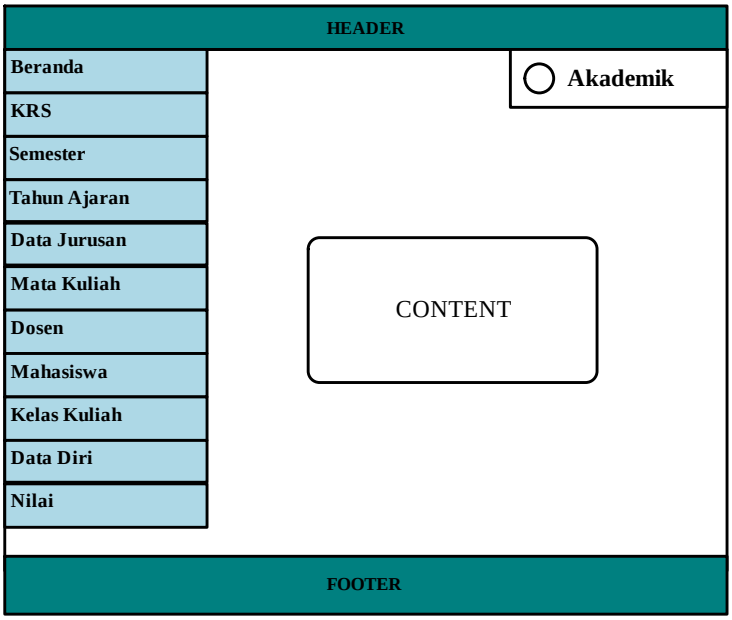
Desain *interface* ini merupakan perencanaan aplikasi *web* yang akan dibuat. Berikut adalah gambaran desain *interface.*



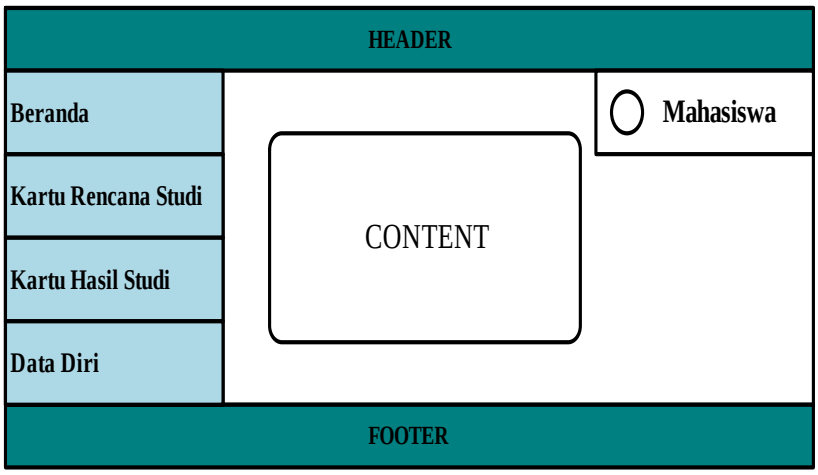
Gambar 30. *Interface* Sebelum *Login*



Gambar 3.36 *Interface Dengan* Hak Akses Sebagai Admin



Gambar 3.37 *Interface* dengan Hak Akses Akademik



Gambar 3.38 Interface Dengan Hak Akses Mahasiswa

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

A. Implementasi

1. Pengujian

   Proses pengujian dilakukan sebagai bagian dari tahap pengembangan program aplikasi yang digunakan untuk menemukan kesalahan yang mungkin akan terjadi pada tahap analisis kebutuhan, perancangan, ataupun pada tahap pembuatan kode program.

   Halaman ini dapat diakses oleh *admin*, akademik, dan mahasiswa yang berisikan informasi, dimana semua datanya disimpan di dalam *database*. Pengujian program adalah menguji konektivitas antara *PHP* dan *Mysql*.

   a. Pengujian Fasilitas *web*

      Fasilitas web dibuat dengan menggunakan *dreamweaver* cs5. Maka proses pengujiannya langsung melalui *browser*, dilakukan sebagai berikut :

      1) Jalankan *file PHP* yang dimaksud, dengan mengetikkan alamat *URL*-nya, *misalnya http://localhost/krs_online.php*
      2) Apabila terjadi kesalahan pada nama *database* atau nama tabel, maka akan keluar peringatan pada *browser* dan ditunjukan letak kesalahannya.
      3) Periksa dan perbaiki sintaks yang salah, maka lakukan kembali pengujian dengan langkah yang sama.

b. Kesalahan proses Informasi

Terdapat beberapa kesalahan yang terjadi, diantaranya disebabkan oleh:

1) Kesalahan dalam menulis nama *database* dan tabel pada *file PHP* dan *file Mysql*.
2) Kesalahan sintaks program, karena tidak sesuai ketentuan atau aturan bahasa pemrograman.
3) Kesalahan dalam logika program.
4) Kesalahan dalam meng-*include file*.

c. Tabel Hasil Pengujian

Sebelum aplikasi ini dibuat, maka harus dilakukan pengujian. Beberapa pengujian yang dilakukan oleh penulis sendiri. Pengujian ini dilakukan dengan metode *Blackbox Testing*.

Tabel 6. Hasil Pengujian Otentifikasi

No	Menu	Unjuk kerja yang diinginkan	Hasil Pengujian	Ket
1	-Login Admin	Dapat menampilkan halaman *login.*	Dapat masuk ke halaman admin dan akademik dengan login nip dan *password*.	
	-Logout	Dapat melepas koneksi user dengan *database.*	Dapat melepas koneksi user dengan *database.*	

2	-Login (Akademik) -Logout	Dapat menampilkan halaman *login.* Dapat melepas koneksi user dengan *database.*	Dapat masuk ke halaman admin dan akademik dengan login nim dan *password.* Dapat melepas koneksi user dengan *database.*	
3	-Login (Akademik) -Logout	Dapat masuk ke halaman admin dan akademik dengan login nim dan *password.* Dapat melepas koneksi user dengan *database.*	Dapat masuk ke halaman admin dan akademik dengan login nim dan *password.* Dapat melepas koneksi user dengan *database.*	

Tabel 7. Hasil Pengujian Data oleh pengguna mahasiswa

No	Menu	Hasil Pengujian	Ket
1	Home	Menampilkan halaman utama *user*	✓
2	Melihat Data Pribadi	Dapat melihat. Dan mengubah data diri	✓
3	Data KRS	Mengisi KRS pada semester tertentu	✓
4	Data KHS	Menampilkan data KHS sesuai dengan semester terakhir	✓

Tabel 8. Hasil Pengujian Data oleh Pengguna Akademik

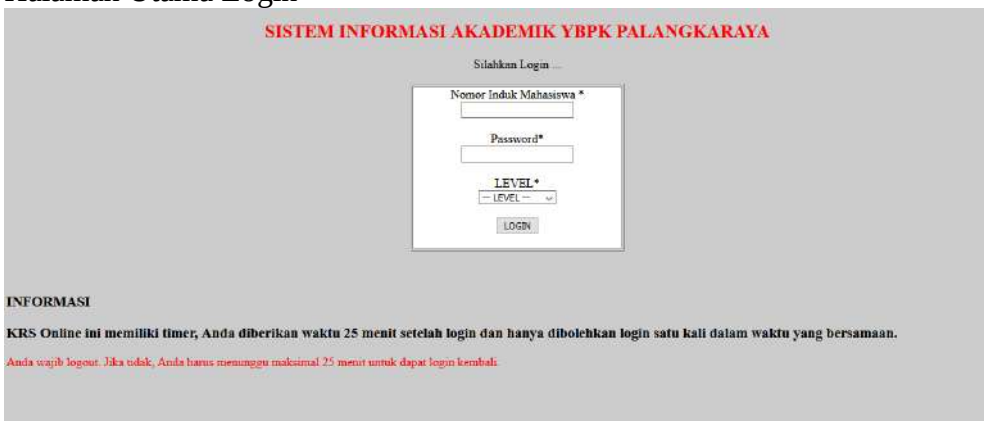
No	Menu	Hasil Pengujian	Ket
1	Home	Menampilkan halaman utama *user.*	✓
2	Data Mata Kuliah	Dapat menampilkan, menambah, mengubah dan menghapus mata kuliah.	✓
3	Data Dosen	Dapat menampilkan, menambah, mengubah dan menghapus data dosen.	✓
4	Data Mahasiswa	Dapat menampilkan, menambah, mengubah dan menghapus mata mahasiswa.	✓
5	Data KRS	Dapat menampilkan KRS mahasiswa sesuai dengan NIM, Semester, dan Tahun Akademik.	✓

6	Data KHS	Dapat menampilkan KHS mahasiswa sesuai dengan NIM, Semester, dan Tahun Akademik.	✓

Tabel 9. Hasil Pengujian Data oleh Pengguna *Admin*

No	Menu	Hasil Pengujian	Ket
1	Home Admin	Menampilkan halaman utama *user*.	✓
2	Data Prodi	Dapat menampilkan program studi.	✓
3	Data Tahun Ajaran	Dapat menampilkan tahun ajaran.	✓
4	Data User	Dapat menampilkan data pengguna.	✓

2. Implementasi Tampilan Web
   a. Halaman Utama Login



Gambar 31. *Halaman* Utama *Login*

b. Home Admin



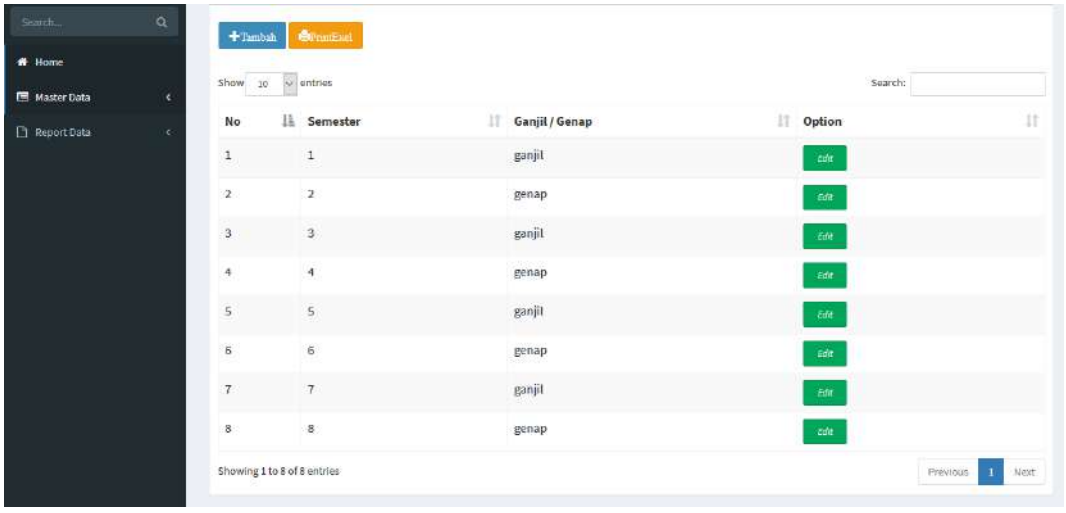
Gambar 32. *Interface* sesudah *Login*

c. Halaman Prodi



Gambar 33. *Interface* Program Studi

d. Halaman Semester



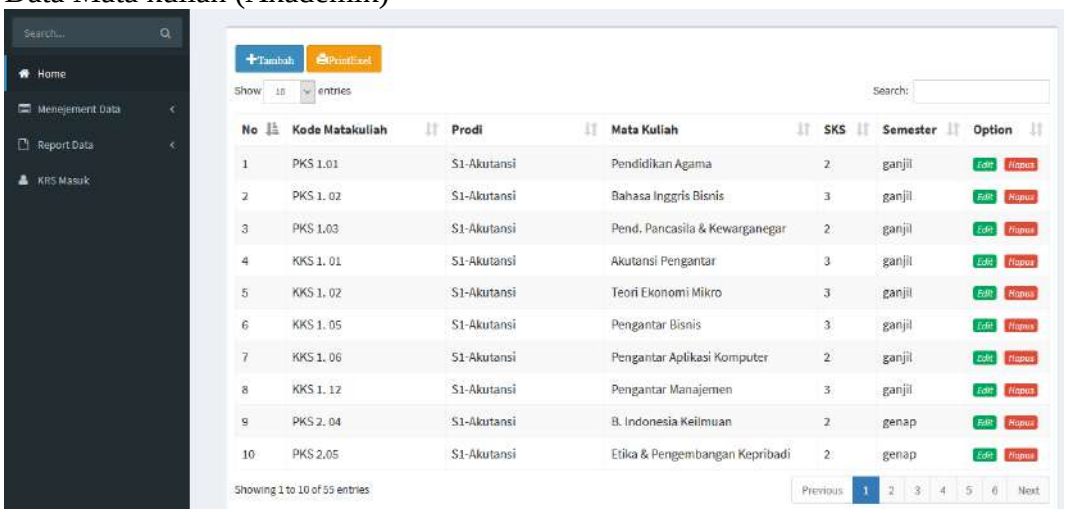
Gambar 34. *Interface* Semester

e. Data User



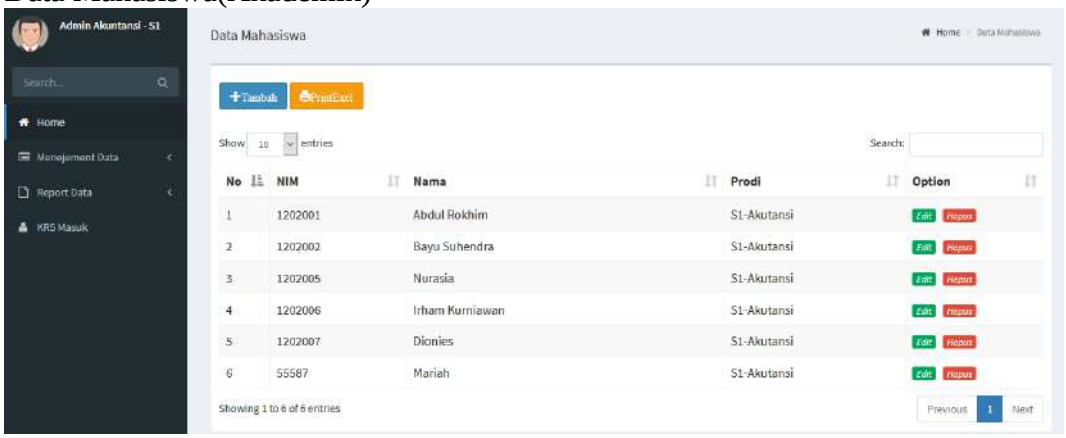
Gambar 35. *Interface* Data *User*

f. Data Mata kuliah (Akademik)



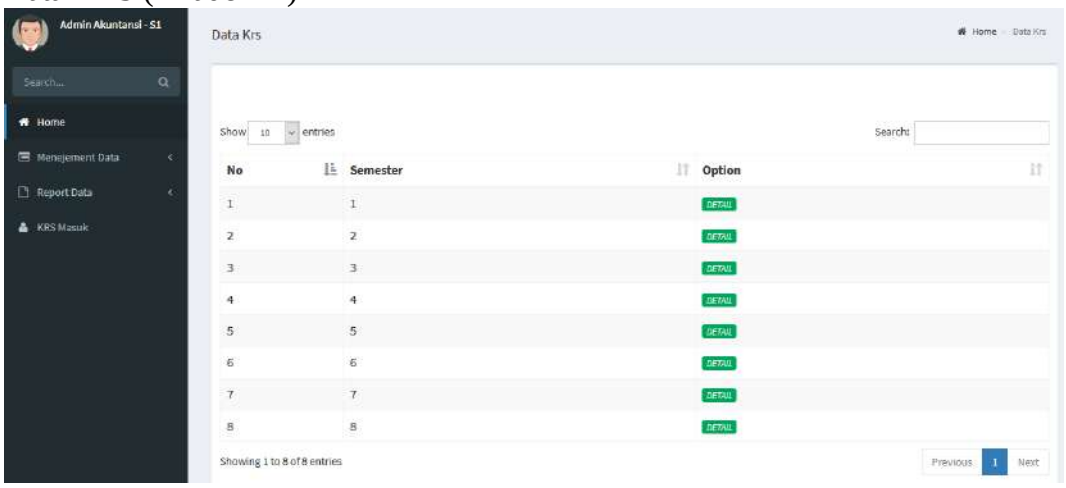
Gambar 36. *Interface* Data Mata Kuliah

g. Data Mahasiswa(Akademik)



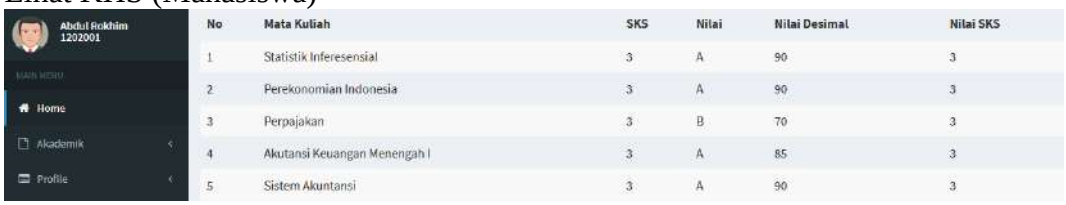
Gambar 37. *Interface* mahasiswa

h. Data KRS (Akademik)



Gambar 38. *Interface* Data KRS

i. Lihat KHS (Mahasiswa)



Gambar 39. *Interface* Lihat KHS Mahasiswa

BAB V
KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan analisis masalah pada perancangan website ini dapat diambil beberapa kesimpulan, diantaranya:

1. Penggunaan *PHP* dan *Mysql* sangat memungkinkan membangun website yang bersifat dinamis lainnya.
2. Mendapatkan keefesienan waktu pada mahasiswa.
3. Dengan adanya *website* dilengkapi login admin dan akademik, data lebih terjamin karena tidak adanya perubahan data.

B. Saran

Setelah melihat hasil dan analiss masalah yang ada, penulis memberikan saran:

1. Perlunya persiapan dan peningkatan sumber daya manusia yang mampu mengoperasikan komputer secara optimal dan meningkatkan keamanan.
2. Diharapkan adanya keamanan terhadap perangkat lunak dan perangkat keras dapat menunjang sistem yang sedang berjalan.

DAFTAR PUSTAKA

Jurnal Khatulistiwa Informatika, Vol. Iv, No. 2 Desember 2016, *Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak),* Hendini, A : 107-116

Jurnal Networking and Security, 2013, *Pembuatan Sistem Informasi Kartu Rencana Studi (KRS) Dan Kartu Hasil Studi (KHS) Pada Program Studi Teknik Informatika Universitas Surakarta, Siti,* Bambang dan Tri : 1-7

Nugroho, Y. A, 2014. *Rancang Bangun Krs Online Pada STMIK Palangkaraya Menggunakan Framework Codeigniter,* Palangkaraya

Pressman, R.S. 2015. *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I.* Yogyakarta: Andi

STMIK Palangkaraya, 2015. *Pedoman Penulisan Proposal dan Skripsi*. Palangkaraya

Sugiarti, Y. 2013. *Analisis dan Perancangan UML(Unified Modeling Language) Generated VB.6,* Graha Ilmu, Yogyakarta