**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Tujuan dari suatu pembelajaran adalah agar peserta didik dapat memahami materi yang diajarkan oleh pengajar melalui metode yang berbeda-beda, namun tujuan dari metode-metode tersebut adalah pemahaman yang baik dari peserta didiknya dan mereka aktif serta menerapkan segala sesuatu yang telah dipelajarinya berdasarkan pengetahuan maupun pemahaman yang dimilikinya. Banyak cara yang dilakukan oleh pengajar seperti membuat quiz, menambah waktu belajar, memperbanyak latihan soal, sampai memberi tugas yang banyak. Hal tersebut diharapkan membuat peserta didik tetap belajar di lingkungan sekolah atau di luar sekolah dan mempersiapkan untuk menghadapi soal-soal yang diberikan pengajar sehingga mendapatkan hasil yang baik, serta  kemampuan peserta didik pun dapat diketahui dari hal tersebut.

Pemahaman peserta didik yang baik itu sendiri berasal dari cara belajar dan lingkungan yang sesuai dengan peserta didik. Dengan ini pengajar perlu mengikuti cara belajar peserta didik sehingga mereka dapat memahami materi dengan mengetahui tingkat kemampuan dan cara belajar serta  bagaimana untuk meningkatkannya. Masalah yang dihadapi pengajar sendiri jika pembelajaran dilakukan di kelas memiliki peserta didik yang banyak, pengajar sulit untuk mengetahui dan menyesuaikan cara belajar masing-masing dari peserta didik Dari cara belajar dan lingkungannya peserta didik juga akan beragam dan pemahaman menjadi tidak seimbang antar peserta didik.

Salah satu contoh masalah berikut terdapat di salah satu Universitas di Bandung yaitu Universitas Nasional Pasim yang beralamat di Jalan Dakota No. 8A. Dimana terdapat suatu jurusan yaitu Manajemen Informatika dimana terdapat dua jenis mahasiswa, yaitu mahasiswa beasiswa PUB (Pemberdayaan Umat Berkelanjutan) dan mahasiswa Reguler. Mahasiswa beasiswa PUB sendiri adalah mahasiswa yang mengikuti beberapa tahap penyeleksian dalam penerimaan mahasiswanya. Terdapat pelatihan pemograman yang dapat membantu mahasiswa PUB dalam pembelajaran pemograman di perkuliahan. Sedangkan untuk mahasiswa Reguler hanya mengikuti pembelajaran di kuliah dan belajar mandiri, namun terkadang tertinggal dalam memahami suatu materi dikarenakan mahasiswa kurang berlatih dan  tidak aktif untuk mencari tahu tentang materi tersebut.

Padahal pembelajaran sangat bergantung dari peserta didik karena setiap manusia pasti memiliki cara sendiri untuk memahami suatu pelajaran. Namun dapat diketahui juga, jika lingkungan seseorang dapat mempengaruhi dalam belajar. Seperti mahasiswa PUB melakukan pelatihan pemrograman, sedangkan Mahasiswa Reguler tidak mendapatkan pelatihan seperti PUB. Pengajar juga menganalisis selama beberapa tahun hasil dari pembelajaran mahasiswa jurusan Manajemen Informatika yang  memperlihatkan data bahwa mahasiswa beasiswa PUB lebih baik dibandingkan mahasiswa Reguler.

Menurut datanya di Jurusan Manajemen Informatika, beberapa mahasiswa dapat meningkatkan kemampuannya. Istilah yang kami berikan bagi mahasiswa yang mengalami peningkatan kemampuan secara signifikan adalah *Good Learner*. Misalkan ada mahasiswa yang awalnya dia selalu mendapatkan nilai yang rendah dan kemudian dia mengalami peningkatan kemampuan sehingga mendapatkan nilai yang tinggi. Bagi mahasiswa yang belum mengalami peningkatan atau mahasiswa yang nilainya masih di bawah akan diarahkan untuk mengikuti jalur belajar *Good Learner*.

Pengajar juga berharap dapat meningkatkan kemampuan mahasiswanya agar memiliki kemampuan yang seimbang dan dapat mengikuti cara belajar *Good Learner*  saat mengerjakan soal atau latihan dari dosen sehingga dosen dapat melihat peningkatan dari mahasiswa tersebut dan juga akan mudah untuk memberi bimbingan bagi mahasiswa yang sulit memahami materi dari pengajar.

Untuk membantu pengajar maka dibutuhkan sistem yang akan digunakan dalam kegiatan mengajar sehingga dapat lebih mudah untuk menilai perkembangan mahasiswa dalam belajar. Mengikuti kemajuan teknologi saat ini, dikenal istilah *E-Learning*atau Pembelajaran Elektronik yang memiliki definisi sebagai sebuah bentuk teknologi informasi yang diterapkan di bidang pendidikan berupa website yang dapat diakses di mana saja. Pemanfaatan e-learning dapat memotivasi peningkatan kualitas pembelajaran dan materi ajar, kualitas aktivitas dan kemandirian mahasiswa, serta komunikasi antara dosen dengan mahasiswa maupun antar mahasiswa. Dan hal yang dibutuhkan adalah aplikasi yang akan kami bangun dapat menemukan cara belajar Good Learner sehingga mahasiswa yang bukan Good Learner dapat ditunjukan untuk mengikuti jalur belajar Good Learner.

E-Learning ini akan menyediakan beberapa fitur dan latihan soal yang disesuaikan dengan kemampuan mahasiswa sehingga mahasiswa dapat menjawab soal tersebut, yang artinya sistem memiliki sifat Adapatif. Dan untuk memudahkan pengajar mengetahui hasil belajar mahasiswa pada E-Learning ,maka Sistem ini juga akan menghasilkan ciri - ciri dari  mahasiswa dengan beberapa parameter tertentu, dapat berupa seberapa rajin mahasiswa belajar dengan menggunakan aplikasi yang penulis bangun dan berapa besar nilai quiz yang didapatkan. Sistem tersebut dinamakan Terpersonalisasi dikarena kan menggambarkan kemampuan dari masing-masing personal.

Untuk membantu hal tersebut, dibutuhkan metode belajar atau pengujian kemampuan mahasiswa dalam memahami materi. Maka penulis akan mengajukan judul  “GOOD LEARNER-BASED ADAPTIVE AND PERSONALIZED E-LEARNING”. Diharapkan dosen mengetahui dan memahami cara belajar peserta didiknya serta meningkatan pemahaman mahasiswa dalam suatu pembahasan sehingga mahasiswa dapat mengerti materi dan dosen dapat mencapai tujuan pengajarannya.

**1.2 Identifikasi Masalah**

Berdasarkan latar belakang yang telah penulis kemukakan, maka penulis

mengidentifikasi beberapa masalah sebagai berikut :

1. Bagaimana membangun sistem yang dapat mengetahui kemampuan mahasiswa di Jurusan Manajemen Informatika, Universitas Nasional Pasim, secara adaptif dan terpersonalisasi.
2. Bagaimana menerapkan aplikasi E-learning adatif dan terpersonalisasi di

Universitas Nasional Pasim sehingga mampu mendukung kegiatan belajar mengajar.

1. Bagaimana meningkatkan kemampuan mahasiswa Reguler atau mahasiswa beasiswa PUB dengan mengikuti cara belajar *Good Learner.*

**1.3** **Maksud dan Tujuan Penelitian**

**1.3.1 Maksud Penelitian**

Dosen dapat mengetahui cara belajar good learner melalui perekaman belajar mahasiswa dalam aplikasi E-Learning yang adaptif dan terpersonalisasi dengan memberi quiz - quiz dan materi yang sesuai diajarkan dosen sehingga cara belajar good learner tersebut dapat diterapkan pada mahasiswa Manajemen Informatika yang lain.

**1.3.2 Tujuan Penelitian**

Berdasarkan Rumusan Masalah yang penulis kemukakan, Penulis menentukan beberapa tujuan dari pembuatan Tugas Akhir ini yaitu :

1. Membangun sistem yang dapat mengetahui kemampuan mahasiswa di jurusan Manajemen Informatika, Universitas Nasional Pasim, secara adaptif dan terpersonalisasi.
2. Memudahkan pengajar untuk mengetahui cara belajar mahasiswa dalam memahami materi yang disampaikan dosen.
3. Meningkatkan kemampuan mahasiswa Reguler atau mahasiswa beasiswa PUB  dengan mengikuti cara belajar *Good Learner.*

**1.4 Batasan Masalah**

Adapun batasan masalah dalam pembangunan aplikasi ini adalah :

1. Data yang digunakan diambil dari Universitas Nasional Pasim.
2. Mata kuliah yang digunakan merupakan mata kuliah dasar pemrograman yaitu Logika dan Algoritma.
3. Aplikasi memberikan informasi (hasil) pencapaian tingkat kemampuan setiap mahasiswa kepada dosen dan tidak memberikan penanganan dari hasil tersebut.
4. Menerapkan teori Pembelajaran Modeling dalam menghasilkan generasi *Good Learner*.
5. Menerapkan teori K+1 dalam menangani mahasiswa yang kurang memahami suatu materi. K merupakan singkatan dari *Knowledge,* yaitu ilmu pengetahuan seseorang. Jika seseorang tersebut ingin belajar, menurut teori K+1 itu paling efektif adalah jika tingkat ilmu yang dipelajarinya adalah +1 dari *Knowledge* yang dia miliki saat itu.

**1.5 Kegunaan Penelitian**

**1.5.1 Bagi Penulis**

Penelitian ini akan menambah wawasan penulis dalam mengetahui cara belajar *Good Learner* dan sistem pembelajaran elektronik yang adatif dan terpersonalisasi serta meningkatkan kemampuan penulis dalam bidang pemrograman.

**1.5.2 Bagi Dosen Logika dan Algoritma Universitas Nasional Pasim**

Penelitian ini sangat diharapkan dapat membantu dosen Logika dan Algoritma di Universitas Nasional Pasim menjalankan tugasnya sebagai pengajar, memudahkan dalam menilai kemampuan mahasiswa, mengarahkan, dan mengetahui cara belajar *Good Learner.*

**1.5.3 Bagi Mahasiswa Jurusan Manajemen Informatika**

Dapat meningkatkan cara belajar dan meningkatkan kemampuan seperti *Good Learner* serta memudahkan mahasiswa untuk belajar materi yang diajarkan.

**1.6 Metodologi Penelitian**

**1.6.1 Metode Pengumpulan Data**

Dalam menyusun Tugas Akhir ini, penulis melakukan penerapan metode penelitian  dalam memperoleh data-data yang dibutuhkan sehingga penyusunan Tugas Akhir dapat diselesaikan dengan baik. Adapun metode penelitian yang penulis lakukan adalah :

1. **Studi Lapangan**

Merupakan metode yang dilakukan oleh penulis secara langsung pada Universitas Nasional Pasim sebagai objek penelitian khususnya data mahasiswa jurusan Manajemen Informatika, data nilai matakuliah Logika dan Algoritma.

**2.  Studi Pustaka**

Merupakan metode yang penulis lakukan untuk mendapatkan data-data teoritis baik yang bersumber dari buku maupun artikel yang berhubungan dengan masalah yang penulis hadapi.

**3.   Wawancara**

Pengumpulan data dengan wawancara ini digunakan untuk meyakinkan bahwa data yang diperoleh benar-benar akurat dan pada kesempatan ini penulis mewawancarai dosen mata kuliah Logika dan Algoritma di Universitas Nasional Pasim untuk mendapatkan data.

**4.    Studi Literatur**

Merupakan metode yang dilakukan oleh penulis dengan mengunjungi dan mempelajari website atau situs - situs yang berhubungan dengan penelitian ini .

**1.6.2 Model Proses Pengembangan Sistem**

**1.6.2.1  Prototype**

Penulis memilih untuk menggunakan model proses *Prototype*. Model proses ini dipilih dengan harapan mendapatkan umpan balik yang lebih awal dari pemakai. Dengan metode *Prototype* ini pengembang dan pemakai dapat saling berinteraksi selama proses pembuatan sistem.

Berikut adalah penjelasan dari tahap-tahap pemodelan proses *Prototype* :

1. Mengidentifikasi kebutuhan pemakai

Pengembang mewawancarai pemakai untuk mendapatkan gagasan dari apa yang diinginkan pemakai terhadap sistem.

1. Membangun *Prototype*

**Pengembang membangun *prototyping* dengan membuat perancangan semantara yang berfokus pada penyajian kepada pemakai (misalnya dengan membuat format *input* dan *output*).**

1. Evaluasi *Prototype*

Evaluasi dilakukan oleh pemakai dengan menilai *prototype* yang telah dibuat oleh pengembang apakah sudah sesuai dengan keinginan pemakai. Jika sudah sesuai maka langkah selanjutnya dilakukan tetapi jika tidak sesuai maka *prototype* akan diperbaiki kembali dengan mengulang langkah sebelumnya.

1. Pengkodean

**Dalam tahap ini *Prototype* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.**

1. Menguji Sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, maka harus di tes dahulu sebelum digunakan. Pengujian ini dilakukan dengan *Black Box*.

1. Evaluasi Sistem

Pada tahap ini, pemakai mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika sudah sesuai maka langkah selanjutnya dilakukan, akan tetapi jika tidak maka akan diperbaiki kembali dengan mengulang langkah Pengkodean dan Menguji Sistem.

1. Menggunakan Sistem

Perangkat lunak yang telah diuji dan diterima pemakai siap untuk digunakan.



**Gambar 1.1** Metode *Prototype*

(Sumber: Raymond Mc.Leod, Jr, 2001)

**1.6.3 Metode Pendekatan Pembangunan Sistem**

Metode pendekatan pengembangan sistem yang digunakan dalam Tugas Akhir  ini   adalah  dengan melakukan pendekatan *Object Oriented*.  Adapun   cara   menggambarkan   hasil  analisis dengan pendekatan *Object Oriented* ini menggunakan UML (*Unified Modeling Language*) yang digambarkan dalam bentuk :

* Use Case Diagram
* Sequence Diagram
* Class Diagram
* Activity Diagram

**1.7 Lokasi dan Waktu Penelitian**

 Lokasi penelitian dilakukan di Universitas Nasional Pasim, Jl. Dakota No. 8A, Cicendo, Sukaraja, Bandung. Adapun rincian kegiatan penyusunan Tugas Akhir yang direncanakan penulis adalah sebagai berikut :

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kegiatan | Bulan | | | | | | | | | |
| Oktober | | November | | | | Desember | | | |
| 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | Komunikasi (Analisis Kebutuhan Sistem) |  |  |  |  |  |  |  |  |  |  |
| 2 | Perencanan (Estimasi dan Penjadwalan) |  |  |  |  |  |  |  |  |  |  |
| 3 | Pemodelan (Analisis dan Desain) |  |  |  |  |  |  |  |  |  |  |
| 4 | Konstruksi (Pengkodean dan Pengujian) |  |  |  |  |  |  |  |  |  |  |
| 5 | Penerapan Aplikasi |  |  |  |  |  |  |  |  |  |  |
| 6 | Penyusunan Dokumen |  |  |  |  |  |  |  |  |  |  |

**Tabel 1.1** Rincian Kegiatan Penyusunan Tugas Akhir

**BAB II**

**LANDASAN TEORI**

**2.1** ***Good Learner***

Seseorang dikatakan *Good Learner* apabila ia dapat meningkatkan kemampuannya secara signifikan. Perubahan ini biasanya dikarenakan adanya keinginan dari orang tersebut dan juga usaha yang dilakukan untuk perubahan yang lebih baik. *Good* dalam Bahasa inggris sendiri berarti bagus atau baik sedangkan *Learner* adalah berasal dari kata *Learn* mempelajari jika diberi imbuhan -er dibelakang kata kerja maka akan menjadi kata benda atau pelaku yaitu Pelajar. Jika dihubungkan maka akan diartikan Pelajar yang baik. Kami menyimpulkan jika yang disebut *Good Learner* ialah dapat mengikuti Pembelajaran dan meningkatkannya.

**2.2** Definisi *E-Learning*

Jaya Kumar C. Koran (2002) dalam Hasbullah (2006), mendefinisikan E-learningsebagai pembelajaran yang menggunakan rangkaian elektronik (LAN, WAN, atau *internet*) untuk menyampaikan isi pembelajaran, interaksi, atau bimbingan. Sedangkan Dong (dalam Kamarga, 2002) mendefinisikan E-learning sebagai kegiatan belajar asynchronousmelalui perangkat elektronik komputer yang memperoleh bahan belajar yang sesuai dengan kebutuhannya.

Onno W. Purbo (2002) dalam Hasbullah (2006) menjelaskan bahwa istilah “e” atau singkatan dari elektronik dalam e-learningdigunakan sebagai istilah untuk segala teknologi yang digunakan untuk mendukung usaha-usaha pengajaran lewat teknologi elektronik internet. Internet, Intranet, satelit, tape audio/video, TV interaktif dan CD-ROM adalah sebagian dari media elektronik yang digunakan Pengajaran boleh disampaikan secara ‘synchronously’ (pada waktu yang sama) ataupun ‘asynchronously’ (pada waktu yang berbeda). Materi pengajaran dan  
pembelajaran yang disampaikan melalui media ini mempunyai teks, grafik, animasi, simulasi, audio dan video. *E-Learning* juga harus menyediakan kemudahan untuk ‘discussiongroup’ dengan bantuan profesional dalam bidangnya.  
Perbedaan Pembelajaran Tradisional dengan e-learning yaitu kelas ‘tradisional’,  
dosen/guru dianggap sebagai orang yang serba tahu dan ditugaskan untuk menyalurkan ilmu pengetahuan kepada pelajarnya. Sedangkan di dalam pembelajaran ‘e-learning’fokus utamanya adalah mahasiswa/siswa. Mahasiswa mandiri pada waktu tertentu dan bertanggung-jawab untuk pembelajarannya. Hasbullah (2006) menjelaskan filosofis e-learning sebagai berikut :

1. E-learning merupakan penyampaian informasi, komunikasi, pendidikan, pelatihan secara on-line.
2. E-learning menyediakan seperangkat alat yang dapat memperkaya nilai belajar secara konvensional (model belajar konvensional, kajian terhadap buku teks, CD-ROM, dan pelatihan berbasis komputer) sehingga dapat menjawab tantangan perkembangan globalisasi.
3. E-learning tidak berarti menggantikan model belajar konvensional di dalam kelas,tetapi memperkuat model belajar tersebut melalui pengayaan content dan pengembangan teknologi Pendidikan.
4. Kapasitas mahasiswa amat bervariasi tergantung pada bentuk isi dan cara  
    penyampaiannya. Makin baik keselarasan antar konten dan alat penyampai dengan gaya belajar, maka akan lebih baik kapasitas siswa yang pada gilirannya akan memberi hasil yang lebih baik.

Menurut Onno W. Purbo (dalam Hasbullah, 2006) mensyaratkan tiga hal yang wajib dipenuhi dalam merancang e-learning, yaitu : sederhana, personal, dan cepat. Sistem yang sederhana akan memudahkan peserta didik dalam memanfaatkan teknologi dan menu yang ada, dengan kemudahan pada panel yang disediakan, akan mengurangi pengenalan sistem e-learning itu sendiri, sehingga waktu belajar peserta dapat diefisienkan untuk proses belajar itu sendiri dan bukan pada belajar menggunakan sistem e-learning-nya. Syarat personal berarti pengajar dapat berinteraksi dengan baik seperti layaknya seorang guru yang berkomunikasi dengan murid di depan kelas. Dengan pendekatan dan interaksi yang lebih personal, peserta didik diperhatikan kemajuannya, serta dibantu segala persoalan yang dihadapinya.

**2.2.1** **Komponen E-learning**

Secara garis besar, ada 3 komponen utama yang menyusun e-learning:

1. *E-learning System*

Sistem perangkat lunak yang memvirtualisasi proses belajar mengajar konvensional. Bagaimana manajemen kelas, pembuatan materi atau konten, forum diskusi, sistem penilaian (rapor), sistem ujian online dan segala fitur yang berhubungan dengan manajemen proses belajar mengajar. Sistem perangkat lunak tersebut sering disebut dengan *Learning Management System* (LMS).

2. *E-learning Content* (Isi)

Konten dan bahan ajar yang ada pada *e-learning system* (*learning management system*). Konten dan bahan ajar ini bisa dalam bentuk *Multimedia-based Content* (konten berbentuk multimedia interaktif) atau *Text-based Content* (konten berbentuk teks seperti pada buku pelajaran biasa)

3.  *E-learning Infrastructure* (Peralatan)

Infrastruktur e-learning dapat berupa *personal computer* (PC), jaringan komputer dan perlengkapan multimedia. Termasuk di dalamnya peralatan *teleconference* apabila kita memberikan layanan *synchronous* learning melalui teleconference.

**2.2.2** **Kategori E-Learning**

Ditinjau dari segi interaksi antara sistem dengan manusia maka ada tiga kategori dasar dari e-learning, yaitu:

1.  *Synchronous Learning*

Pada  pembelajaran  synchronous  kondisinya  mirip  dengan  pembelajaran konvensional hanya saja pada e-learning hal ini tidak ditandai dengan kehadiran secara fisik. Pada bentuk synchronous ini pendidik (instruktur), peserta didik dan rekan rekannya   melakukan  “pertemuan” secara  online di  internet.  Melakukan proses belajar mengajar seolah sedang berada pada ruang fisik yang sama.

1. *Self-directed Learning*

Pada kategori  ini peserta didik melakukan pembelajaran  secara mandiri

Denganmengakses  berbagai  referensi  dan  bahan  belajar  yang  disediakanTidak  ada instruktur ataupun waktu khusus untuk berdiskusi dengan sesama peserta didik. Masing - masing peserta didik melakukan  proses belajar  sesuai  dengan kebutuhannya.

3.  *Asynchronous (Collaborative) Learning*

Kategori  ini  mengkombinasikan  karakteristik  dari  kedua  kategori  sebelumnya. Peserta didik belajar secara mandiri namun tetap berkomunikasi dengan peserta didik  lainnya maupun dengan pendidik walaupun  tidak harus di waktu khusus.. Penggunaan email, instant  message (Yahoo!  Messenger,  Gtalk)  ataupun  board pada forum dapat digunakan sebagai media komunikasi dan interaksi baik dengan pendidik maupun sesama peserta didik.

**2.3** **Pengembangan sistem *adaptive* *e-learning***

Pengembangan sistem *adaptive* *e-learning* berbeda dalam beberapa cara dari pengembangan perangkat lunak umum. Perbedaan ini terutama terkait dengan fasilitas navigasi, peran pengguna, adaptasi dinamis isi, navigasi dan presentasi, (Surjono, 2013). Pengembangan sistem *adaptive* *e-learning*  ini berawal dari adanya kebutuhan proses pembelajaran berbasis web terhadap sistem adaptif. Dengan menggunakan pendekatan sistem adaptif, diharapkan proses pembelajaran melalui *web* akan menjadi lebih optimal. Hal ini, karena sistem adaptif dapat menyesuaikan diri dengan proses adaptasi berdasarkan berbagai tujuan dan preferensi pengguna, sebagaimana dijelaskan oleh Fröschl (2005:12)  berikut ini:

*“An adaptive system adapts itself or another system to various circumstances. The process of adaptation is based on user’s goals and preferences”* yang artinya Sebuah system adaptif akan mengadaptasi dengan dirinya sendiri atau sistem yang lain dari berbagai lingkup. Proses adaptasi didasari oleh tujuan dan pilihan pengguna.

Dengan demikian *e-learning* dapat difasilitasi untuk menyampaikan materi pembelajaran sesuai dengan kebutuhan dan preferensi pengguna.  Sistem yang demikian bisa dikatakan sistem  *e-learning* adaptif.  Lebih lanjut Oppermann, dkk., (dalam Surjono 2008) menyatakan bahwa, sistem *e-learning* disebut bersifat adaptif apabila sistem mampu menyesuaikan secara otomatis kepada pengguna berdasarkan asumsi tentang pengguna. Disamping itu, *e-learning* adaptif, seharusnya mampu memberi kesempatan kepada pengguna untuk mengubah perilaku sistem sesuai dengan keinginan pengguna itu sendiri. Sistem *e-learning* adaptif perlu mengakomodasi kondisi atau karakteristik pengguna dan menyimpan semua informasi ini dalam model pengguna dan selanjutnya sistem akan memanfaatkan informasi ini sebagai dasar untuk menyampaikan materi pembelajaran. Model pengguna memperoleh informasi tentang pengguna dengan cara memonitor interaksi, tingkah laku browsing, dan mengetes, (De Bra, et al, 2002).

Menurut Brusilovsky (dalam Surjono, 2008), teknik melakukan adaptasi secara garis besar dapat dikategorikan menjadi dua, yaitu *adaptive presentation* (adaptasi isi) dan *adaptive* *navigation support* (adaptasi link). *adaptive presentation* adalah teknik yang digunakan untuk mengadaptasikan isi halaman web berdasarkan model pengguna. Teknik ini meliputi: presentasi teks adaptif dan presentasi multimedia adaptif. Sedangkan adalah *adaptive navigation support* teknik yang digunakan untuk memodifikasi link yang dapat diakses pengguna pada waktu tertentu.

Selanjutnya tingkat adaptasi dalam sistem *e-learning* adaptif dapat dibedakan tergantung pada siapa yang mengambil inisiatif, sistem atau siswa. Pendekatan ini dibedakan menjadi dua bentuk dalam adaptasi yaitu, *adaptivity* dan *adaptability*, (Kay, 2001). Menurut Papanikolaou (dalam Surjono, 2013) bahwa, adaptivity mengacu pada sistem *e-learning* adaptif yang menyesuaikan presentasi menggunakan data tentang siswa dengan cara sistem yang dikontrol. Sedangkan adaptability adalah suatu sistem *e-learning* adaptif yang mendukung modifiability pengguna akhir dengan menyediakan control siswa atas fungsi tertentu. Dalam merancang sistem *e-learning* adaptif, penting untuk menerapkan kedua tingkat adaptasi tersebut

**2.4** **Personalisasi *E-Learning***

Personalisasi *e-learning* adalah sebuah strategi dimana sistem *e-learning* akan mampu melayani pengguna sesuai kebutuhannya. Dengan kata lain situs *e-learning* tersebut akan tampil dan menyapa pengguna seolah-olah tahu keinginan, kebutuhan dan apa-apa yang ingin diperoleh setiap pengguna. Walaupun jumlah pengguna sistem tersebut banyak maka sistem *e-learning* dengan personalisasi tetap akan tampil dinamis sesuai kebutuhan dan karakter tiap pengguna. Sistem tersebut akan bekerja sebagaimana sistem *e-learning* pada umumnya. Sistem akan mengamati perilaku dari pengguna dan dicatat dalam *database*. Pengamatan dan pencatatan sistem *e-learning* meliputi topik materi belajar, diktat, buku dan bahan kuliah digital apa saja yang sering diakses oleh seorang pengguna. Diharapkan cara ini akan meningkatkan kedekatan sistem terhadap pengguna, memberikan kesan sistem tersebut ramah terhadap pengguna dan mengerti kebutuhan dan keinginan pengguna. Sehingga *e-learning* akan makin diminati dan akan memunculkan ketergantungan pengguna terhadap sistem *e-learning* tersebut.

**2.5 Teori Knowledge (i + 1)**

Teori ini merupakan salah satu teori yang telah diungkapkan oleh seorang profesor ahli bahasa yang bernama Stephen Krashen. Stephen Krashen adalah seorang profesor di Universitas Southern California. Dia adalah seorang ahli dalam ilmu kebahasaan yang sangat diakui oleh dunia. Dia juga seorang peneliti dan aktivis pendidikan. Stephen Krashen terkenal karena kontribusinya terhadap ide pemerolehan bahasa kedua (Second Language Acquisistion), Bilingual Education atau pendidikan dua bahasa, serta pendidikan membaca.

Stephen Krashen membuat beberapa hipotesis dan memabagi hipotesis tersebut

menjadi beberapa bagian sebagai berikut :

1. *Acquisition-learning Hipotesis,*

Dalam bukunya yang berjudul “Principle and Practice in Second Language Acquisition”, Krashen menekankan bahwa pemerolehan adalah proses tidak sadar “Acquisition is a subconscious procces”. Lebih rincinya, Krashen menjelaskan bahwa, pelajar tidak akan menyadari bahwa ia belajar bahasa, tetapi mereka hanya meyadari bahwa mereka sedang berkomunikasi. Singkat kata, pemerolehan bahasa terjadi ketika pelajar berkomunikasi dan terus berkomunikasi secara natural/alami, tidak terfokus kepada aturanaturan kebahasaan “not consciously aware of the rules”.

Sedangkan pengkoreksiannya/evaluasinya juga terjadi secara alami sesuai dengan konteksnya. Selanjutnya, kemampuan pendapatan bahasa ini tidak akan musnah dengan bertambahnya usia atau pada masa pubertas, “…the ability to pick up the language does not disappear at puberity” walaupun sudah berusia dewasa, pemerolehan masih sangat mungkin dilakukan dan terjadi. Malahan Krashen menganggap bahwa proses pemerolehan akan sangat kuat bila diterapkan sewaktu dewasa.

Berbeda dengan sebelumnya, sistem yang dipelajarai (pembelajaran) mengandung maksud kebalikannya, yaitu bahasa dikuasai melalui proses sadar, hal ini diamini oleh Krashen, ia berpendapat bahawa istilah belajar merujuk kepada pengetahuan secara sadar “…. The term (learning) henceforth to refer to conscious knowledge of second language”. Dengan kata lain bahasa dikuasai melalui proses dan pengkondisian yang terjadi secara formal, seperti belajar di kelas, kursus dll dengan mengetahui aturan kebahasaan, sinonom kata, dan belajar secara kontekstual. Adapun pengoreksiannya terjadi dengan melakukan latihan-latihan dan pembiasaan. Hal - hal yang telah tersebut tadi, akan berguna pada pelajar sebagai sensor ucapan-ucapan mereka sebelum memproduksi kata.

1. *Monitor Hypothesis,*

Maksud dari hipotesis ini adalah, setiap manusia dalam proses internal bahasa memiliki monitor yang berfungsi sebagai editing serta pengoreksi. Contohnya dalam belajar bahasa Arab terdapat pemakain ism mu’annats dan mudzakkar, monitor akan muncul dalam pikiran seseorang untuk mempertimbangkan kapan pelajar menggunakan Hadza atau Hadzihi. Hipotesis monitor berpendapat bahwa pemerolehan dan pembelajaran digunakan dengan cara yang sangat kompleks dan spesifik. Biasanya pemerolehan memulai dengan membuat para pelajar berucap/berbicara bahasa kedua (bahasa target) dan bertanggung jawab atas kefasihan dalam berbicara “acquisition “initiates” our utterances in a second language and is responsible for our fluency”. 50 Sedangkan belajar memiliki hanya satu fungsi, yaitu sebagai monitor atau editor “Learning has only one function, and that is as a Monitor.” Walaupun dimasukkan di dalamnya permainan (games) atau belajar sambil bermain, tetap saja ia hanya melakukan perubahan dalam ucapan.

1. *Natural Hypothesis Order,*

Dalam hipotesis ini Krashen menyatakan bahwa struktur bahasa diperoleh dengan urutan ilmiah yang dapat diperkirakan, beberapa struktur tertentu cenderung muncul lebih awal dari struktur yang lain dalam pemerolehan bahasa. Contohnya ada pada Struktur fonologi, dalam struktur fonologi anak cenderung memperoleh vokal-vokal seperti (a) sebelum akhirnya menyentuh vokal (i) dan (u). Konsonan depan lebih dahulu dikuasai oleh anak daripada konsonan belakang. Urutan alamiah seperti ini tidak saja terjadi pada masa kanak-kanak tapi juga terjadi pada masa dewasa.

1. *Input Hypothesis,*

Hipotesis ini menjelaskan bahwa pembelajaran bahasa kedua dinggap akan terjadi jika siswa yang mendapatkan informasi/ pengetahuan setingkat lebih tinggi dari pada yang telah dikuasainya. Dengan kata lain pelajar harus mendapatkan setingkat hal baru yang belum diketahuinya. Hipotesis ini memiliki rumusan (i+1). (i) memiliki maksud sabagai input sedangkan (1) memiliki maksud sebagai kompetensi setingkat dari sebelumnya. Jika (i+2) maka pelajar akan merasakan kesulitan dalam belajar bahasa, beda lagi jika (i+0) pelajar akan malas belajar, sebab pembelajaran dilakukan dengan pengetahuan sebagai input yang sudah dikuasai oleh siswa

1. *Effective Filter Hypothesis.*

Dalam hipotesis ini Stephen Krashen menjelaskan bahwa setiap manusia memiliki saringan efektif atau yang biasa disebut dengan (Effective Filter). Saringan inilah yang memberikan rasa takut, malu pada seorang pelajar. Seorang pelajar bahasa yang memiliki motivasi tinggi, kepercayaan tinggi, dan kecemasan lebih rendah, akan lebih mungkin untuk berhasil dalam pemerolehan bahasa, tapi sebaliknya jika pelajar bahasa tidak memiliki beberapa hal yang telah tersebut diatas dalam dirinya maka terwujudlah sebuah variabel emosional yang positif. Selanjutnya, menurut Krashen, saringan/filter ini akan menghambat siswa menerima/ mereproduksi bahasa.

Teori yang sangat erat hubunganya dengan teori yang akan kami terapkan dalam penyelesaian tugas akhir ini merupakan salah satu hipotesis yang di ungkapkan oleh Stephen Krashen yaitu *Input Hypothesis,* dalam hipotesisnya dijelaskan bahwa pembelajaran bahasa kedua dinggap akan terjadi jika siswa yang mendapatkan informasi/ pengetahuan setingkat lebih tinggi dari pada yang telah dikuasainya. Dengan kata lain pelajar harus mendapatkan setingkat hal baru yang belum diketahuinya. Hipotesis ini memiliki rumusan (i+1). i memiliki maksud sabagai input sedangkan (1) memiliki maksud sebagai kompetensi setingkat dari sebelumnya. Jika (i+2) maka pelajar akan merasakan kesulitan dalam belajar bahasa, beda lagi jika (i+0) pelajar akan malas belajar, sebab pembelajaran dilakukan dengan pengetahuan sebagai input yang sudah dikuasai oleh siswa.

2.6 **Tinjauan Perangkat Lunak**

**2.6.1 Moodle**

Cole dan Foster (2008) mendefinisikan *Moodle* sebagai singkatan dari *Modular Object-Oriented Dynamic Learning Environment* yang berarti tempat belajar dinamis dengan menggunakan model berorientasi objek. Aplikasi *Moodle* pertama kali dikembangkan oleh Martin Dougiamas pada Agustus 2002 dengan *Moodle* versi 1.0. Saat ini, *Moodle* bisa dipakai oleh siapa saja secara *open source(*Amiroh 2012).

*Moodle* dapat diinstalasi secara *online* maupun *offline*. Sistem yang  
dibutuhkan agar aplikasi *Moodle* dapat berjalan dengan baik secara *offline* adalah *Apache Web Server, PHP, database MySQL atau PostgreSQL.* Ketiganya dapat diperoleh dengan mengunduh *Xampp*. *Moodle* yang diintalasi langsung secara *online* membutuhkan *hosting, domain*, dan *file Moodle. Control panel* yang dibutuhkan tidak lagi secara *offline* dalam bentuk *xampp control panel* tapi diilakukan melalui *control panel online*, yaitu dengan menggunakan *cPanel.* Instalasi *Moodle* dilakukan di *cPanel.*

1. *Hosting*

*Hosting* adalah *space* dalam *server* komputer yang di gunakan sebagai  
penempatan data dan *file* yang ada. Purwanto (2010) mendefinisikan *hosting* sebagai ruangan yang terdapat dalam *harddisk* tempat menyimpan berbagai data, *file-file*, gambar dan lain sebagainya yang akan ditampilkan di situs. *Hosting* memiliki ukuran yang bermacam-macam. Semakin besar *hosting,* semakin besar data yang dapat disimpan.

1. *Domain*Nama *domain* adalah alamat permanen situs di dunia internet yang  
   digunakan untuk mengidentifikasi sebuah situs atau dengan kata lain *domain name* adalah alamat yang digunakan untuk menemukan situs kita pada dunia internet (Purwanto 2010). Domain diberikan untuk mengidentifikasi nama *server* komputer seperti *web server* atau *email server* di internet.
2. *cPanel*

*cPanel* merupakan *control panel online* yang dapat digunakan untuk  
mengatur *website*, membuat *email account* dan banyak hal lainnya seperti instalasi *script*. Anfidz (2010) mendefinisikan *cPanel* sebagai sebuah *Control Panel* untuk mengelola layanan *web hosting,* mudah digunakan dan kaya akan *feature*, seperti pengelolaan *e-mail.* Pengubahan format standar *Moodle* yang tersedia juga dilakukan melalui *cPanel.*

1. *Moodle*

*Moodle* yang dimaksud adalah *Moodle* terbaru yang kompatibel untuk windows (pada penelitian ini dikembangkan *Moodle* 2.4). *Moodle* dapat diunduh dalam bentuk .zip di www.Moodle.org. Kelebihan *Moodle* menurut Amiroh (2012) yaitu :

1. Sederhana, efisien dan ringan, serta kompatibel dengan banyak *browser*
2. Instalasi yang sangat mudah dengan dukungan dengan berbagai bahasa,

termasuk Bahasa Indonesia

1. Tersedianya manajemen situs untuk pengaturan situs secara keseluruhan, perubahan modul, dan lain sebagainya
2. Tersedianya manajemen pengguna (*user management*) dan manajemen  
   *course* yang baik

**2.6.1.1 *Social constructionist pedagogy***

## Berdasarkan *social constructionist pedagogy*, cara terbaik untuk [belajar](https://id.wikipedia.org/wiki/Belajar) adalah dari sudut pandang murid itu sendiri. [Model](https://id.wikipedia.org/wiki/Model) pengajaran berorientasi objek (murid) ini berbeda dengan sistem pengajaran tradisional yang biasanya memberikan [informasi](https://id.wikipedia.org/wiki/Informasi) atau materi yang dianggap perlu oleh pengajar untuk diberikan kepada murid. Tugas pengajar akan berubah dari sumber informasi menjadi orang yang memberikan pengaruh (*influencer*) dan menjadi contoh dari budaya kelas. Peran pengajar dalam [sistem](https://id.wikipedia.org/wiki/Sistem) Moodle ini antara lain: berhubungan dengan murid-murid secara perorangan untuk memahami kebutuhan belajar mereka dan memoderatori [diskusi](https://id.wikipedia.org/wiki/Diskusi) serta aktivitas yang mengarahkan murid untuk mencapai tujuan belajar dari kelas tersebut.

## 2.6.1.2 Penggunaan

## Di dunia *e-learning* Indonesia, Moodle lebih dikenal fungsinya sebagai *Course Management System* atau "[*Learning Management System*](https://id.wikipedia.org/wiki/Learning_Management_System)" (LMS). Dengan tampilan seperti halaman web pada umumnya, Moodle memiliki fitur untuk menyajikan kursus (*course*), dimana pengajar bisa mengunggah materi ajar, soal dan tugas. Murid bisa masuk log ke Moodle kemudian memilih kursus yang disediakan atau di-*enroll* untuknya. Aktivitas murid di dalam Moodle ini akan terpantau progress dan nilainya. Di Indonesia sendiri, diketahui bahwa Moodle telah dimanfaatkan untuk sekolah menengah, perguruan tinggi dan perusahaan.

## 2.6.1.3 Fitur

Sebagai LMS, Moodle memiliki fitur yang tipikal dimiliki LMS pada umumnya ditambah beberapa fitur unggulan. Fitur-fitur tersebut adalah:

1. Assignment submission
2. Forum diskusi
3. Unduh arsip
4. Peringkat
5. Chat
6. Kalender online
7. Berita
8. Kuis online
9. Wiki

Developer dapat meningkatkan konstruksi modular Moodle dengan menciptakan plugin untuk fungsi-fungsi baru yang lebih spesifik.

## 2.6.1.4 Penyebaran

Pengguna dapat menginstall *Moodle* dari banyak sumber, namun tetap membutuhkan keahlian teknis daripada penginstalan otomatis. Beberapa penyedia *hosting Moodle* gratis mengizinkan pendidik untuk menciptakan kelas *online* berbasis *Moodle* tanpa harus memiliki pengetahuan *server* atau instalasi. Beberapa penyedia *hosting* berbayar menyediakan layanan tambahan seperti kostumisasi dan penyebaran konten.

1. Aktifitas (termasuk permainan matematika dan kata)
2. Jenis-jenis sumber daya
3. Jenis-jenis pertanyaan (pilihan berganda, benar dan salah, mengisi

titik-titik, dll)

1. Jenis-jenis pengisian data (untuk aktifitas database)
2. Tema bergambar
3. Metode autentikasi (yang membutuhkan akses menggunakan

username dan password)

1. Metode pengambilan pembelajaran
2. Penyaring konten
3. Banyak sekali plugin pihak ketiga Moodle yang dapat dicari secara

gratis untuk membuat infrastrukturnya.

## 2.6.1.5 Penggunaan Moodle

Moodle berjalan tanpa harus dimodifikasi di [*Unix*](https://id.wikipedia.org/wiki/Unix)*,* [*Linux*](https://id.wikipedia.org/wiki/Linux)*,* [*FreeBSD*](https://id.wikipedia.org/wiki/FreeBSD)*,* [*Windows*](https://id.wikipedia.org/wiki/Windows)*,* [*Mac OS X*](https://id.wikipedia.org/wiki/Mac_OS_X)*,* [*NetWare*](https://id.wikipedia.org/wiki/NetWare) dan sistem operasi lainnya yang mendukung [*PHP*](https://id.wikipedia.org/wiki/PHP) dan *database*, termasuk penyedia *hosting web.*

Data bekerja pada *database* tunggal. *Moodle* versi 1.6 dapat menggunakan [*MySQL*](https://id.wikipedia.org/wiki/MySQL) atau [*PostgreSQL*](https://id.wikipedia.org/wiki/PostgreSQL). Versi 1.7, rilis pada November 2006, dapat menggunakan pemisahan *database* secara penuh sehingga penginstall *Moodle* dapat memilih salah satu jenis *server* database seperti [*Oracle*](https://id.wikipedia.org/wiki/Oracle) dan [*Microsoft SQL Server*](https://id.wikipedia.org/wiki/Microsoft_SQL_Server)*.*

Sistem *e-learning* memiliki banyak sekali dimensi penggunaannya. Fitur Penggunaan Moodle termasuk:

1. Autetikasi, menggunakan *Lightweight Directory Access Protokol* (LDAP),
2. *Shibbloleth*, atau metode standar lainnya seperti *Internet Message Access Protocol* (IMAP)
3. Penerimaan kelas, menggunakan IMS *Enterprise* daripada metode standar lainnya, atau dengan interaksi langsung dengan *database* eksternal
4. Pertanyaan kuis, menyediakan impor/ekspor banyak format: GIFT (format Moodle sendiri), IMS, [XML](https://id.wikipedia.org/wiki/XML) dan [XHTML](https://id.wikipedia.org/wiki/XHTML) *(walaupun ekspor berjalan dengan baik, impor sering kali tidak berhasil)*. *Moodle* menyediakan bermacam jenis pertanyaan - Perhitungan, Deskripsi, Essai, Pencocokan, Pilihan berganda, Pilihan singkat, angka, Pencocokan jawaban singkat acak, Benar/Salah.
5. Sumber, menggunakan IMS *Content Packaging* seperti [*Drupal*](https://id.wikipedia.org/wiki/Drupal)*,* [*Joomla*](https://id.wikipedia.org/wiki/Joomla) *atau* [*Postnuke*](https://id.wikipedia.org/w/index.php?title=Postnuke&action=edit&redlink=1) (via ekstensi pihak ketiga)
6. Sindikasi, menggunakan RSS atau *NewsFeed* - *newsfeed* dapat ditampilkan pada mata pelajaran, dan forum, blog, serta fitur lainnya yang dapat menyediakan *newsfeed*

*Moodle* juga memiliki fitur impor untuk pengguna dengan sistem spesifik lainnya, seperti mengimpor kuis atau seluruh mata pelajaran dari *Blackboard* atau *WebCT*. Bagaimanapun, teknis impor masih belum sempurna. Saat menulis ini (Feb 2010), Moodle tidak akan mengimport lagi mata pelajaran *Blackboard*, dikarenakan perubahan dari *source-code* php. Beberapa kegunaan yang tersedia dapat menolong mengubah mata pelajaran *Blackboard* menjadi format yang Moodle mengerti.["Blackboard migration"](http://docs.moodle.org/22/en/Blackboard_migration). moodle.or .

**2.7 PHP**

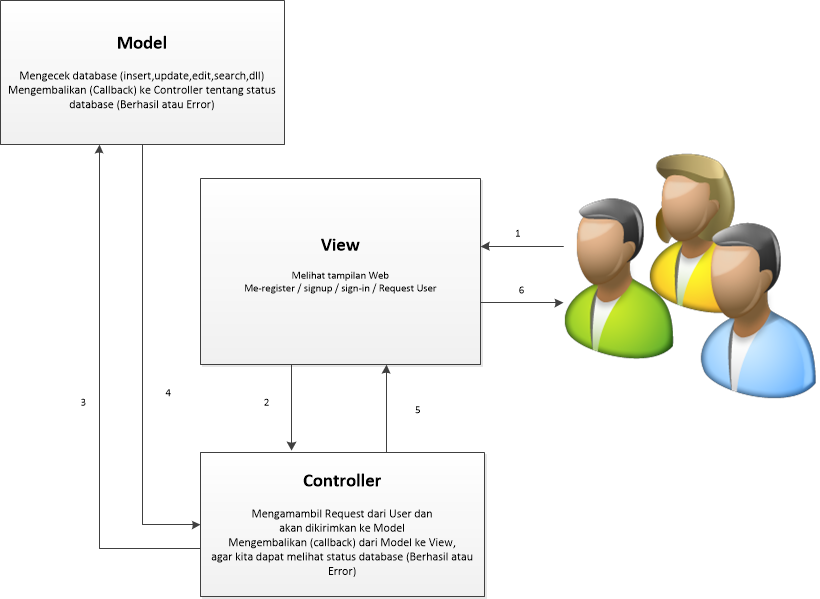
PHP merupakan sebagai salah satu server side  
programming.PHP atau merupakan singkatan rekursif dari PHP : *Hypertext Preprocessor* adalah suatu bahasa pemrograman yang termasuk kategori *server side*.*Server side programming* adalah jenis bahasa pemrograman yang nantinya *script*/program tersebut akan dijalankan oleh *server*. Selanjutnya hasil pengolahan *script/*program tersebut akan dikirim ke *client* sebagai output. Kelebihan antara lain:

1. Bahasa pemrograman *PHP* sangat mudah dipelajari karena mirip dengan bahasa C/C++
2. PHP mudah diimplementasikan karena *software* PHP servernya 100% *free*
3. Kebanyakan *server web hosting* mensupport PHP
4. PHP paling banyak digunakan oleh *web programmer* di seluruh dunia
5. Integrasi antara PHP dan database mudah untuk diimplementasikan karena PHP mensupport banyak *database* (DBMS), seperti *MySQL*, *MS. Access, Oracle*, DB2,Dbase*, Interbase* dll. Berikut ini beberapa tool yang sebaiknya Anda miliki:
6. *Software web server*, bisa dipilih salah satu antara *Apache*, IIS, PWS (100 % *free*)
7. *Software* PHP *server* (100% free)
8. *Software database*, sebaiknya pilih *MySQL* karena 100% *free*. Bila sudah  
   menguasai PHP + *MySQL* bisa dikembangkan dengan *database* yang lain.
9. *Text Editor*, misalkan Notepad atau Notepad++. Dalam hal ini saya  
   merekomendasikan Notepad++ karena selain software ini 100% *free*, juga  
   mensupport untuk memudahkan penulisan bahasa program.
10. *Web Editor*, misalnya *Dreamweaver* atau *Frontpage* (sifatnya optional) sebagai pendukung saja.
    1. **MVC**

*Model View Controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi 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 MVC *pattern* dalam suatu aplikasi yaitu :

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi *web* bagian ini biasanya berupa *file* *template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. Model, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, *controller* berfungsi untuk menerima *reques*t dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Singkat kata Model untuk mengatur alur *database*, *View* untuk menampilkan *web*, sedangkan *Controller* untuk mengatur alur kerja antara *Model* dan *View*.

[](http://situsali.com/wp-content/uploads/2015/07/mvc_konsep.png)

**Gambar 2.1** Proses Model-View-Controller

**2.9 HTML *(Hypertext markup Language)***

*Hypertext Markup Language* (HTML) adalah bahasa yang digunakan untuk menulis halaman web. HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language* (SGML). HTML sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

HTML dibuat oleh Tim *Berners-Lee* ketika masih bekerja untuk CERN dan dipopulerkan pertama kali oleh browser Mosaic. Selama awal tahun 1990 HTML mengalami perkembangan yang sangat pesat. Setiap pengembangan HTML pasti akan menambahkan kemampuan dan fasilitas yang lebih baik dari versi sebelumnya. Namun perkembangan resmi dikeluarkan pada bulan November 1995 oleh IETF (*Internet Engineering Task Force*).

HTML 2.0 ini merupakan penyempurnaan dari HTML+ (1993).HTML 3.0 (1995) memberikan kemampuan lebih daripada versi sebelumnya. Sebuah usaha dari *World Wide Web Consortium’s* (W3C) HTML Working Group pada tahun 1996 menghasilkan HTML 3.2. HTML versi ini secara resmi diterbitkan pada bulan Januari 1997.HTML versi terbaru adalah HTML 4.01 yang dikeluarkan secara resmi oleh W3C pada tanggal 24 April 1998. HTML merupakan perbaikan dari HTML 4.0 yang lebih dulu diterbitkan. (18 Desember 1997).

**2.10   CSS (*Cascading Style Sheet*)**

*Cascading Style Sheet* (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam.

Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

**2.11    Javascript**

JavaScript merupakan bahasa scripting sederhana yang secara khusus diciptakan untuk digunakan di dalam *web browser* yang membuat website menjadi lebih dinamis. Bila HTML berjalan sendirian, hanya mampu menghasilkan halaman-halaman statis. Setelah diload, tampilan halaman HTML tak banyak berubah sampai kamu mengklik sebuah link menuju halaman lain.

Banyak yang mengira JavaScript berhubungan dengan Java padahal keduanya tidak sama.  Java, yang diciptakan oleh  Sun Microsystems, merupakan bahasa pemrogaman komputer sepenuhnya seperti C++, cocok untuk penulisan program skala besar dan lengkap. JavaScript, di sisi lain, diciptakan oleh  Netscape. Ini berbasis pada suatu tingkatan di dalam Java – *syntax codenya* amat mirip, namun sangat jarang dipergunakan untuk keperluan di luar lingkup *browser*.

JavaScript ditulis dengan cara yang sama dengan HTML dalam *editor teks* (Wordpad, Notepad). Penerapan JS sama  persis dengan CSS kamu bisa mentautkan *link* ke file eksternal dengan ekstensi file .js, atau menulis blok kodenya langsung dalam dokumen HTML dengan tag. Jika kamu menggunakan script yang sama di dalam beberapa halaman, kamu dapat melinknya ke sebuah file JS *eksternal* atau bisa juga mengembed meletakkan kodenya langsung dalam halaman.

* 1. **Model Proses *Prototype***

Model proses ini dipilih dengan harapan mendapatkan umpan balik yang lebih awal dari pemakai. Dengan metode *Prototype* ini pengembang dan pemakai dapat saling berinteraksi selama proses pembuatan sistem. Berikut adalah penjelasan dari tahap-tahap pemodelan proses *Prototype* :

1. Mengidentifikasi kebutuhan pemakai

Pengembang mewawancarai pemakai untuk mendapatkan gagasan dari apa yang diinginkan pemakai terhadap sistem.

1. Membangun *Prototype*

**Pengembang membangun *prototyping* dengan membuat perancangan semantara yang berfokus pada penyajian kepada pemakai (misalnya dengan membuat format *input* dan *output*).**

1. Evaluasi *Prototype*

Evaluasi dilakukan oleh pemakai dengan menilai *prototype* yang telah dibuat oleh pengembang apakah sudah sesuai dengan keinginan pemakai. Jika sudah sesuai maka langkah selanjutnya dilakukan tetapi jika tidak sesuai maka *prototype* akan diperbaiki kembali dengan mengulang langkah sebelumnya.

1. Pengkodean

**Dalam tahap ini *Prototype* yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.**

1. Menguji Sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, maka harus di tes dahulu sebelum digunakan. Pengujian ini dilakukan dengan *Black Box*.

1. Evaluasi Sistem

Pada tahap ini, pemakai mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika sudah sesuai maka langkah selanjutnya dilakukan, akan tetapi jika tidak maka akan diperbaiki kembali dengan mengulang langkah Pengkodean dan Menguji Sistem.

1. Menggunakan Sistem

Perangkat lunak yang telah diuji dan diterima pemakai siap untuk digunakan.

**Gambar 2.2** Metode *Prototype* (Sumber: Raymond Mc.Leod, Jr, 2001)

**2.13 UML (Unified Modeling Language)**

Unified Modeling Language (UML) adalah himpunan [struktur](https://id.wikipedia.org/wiki/Struktur) dan [teknik](https://id.wikipedia.org/wiki/Teknik) untuk pemodelan [desain](https://id.wikipedia.org/wiki/Desain) program berorientasi objek ([OOP](https://id.wikipedia.org/wiki/Object_Oriented_Programming)) serta [aplikasinya](https://id.wikipedia.org/wiki/Aplikasi). UML adalah [metodologi](https://id.wikipedia.org/wiki/Metodologi) untuk mengembangkan [sistem](https://id.wikipedia.org/wiki/Sistem) OOP dan sekelompok perangkat [*tool*](https://id.wikipedia.org/wiki/Tool) untuk mendukung pengembangan sistem tersebut.[[1]](https://id.wikipedia.org/wiki/Unified_Modeling_Language#cite_note-dav-1) UML mulai diperkenalkan oleh [*Object Management Group*](https://id.wikipedia.org/w/index.php?title=Object_Management_Group&action=edit&redlink=1), sebuah [organisasi](https://id.wikipedia.org/wiki/Organisasi) yang telah mengembangkan [model](https://id.wikipedia.org/wiki/Model), [teknologi](https://id.wikipedia.org/wiki/Teknologi), dan standar OOP sejak tahun [1980](https://id.wikipedia.org/wiki/1980)-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari [IBM](https://id.wikipedia.org/wiki/IBM).

UML adalah suatu [bahasa](https://id.wikipedia.org/wiki/Bahasa) yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu [sistem informasi](https://id.wikipedia.org/wiki/Sistem_informasi). UML dikembangkan sebagai suatu alat untuk [analisis](https://id.wikipedia.org/wiki/Analisis) dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam [industri](https://id.wikipedia.org/wiki/Industri) terus meningkat.[[2]](https://id.wikipedia.org/wiki/Unified_Modeling_Language#cite_note-rj-2) Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri [peranti lunak](https://id.wikipedia.org/wiki/Peranti_lunak) dan pengembangan sistem.

UML menyediakan [10](https://id.wikipedia.org/wiki/10) macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

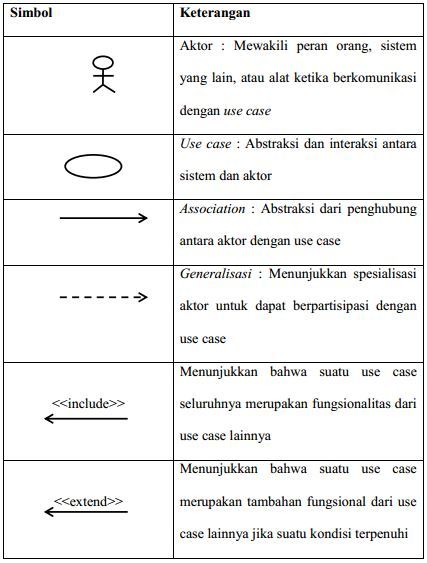
1. Use Case Diagram untuk memodelkan proses bisnis.
2. Sequence Diagram untuk memodelkan pengiriman pesan (*message*) antar *objects*.
3. [Activity Diagram](https://id.wikipedia.org/wiki/Diagram_Aktivitas) untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
4. [Class Diagram](https://id.wikipedia.org/wiki/Class_Diagram) untuk memodelkan struktur kelas.

Berikut akan dijelaskan 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi object, yaitu *use case diagram, sequence diagram, collaboration diagram, dan class diagram*.

1. Use Case Diagram

Use case diagram digunakan untuk memodelkan [semua bisnis](https://id.wikipedia.org/wiki/Bisnis) proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan *actor*. *Actor* merepresentasikan [orang](https://id.wikipedia.org/wiki/Orang) yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. Use case digambarkan berbentuk [elips](https://id.wikipedia.org/wiki/Elips) dengan [nama](https://id.wikipedia.org/wiki/Nama) [operasi](https://id.wikipedia.org/wiki/Operasi) dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke use case. Berikut symbol dalam Use Case Diagram



**Gambar 2.3** Simbol Use Case Diagram

1. Activity Diagram

**Activity diagram**, sesuai dengan namanya diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Struktur diagram ini juga mirip dengan flowchart. Fungsi Activity Diagram diagram :

### Menggambarkan  proses bisnis dan urutan aktivitas dalam sebuah proses

### Memperlihatkan urutan aktifitas proses pada sistem

1. Activity diagram dibuat berdasarkan sebuah atau beberapa use case pada use case diagram. Berikut adalah Element - Element Pada Activity Diagram.

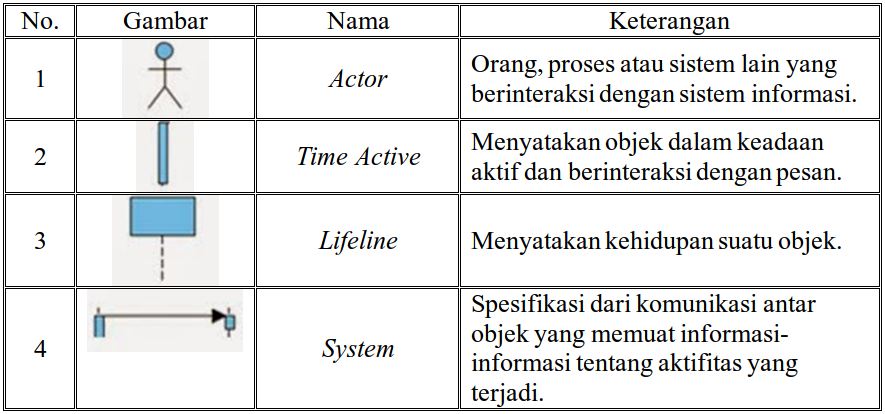


**Gambar 2.4** Simbol Acivity Diagram

1. Sequence Diagram

**Sequence Diagram** adalah salah satu dari diagram - diagram yang ada pada UML, sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

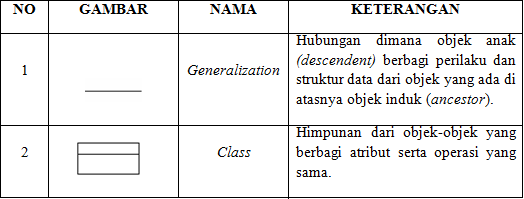
Dalam UML, object  pada  sequence diagram digambarkan dengan segi empat yang berisi nama dari object yang digarisbawahi. Pada object terdapat 3 cara untuk menamainya yaitu : nama object, nama object dan class, dan nama class.



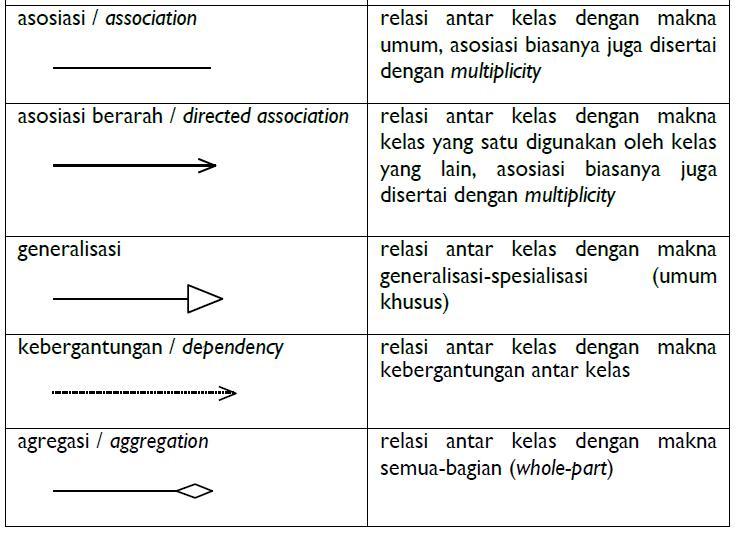
**Gambar 2.5** Simbol Sequence Diagram

## Class diagram

Class diagram menggambarkan struktur statis class di dalam sistem. class merepresentasikan sesuatu yang ditangani oleh sistem. class dapat berhubungan dengan yang lain melalui berbagai cara: associated (terhubung satu sama lain), dependent (satu class tergantung/menggunakan class yang lain), specialed (satu class merupakan spesialisasi dari class lainnya), atau package (group bersama sebagai satu unit). sebuah sistem biasanya mempunyai beberapa class diagram.



**Gambar 2.6** Simbol Class Diagram



**Gambar 2.7** Simbol Panah dalam Class Diagram

**2.14 Basis Data**

Basis data ([bahasa Inggris](https://id.wikipedia.org/wiki/Bahasa_Inggris): *database*) adalah kumpulan [informasi](https://id.wikipedia.org/wiki/Informasi) yang disimpan di dalam [komputer](https://id.wikipedia.org/wiki/Komputer) secara sistematik sehingga dapat diperiksa menggunakan suatu [program komputer](https://id.wikipedia.org/wiki/Program_komputer) untuk memperoleh informasi dari basis data tersebut. [Perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) yang digunakan untuk mengelola dan memanggil [kueri](https://id.wikipedia.org/wiki/Kueri) (*query*) basis data disebut [sistem manajemen basis data](https://id.wikipedia.org/wiki/Sistem_manajemen_basis_data) (*database management system*, DBMS). Sistem basis data dipelajari dalam [ilmu informasi](https://id.wikipedia.org/wiki/Ilmu_informasi).

Istilah *basis data* mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai *sistem manajemen basis data* (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti tersebut.

Jadi secara konsep basis data atau database adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tatacara yang tertentu untuk membentuk data baru atau informasi. Atau basis data (database) merupakan kumpulan dari data yang saling berhubungan (relasi) antara satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu. Pada komputer, basis data disimpan dalam perangkat hardware penyimpan, dan dengan software tertentu dimanipulasiunruk kepentingan atau keguanaan tertentu. Hubungan atau relasi data biasanya ditunjukkan dengan kunci (*key*) dari tiap file yang ada. Data merupakan fakta atau nilai (*value*) yang tercatat atau merepresentasikan deskripsi dari suatu objek. Data yang merupakan fakta yang tercatat dan selanjutnya dilakukan pengolahan (proses) menjadi bentuk yang berguna atau bermanfaat bagi pemakainya akan membentuk apa yang disebut informasi. Bentuk informasi yang kompleks dan teritegrasi dan pengolahan sebuah database dengan komputer akan digunakan untuk proses pengambilan keputusan pada manajemen akan membenuk Sistem Informasi Manajemen (SIM), data dalam basis data merupan item terkecil dan terpenting untuk membangun basis data yang baik dan valid. Data dalam basis data bersifat *integrated* dan *shared*:

1. Terpadu (*integrated*), berkas-berkas data yang ada pada basis data saling terkait (terjadi dependensi data);
2. Berbagi data (*shared*), data yang sama dapat dipakai oleh sejumlah pengguna dalam waktu yang bersamaan. Sering dinamakan sebagi sistem *multiuser*

Data merupakan suatu sumber yang sangat berguna bagi hampir disemua organisasi. Dengan tersedianya data yang melimpah, maka masalah pengaturan data secara efektif menjadi suatu hal yang sangat penting dalam pengembangan sistem informasi manajemen. Oleh karena itu, tujuan dari diadakannya pengaturan data adalah sebagai berikut:

1. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi saat sekarang dan masa akan datang.
2. Sebagai cara pemasukan data sehingga sehingga memudahkan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani
3. Pengendalian data untuk setiap siklus agar data selalu *up to date* dan dapat mencerminkan perubahan spesifik yang terjadi di setiap sistem.
4. Pengamanan data terhadap kemungkinan penambahan, modifikasi, pencurian, dan gangguan-gangguaan lain.

Suatu bangunan basis data memiliki jenjang sebagai berikut:

1. Karakter, merupakan bagian data terkecil yang berupa angka, huruf, atau karakter khusus yang membentuk sebuah item data atau field. Contoh A,B,X,Y,2,1,2,9,0,=,<,> dan sebagainya.
2. Field/item, merupakan representasi suatu atribut dan record (rekaman/tupel) yang sejenis yang menunjukkan suatu item dari data. Contoh *field* nama (berisi data nama-nama pegawai), field departemen (berisi data bagian atau spesifikasi pekerjaan), dan lain sebagainya.
3. Record/rekaman/tupel: Kumpulan dari *field* membentuk suatu *record* atau rekaman. Record menggambarkan suatu unit data individu yang tertentu. Contoh: file pegawai, dimana tiap-tiap *recordnya* berisi kumpulan data nama, alamat, departemen, yang dapat mewakili tiap-tiap data.
4. File, merupakan kumpulan dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Contoh file pegawai berisi data tentang semua yang berhubungan dengan pegawai seperti nama pegawai, alamat pegawai, departemen, yang dapat mewakili tiap-tiap data.
5. *Database*, merupakan kumupan dari file atau tabel yang membentuk suatu database. Contoh *database* pegawai PT Maju Terus terdiri atas file pegawai, file gaji, file golongan, dan sebagainya.

Dalam satu file terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari field yang saling berhubungan menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. Setiap nilai atau isi field memiliki kapasitas ruang atau lebar yang sama. Jenis isi data sebuah field harus sesuai dengan tipe datanya. Nama sebuah file harus menggambarkan isi dari data file tersebut. Untuk melengkapi definisi tentang file, dalam database dikenal nama entitas (*entity*) dan atribut. Entetis adalah orang, tempat, kejadian, atau konsep yang informasinya direkam. setiap entitas memiliki atribut atau sebutan untuk mewakili suatu entitas. Sebagai contoh dalam sistem perkuliahan; mahasiwa, matakuliah, pembayaran, dosen adalah sebagai entitas. Sedangkan entitas mahasiswa memiliki atribut nomor induk, nama, jurusan, dan sebagainya. Atau dari contoh di atas entitasnya adalah pegawai, ang memilik atribut NIP, nama, alamat, tgl\_lahir, jns\_kel Sistem basis data merupakan perpaduan antara basis data dan sistem manajemen basis data (SMBD). Database yang kompleks dan disertai dengan teknik pendokumentasian dan prosedur manipulasinya akan membentuk Sistem Manajemn Basis Data (*Database Management System-DBMS*). Singkatnya DBMS adalah database dan program untuk mengaksesnya.

Definisi di atas dapat menggambarkan pada kita bahwa basis data mempunyai beberapa kriteria penting, yaitu:

1. Berorientasi data dan bukan berorientasi program.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
3. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah.
5. Dapat digunakan dengan cara-cara yang berbeda.

Komponen-komponen sistem basis data adalah:

1. Hardware, sebagai pendukung operasi pengolahan data seperti CPU, memori, disk, terminal, dan sebagainya.
2. Software sistem operasi , (Windows 9x, Windows 2000/XP, Linux, Unix.)
3. Software pengelola basis data (DBMS) seperti MS-Access, SQL, Oracle.
4. Software program aplikasi misalnya Visual Basic, Delphi, Visual Foxpro.
5. Basis data (semua data yang diperlukan, dipelihara, dikelola oleh sistem Basis Data)
6. Pemakai/pengguna basis data (*user*).

Pengguna (user) basis data meliputi:

1. Database Administrator, yaitu pengguna yang memiliki kewenangan sebagai pusat pengendali seluruh sistem baik basis data maupun program-program yang mengaksesnya, menentukan pola struktur basis data, memodifikasi, membagi tugas pengolahan dan memberikan orientasi tertentu, dan sebagainya.
2. Aplication Programmers, yaitu programmer aplikasi yang berinteraksi dengan sistem melalui pemanggilan *Data Manipulation Language* (DML) yang dimasukkan ke dalam program yang tulis.
3. Sophisticated Users, yaitu pengguna yang berintraksi dengan sistem tapa harus menuliskan sendiri programnya, tetapi diganti dengan melakukan permintaan (*request*) dalam bentuk bahasa *query* basis data. Seperti menggunakan MS-Accsess, SQL, dan sebagainya.
4. Specialized Users, yaitu pengguna yang menuliskan program aplikasi basis data khusus yang tidak sesuai dengan *framework* pemrosesan data tradisional. Contoh: sistem pakar, multimedia, dan sebagainya.
5. Naive Users, yaitu kebanyakan pengguna yang berintraksi dengan sistem dengan cara memanggil salah satu program aplikasi yang telah disediakan. Contoh: operator pada bagian teller, personalia, dan sebagainya.
   * 1. **Peran Basis Data**

Basis data penting dalam pengolaan data dalam menggunakan komputer, karena bebrapa alasan yaitu:

1. Sebagai komponen utama atau penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
2. Menentukan kualitas informasi yang cepat, akurat, relevan, sehingga informasi yang disajikan tidak basi. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

Penyusunan suatu database digunakan untuk mengatasi masalah-masalah pada penyusunan data. Manfaat yang diperoleh dari penyusunan database yaitu untuk:

1. Mengatasi kerangkapan (*redundancy*) data. Penyimpanan data yang sama pada beberapa tempat selain bisa meyulitkan pemakai tentang aktualisasi data juga memboroskan tempat penyimpanan, maka basis data akan mendeteksi dan menghindari jika terjadi kerangkapan data.
2. Menghindari terjadinya inkonsistensi data. Akibat lain jika terjadi kerangkapan data, maka jika terjadi perubahan pada data yang satu sedangkan yang lain tidak dirubah akan terjadi ketidakonsistenan data. Untuk itu database akan menyesuaikan terhadap data yang sama apabila terjadi perubahan, jika semua data akan selalu ter-*update*
3. Mengatasi kesulitan dalam mengakses data. Memudahkan jika suatu saat akan diambil atau dicetak data yang memiliki kriteria tertentu, misalnya pada tanggal tertentu, alamat tertentu, jumlah tertentu, dan sebagainya.
4. Menyusun format yang standar dari sebuah data. Dta ayang sama pada file yang berbeda harus memiliki format data berupa tipe dan jangkauannya harus sama. Kedidaksamaan format data akan mengakibatkan sulit atau tidak bisanya pengaksesan data yang lain.
5. Penggunaan oleh banyak pemakai (*multiple user*). Sebuah database bisa dimanfaatkan sekaligus secara bersama oleh banyak pengguna (*multiuser*)
6. Melakukan perlindungan dan pengamanan data (*data security*). Setiap data hanya bisa diakses atas dimanipulasi oleh pihak yang diberi otoritas dengan memberikan *login* dan *password* terhadap masing-masing data.
7. Menyusun integritas dan independensi data. Basis data merupakan data kompleks yang bisa diintegrasikan, sehingga kita bisa memanipulasi untuk mendapatkan berbagai bentuk lembar kerja dan laporan yang kita inginkan. Namun demikian masing-masing data tidak saling tergantung terhadap data yang lain. Data-data dalam database bisa saling berdiri sendiri.
   * 1. **Abstraksi Data**

Kegunaan utama sistem basis data adalah agar pemakai mampu menyusun suatu pandangan (*view*) abstraksi data. Hal ini bertujuan untuk menyederhanakan intraksi antara pengguna dengan sistemnya dan basis data dapat mempresentasikan pandangan yang berbeda kepada para pengguna, programmer, dan administratornya. Karena tidak semua pengguna basis data terlatih dengan baik dan penggunanya terbagi dalam berbagai tingkatan, maka kompleksitas basis data akan tersembbunyi dari para pengguna melalui beberapa level abstraksi data. Ketika memandang basis data, pemakai dapat dikelompokkan menjadi 3 tingkatan (*level*) yaitu:

1. Level Fisik (*physical view/internal view*). Merupakan tingkatan terendah dalam abstraksi data yang menunjukkan bagaimana data disimpan dalam kondisi sebenarnya. Level ini merupakan bentuk paling kompleks, dimana struktur data level terendah digambarkan pada level ini.
2. Level Konseptual. Merupakan level yang menggambarkan data apa yang sebenarnya (secara fungsional) disimpan dalam basis data, beserta relasi yang terjadi antara data. Level ini menggambarkan keseluruhan database, dimana administrator basis data (DBA) membangun dan mengolah basis data, sedangkan pemakai tidak memperdulikan kerumitan dalam struktur level fisik lagi. Contohnya: pengguna akan mengetahui bahwa penjualan disimpan di dalam tabel barang, produksi, keuangan, marketing.
3. Level Pandangan Pemakai. Merupakan level dengan tingkatan tertinggi, yang menggambarkan hanya satu bagian dari keseluruhan database. Beberapa pengguna basis data tidak membutuhkan semua isi basis data misalkan bagian personalia hanya membutuhkan data file karyawan dan gaji, tidak membutuhkan data file gudang, transaksi barang masuk.

Level abstraksi data tersebut bisa memberikan gambaran atau pandangan mengenai arsitektur *Database Management System (DBMS), yang akan menambah pengertian mengenai independensi data (*data independence*)* Independensi data dapat dibagi menjadi dua bagian yaitu:

1. *Physical Data Independence*

Kemampuan untuk mengubah pola fisik database tanpa mengakibatkan suatu aplikasi program ditulis kembali. Modifikasi pada level fisik biasanya pada saat meningkatkan daya guna.

1. *Logical Data Independence*ata model Data model merupakan kumpulan konsep yang dapat digunakakn untuk menggambar struktur data. Struktur basis data meliputi tipe data, hubungan, dan beberapa syarat yang harus dipenuhi basis data.
   1. **MYSQL**

MySQL adalah sebuah [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) sistem manajemen [basis data](https://id.wikipedia.org/wiki/Basis_data) [SQL](https://id.wikipedia.org/wiki/SQL) ([bahasa Inggris](https://id.wikipedia.org/wiki/Bahasa_Inggris): *database management system*) atau DBMS yang [multialur](https://id.wikipedia.org/w/index.php?title=Multialur&action=edit&redlink=1), [multipengguna](https://id.wikipedia.org/wiki/Multipengguna), dengan sekitar 6 juta instalasi di seluruh dunia. [MySQL AB](https://id.wikipedia.org/w/index.php?title=MySQL_AB&action=edit&redlink=1) membuat MySQL tersedia sebagai [perangkat lunak gratis](https://id.wikipedia.org/wiki/Perangkat_lunak_gratis) di bawah lisensi [GNU General Public License](https://id.wikipedia.org/wiki/GNU_General_Public_License) (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti [Apache](https://id.wikipedia.org/wiki/Apache_HTTP_Server), di mana perangkat lunak dikembangkan oleh komunitas umum, dan [hak cipta](https://id.wikipedia.org/wiki/Hak_cipta) untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial [Swedia](https://id.wikipedia.org/wiki/Swedia) [MySQL AB](https://id.wikipedia.org/w/index.php?title=MySQL_AB&action=edit&redlink=1), di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang [Finlandia](https://id.wikipedia.org/wiki/Finlandia) yang mendirikan MySQL AB adalah: [David Axmark](https://id.wikipedia.org/w/index.php?title=David_Axmark&action=edit&redlink=1), [Allan Larsson](https://id.wikipedia.org/w/index.php?title=Allan_Larsson&action=edit&redlink=1), dan [Michael "Monty" Widenius](https://id.wikipedia.org/w/index.php?title=Michael_(Monty)_Widenius&action=edit&redlink=1).

## 2.15.1 Sistem manajemen basis data [relasional](https://id.wikipedia.org/w/index.php?title=Relasional&action=edit&redlink=1)

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional ([RDBMS](https://id.wikipedia.org/wiki/RDBMS)) yang didistribusikan secara gratis di bawah [lisensi GPL](https://id.wikipedia.org/wiki/GPL) (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; [SQL](https://id.wikipedia.org/wiki/SQL) (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata ([DBMS](https://id.wikipedia.org/wiki/DBMS)) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web ([wordpress](https://id.wikipedia.org/wiki/Wordpress)), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

* 1. **Xampp**

XAMPP adalah [perangkat lunak bebas](https://id.wikipedia.org/wiki/Perangkat_lunak_bebas), yang mendukung banyak [sistem operasi](https://id.wikipedia.org/wiki/Sistem_operasi), merupakan kompilasi dari beberapa [program](https://id.wikipedia.org/wiki/Program). Fungsinya adalah sebagai [server](https://id.wikipedia.org/wiki/Server) yang berdiri sendiri ([localhost](https://id.wikipedia.org/w/index.php?title=Localhost&action=edit&redlink=1)), yang terdiri atas program [Apache HTTP Server](https://id.wikipedia.org/wiki/Apache_HTTP_Server), [MySQL](https://id.wikipedia.org/wiki/MySQL) [database](https://id.wikipedia.org/wiki/Database), dan [penerjemah](https://id.wikipedia.org/wiki/Penerjemah) [bahasa](https://id.wikipedia.org/wiki/Bahasa) yang ditulis dengan [bahasa](https://id.wikipedia.org/wiki/Bahasa) [pemrograman](https://id.wikipedia.org/wiki/Pemrograman) [PHP](https://id.wikipedia.org/wiki/PHP) dan [Perl](https://id.wikipedia.org/wiki/Perl). Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), [Apache](https://id.wikipedia.org/wiki/Apache), [MySQL](https://id.wikipedia.org/wiki/MySQL), [PHP](https://id.wikipedia.org/wiki/PHP) dan [Perl](https://id.wikipedia.org/wiki/Perl). [Program](https://id.wikipedia.org/wiki/Program) ini tersedia dalam [GNU General Public License](https://id.wikipedia.org/wiki/GNU_General_Public_License) dan [bebas](https://id.wikipedia.org/wiki/Bebas), merupakan [web server](https://id.wikipedia.org/wiki/Web_server) yang mudah digunakan yang dapat melayani tampilan halaman [web](https://id.wikipedia.org/wiki/Web) yang dinamis. Untuk mendapatkanya dapat men[download](https://id.wikipedia.org/wiki/Download) langsung dari web resminya.

**BAB III**

**OBJEK PENELITIAN**

* 1. **Objek Penelitian**

**3.1.1 Sejarah Universitas Nasional Pasim**

YAPASIM berdiri pada tanggal [11](https://id.wikipedia.org/wiki/11) [Juni](https://id.wikipedia.org/wiki/Juni) [1996](https://id.wikipedia.org/wiki/1996) dengan akta notaris No 41 didepan [notaris](https://id.wikipedia.org/wiki/Notaris) Dr Wiratni Ahmadi SH. Pendirian YAPASIM ini ditujukan untuk menjadi lembaga pengelola [pendidikan](https://id.wikipedia.org/wiki/Pendidikan) tinggi. Oleh karena itu, setelah YAPASIM berdiri tidak berapa lama sesudah itu pendiri mengajukan izin kepada Direktur Jenderal Pendidikan Tinggi Departemen [Pendidikan](https://id.wikipedia.org/wiki/Pendidikan) dan Kebudayaan untuk mendirikan Sekolah Tinggi Ilmu Ekonomi yang diberi nama STIE PASIM.

Dan pada akhirnya permohonan tersebut dikabulkan oleh Menteri pendidikan dan kebudayaan pada tanggal [4](https://id.wikipedia.org/wiki/4) [Desember](https://id.wikipedia.org/wiki/Desember) [1996](https://id.wikipedia.org/wiki/1996) dengan dikeluarkannya SK Menteri Pendidikan dan Kebudayaan RI No.80/D/0/1996 tentang izin STIE PASIM yang terdiri dari dua jurusan yaitu jurusan [manajemen](https://id.wikipedia.org/wiki/Manajemen) dan jurusan [akuntansi](https://id.wikipedia.org/wiki/Akuntansi).

Seiring dengan berjalannya waktu dan perkembangan [teknologi informasi](https://id.wikipedia.org/wiki/Teknologi_informasi), di [Indonesia](https://id.wikipedia.org/wiki/Indonesia) khususnya banyak berdiri [sekolah tinggi](https://id.wikipedia.org/wiki/Sekolah_tinggi) bidang [informatika](https://id.wikipedia.org/wiki/Informatika) dan [komputer](https://id.wikipedia.org/wiki/Komputer). Untuk merespon perkembangan teknologi informasi tersebut serta memenuhi kebutuhan masyarakat, YAPASIM mengajukan izin kepada menteri pendidikan [nasional](https://id.wikipedia.org/wiki/Nasional) untuk mendirikan Sekolah Tinggi Manajemen Informatika dan Komputer yang diberi nama STMIK PASIM. Berdasarkan pada SK Menteri Pendidikan Nasional RI No 164/D/0/2000 tanggal [23](https://id.wikipedia.org/wiki/23) [Agustus](https://id.wikipedia.org/wiki/Agustus) [2000](https://id.wikipedia.org/wiki/2000) secara resmi berdirilah STMIK PASIM, sehingga di bawah pengelolaan YAPASIM ada dua sekolah tinggi yaitu STIE PASIM dan STMIK PASIM.

Tidak hanya berhenti sampai disitu, setelah beberapa tahun kemudian pendiri YAPASIM sangat berkeinginan untuk mendirikan [universitas](https://id.wikipedia.org/wiki/Universitas), namun pada saat itu izin untuk mendirikan universitas baru tidak dikeluarkan lagi oleh [Departemen Pendidikan Nasional](https://id.wikipedia.org/wiki/Departemen_Pendidikan_Nasional). Namun hal ini tidak menghalangi keinginan pendiri YAPASIM untuk mengelola universitas, tepatnya pada tanggal [25](https://id.wikipedia.org/wiki/25) [September](https://id.wikipedia.org/wiki/September) [2006](https://id.wikipedia.org/wiki/2006) pengelola Yayasan Pendidikan Kader Keuangan dan Perbankan yang mengelola Universitas Nasional Bandung menyerahkan pengelolaan Universitas Nasional Bandung kepada pendiri YAPASIM. Penyerahan tersebut dituangkan dalam akta notaris No [25](https://id.wikipedia.org/wiki/25) tanggal 25 [September](https://id.wikipedia.org/wiki/September) [2006](https://id.wikipedia.org/wiki/2006) didepan Notaris R. Sabar Partakoesoema, SH.MH.

Setelah berada di bawah pengelolaan YAPASIM, Universitas Nasional Bandung diajukan perubahan namanya kepada Menteri Pendidikan Nasional [RI](https://id.wikipedia.org/wiki/RI) menjadi Universitas Nasional Pasim (*Pengembangan Sistem Informasi Manajemen*). Pengajuan perubahan nama tersebut disetujui oleh Menteri Pendidikan Nasional RI dengan dikeluarkannya SK Menteri Pendidikan Nasional RI No 172/D/0/2007 pada tanggal [6](https://id.wikipedia.org/wiki/6) [September](https://id.wikipedia.org/wiki/September) [2007](https://id.wikipedia.org/wiki/2007) di bawah Yayasan Pendidikan Kader Keuangan Perbankan.

Merger dilakukan dikarenakan pada tahun sebelumnya UNB (YPKKP) memiliki defisit mahasiswa yang cukup besar. seluruh [mahasiswa](https://id.wikipedia.org/wiki/Mahasiswa) UNB yang terbagi kedalam 5 fakultas tidak mencapai 150 orang. Dalam keadaan tersebut pemilik yayasan (YPKKP) menjual salah satu gedung milik UNB kepada Infomedia. keadaan ini makin memperuncing konflik antara mahasiswa UNB dan pemilik yayasan, dengan dijualnya aset tersebut ada kemungkinan aset-aset lain akan dijual jika mahasiswa UNB kembali berkurang pada tahun berikutnya.

Wancana menjual UNB pun Bergulir, beberapa tawaran semisal, dari UK (Uniersitas Kebangsaan) untuk menjual fakultas ekonomi yang dimiliki UNB ditolak karena yayasan ingin menjual seluruh fakultas yang dimilikinya. Pada perkembangan selanjutnya, UNB akhirnya dibeli oleh Yapasim. seluruh mahasiswa ex-UNB diberikan opsi untuk ikut bergabung dengan management baru atau keluar selamanya dari UNB (atau Universitas Nasional Pasim).

**3.1.2 Mahasiswa D3 Manajemen Informatika**

    Objek Penelitian yang kami lakukan di Universitas Nasional Pasim ditujukan unutk Mahasiswa D3 Manajemen Informatika. Dalam jurusan ini terdapat dua jalur mahasiswa. Yang pertama yaitu Mahasiwa jalur Beasiswa Pemberdayaan Umat Berkelanjutan  dan yang kedua adalah Mahasiswa jalur Reguler .

1. Mahasiswa Pemberdayaan Umat Berkelanjutan (PUB)

Program pemberdayaan umat berkelanjutan (PUB) adalah program kerjasama UNIVERSITAS NASIONAL PASIM dengan Yayasan Marlina Buchari untuk menampung lulusan setingkat SMA yang Kurang mampu untuk melanjutkan pendidikannya di perguruan tinggi tetapi memiliki kecerdasan yang baik.

Untuk diterima sebagai mahasiswa program PUB, mereka harus mengikuti seleksi yang sangat ketat dan melibatkan para psikolog, mengingat program PUB ini dirancang untuk mencetak lulusan yang unggul dan memiliki mental yang baik.

Selama mengikuti program PUB, mahasiswa PUB ditempatkan di asrama dan seluruh biaya pendidikan dan akomodasi ditanggung Universitas Nasional Pasim, dalam mengikuti program PUB, mereka diwajibkan mengikuti aturan-aturan dan pendidikan ketat mulai dari kedisiplinan, mental spiritual, kemampuan khusus dibidang Teknologi Informasi, kemampuan bahasa asing, program magang, kemampuan bekerja dan kemampuan lainnya.

    Setiap harinya mahasiswa PUB melakukan pelatihan pemograman , bahasa inggris dan lain-lain. Pelatihan ini ditujukan untuk membantu mahasiswa dalam belajar di perkuliahan agar mahasiswa dapat mendapatkan IPK diatas 3.25.

Mahasiswa Reluger adalah mahasiswa yang mengikuti pembelajaran seperti biasa namun yang penulis maksud yang tidak mengikuti jalur beasiswa melainkan jalur umum. Dilihat dari kegiatannya sendiri mahasiswa PUB lebih banyak kegiatan belajar seperti pelatihan , namun untuk mahasiswa Reguler tidak. Sehingga mahasiswa memiliki lingkungan yang berbeda yang dapat mempengaruhi cara belajar.

**3.1.3 Visi**

Visi Universitas Nasional Pasim adalah : “To Make World Class Quality of Human Resources Development in Global Era of Information System” yang artinya membangun Sumber daya manusia berkelas dunia di era global sistem informasi.

**3.1.4 Misi**

Misi Universitas Nasional Pasim adalah : “The best curriculum, the best lecturers, the best facilities” yang artinya menyediakan kurikulum terbaik, pengajar terbaik, dan fasilitas terbaik.

**3.1.5 Struktur Organisasi**

Dibawah ini adalah struktur organisasi Universitas Nasional Pasim :



**Gambar 3.1** Struktur Organisasi Universitas Nasional Pasim

**3.2 Analisis Sistem**

Analisis sistem merupakan pendefinisian suatu sistem menjadi beberapa bagian dengan tujuan untuk mengidentifikasi suatu permasalahan serta mengevaluasinya agar didapatkan suatu sistem yang sesuai dengan kebutuhan.

Berdasarkan data-data yang telah diperoleh, kemudian dilakukan analisis pada sistem yang berjalan saat ini, kelemahan sistem yang berjalan, dan analisis sistem yang diinginkan serta analisis kebutuhan sistem.

**3.2.1 Analisis Sistem yang Berjalan**

Sistem yang sedang berjalan merupakan sistem yang sedang di pakai selama ini, Sedangkan Analisis Sistem yang sedang berjalan merupakan cara penulis untuk memahami sistem yang sedang berjalan, kalau penulis sudah memahami tentang sistem yang sedang berjalan maka akan terpikirkan oleh penulis mengenai apa yang dibutuhkan untuk membangun sistem yang memudahkan dan membantu sistem yang berjalan saat ini.

Salah satu manfaat dari Analisis yang sedang berjalan adalah untuk mendefinisikan kebutuhan fungsional dari sistem dan mencari tahu kebutuhan apa saja yang di butuhkan oleh pemakai dan kelemahan oleh sistem yang sedang berjalan tersebut.

Berdasarkan penjelasan dari dosen yang menjadi objek penelitian penulis sekaligus aktor yang menjalankan sistem, penulis mendapatkan beberapa objek inti dari penelitian masalah pada universitas dalam membangun e-learning dan fitur standar yang dibutuhkan dalam implementasi e-learning pada universitas. Berikut ini adalah analisis dari sistem yang sedang berjalan saat ini :

**3.2.1.1 Analisis Kegiatan Belajar Mengajar yang sedang Berjalan.**

1. Prosedur Kegiatan Belajar Mengajar di Kelas.

* Mahasiswa memasuki kelas.
* Apakah dosen ingin memberikan quiz?
* Jika iya maka dosen mulai memasuki ruangan Kelas dan memberikan quiz kepada mahasiswa.
* Mahasiswa mengerjakan quiz dalam waktu tertentu dan mengumpulkan quiz ketika waktunya sudah selesai.
* Jika quiz perlu dibahas maka dosen membahasnya. Tapi jika tidak, maka soal quiz di arsipkan dan
* Jika dosen tidak ingin memberikan quiz maka dosen langsung memberikan materi dan latihan kepada mahasiswa.
* Mahasiswa menyimak dan mengerjakan latihan dari dosen.

 **Gambar 3.2**

Flowmap Prosedur Kegiatan Belajar Mengajar di Kelas

1. Prosedur Kegiatan Belajar Mengajar di Lab

* Mahasiswa memasuki Lab.
* Asisten Dosen memasuki Lab dan memberi latihan dan juga materi
* Mahasiswa menerima materi , menyalakan komputer masing – masing dan mulai mengerjakan latihan yang diberikan Asisten Dosen.
* Asisten dosen memeriksa output latihan mahasiswa di komputer.

 **Gambar 3.3**

Flowmap Prosedur Kegiatan Belajar Mengajar di Lab

1. Prosedur pelaksanaan pemberian tugas

* Dosen membuat tugas untuk mahasiswa.
* Kemudian mahasiswa mengerjakan tugas yang diberikan oleh dosen.
* Dosen mengoreksi tugas dan mendapatkan nilai mahasiswa dan diarsipkan.



**Gambar 3.4** Flowmap Prosedur pelaksanaan pemberian tugas

1. Prosedur Mengoreksi Quiz

* Dosen mengambil soal quiz dan mulai mengoreksi
* Lalu soal quiz dan hasil quiz di arsipkan



**Gambar 3.5** Flowmap Prosedur Mengoreksi Quiz.

1. Hubungan dosen dengan Asisten dosen.

Asumsikan jika Asisten dosen sudah mendapatkan nilai ujian lab, yang didapatkan dari hasil kegiatan belajar di lab.

* Asisten dosen menyerahkan nilai ujian lab kepada dosen.
* Jika menurut dosen, banyak mahasiswa mendapatkan nilai yang bagus maka nilai tersebut akan digabungkan sebagai komponen nilai akhir
* Jika tidak maka dosen tidak memasukkan sebagai komponen tugas akhir.



**Gambar 3.6** Flowmap Hubungan dosen dengan Asisten dosen.

1. Prosedur pelaksanaan Asisten dosen membantu mengoreksi quiz

* Dosen mengambil quiz dan memberikan kepada asisten dosen.
* Asisten dosen mengoreksi soal dan memberi nilai quiz yang sudah dikoreksi.
* Dosen mengarsipkan nilai quiz dan soal quiz
* Dan dosen juga memasukkan nilai quiz ke daftar nilai mahasiswa.



**Gambar 3.7** Flowmap pelaksanaan Asisten dosen

membantu mengoreksi quiz

1. Prosedur pelaksanaan mengamati kemampuan mahasiswa berdasarkan nilai mahasiswa.

* Dosen menilai quiz, uts, dan uas.
* Dosen menyorting nilai secara descending.



**Gambar 3.8** Flowmap Pelaksanaan mengamati kemampuan mahasiswa berdasarkan nilai mahasiswa.

1. Prosedur pelaksanaan mengamati kemampuan mahasiswa berdasarkan pemahaman mahasiswa.

* Dosen memberikan Soal dan Menyuruh Mahasiswa Lagsung Mengerjakan Soal di Kelas.
* Mahasiswa langsung mengerjakan soal di papan tulis.



**Gambar 3.9** Flowmap pelaksanaan mengamat kemampuan mahasiswa berdasarkan pemahaman mahasiswa.

1. Prosedur pelaksanaan mengamati kemampuan mahasiswa secara langsung.

* Dosen mengamati mahasiswa secara langsung, apakah mahasiswa itu aktif atau pasif, dan suka mengajari yang lainnya atau tidak



**Gambar 3.10** Flowmap pelaksanaan mengamati kemampuan mahasiswa secara langsung

1. Prosedur penarikan kesimpulan baik buruk mahasiswa dalam

belajar.

* Mengumpulkan variabel dari prosedur 9,10,11
* Jika Nilai bagus dan mahasiswa bisa mengerjakan soal dengan baik, dan mahasiswa aktif , suka mengajari temannya. Maka dosen menilai mahasiswa itu bagus
* Jika tidak , maka mahasiswa kurang baik.



**Gambar 3.11** Flowmap penarikan kesimpulan baik buruk mahasiswa

**3.2.1.2 Analisis Kelemahan Sistem yang Berjalan**

1.Tidak ada modul khusus yang disediakan oleh universitas untuk latihan atau belajar mahasiswa dalam mata kuliah Logika dan Algoritma sehingga arah pembelajaran tidak terarah dengan baik.

2.    Tidak adanya SOP (Standar Operasional Prosedur) khusus untuk Asisten Dosen. Sehingga, asisten dosen tidak terlalu berkonsentrasi ke mahasiswa sebanyak sebagaimana asisten dosen yang membantu dosen dengan adanya SOP (read: profesional). Kurangnya feedback (tentang personal mahasiswa) yang didapatkan oleh dosen dari asisten dosennya. Sehingga hal tersebut menjadi salah satu faktor dosen tidak mengetahui secara pasti kemampuan masing-masing mahasiswa. Sedangkan untuk mata kuliah lanjutan, Pemrograman Teks 2, sudah tidak ada mahasiswa yang mempunyai kualifikasi untuk menjadi asisten dosen Proteks 2 yang masih tinggal di Pasim. Karena para mahasiswa PUB tahun akhir yang biasanya menjadi asdos, sudah selesai berkuliah. Hal tersebut bisa ditangani dengan meminta orang-orang IT alumni PUB untuk menjadi asisten dosen, tapi itu akan lebih menyusahkan dosen, yaitu mengajari mereka terlebih dahulu untuk mengajar peserta didik.

3.    Kelas laboratorium tidak profesional dan kurangnya fasilitas pendukung.

4.    Tahun demi tahun, penambahan mahasiswa semakin banyak. Hal tersebut membuat ruang kelas semakin terisi. Meskipun kelas Algoritma dan Logika sudah dipecahkan menjadi dua, mahasiswa masih tetap banyak. Akibatnya adalah kurangnya personalisasi pembelajaran terhadap mahasiswa. Semakin banyak mahasiswa dan semakin menurunnya personalisasi pembelajaran, berdampak pada waktu pembelajaran di kelas. Sangat baik dan merupakan metode yang efektif jika personalisasi pembelajaran diterapkan. Tapi jika personalisasi pembelajaran tetap diterapkan di kondisi tadi, akan membutuhkan waktu lebih. Sehingga terkadang sampai ada kelas lain yang harus masuk ke ruang itu karena waktu pembelajaran sudah selesai. Ketika dibutuhkan untuk pindah kelas, tidak ada kelas yang kosong. Selain itu, akibat yang timbul dari personalisasi pembelajaran yang rendah adalah dosen yang tidak mengetahui secara pasti mana mahasiswa *Good Learner*.

**Analisis PIECES**

|  |  |  |
| --- | --- | --- |
| **Analisis** | **Sistem Lama** | **Sistem Baru** |
| Kinerja (Performance) | * Tahun demi tahun, penambahan mahasiswa semakin banyak. Hal tersebut membuat ruang kelas semakin terisi. Meskipun kelas Algoritma dan Logika sudah dipecahkan menjadi dua, mahasiswa masih tetap banyak. Akibatnya adalah kurangnya personalisasi pembelajaran terhadap mahasiswa. Semakin banyak mahasiswa dan semakin menurunnya personalisasi pembelajaran, berdampak pada kurangnya pengetahuan dosen terhadap personal masing-masing mahasiswa. * Tidak adanya SOP (Standar Operasional Prosedur) khusus untuk Asisten Dosen. Sehingga, asisten dosen tidak terlalu berkonsentrasi ke mahasiswa sebanyak sebagaimana asisten dosen yang membantu dosen dengan adanya SOP (read: profesional). * Kurangnya feedback (tentang personal mahasiswa) yang didapatkan oleh dosen dari asisten dosennya. Sehingga hal tersebut menjadi salah satu faktor dosen tidak mengetahui secara pasti kemampuan masing-masing mahasiswa. * Kelas laboratorium tidak profesional dan kurangnya fasilitas pendukung. | Penerapan personalisasi dan adaptif pembelajaran pada sistem. |
| Informasi (Information) | * Kurangnya informasi yang diterima oleh dosen tentang performa dan kemampuan masing-masing mahasiswa. | Sistem dapat merekam data-data interaksi mahasiswa dengan sistem dan dengan adanya data tambahan dimana akan menjadi laporan untuk dosen. |
| Ekonomi (Economic) | * Jika dosen mendekati mahasiswa dan/atau bertanya pada asisten dosen untuk mendapatkan informasi dari masing-masing mahasiswa akan memakan biaya yang banyak. Misalnya transportasi, dll. | Penerapan personalisasi dan adaptif pembelajaran pada sistem. |
| Pengendalian (Control) | * Tidak ada modul khusus yang disediakan oleh universitas untuk latihan atau belajar mahasiswa dalam mata kuliah Logika dan Algoritma sehingga arah pembelajaran tidak terarah dengan baik. | Sistem menyediakan materi atau bahan ajar khusus sehingga mahasiswa akan terarah. |
| Efisiensi (Efficiency) | * Penerapan personalisasi pembelajaran di dalam kelas yang memakan waktu jika hanya dengan seorang dosen dan mahasiswa yang terlampau banyak. * Membutuhkan waktu yang sangat banyak jika dosen mendekati mahasiswa dan/atau bertanya pada asisten dosen untuk mendapatkan informasi dari masing-masing mahasiswa. | Penerapan personalisasi dan adaptif pembelajaran pada sistem. |
| Pelayanan (Service) | * Kelas laboratorium kurang profesional dan kurangnya fasilitas pendukung. | Sistem menyediakan latihan-latihan soal dimana akan membantu kelas laboratorium. |

Tabel 3.1 Analisis PIECES

**3.2.2 Analisis Kebutuhan Sistem**

**3.2.2.1 Analisis Kebutuhan Fungsional**

Dibutuhkan sebuah aplikasi *E-Learning* untuk mendampingi dosen. Dimana aplikasi tersebut memiliki kemampuan sebagai

berikut :

* + 1. Menyediakan bahan ajar (modul) yang dibutuhkan mata kuliah

Logika dan Algoritma dasar sehingga mahasiswa belajar lebih

terarah.

1. Memberikan *feedback* personal mahasiswa yang akan membantu dosen mengetahui kemampuan personal mahasiswa. Sehingga dosen dapat mengajar lebih baik.
2. Aplikasi mampu beradaptasi dan bersifat personal terhadap

penggunanya.

4.     Menemukan jalur dari *Good Learner*. Sehingga aplikasi bisa mengetahui mana yang *Good Learner* dan merekomendasikan kepada dosen, jalur *Good Learner* mana yang sesuai bagi seorang mahasiswa *Low Learner*.

**3.2.2.2 Analisis Kebutuhan Non Fungsional**

Analisis kebutuhan Non Fungsional yang dibutuhkan untuk membangun aplikasi adalah sebagai berikut :

* 1. **Kebutuhan Perangkat Keras**

Perangkat keras yang dibutuhkan adalah sebagai berikut:

1. Personal Computer(PC), untuk menjalankan aplikasi kita membutuhkan PC yang minimal memiliki RAM sebesar 1GB.
2. Disk sekitar 160Mb untuk menyimpan kode *moodle* dan 5 GB untuk menyimpan materi.
3. Personal Computer (PC) untuk server.
4. Koneksi internet yang bagus.
   1. **Kebutuhan Perangkat Lunak**
   2. Sublime/Notepad++, Untuk mengetikkan kode - kode yang dibutuhkan untuk membuat aplikasi.
   3. Moodle, Merupakan Open Source yang akan penulis kembangkan
5. Xampp, sebagai server database  lokal yang penulis gunakan
6. Microsoft Visual C++.
7. Browser : Mozila Firefox , Chrome ataupun yang lainnya.
   1. **Kebutuhan Pengguna**

Merupakan penjelasan tentang pengguna yang akan menjalankan aplikasi yang akan di bangun, dan kritetria pengguna tersebut adalah sebagai berikut :

* Mahasiswa / Dosen.
* Bisa mengoperasikan komputer.
* Memiliki pengalaman dan bisa mengoperasikan web browser.
* Mempunyai akses internet.

**BAB IV**

**PERANCANGAN SISTEM**

1. **Perancangan Sistem**

Perancangan sistem merupakan suatu proses pembuatan rancangan bangun suatu sistem yang bertujuan untuk memetakan setiap kebutuhan sistem sehingga proses pembangunannya dapat berjalan dengan baik. Dalam proses ini didefinisikan kebutuhan-kebutuhan fungsional dan teknikal dari sistem yang akan dibangun.

Pembangunan sistem informasi HRD Manajemen yang diusulkan disesuaikan dengan sistem yang telah ada sebelumnya. Perubahan yang dilakukan tidak banyak mempengaruhi prosedur kerja sistem yang telah berjalan. Perubahan lebih difokuskan pada proses, penyimpanan data yang terkomputerisasi dan efektifitas data.

* + 1. **Prosedure Pengaksesan Aplikasi**

1. **Prosedur Kategori Administrator**

Hak akses administrator diberikan kepada petugas untuk mengelola sistem secara keseluruhan. Administrator mempunyai akses untuk mengelola setiap data master antara lain data course, dan data user,dll.

1. **Prosedur Kategori *Dosen***

Hak akses Dosen adalah membuat quiz, mengupload materi, mengelola report dan mengupload nilai. Dosen juga dapat membuat course seperti yang dilakukan Administrator.

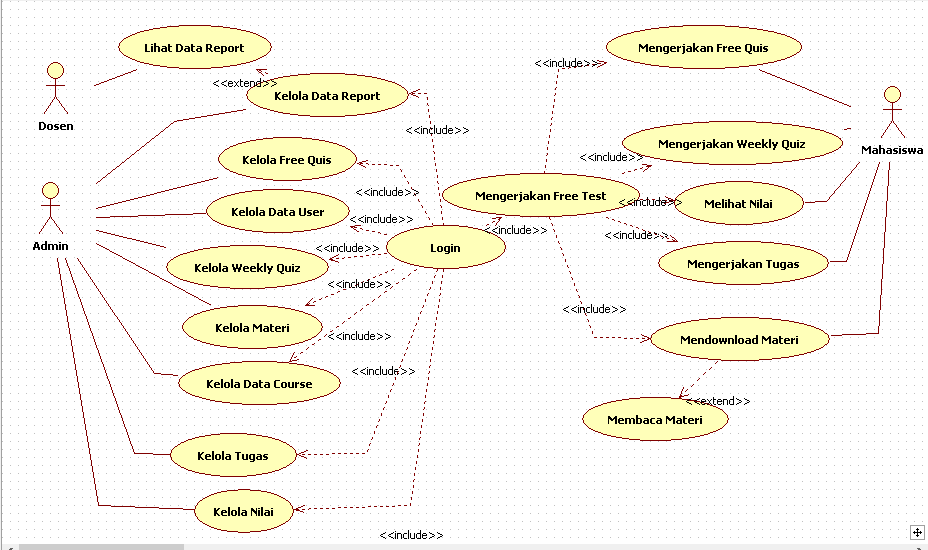
1. **Mahasiswa**

Hak Akses bagi Mahasiswa adalah mengerjakan quiz-quiz dan tugas yang terdapat dalam aplikasi, membaca materi dan mendownload materi dan juga melihat nilai.

* + 1. **Metode Perancangan Sistem**

Perancangan apliakasi desktop menggunakan desain berorientasi objek dengan menggunakan pendekatan object oriented method. Untuk membantu proses perancangan sistem berorientasi objek, maka digunakan 4 (empat) diagram yaitu use case diagram, activity diagram, sequence diagram dan class diagram.

1. **Use Case Diagram**



**Gambar 4.1** *Use Case Diagram*

**Use Case Spesification**

1. **Nama Use Case :** *Login*

**Deskripsi Singkat**

Use Case *Login* dapat diakses oleh Administrator, Dosen dan Mahasiswa. Namun untuk menunya berbeda untuk masing-masing user.

**Basic Flow**

1. Ketika aplikasi pertama kali dijalankan maka aplikasi akan menampilkan form login.
2. User mengisi *username* dan *password*.
3. *Username* dan *password* valid, sistem menampilkan halaman utama dengan beberapa menu.

**Alternate Flow :** Jika *username* dan *password* tidak valid maka

pengguna tidak dapat mengakses sistem.

**Pre-Condition :** Pengguna dalam keadaan logout.

**Post-Condition :** Tampil halaman utama.

1. **Nama Use Case :** Mengelola *Course*

**Deskripsi Singkat**

Use case ini digunakan oleh Administrator untuk mengelola data-data *course* yang mempunyai hak akses untuk dapat mengelola sistem. Fungsi-fungsi dalam use case ini yaitu tambah data, ubah data dan hapus data.

**Basic Flow**

1. **Tambah Data**
2. Administrator memilih Menu Admin pada menu Master Data.
3. Sistem memuat dan menampilkan data-data *course* dan yang terdapat dalam sistem.
4. Administrator mengklik tombol tambah.
5. Sistem menampilkan form tambah *course*.
6. Administrator mengisi data *course*.
7. Sistem menyimpan data.
8. **Ubah Data**
9. Administrator memilih Menu Admin pada menu Master Data.
10. Sistem memuat dan menampilkan data-data *course* yang terdapat dalam sistem.
11. Administrator mengklik tombol ubah.
12. Sistem menampilkan form ubah data *course*.
13. Administrator mengubah data *course*.
14. Sistem berhasil mengubah data *course*.
15. **Hapus Data**
16. Administrator memilih Menu Admin pada menu Master Data.
17. Sistem memuat dan menampilkan data-data *course* yang terdapat dalam sistem.
18. Administrator mengklik tombol hapus.
19. Sistem menampilkan konfirmasi hapus data *course*.
20. Administrator mengklik tombol ‘*Yes*’.
21. Sistem menghapus data *course*.

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Data *Course* sudah dimanipulasi.

1. **Nama Use Case :** Kelola Data *User*

**Deskripsi Singkat**

Use case ini digunakan oleh Administrator untuk mengelola data-data *user* pada sistem. Fungsi-fungsi dalam use case ini yaitu tambah data, ubah data dan hapus data.

**Basic Flow**

1. **Tambah Data**
2. Administrator memilih Menu *User* pada menu Master Data.
3. Sistem memuat dan menampilkan data-data *user* yang terdapat dalam sistem.
4. Administrator mengklik tombol tambah.
5. Sistem menampilkan formulir tambah *user*.
6. Administrator mengisi data *user*.
7. Sistem menyimpan data *user*.
8. **Ubah Data**
9. Administrator memilih Menu User pada menu Master Data.
10. Sistem memuat dan menampilkan data-data *user* yang terdapat dalam sistem.
11. Administrator mengklik tombol ubah.
12. Sistem menampilkan formulir ubah data *user*.
13. Administrator mengubah data *user*.
14. Sistem berhasil mengubah data *user*.
15. **Hapus Data**
16. Administrator memilih Menu *User* pada menu Master Data.
17. Sistem memuat dan menampilkan data-data *user* yang terdapat dalam sistem.
18. Administrator mengklik tombol hapus.
19. Sistem menampilkan konfirmasi hapus data *user*.
20. Administrator mengklik tombol ‘Yes’.
21. Sistem menghapus data *user*.

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Data *user* sudah dimanipulasi.

1. **Nama Use Case :** Kelola Data *Report*

**Deskripsi Singkat**

Use case ini digunakan oleh Dosen untuk mengelola data-data hasil / nilai latihan yang dikerjakan oleh Mahasiswa. *Report* tersebut dapat di*download* dari sistem. Fungsi-fungsi dalam use case ini yaitu Buat *report* berdasarkan waktu dan men*download* *report* tersebut.

**Basic Flow**

* 1. **Membuat Report**
     1. Dosen membuka tampilan nilai berdasarkan latihan yang dilakukan mahasiswa.
     2. Dosen Memilih berdasarkan waktu
     3. Dosen memilih tombol Buat *Report*
  2. **Mendownload Report**
     1. Setelah dosen membuat *report* maka Dosen memilih tombol *Download*
     2. Sistem akan memunculkan *Directory* yang harus dipilih oleh Dosen sebelum menyimpan *Report*
     3. Dosen memilih Directory dan memilih “*Save*”
     4. File *Report* yang sesuai dengan waktu dan latihan yang dipilih oleh Dosen akan di*download* dan tersimpan di file

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** File Report sudah didownload.

1. **Nama Use Case :** Kelola Materi

**Deskripsi Singkat**

Use case ini digunakan oleh Dosen untuk mengelola Materi yang sesuai dengan yang diberikan Dosen untuk membantu mengerjakan Quiz maupun Tugas. Dalam use case ini kegiatannya adalah menambah/mengupload menghapus, dan mengupdate materi

**Basic Flow**

* 1. **Membuat Materi**
     1. Dosen membuka tampilan course ,kemudian Dosen memilih Pembahasan yang sesuai dengan materi yang akan di upload ke system.
     2. Dosen memilih Tambah materi dan mengupload materi
     3. Setelah memilih materi , dosen memilih simpan
  2. **Menghapus Materi**
     1. Dosen di tampilan *course* dan memlih pembahasan yang terdapat materi yang ingin dihapus
     2. Dosen memilih Materi yang ingin dihapus
     3. Dosen memilih hapus
  3. **Mengedit Materi** 
     1. Sistem menampilkan tampilan course dann pembahasan yang sesuai dengan materi yang ingin diedit
     2. Jika materi bentuk Microsoft word atau Microsoft Powerpoint maka Dosen dapt memilih edit
     3. Dosen mengedit materi
     4. Dosen memilih simpan

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Data materi dapat dimanipulasi.

1. **Nama Use Case :** Kelola Nilai

**Deskripsi Singkat**

Use case ini digunakan oleh Dosen untuk mengelola Akumulasi dari keseluruhan quiz maupun tugas oleh mahasiswa. Dosen juga kan mnegikutsertkakan nilai UTS dan UAS yang akan diinputkan oleh Dosen . Kegiatannya sendiri seperti Menambah Nilai UTS dan UAS, Menambah Nilai Quiz dan Tugas yang akan ditampilkan dan dihitung oleh Sistem, dan Mengupload Nilai Akhir.

**Basic Flow**

* 1. **Menambah Nilai UTS dan UAS**
     1. Dosen membuka tampilan Kelola Nilai
     2. Dosen memilih Tambah nilai UTS
     3. Kemudian Dosen mneginputkan dan memilih save
     4. Dosen memilih Tambah nilai UAS
     5. Dosen menginputkan Nilai UAS dan memilih Simpan
  2. **Menambah Nilai Quiz dan Tugas**
     1. Dosen di tampilan Kelola Nilai
     2. Dosen memilih Tambah Akumulasi Nilai Quiz
     3. Sistem menampilkan nilai rata-rata Quiz
     4. Dosen memilih Tambah Akumulasi Nilai Tugas
     5. Sistem menampilkan nilai rata-rata tugas
     6. Dosen memilih Simpan
  3. **Mengupload Nilai**
     1. Dosen di Tampilan Nilai
     2. Dosen memilih tambah persentase dan kemudian menambahkan persentase yang akan diambil dari nilai Quiz,Tugas,UTS dan UAS
     3. Dosen memilih Simpan
     4. Sistem menampilkan nilai sesuai persentase yang diinputkan dosen
     5. Dosen memilih Upload Nilai

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Nilai telah di upload

1. **Nama Use Case :** Kelola Weekly Quiz

**Deskripsi Singkat**

Use case ini digunakan oleh Dosen untuk mengelola Weekly Quiz yang akan diberikan kepada Mahasiswa. Kegiatannya adalah Menambah , dan Mengedit Weekly Quiz.

**Basic Flow**

* 1. **Membuat Weekly Quiz**
     1. Dosen membuka tampilan course ,kemudian Dosen memilih Weekly Quiz
     2. Dosen menginputkan atau dapat juga mengupload Weekly Quiz
     3. Dosen memilih simpan
  2. **Mengedit Weekly Quiz**
     1. Sistem menampilkan tampilan course dan memilih Weekly Quiz
     2. Dosen memilih Bank Soal
     3. Dosen melakukan pencarian
     4. Dosen mengedit soal
     5. Dosen memilih simpan

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Data telah dimanipulasi

1. **Nama Use Case :** Kelola Free Quiz

**Deskripsi Singkat**

Use case ini digunakan oleh Dosen untuk mengelola Free Quiz yang akan diberikan kepada Mahasiswa. Kegiatannya adalah Menambah dan Mengedit Free Quiz

**Basic Flow**

* 1. **Membuat Free Quiz**
     1. Dosen membuka tampilan course ,kemudian Dosen memilih Free Quiz
     2. Dosen menginputkan soal Free Quiz
     3. Dosen memilih simpan
  2. **Mengedit Free Quiz**
     1. Sistem menampilkan tampilan course dan memilih Free Quiz
     2. Dosen memilih Bank Soal
     3. Dosen melakukan pencarian
     4. Dosen mengedit soal
     5. Dosen memilih simpan

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Data telah dimanipulasi

1. **Nama Use Case :** Kelola Tugas

**Deskripsi Singkat**

Use case ini digunakan oleh Dosen untuk mengelola Tugas yang akan diberikan kepada Mahasiswa. Kegiatannya adalah Menambah/Upload , Menghapus dan Mengedit Tugas.

**Basic Flow**

* 1. **Membuat Tugas**
     1. Dosen membuka halaman utama
     2. Dosen memilih Tambah / mengupload Tugas
     3. Dosen melakukan pengaturan seperti Kelas dan Waktu
     4. Dosen memilih simpan/Upload Tugas akan dishare kepada Mahasiswa
  2. **Menghapus Tugas**
     1. Dosen di tampilan halaman utama dan Dosen memilih Menu tugas
     2. Dosen memilih Kumpulan Tugas
     3. Dosen memilih hapus
     4. Dosen memilih “Yes” untuk menghapus
     5. Data Tugas telah dihapus
  3. **Mengedit Tugas**
     1. Sistem menampilkan tampilan halaman utama Dosen memilih Menu Tugas
     2. Dosen memilih Menu Kumpulan Tugas
     3. Dosen mencari Tugas
     4. Dosen memilih edit
     5. Dosen memilih simpan

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Data telah dimanipulasi

1. **Nama Use Case :** Mengerjakan Tugas

**Deskripsi Singkat**

Use case ini digunakan oleh Mahasiswa dalam mengerjakan Tugas. Kegiatannya adalah Mengerjakan dan Submit soal.

**Basic Flow**

* 1. **Mengerjakan dan Submit soal**
     1. Mahasiswa membuka halaman utama
     2. Mahasiswa memilih menu Tugas
     3. Jika terdapat tugas maka Mahasiswa membuka tugas yang ada di tampilan menu Tugas
     4. Mahasiswa membuka soal dan mengerjakan soal
     5. Tugas selesai, Mahasiswa Submit Tugas

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Tugas telah disubmit

1. **Nama Use Case :** Mengerjakan Free Quiz

**Deskripsi Singkat**

Use case ini digunakan oleh Mahasis dalam mengerjakan Free Quiz. Kegiatannya adalah Mengerjakan dan Submit Quiz.

**Basic Flow**

* 1. **Mengerjakan dan Submit soal**
     1. Mahasiswa membuka halaman utama
     2. Mahasiswa memilih menu Free Quiz
     3. Mahasiswa membuka Menu Free Quiz yang belum dikerjakan dan sesuai levelnya
     4. Mahasiswa mengerjakan soal sesuai waktu
     5. Mahasiswa dapat melihat materi yang tersediasesuai dengan soal
     6. Jika waktu habis maka sistem akan langsung mensubmit dan menutup soal

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Free Quiz telah disubmit.

1. **Nama Use Case :** Mengerjakan Weekly Quiz

**Deskripsi Singkat**

Use case ini digunakan oleh Mahasis dalam mengerjakan Weekly Quiz. Kegiatannya adalah Mengerjakan dan Submit Quiz.

**Basic Flow**

* + 1. **Mengerjakan dan Submit soal**
       1. Mahasiswa membuka halaman utama
       2. Mahasiswa memilih menu Weekly Quiz
       3. Mahasiswa membuka Menu Weekly Quiz sesuai dengan pertemuan
       4. Mahasiswa mengerjakan soal sesuai waktu
       5. Jika waktu habis mahasiswa maka akan langsung mensubmit dan menutup soal

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Weekly Quiz Telah di submit .

1. **Nama Use Case :** Mendownload Materi

**Deskripsi Singkat**

Use case ini digunakan oleh Mahasis dalam mendapatkan Materi mengenai Course Kegiatannya adalah Mendownload Materi

**Basic Flow**

1. **Membaca dan mendownload Materi**
   * 1. Mahasiswa membuka halaman utama
     2. Mahasiswa memilih menu Materi
     3. Mahasiswa mencari materi membuka Menu Free Quiz yang belum dikerjakan dan sesuai levelnya
     4. Mahasiswa mengerjakan soal sesuai waktu
     5. Mahasiswa dapat melihat materi yang tersedia sesuai dengan soal
     6. Sistem menampilkan materi
     7. Mahasiswa memilih download
     8. Mahasiswa menentukan directory penyimpanan untuk materi yang akan didownload
     9. Mahasiswa memilih simpan

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** File Materi telah di Download.

1. **Nama Use Case :** Melihat Nilai

**Deskripsi Singkat**

Use case ini digunakan oleh Mahasiswa dalam Melihat nilai dari keseluruhan latihan yang dikerjakan siswa. Kegiatannya adalah Melihat Nilai.

**Basic Flow**

1. **Melihat Nilai**
   * 1. Melihat nilai Mahasiswa membuka halaman utama
     2. Mahasiswa memilih menu Penilaian
     3. Sistem menampilkan penilaian latihan yang dilakukan mahasiswa sesuai dengan persentase yang diberikan oleh Dosen

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Sistem menampilkan nilai mahasiswa

1. **Nama Use Case :** Membaca Materi

**Deskripsi Singkat**

Use case ini digunakan oleh Mahasiswa dalam membaca materi. Kegiatannya adalah Membaca Materi.

**Basic Flow**

1. **Membaca Materi**
   * 1. Mahasiswa membuka halaman utama
     2. Mahasiswa memilih menu Materi
     3. Mahasiswa mencari materi membuka Menu Free Quiz yang belum dikerjakan dan sesuai levelnya
     4. Mahasiswa mengerjakan soal sesuai waktu
     5. Mahasiswa dapat melihat materi yang tersedia sesuai dengan soal
     6. Sistem menampilkan materi
     7. Mahasiswa memilih buka materi dan mahasiswa membaca materi

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Sistem menampilkan materi

1. **Nama Use Case :** Mengerjakan *Free Test*

**Deskripsi Singkat**

Use case ini digunakan oleh Mahasiswa dalam mengerjakan *Free Test* yang di sediakan di aplikasi.

**Basic Flow**

1. **Mengerjakan *Free Test***
   * 1. Mahasiswa membuka halaman utama
     2. Mahasiswa memilih *Free Test*
     3. Mengerjakan soal *Free Test* sesuai dengan waktu yang di tentukan.

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Administrator dalam keadaaan login.

**Post-Condition :** Sistem menampilkan halaman *Free Test*

1. **Nama Use Case :** Lihat Laporan

**Deskripsi Singkat**

Use case ini digunakan oleh Dosen dalam melihat laporan dari setiap mahasiswa yang terdaftar.

**Basic Flow**

1. **Lihat Laporan**
   * 1. Dosen membuka halaman utama
     2. Dosen memilih menu laporan
     3. Dosen melihat laporan aktifitas mahasiswa.

**Alternate Flow :** Tidak ada.

**Pre-Condition :** Dosen dalam keadaaan login.

**Post-Condition :** Sistem menampilkan laporan

* + 1. **Activity Diagram**

**4.2.2.1 Activity - Login**

****

**Gambar 4.2** *Activity Diagram Login*

**4.2.2.2 Activity – Kelola Tugas**

****

**Gambar 4.3** *Activity Diagram Kelola tugas*

**4.2.2.3 Activity – Kelola Free Quiz**

****

**Gambar 4.4** *Activity Diagram Kelola Free Quiz*

**4.2.2.4 Activity – Kelola Weekly Quiz**

****

**Gambar 4.5** *Activity Diagram Kelola Weekly Quiz*

**4.2.2.5 Activity – Kelola Report**

****

**Gambar 4.6** *Activity Diagram Kelola Report*

**4.2.2.6 Activity – Kelola User**

****

**Gambar 4.7** *Activity Diagram Kelola User*

**4.2.2.7 Activity – Kelola Course**

****

**Gambar 4.8** *Activity Diagram Kelola Course*

**4.2.2.8 Activity – Mengerjakan Weekly Quiz**

****

**Gambar 4.9** *Activity Diagram Mengerjakan Weekly Quiz*

**4.2.2.9 Acitivity – Mengerjakan Tugas**

****

**Gambar 4.10** *Activity Diagram Mengerjakan Tugas*

**4.2.2.10 Activity – Mengerjakan Free Quiz**

****

**Gambar 4.11** *Activity Diagram Mengerjakan Free Quiz*

**4.2.2.11 Activity – Mendownload Materi**

****

**Gambar 4.12** *Activity Diagram Mendownload Materi*

**4.2.2.12 Activity – Melihat Nilai**

****

**Gambar 4.13** *Activity Diagram Melihat Nilai*

**4.2.2.13 Activity – Kelola Materi**

****

**Gambar 4.14** *Activity Diagram Kelola Materi*

**4.2.2.14 Activity – Kelola Nilai**

****

**Gambar 4.15** *Activity Diagram Kelola Nilai*

**4.2.2.14 Activity – Membaca Materi**

****

**Gambar 4.15** *Activity Diagram Membaca Materi*

**4.2.2.15 Activity – Jalankan Fungsi**

****

**Gambar 4.16** *Activity Jalankan fungsi*

* + 1. **Sequence Diagram**
       1. **Sequence Diagram – Login**

****

**Gambar 4.17** *Sequence Diagram Login*

* + - 1. **Sequence Diagram – Tambah**



Sistem E-Learning

**Gambar 4.18** *Sequence Diagram Tambah*

* + - 1. **Sequence Diagram – Ubah**



Sistem

E-Learning

**Gambar 4.19** *Sequence Diagram Ubah*

* + - 1. **Sequence Diagram – Hapus**



Sistem

E-Learning

**Gambar 4.20** *Sequence Diagram Hapus*

**4.2.3.5 Sequence Diagram - Melihat Nilai**

****

**Gambar 4.21** *Sequence Diagram Melihat Nilai*

**4.2.3.6 Sequence Diagram – Mengelola Profile**

****

Sistem

E-Learning

Sistem

E-Learning

Sistem

E-Learning

Sistem

E-Learning

**Gambar 4.22** *Sequence Diagram Mengelola Profile*

* + - 1. **Sequence Diagram Menjawab Soal**



**Gambar 4.23** *Sequence Diagram Menjawab Soal*

* + - 1. **Sequence Diagram – Mendownload Materi**



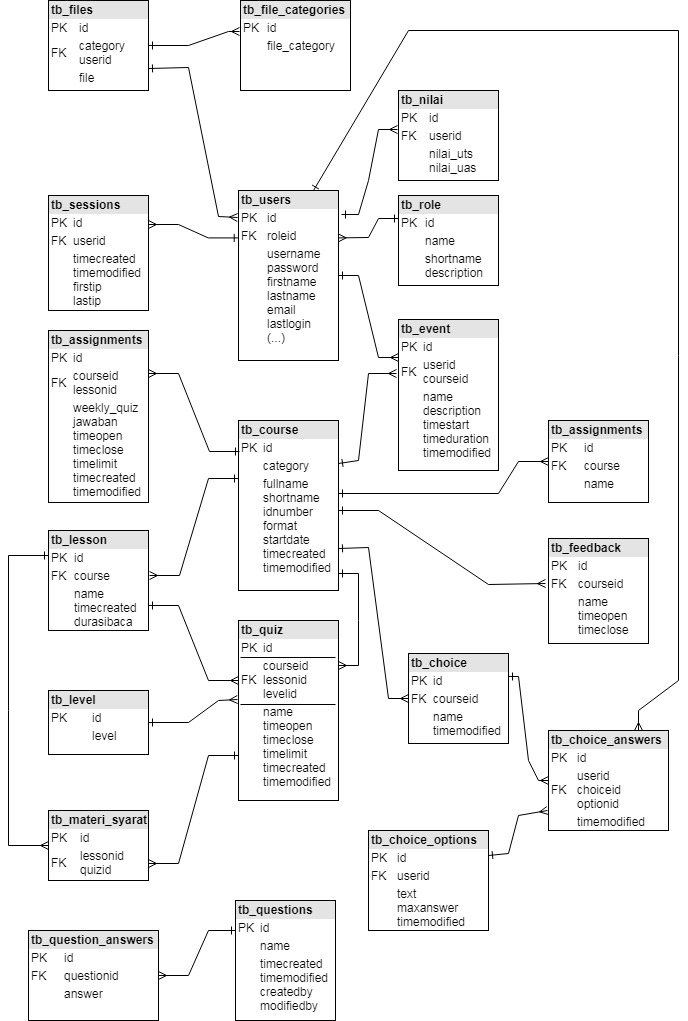
**Gambar 4.24** *Sequence Diagram Mendownload Materi*

* + 1. **Class Diagram**

****

**Gambar 4.25** *Class Diagram Sistem E-Learning*

* 1. **Perancangan Basis Data**
     1. **Skema Relasi Basis Data**



**Gambar 4.26** *Schema Sistem E-Learning*

* + 1. **Deskripsi Tabel**
  1. **Tabel 1 – tb\_course**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| Id | BIGINT | 10 | PRIMARY KEY,  AUTO INCREMENT |
| Category | BIGINT | 10 | - |
| Fullname | VARCHAR | 254 | - |
| Shortname | VARCHAR | 255 | - |
| Idnumber | VARCHAR | 100 | - |
| Format | VARCHAR | 21 | - |
| Startdate | BIGINT | 10 | - |
| Timecreated | BIGINT | 10 | - |
| Timemodified | BIGINT | 10 | - |

**Tabel 4.1** Deskripsi Tabel tb\_course

* 1. **Tabel 2 – tb\_user**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| Id | BIGINT | 10 | PRIMARY KEY,  AUTO INCREMENT |
| Username | VARCHAR | 100 | - |
| Password | VARCHAR | 255 | - |
| Firstname | VARCHAR | 100 | - |
| Lastname | VARCHAR | 100 | - |
| Email | VARCHAR | 100 | - |
| Skype | VARCHAR | 50 | - |
| Yahoo | VARCHAR | 50 | - |
| phone1 | VARCHAR | 20 | - |
| phone2 | VARCHAR | 20 | - |
| Institution | VARCHAR | 255 | - |
| Department | VARCHAR | 255 | - |
| Address | VARCHAR | 255 | - |
| City | VARCHAR | 120 | - |
| Country | VARCHAR | 2 | - |
| Timezone | VARCHAR | 100 | - |
| Firstaccess | BIGINT | 10 | - |
| Lastaccess | BIGINT | 10 | - |
| Lastlogin | BIGINT | 10 | - |
| Currentlogin | BIGINT | 10 | - |
| Picture | BIGINT | 10 | - |
| Description | LONGTEXT | - | - |
| Timecreated | BIGINT | 10 | - |
| Timemodified | BIGINT | 10 | - |
| Imagealt | VARCHAR | 255 | - |
| Lastnamephonetic | VARCHAR | 255 | - |
| Firstnamephonetic | VARCHAR | 255 | - |
| Middlename | VARCHAR | 255 | - |
| Alternatename | VARCHAR | 255 | - |

**Tabel 4.2** Deskripsi Tabel tb\_user

* 1. **Tabel 3 – tb\_role**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| Id | BIGINT | 10 | PRIMARY KEY,  AUTOINCREMENT |
| name | BIGINT | 10 | - |
| Shortname | VARCHAR | 254 | - |
| Description | VARCHAR | 255 | - |

**Tabel 4.3** Deskripsi Tabel tb\_role

1. **Tabel 4 – tb\_session**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| Id | BIGINT | 10 | PRIMARY KEY,  AUTO INCREMENT |
| Userid | BIGINT | 10 | - |
| Timecreated | BIGINT | 10 | - |
| Timemodified | BIGINT | 10 | - |
| Firstip | VARCHAR | 45 | - |
| Lastip | VARCHAR | 45 | - |

**Tabel 4.4** Deskripsi Tabel tb\_session

1. **Tabel 5 – tb\_assignment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| Id | BIGINT | 10 | PRIMARY KEY,  AUTO INCREMENT |
| Course | BIGINT | 10 | - |
| Name | VARCHAR | 255 | - |

**Tabel 4.5** Deskripsi Tabel tb\_assignment

1. **Tabel 6 – tb\_lesson**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| Id | BIGINT | 10 | PRIMARY KEY,  AUTO INCREMENT |
| Course | BIGINT | 10 | - |
| Name | VARCHAR | 255 | - |

**Tabel 4.6** Deskripsi Tabel tb\_lesson

1. **Tabel 7 – tb\_quiz**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| Id | BIGINT | 10 | PRIMARY KEY,  AUTO INCREMENT |
| Course | BIGINT | 10 | - |
| Name | VARCHAR | 255 | - |
| Timeopen | BIGINT | 10 | - |
| Timeclose | BIGINT | 10 | - |
| Timelimit | BIGINT | 10 | - |
| Timecreated | BIGINT | 10 | - |
| Timemodified | BIGINT | 10 | - |

**Tabel 4.7** Deskripsi Tabel tb\_user

1. **Tabel 8 – tb\_daily\_quiz**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id\_daily\_quiz | CHAR | 10 | PRIMARY KEY,  AUTO INCREMENT |
| daily\_quiz | VARCHAR | 255 | - |
| Jawaban | VARCHAR | 255 | - |
| Timeopen | DATE |  | - |
| Timeclose | DATE |  | - |
| Timelimit | DATE |  | - |
| Timecreated | DATE |  | - |
| Timemodified | DATE |  | - |
| id\_course | CHAR | 10 | id\_course\_fk |
| id\_materi | CHAR | 10 | id\_lesson\_fk |
| id\_level | CHAR | 10 | id\_level\_fk |

**Tabel 4.8** Deskripsi Tabel tb\_daily\_quiz

1. **Tabel 9 – tb\_level**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id\_level | CHAR | 10 | PRIMARY KEY,  AUTO INCREMENT |
| Level | CHAR | 2 | - |

**Tabel 4.9** Deskripsi Tabel tb\_level

1. **Tabel 10 – tb\_weekly\_quiz**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id\_weekly\_quiz | CHAR | 10 | PRIMARY KEY,  AUTO INCREMENT |
| weekly \_quiz | VARCHAR | 255 | - |
| Jawaban | VARCHAR | 255 | - |
| Timeopen | DATE |  | - |
| Timeclose | DATE |  | - |
| Timelimit | DATE |  | - |
| Timecreated | DATE |  | - |
| Timemodified | DATE |  | - |
| id\_course | CHAR | 10 | id\_course\_fk |
| id\_materi | CHAR | 10 | id\_lesson\_fk |

**Tabel 4.10** Deskripsi Tabel tb\_weekly\_quiz

1. Tabel 11 – **tb\_jurusan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 10 | PRIMARY KEY,  AUTO INCREMENT |
| jurusan | VARCHAR | 255 | - |

**Tabel 4.11** Deskripsi Tabel tb\_jurusan

1. Tabel 12 – **tb\_nilai**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 10 | PRIMARY KEY,  AUTO INCREMENT |
| userid | CHAR | 10 | iduser\_fk |
| nilai\_uts | DECIMAL | 5 | - |
| nilai\_uas | DECIMAL | 5 | - |

**Tabel 4.12** Deskripsi Tabel tb\_nilai

1. Tabel 13 – **tb\_materi\_syarat**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 10 | PRIMARY KEY,  AUTO INCREMENT |
| lessonid | CHAR | 10 | idlesson \_fk |
| quizid | CHAR | 10 | idquiz\_fk |

**Tabel 4.13** Deskripsi Tabel tb\_materi\_syarat

1. Tabel 14 – **tb\_files**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| file | VARCHAR | 255 | - |
| category | CHAR | 4 | idfilecategory\_fk |
| userid | CHAR | 10 | iduser\_fk |

**Tabel 4.14** Deskripsi Tabel tb\_files

1. Tabel 15 – **tb\_file\_categories**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| file\_category | CHAR | 4 | idlesson \_fk |

**Tabel 4.15** Deskripsi Tabel tb\_files\_categories

1. Tabel 16 – **tb\_feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| courseid | CHAR | 4 | idcourse \_fk |
| name | VARCHAR | 255 | - |
| timeopen | DATE |  | - |
| timeclose | DATE |  | - |

**Tabel 4.16** Deskripsi Tabel tb\_feedback

1. Tabel 17 – **event**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| name | VARCHAR | 255 | - |
| description | VARCHAR | 255 | - |
| courseid | CHAR | 10 | idcourse\_fk |
| userid | CHAR | 10 | iduser\_fk |
| timestart | DATE |  | - |
| timeduration | DATE |  | - |
| timemodified | DATE |  | - |

**Tabel 4.17** Deskripsi Tabel event

1. Tabel 18 – **tb\_choice**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| courseid | CHAR | 10 | idcourse \_fk |
| name | VARCHAR | 255 | - |
| timemodified | DATE |  | - |

**Tabel 4.18** Deskripsi Tabel tb\_choice

1. Tabel 19 – **tb\_choice\_answers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| choiceid | CHAR | 4 | idchoice \_fk |
| userid | CHAR | 10 | - |
| optionid | CHAR | 4 | idoption\_fk |
| timemodified | DATE |  | - |

**Tabel 4.19** Deskripsi Tabel tb\_choice\_answers

1. Tabel 20 – **tb\_choice\_options**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| choiceid | CHAR | 4 | idchoice \_fk |
| text | VARCHAR | 255 | - |
| maxanswer | INT | 2 | - |
| timemodified | DATE |  | - |

**Tabel 4.20** Deskripsi Tabel tb\_choice\_option

1. Tabel 21 – **tb\_questions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| name | CHAR | 4 | - |
| timecreated | DATE |  | - |
| timemodified | DATE |  | - |
| createdby | CHAR | 10 | - |
| modifiedby | CHAR | 10 | - |

**Tabel 4.21** Deskripsi Tabel tb\_question

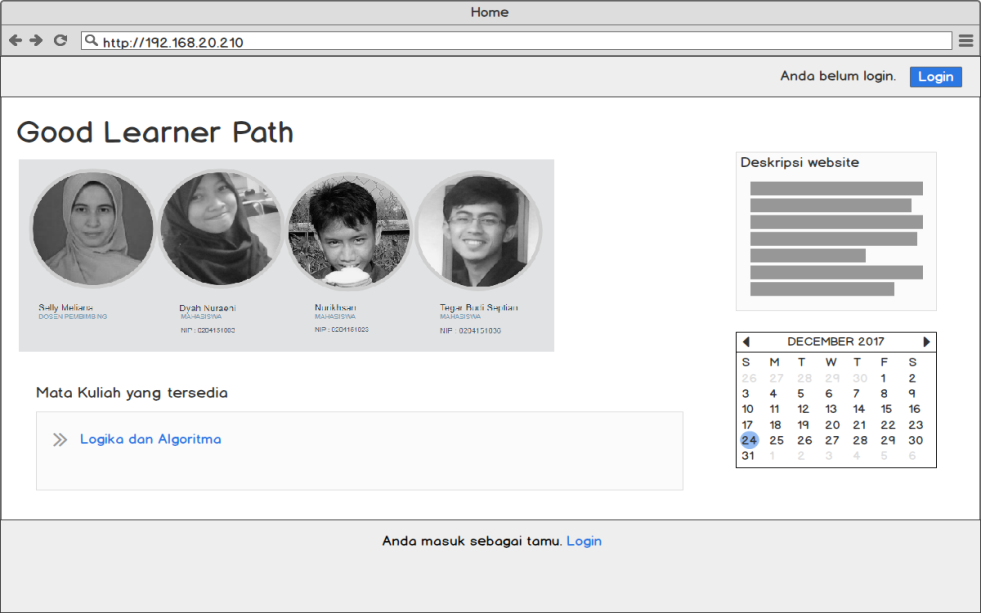
1. Tabel 22 – **tb\_question\_answers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tipe Data** | **Panjang** | **Keterangan** |
| id | CHAR | 4 | PRIMARY KEY,  AUTO INCREMENT |
| questionid | CHAR | 4 | idquestion \_fk |
| answer | VARCHAR | 255 | - |

**Tabel 4.22** Deskripsi Tabel tb\_question\_answers

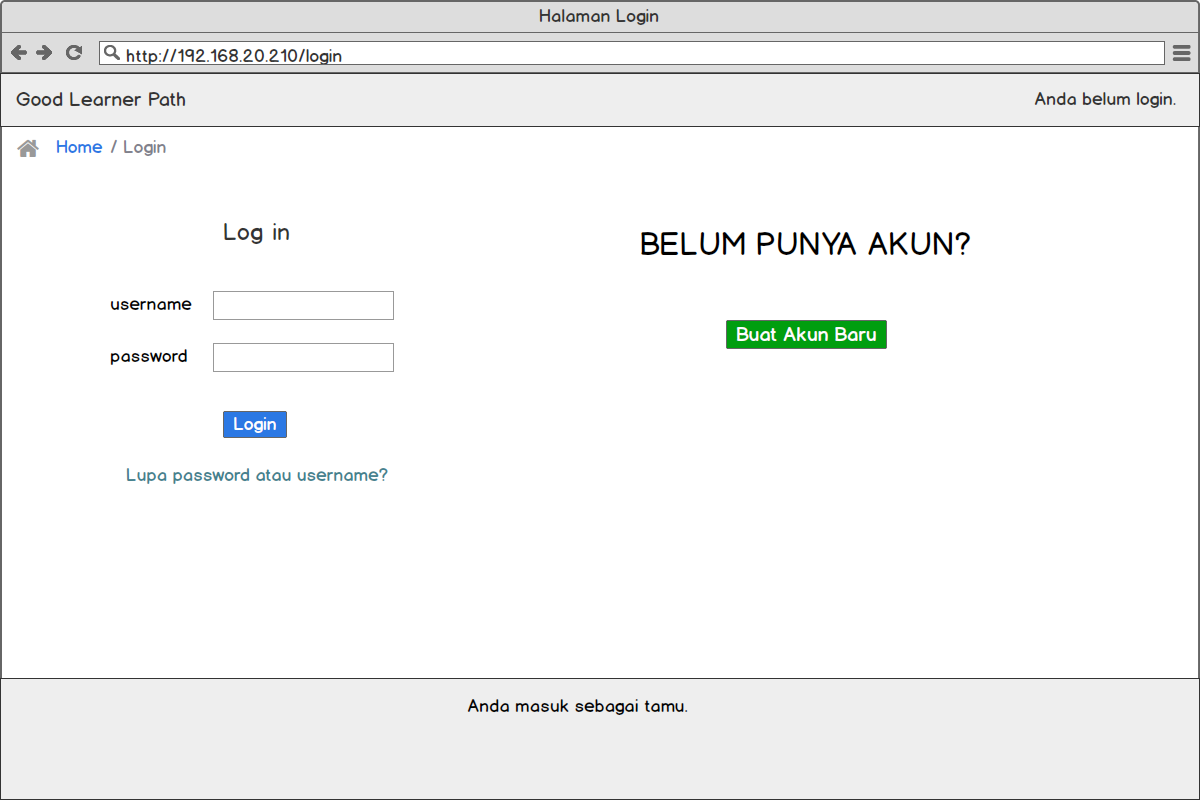
* 1. **Perancangan Antar Muka**

**4.4.1 Tampilan Halaman Utama**



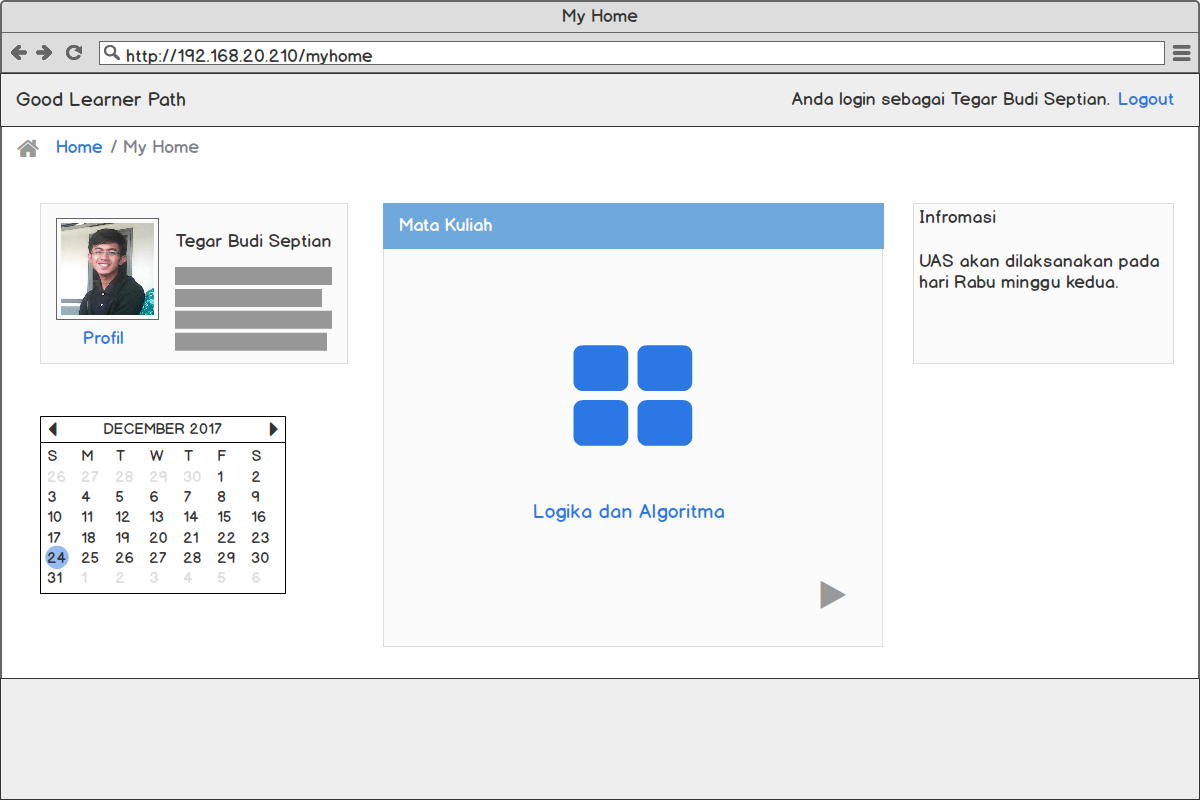
**Gambar 4.27** *Tampilan halaman utama*

**4.4.2 Tampilan Halaman Login**



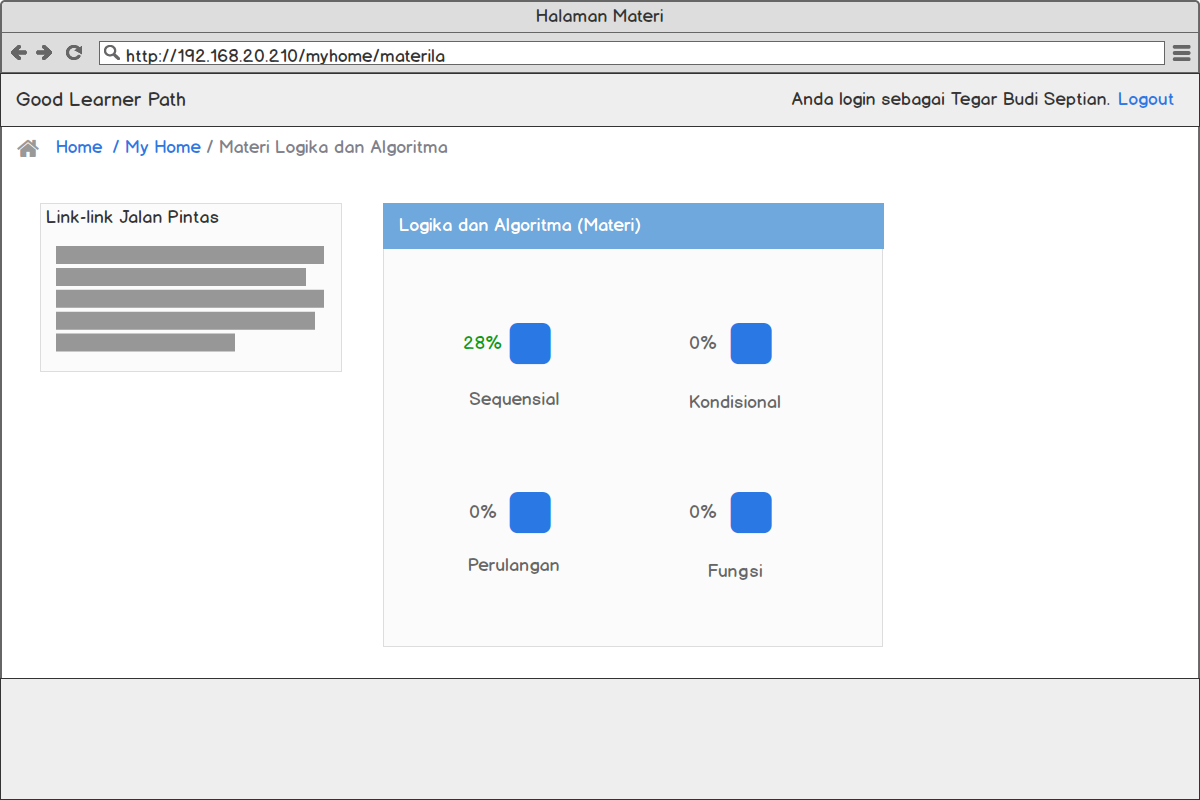
**Gambar 4.28** *Tampilan Halaman Login*

**4.4.3 Tampilan Halaman Profile**



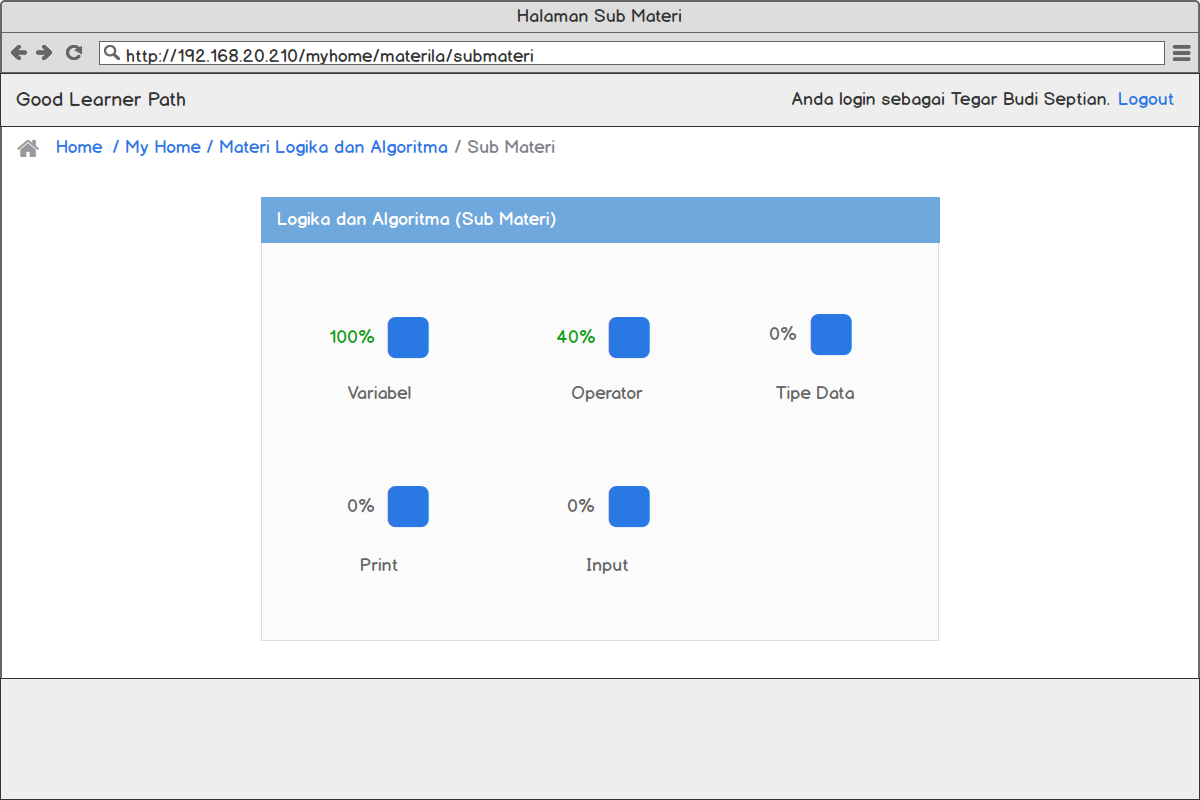
**Gambar 4.29** *Tampilan halaman Profil*

**4.4.4 Tampilan Halaman Course Permateri**



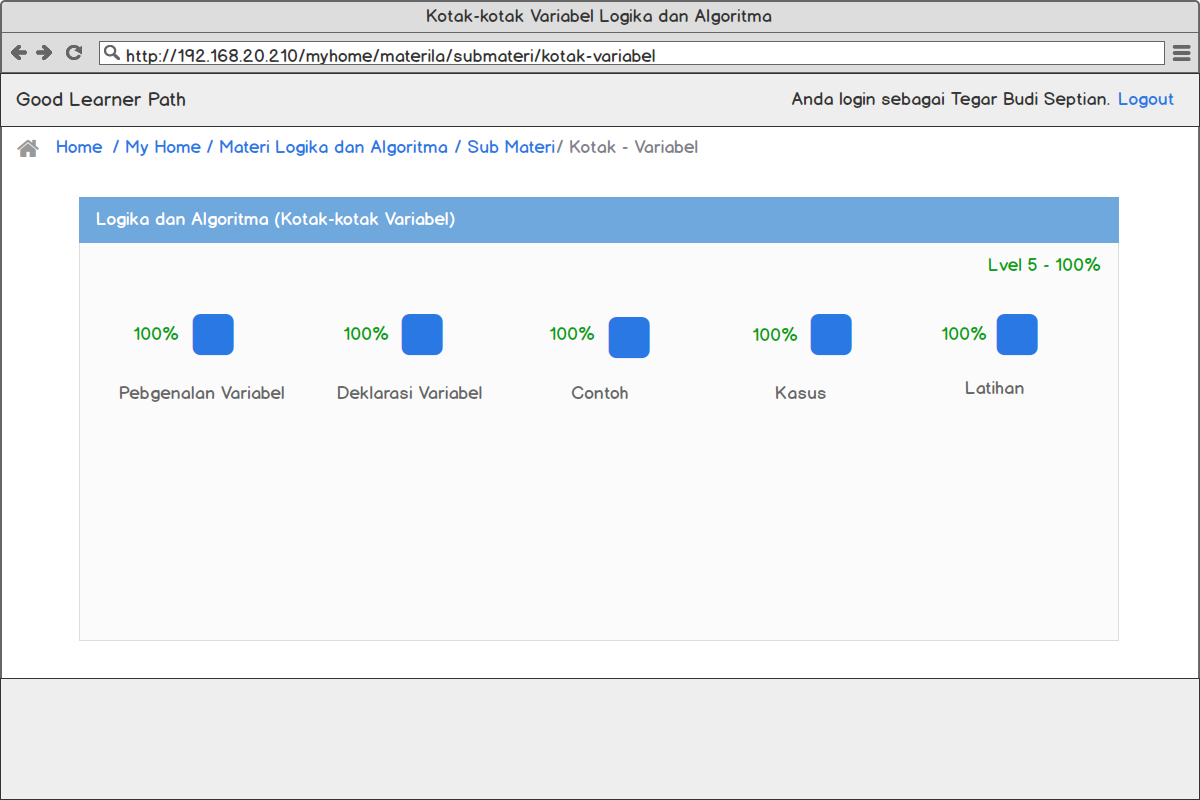
**Gambar 4.30** *Tampilan Halaman Course Materi*

**4.4.5 Tampilan Halaman Sub Materi**

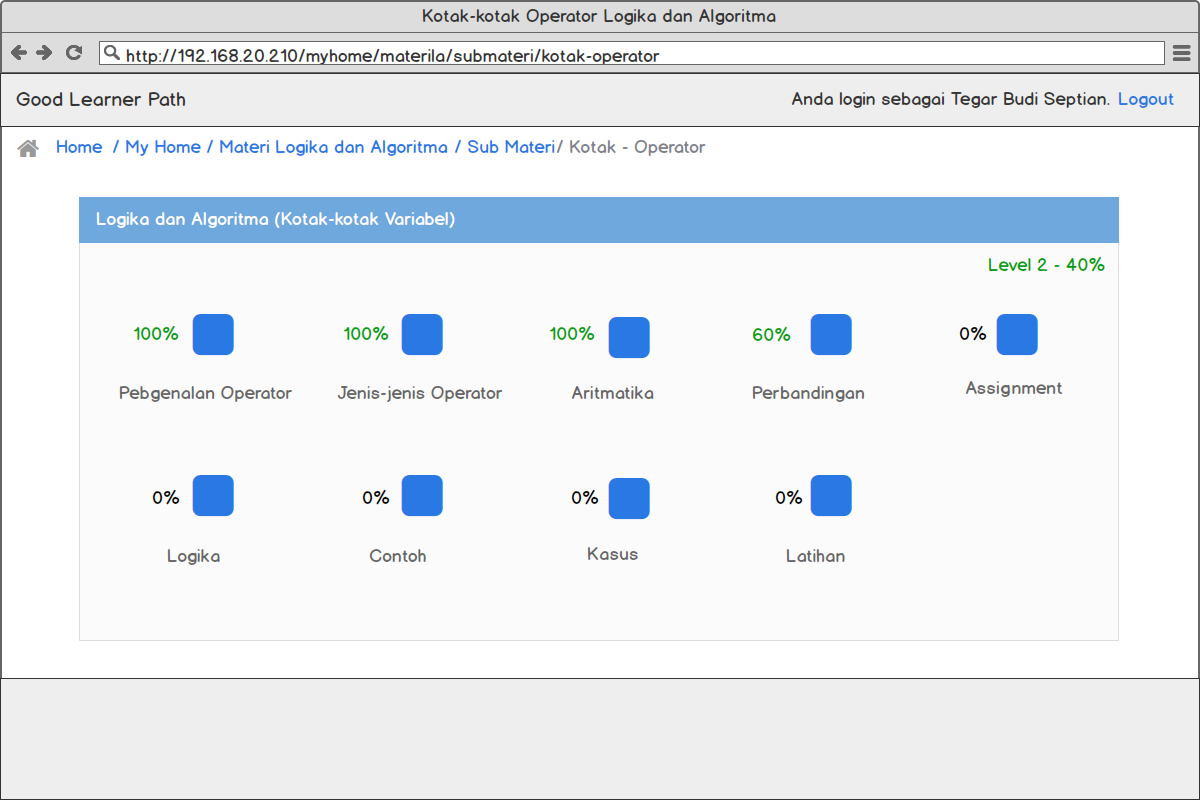


**Gambar 4.31** *Tampilan halaman Sub Materi*

**4.4.6 Tampilan Halaman Di dalam Sub Materi**

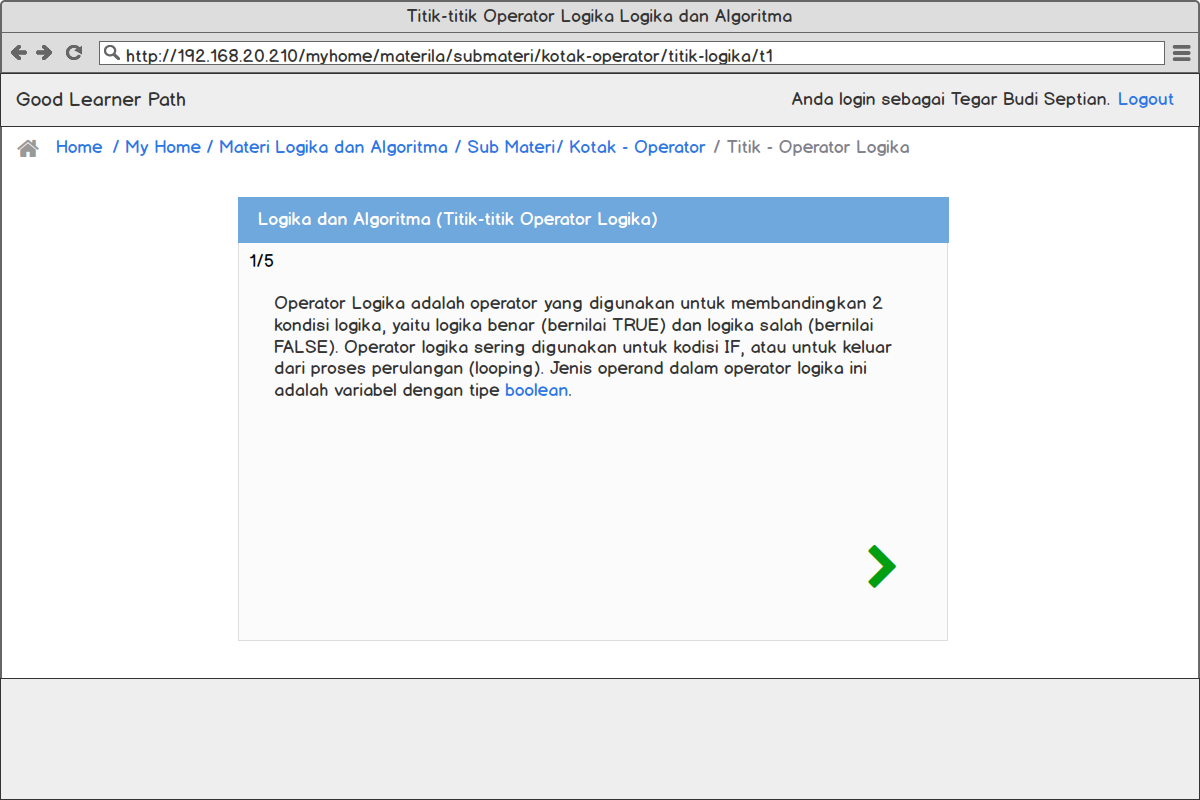


**Gambar 4.32** *Tampilan halaman Sub Materi Variabel*



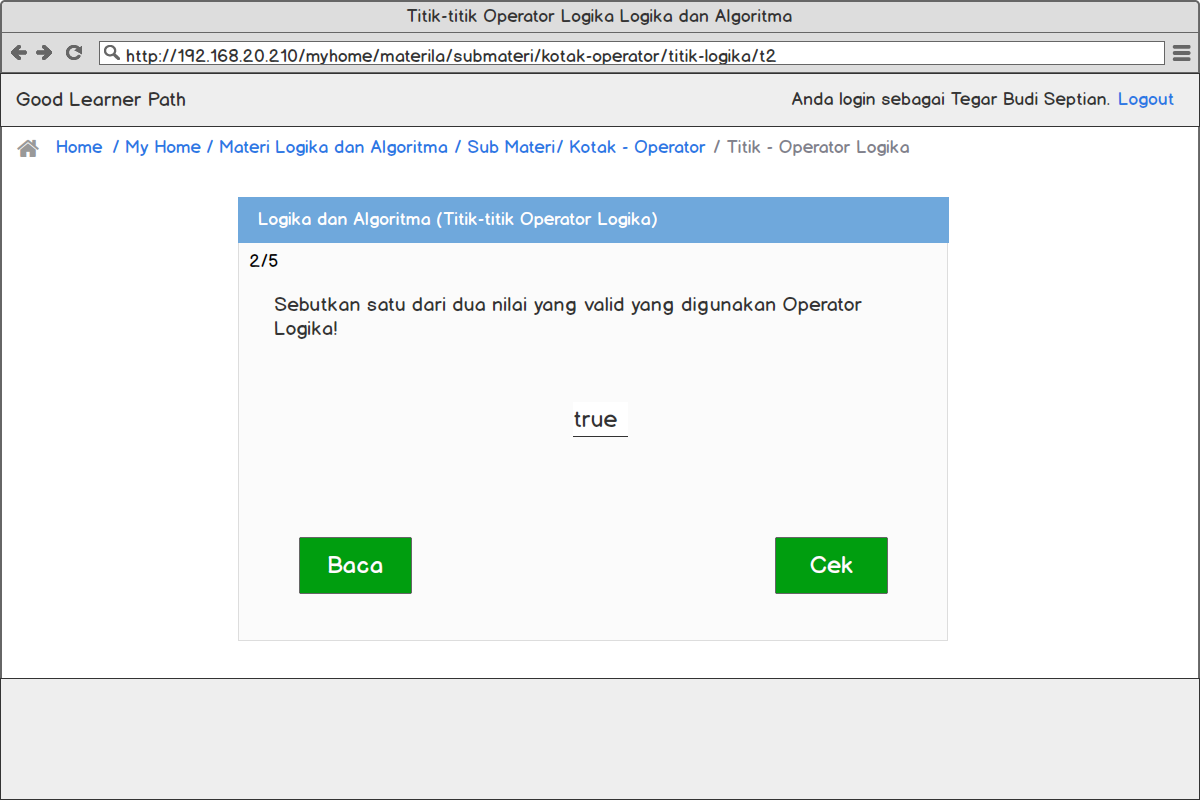
**Gambar 4.33** *Tampilan halaman Sub Materi Operator*

**4.4.7 Tampilan Halaman Materi**

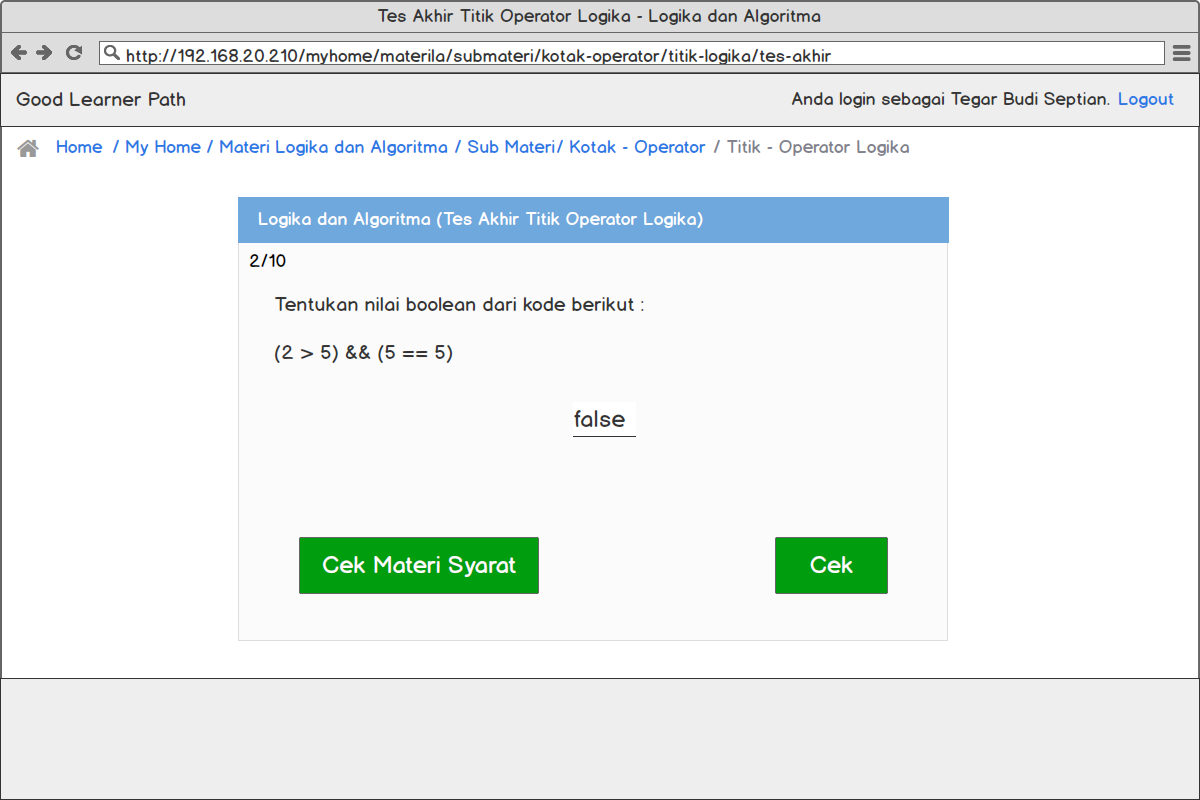


**Gambar 4.34** *Tampilan halaman Materi pada Daily Quiz*

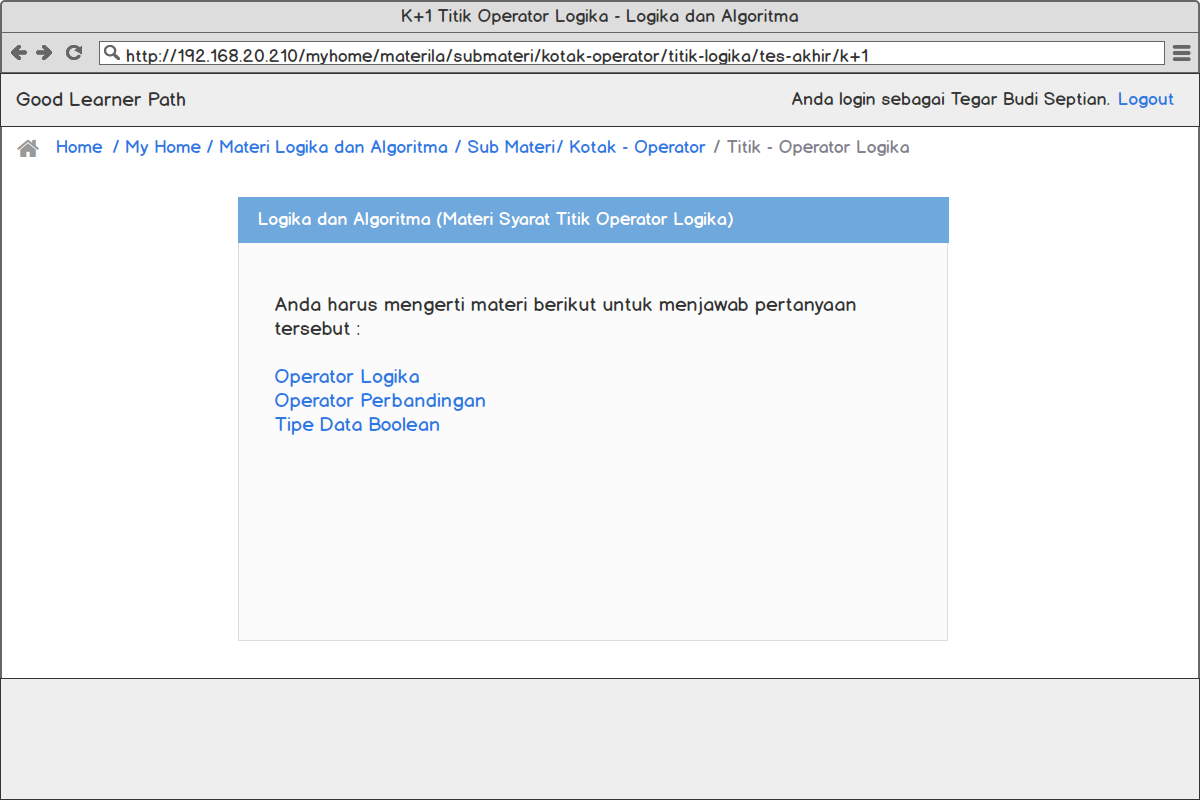
**4.4.8 Tampilan Halaman Soal**



**Gambar 4.35** *Tampilan Halaman Soal pada Dialy Quiz*

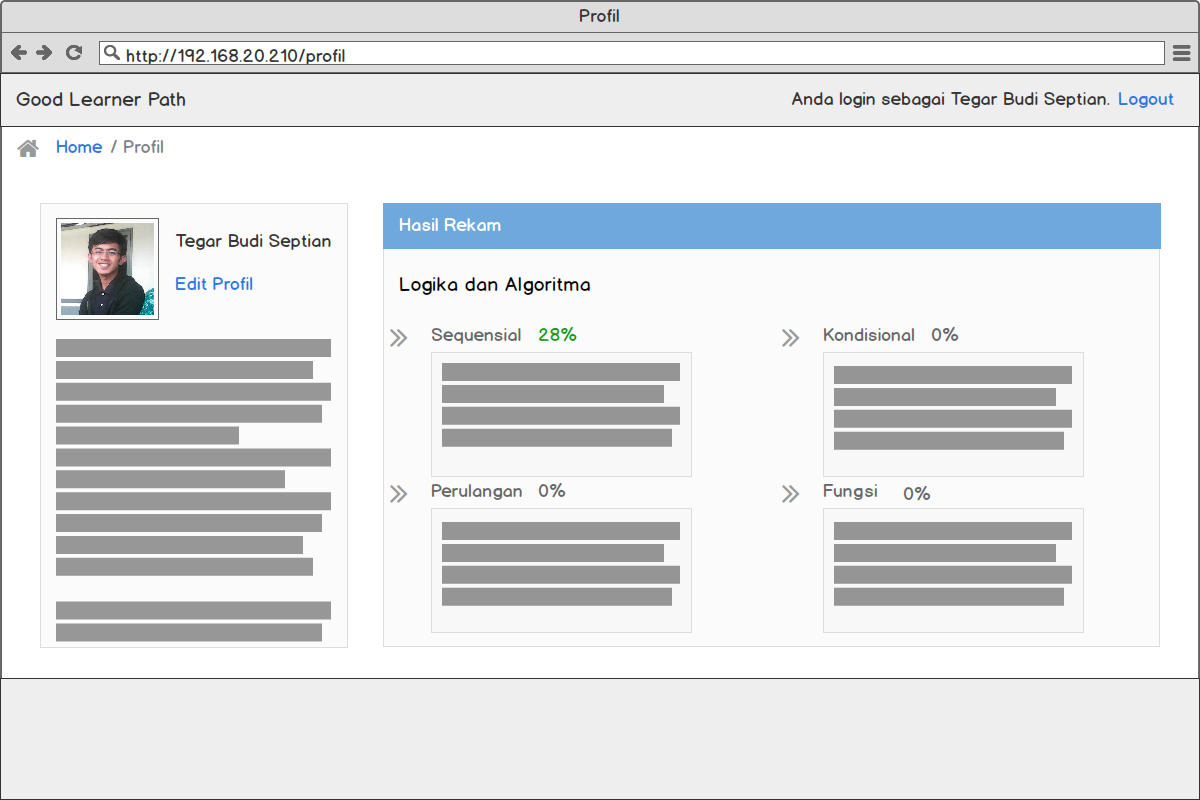


**Gambar 4.36** *Tampilan Halaman Soal Daily Quiz*

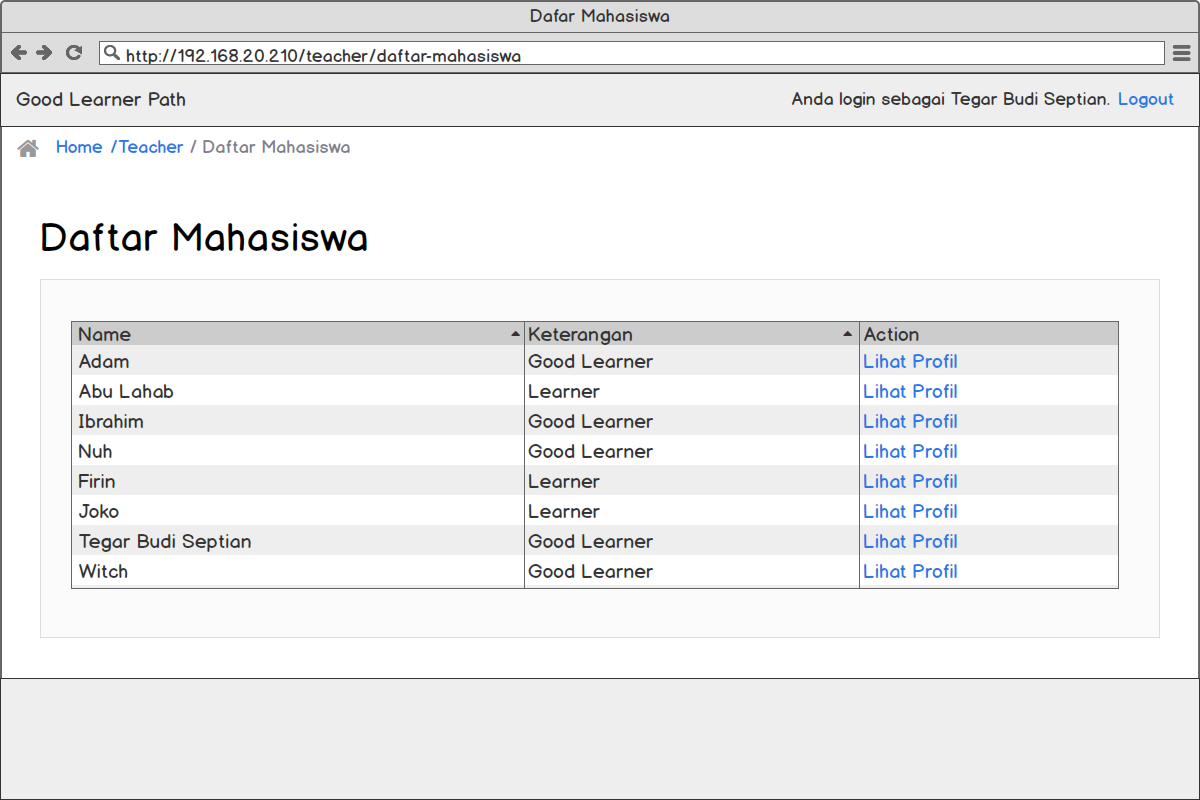


**Gambar 4. 37** *Tampilan Halaman Info K+1*

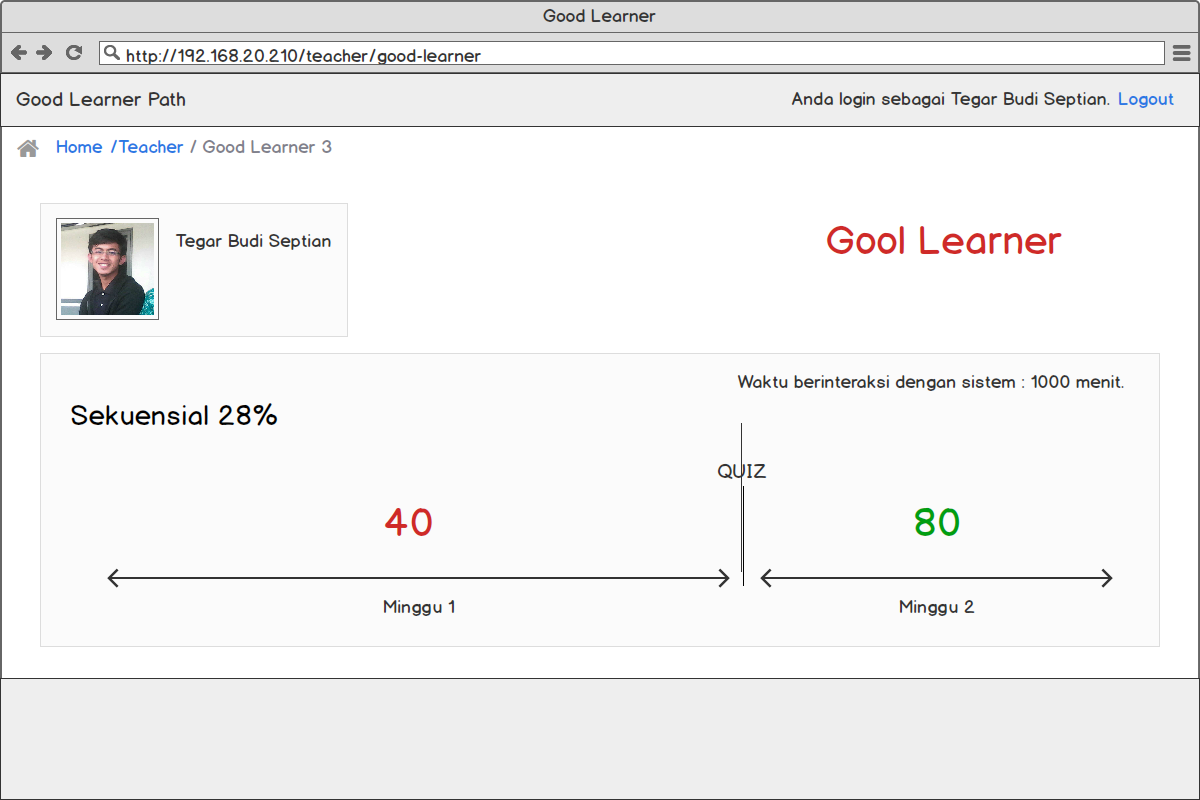
**4.4.9 Tampilan Halaman Pemilaian**



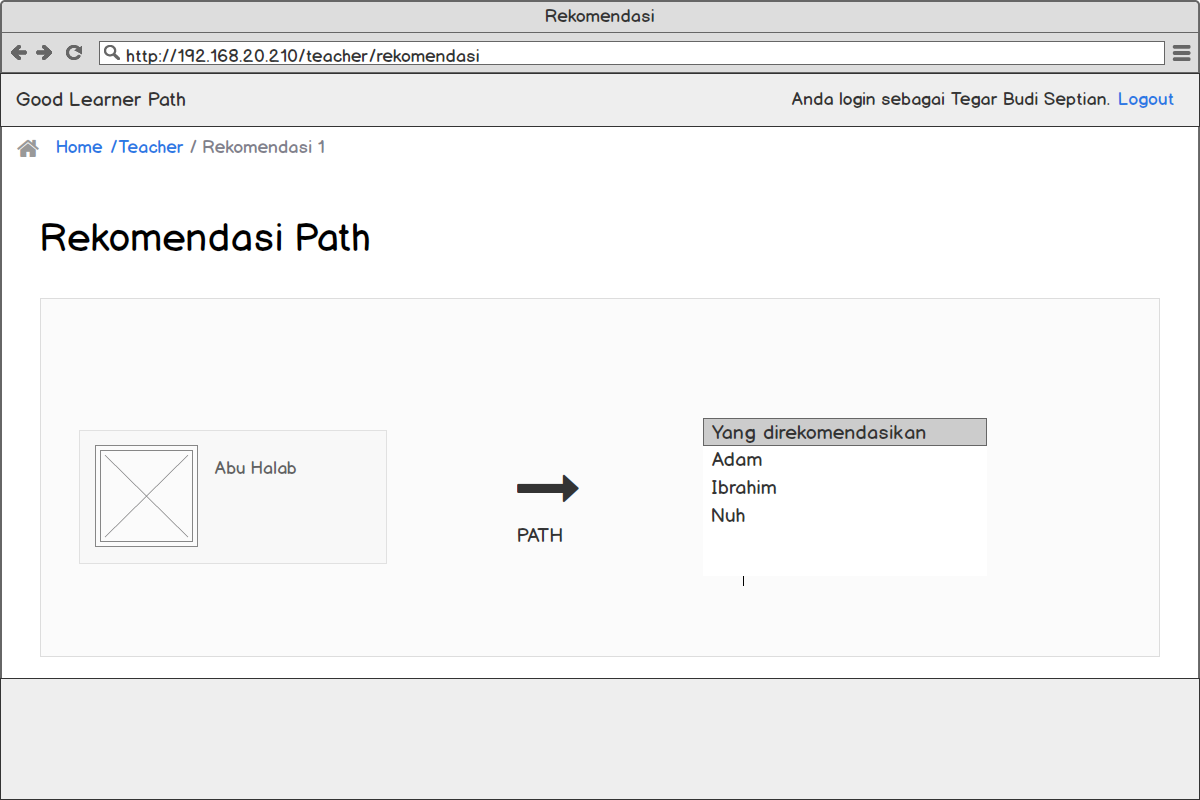
**Gambar 4. 38** *Tampilan Halaman Deskripsi Mahasiswa*



**Gambar 4. 39** *Tampilan Halaman Penilaian Good Learner*



**Gambar 4. 40** *Tampilan Halaman Detail Penilaian Good Learner*



**Gambar 4. 41** *Tampilan Halaman Rekomendasi Path Good Learner*