

# **PROPOSAL TUGAS AKHIR**

## **RANCANG BANGUN “SDKU *ONLINE*” SEBAGAI MEDIA PENDUKUNG BELAJAR MENGAJAR SEKOLAH DASAR BERBASIS WEB**



**Oleh:**

**Lalu Septian Wahyu Adipratama**

**F1B016050**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MATARAM  
2020**

## USULAN TUGAS AKHIR

### RANCANG BANGUN “SDKU *ONLINE*” SEBAGAI MEDIA PENDUKUNG BELAJAR MENGAJAR SEKOLAH DASAR BERBASIS WEB

Oleh:

**LALU SEPTIAN WAHYU ADIPRATAMA**  
**F1B016050**

#### 1. Pembimbing Utama

Lalu A. Syamsul Irfan Akbar, ST., M.Eng.  
NIP. 19830310 200912 1 004

Tanggal: 2020

#### 2. Pembimbing Pendamping

Dwi Ratnasari, S.Kom., MT.  
NIP. 19850901 201903 2 010

Tanggal: 2020

Mengetahui  
Ketua Jurusan Teknik Elektro  
Fakultas Teknik  
Universitas Mataram

Muhamad Syamsu Iqbal, ST., MT., Ph.D.  
NIP. 19720222 199903 1 002

## KATA PENGANTAR

Puji syukur penyusun panjatkan kehadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia yang diberikan-nya dalam penyelesaian Laporan Proposal Tugas Akhir dengan judul **“RANCANG BANGUN SDKU *ONLINE* SEBAGAI MEDIA PENDUKUNG BELAJAR MENGAJAR SEKOLAH DASAR BERBASIS WEB”** ini.

Proposal Tugas Akhir ini merupakan salah satu syarat dalam memperoleh gelar kesarjanaan di Jurusan Teknik Elektro Universitas Mataram. Dimana kegiatan tersebut mahasiswa dituntut dapat memahami dan menerapkan ilmu yang diperoleh saat perkuliahan.

Atas tersusunnya Laporan Proposal Tugas Akhir ini, Penyusun mengucapkan terimakasih yang sebesar-besarnya kepada :

1. Bapak Akmaluddin, ST., M.Sc.(Eng)., Ph.D selaku Dekan Fakultas Teknik Universitas Mataram.
2. Bapak Muhamad Syamsu Iqbal, ST., MT., Ph.D. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Mataram.
3. Bapak Lalu A. Syamsul Irfan Akbar, ST., M.Erg. dan ibu Dwi Ratnasari, S.Kom.,MT. selaku Dosen Pembimbing.
4. Kepada kedua Orang tua dan seluruh keluarga yang telah memberikan dukungan, do'a dan restu.

Penyusun menyadari bahwa laporan proposal Tugas Akhir ini masih jauh dari sempurna, oleh karena itu penyusun selalu terbuka terhadap masukan dan saran dari semua pihak yang sifatnya membangun untuk mendekati kesempurnaan dari laporan ini. Tidak lupa Penyusun menyampaikan permohonan maaf yang sebesar-besarnya jika dalam penyusunan Laporan ini terdapat kesalahan dan kekeliruan. Akhir kata, semoga laporan Tugas Akhir ini dapat memberikan manfaat sebagaimana mestinya.

Mataram, 2020

Penyusun

(Lalu Septian Wahyu Adipratama)

NIM : F1B0161050

## DAFTAR ISI

|   |     |
|---|-----|
| HALAMAN JUDUL                                       |     |
| LEMBAR PENGESAHAN .....                             | ii  |
| KATA PENGANTAR .....                                | iii |
| DAFTAR ISI .....                                    | iv  |
| DAFTAR GAMBAR .....                                 | vi  |
| DAFTAR TABEL .....                                  | vii |
| BAB 1 PENDAHULUAN .....                             | 1   |
| 1.1 Latar Belakang .....                            | 1   |
| 1.2 Rumusan Masalah .....                           | 3   |
| 1.3 Batasan Masalah .....                           | 4   |
| 1.4 Tujuan Penelitian .....                         | 4   |
| 1.5 Manfaat Penelitian .....                        | 4   |
| BAB 2 TINJAUAN PUSTAKA .....                        | 5   |
| 2.1 Penelitian Terdahulu .....                      | 5   |
| 2.2 Dasar Teori .....                               | 6   |
| 2.2.1 Pengertian Sistem .....                       | 6   |
| 2.2.2 Informasi .....                               | 7   |
| 2.2.3 Basis Data .....                              | 7   |
| 2.2.4 Konsep Dasar <i>Electronic Learning</i> ..... | 7   |
| 2.2.5 <i>Waterfall</i> .....                        | 9   |
| 2.2.6 <i>Object Oriented Programming</i> .....      | 10  |
| 2.2.7 <i>Unified Modelling Language (UML)</i> ..... | 11  |
| 2.2.8 <i>Framework</i> .....                        | 15  |
| 2.2.9 <i>Django</i> .....                           | 16  |
| 2.2.10 <i>Python</i> .....                          | 17  |
| 2.2.11 <i>Visual Studio Code</i> .....              | 19  |
| 2.2.12 <i>Xampp Server</i> .....                    | 20  |
| BAB 3 METODOLOGI PENELITIAN .....                   | 21  |
| 3.1 Metode Perancangan .....                        | 21  |
| 3.2 Studi Literatur .....                           | 21  |
| 3.3 Analisis Kebutuhan .....                        | 22  |
| 3.4 Perancangan Sistem .....                        | 22  |

|                                     |    |
|-------------------------------------|----|
| 3.4.1 <i>Use Case Diagram</i> ..... | 22 |
| 3.4.2 <i>Activity Diagram</i> ..... | 24 |
| 3.4.3 <i>Class Diagram</i> .....    | 29 |
| 3.4.4 Perancangan Antar Muka .....  | 30 |

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 <i>Distance Learning</i> .....                         | 9  |
| Gambar 2.2 Ilustrasi <i>Model Waterfall</i> .....                 | 10 |
| Gambar 2.3 <i>Use Case Diagram</i> .....                          | 12 |
| Gambar 2.4 <i>Activity Diagram</i> .....                          | 13 |
| Gambar 2.5 Contoh <i>Sequance Diagram</i> .....                   | 14 |
| Gambar 3.1 <i>Diagram Waterfall</i> .....                         | 21 |
| Gambar 3.2 <i>Use Case</i> Aplikasi SDku Online .....             | 23 |
| Gambar 3.3 Aktivitas <i>Login</i> Sebagai <i>Admin</i> .....      | 25 |
| Gambar 3.4 Aktivitas <i>Login</i> Sebagai Guru .....              | 25 |
| Gambar 3.5 Aktivitas <i>E-learning</i> .....                      | 26 |
| Gambar 3.6 Aktivitas <i>Upload</i> Tugas.....                     | 27 |
| Gambar 3.7 Aktivitas Ujian .....                                  | 28 |
| Gambar 3.8 <i>Class Diagram</i> Aplikasi SDku <i>Online</i> ..... | 29 |
| Gambar 3.9 Rancangan Halaman Utama .....                          | 30 |
| Gambar 3.10 Perancangan Halaman <i>Login</i> Tugas .....          | 31 |
| Gambar 3.11 Perancangan Halaman Tugas .....                       | 31 |
| Gambar 3.12 Kondisi Pada Keadaan Pertama .....                    | 32 |
| Gambar 3.13 Kondisi Pada Keadaan Kedua .....                      | 33 |
| Gambar 3.14 Halaman Ujian Siswa.....                              | 33 |
| Gambar 3.15 Tampilan Halaman <i>E-learning</i> .....              | 34 |
| Gambar 3.16 Tampilan <i>Playlist</i> Video Mata Pelajaran .....   | 34 |

## **DAFTAR TABEL**

|  |    |
|--|----|
| Tabel 3.1 Aktivitas Pada Aktor Admin ..... | 26 |
| Tabel 3.2 Aktivitas Pada Aktor Guru .....  | 27 |
| Tabel 3.3 Aktivitas Pada Aktor Siswa ..... | 27 |
| Tabel 3.4 Kardinalitas Antar Entitas ..... | 32 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dunia saat ini dikejutkan dengan mewabahnya suatu penyakit yang disebabkan oleh virus yang bernama *Corona virus*. *Corona virus* merupakan keluarga besar virus yang menyebabkan penyakit mulai dari gejala ringan sampai berat. Terdapat setidaknya dua jenis *corona virus* yang diketahui menyebabkan penyakit yang dapat menimbulkan gejala berat. *Corona virus Diseases 2019 (Covid-19)* adalah penyakit jenis baru yang belum pernah diidentifikasi sebelumnya pada manusia. Tanda dan gejala umum infeksi *Covid-19* antara lain gejala gangguan pernapasan akut seperti demam, batuk, dan sesak napas. Masa inkubasi rata-rata 56 hari dengan masa inkubasi terpanjang 14 hari.

Pada tanggal 30 Januari 2020 *World Health Organization* (WHO) telah menetapkan sebagai kedaruratan kesehatan masyarakat yang meresahkan dunia. Pada tanggal 2 Maret 2020, Indonesia melaporkan kasus konfirmasi *Covid-19* sebanyak 2 kasus. Sebanyak 3 kasus yang dikutip dari situs resmi Gugus Tugas Percepatan Penanganan *Covid-19* sampai dengan tanggal 16 Mei 2020 ada 17.025 orang yang dinyatakan positif, 3.991 orang sembuh, dan 1.089 orang yang meninggal dunia.

Adanya virus *Covid-19* di Indonesia saat ini berdampak bagi aktivitas masyarakat. Dampak *Covid-19* ini terjadi di berbagai bidang seperti social, ekonomi, pariwisata, pendidikan. Surat Edaran (SE) yang dikeluarkan pemerintah pada 18 Maret 2020 segala kegiatan didalam dan diluar ruangan di semua sektor sementara waktu ditunda demi mengurangi penyebaran corona terutama pada bidang pendidikan. Tanggal 24 maret 2020 Menteri Pendidikan dan Kebudayaan Republik Indonesia mengeluarkan Surat Edaran Nomor 4 Tahun 2020 Tentang Pelaksanaan Kebijakan Pendidikan dalam masa Darurat Penyebaran *Covid-19* dijelaskan bahwa proses belajar dilaksanakan di rumah melalui pembelajaran daring (dalam jaringan)/jarak jauh dilaksanakan untuk membantu memutus penyebaran *Covid-19* serta memberikan kebebasan bagi siswa untuk belajar dirumah. Gagasan pelaksanaan ini mengacu pada penerapan teknologi informasi. Tidak dipungkiri teknologi informasi berkembang dan merambah ke berbagai bidang kehidupan yang turut adil membawa banyak dampak sehingga ini sangat dibutuhkan untuk meningkatkan mutu dan mekanisme pelayanan



pendidikan agar lebih berdaya guna khususnya dalam kasus *Covid-19* ini, sehingga sumber daya manusia (SDM) yang dihasilkan juga berkualitas serta mampu dalam membangun dunia luar sesuai dengan kemampuannya.

Sekolah Dasar merupakan ujung dari pondasi dalam melanjutkan pendidikan, itu mengapa pendidikan sekolah dasar merupakan hal penting untuk ditekuni. Melihat kondisi *pandemic* yang terjadi saat ini, segala aktivitas belajar mengajar dilakukan melalui daring (dalam jaringan)/*online*. Menurut Isman dalam jurnalnya yang berjudul “Pembelajaran Moda Dalam Jaringan (Moda Daring)” mengatakan “Pembelajaran daring (dalam jaringan) merupakan pemanfaatan jaringan internet dalam proses pembelajaran”. Pembelajaran daring (dalam jaringan) siswa memiliki keleluasaan waktu belajar, dapat belajar kapanpun dan dimanapun. Siswa dapat berinteraksi dengan guru menggunakan beberapa aplikasi seperti *classroom*, *video converence*, telepon atau *live chat*, *zoom* maupun melalui *whatsapp group*. Pembelajaran ini merupakan inovasi pendidikan untuk menjawab tantangan akan ketersediaan sumber belajar yang variatif. Keberhasilan dari suatu model ataupun media pembelajaran tergantung dari karakteristik peserta didiknya. Nakayama mengungkapkan bahwa semua *literature* dalam *eleraning* mengindikasikan bahwa tidak semua peserta didik akan sukses dalam pembelajaran *online*, ini dikarenakan faktor lingkungan belajar dan karakteristik peserta didik (Nakayama M, Yamamoto H, 2007).

Metode pembelajaran *online* merupakan unsur yang sama dengan pendidikan seperti biasanya dan tidak bisa dihilangkan dari rumus pendidikan. Proses pembelajaran terdapat berbagai metode yang dapat digunakan sesuai dengan kondisi yang ada. Pendidikan yang dilakukan dengan cara *online*, dapat juga terlaksana dengan menjalankan strategi pembelajaran yang meliputi pengajaran, membaca, diskusi, presentasi, penugasan dan evaluasi yang tergantung pada satu atau lebih tiga model dasar komunikasi, antara lain: komunikasi guru dengan siswa, komunikasi siswa dengan sumber belajar dan komunikasi antara siswa dengan siswa. Pembelajaran secara *online* dapat berjalan secara optimal apabila aspek tersebut dijalankan dengan komposisi yang seimbang. Khususnya pembelajaran dalam hal evaluasi/ujian sekolah. Evaluasi/ujian secara *online* untuk sekolah dasar masih terbilang minim dan menjadi suatu potensi yang besar jika dapat dikembangkan di Indonesia. Evaluasi/ujian *online* juga akan menampilkan informasi mata pelajaran yang diampu oleh siswa serta opsi mengikuti ujian sesuai jadwal dan ujian susulan

serta *box* keterangan yang akan memuat alasan mundurnya ujian. Sistem evaluasi juga harus memperhatikan kehadiran pendidik, peserta didik, materi ujian pada sistem, dan kesiapan sistem yang menjadi media kelancaran proses evaluasi/ujian.

Aplikasi *E-learning* merupakan sebuah inovasi *metode* pembelajaran yang dapat digunakan dan dikembangkan untuk saat ini. Menurut Marlina Kusuma Putri dalam skripsinya yang berjudul “Implementasi *E-Learning* pada SMA Negeri 2 Surakarta Menggunakan *PHP* dan *MYSQL*” mengatakan bahwa *e-learning* merupakan revolusi yang berbasiskan pada teknologi internet dan bergelut di bidang pendidikan *education*. *E-learning* adalah pemanfaatan teknologi atau media elektronik seperti internet, *intranet/extranet*, *satellite broadcast*, audio/video tape, *interactive TV*, CD-ROM, dan *computer-based training* (CBT) untuk penyebaran materi pembelajaran. Sekolah dasar (SD) merupakan target yang ingin dituju, guna untuk memberi bimbingan terkait teknologi yang kini semakin berkembang serta memberikan solusi untuk memberikan edukasi tentang teknologi kepada siswa-siswi dan memberi suasana pembelajaran yang baru dan menarik.

Pembelajaran ini merupakan inovasi pendidikan untuk menjawab tantangan akan ketersediaan sumber belajar yang variatif. Aplikasi ini diharapkan dapat meningkatkan minat belajar serta meningkatkan prestasi siswa. Selain itu dengan aplikasi *E-learning* ini siswa bisa belajar dirumah saja selama terhubung ke jaringan internet. Aplikasi *E-learning* ini dibuat dengan menggunakan bahasa pemrograman *PHP* dan *Mysql* sebagai basis datanya. Pembuatan aplikasi ini difokuskan untuk pengerjaan soal, pengumpulan tugas dan untuk melaksanakan evaluasi belajar siswa (ulangan/ujian sekolah), dan Aplikasi ini dapat digunakan sebagai mengolah data dan sarana penyedia layanan informasi bagi penggunanya baik untuk guru dan siswa secara cepat dan efisien.

## 1.2 Rumusan Masalah

Rumusan masalah dalam pengembangan aplikasi ini adalah mengembangkan sistem aplikasi SD *Online* berbasis *website* untuk menunjang proses belajar yang *user friendly*

### 1.3 Batasan Masalah

Batasan masalah merupakan acuan penelitian untuk membatasi gerak dan terfokus pada tujuan penelitian. Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Ruang lingkup sistem ini hanya untuk sekolah dasar
2. Sistem ini hanya untuk melakukan kegiatan ujian, pengumpulan tugas, dan media belajar
3. Aplikasi ini dirancang berbasis web

### 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu dapat merancang dan menghasilkan aplikasi yang dapat memberikan sarana kepada pendidik untuk menyampaikan materi evaluasi, memberikan sarana kepada siswa untuk mengakses kebutuhan belajar seperti mengikuti evaluasi, serta memberikan metode pembelajaran yang cepat dan praktis serta dapat diakses dari mana saja selama dalam jangkauan jaringan internet dan juga menghasilkan aplikasi yang *user friendly*.

### 1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini yaitu:

1. Mampu memecahkan masalah pembelajaran ditengah *pandemic* khususnya untuk sekolah dasar
2. Mampu memberikan sarana edukasi untuk siswa
3. Memberikan kemudahan dalam mengakses sistem dimana saja selama dalam jangkauan internet
4. Sarana penerapan ilmu atau implementasi pengembangan ilmu yang telah diperoleh di perkuliahan

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Penelitian tentang pembelajaran *online (E-learning)* cukup banyak dilakukan, Penelitian pada tahun 2015 yang dilakukan oleh Risiti Ana Diah dan Umi Fadillah mahasiswa Universitas Muhammadiyah Surakarta yang meneliti tentang *Rancang Bangun Website dan E-Learning di TPQ Al-Fadhillah*. Penelitian bertujuan untuk mempermudah dalam pengksesan pada sarana pendidikan. Siswa bisa dengan mudah mendapatkan informasi, siswa juga bisa mengerjakan kuis, sesuai dengan kelas dan jurusannya, dan juga bisa mendownload materi. Pengunjung biasa atau *user public* bisa dengan gratis mendownload materi yang tersedia di web *e-learning*.

Pengembangan sistem yang dilakukan oleh Abdullah Muammar pada tahun 2018 tentang *Pengembangan E-learning Berbasis Web Di Jurusan Pendidikan Biologi UIN Raden Intan Lampung*. Penelitian dilakukan untuk mencari tahu karakteristik dan kelayakan *e-learning* jika diterapkan pada jurusan Pendidikan Biologi. Serta untuk membantu mahasiswa dalam mencari materi perkuliahan dan mengakses informasi. Penelitian ini juga sebagai sarana untuk meningkatkan mutu pendidikan.

Pengembangan sistem yang dilakukan oleh Iswan Kristiyaningsih pada tahun 2011 tentang *Pembuatan Aplikasi Pembelajaran Oline(E-learning) di MTs Negeri Plupuh Berbasis Web* juga ingin membantu MTsN dalam menerapkan aplikasi *e-learning* sebagai tujuan mengembangkan dan membangun sistem pembelajaran secara online di MTsN Plupuh serta memberikan sarana atau fasilitas kepada guru untuk menyampaikan materi pelajaran. Selain itu dapat memberikan metode pembelajaran yang cepat dan praktis dapat diakses dari mana saja selama dalam jangkauan jaringan internet.

Pengembangan sistem *e-learning* oleh Dian Novitasari dalam penelitiannya tentang *Pembuatan E-learning Fisika SMA Kelas X Berbasis Web dengan Fasilitas Upload untuk Penambahan Mata Pelajaran* yang dilakukan untuk Madrasah Aliyah Negeri Karanganyar memiliki tujuan untuk memberikan media perantara bagi interaksi antara guru dan siswa dalam melakukan proses pembelajaran, guru menyampaikan dan menjelaskan bahan ajar tanpa harus langsung beratap muka dengan siswa, tetapi cukup dengan mengirimkan melalui web, fasilitas ini memungkinkan semua siswa dapat belajar efektif, mendapat bahan ajar dengan mudah, tanpa kendala batasan waktu dan tempat.

## 2.2 Dasar Teori

### 2.2.1 Pengertian Sistem

Sistem adalah sekelompok komponen dan elemen yang digabungkan menjadi satu untuk mencapai tujuan tertentu. Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Kata sistem sering digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.

Adapun manfaat sistem yaitu untuk menyatukan atau mengintegrasikan semua unsur yang ada dalam suatu ruang lingkup, dimana komponen-komponen tersebut tidak dapat berdiri sendiri. Komponen atau sub sistem harus saling berintegrasi dan berhubungan untuk membentuk satu kesatuan sehingga sasaran dan tujuan sistem tersebut bisa tercapai.

Menurut Mulyadi (2001) suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya antara satu dengan lainnya, yang berfungsi bersama-sama untuk mencapai suatu tujuan tertentu.

Menurut McLeod (2001) *"A system is a group of elements that are integrated with the common purpose of achieving an objective."* Yang artinya sistem adalah sekelompok elemen-elemen yang saling terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.

Simpulan yang dapat diambil menjabarkan bahwa sistem adalah kumpulan unsur-unsur yang berhubungan untuk melaksanakan kegiatan-kegiatan perusahaan dalam mencapai suatu tujuan tertentu.

### 2.2.2 Informasi

Menurut Jogiyanto HM (2005:6) informasi adalah hasil dari pengolahan data ke dalam bentuk yang lebih bermanfaat bagi penerimanya yang menggambarkan kejadian-kejadian yang nyata untuk digunakan dalam pengambilan keputusan.

1. Menurut Abdul Kadir (2003:26) Informasi merupakan salah satu sumber daya penting dalam manajemen modern.
2. Menurut Mc Fadden, dkk (Abdul Kadir, 2003:31) Informasi adalah data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut.
3. Menurut Tata Sutabri (2005:23) Informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

Informasi diatas dapat disimpulkan bahwa, Informasi adalah data yang diolah menjadi sebuah bentuk yang lebih berguna bagi penerima dan dapat bermanfaat untuk mengambil keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan nyata. Informasi diperoleh setelah data-data mentah diproses atau diolah.

### 2.2.3 Basis Data

Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti secara *implisit*. Data dapat dinyatakan dalam bentuk angka, karakter atau simbol, sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (*database*) (Ramez:2000). Sedangkan menurut George Tsu-der Chou basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang (Abdul, 1999). Menurut *Encyclopedia of Computer Science and Engineer*, para ilmuwan di bidang informasi menerima definisi standar informasi yaitu data yang digunakan dalam pengambilan keputusan (Haidar, 2007).

### 2.2.4 Konsep Dasar *Electronic Learning*

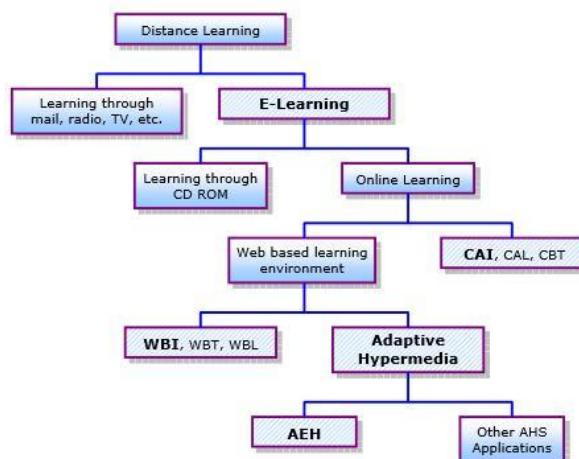
*E-learning* merupakan singkatan dari *Elektronic Learning*, merupakan cara baru dalam proses belajar mengajar yang menggunakan media elektronik khususnya

internet sebagai sistem pembelajarannya. Beberapa ahli mencoba menguraikan pengertian *e-learning* menurut versinya masing-masing, diantaranya:

1. Jaya Kumar C. Koran (2002) : *E-learning* sebagai sembarang pengajaran dan pembelajaran yang menggunakan rangkaian elektronik (LAN, WAN, atau internet) untuk menyampaikan isi pembelajaran, interaksi, atau bimbingan.
2. Dong (dalam Kamarga, 2002) : *E-learning* sebagai kegiatan belajar *asynchronous* melalui perangkat elektronik komputer yang memperoleh bahan belajar yang sesuai dengan kebutuhannya.
3. Rosenberg (2001) : menekankan bahwa *E-learning* merujuk pada penggunaan teknologi internet untuk mengirimkan serangkaian solusi yang dapat meningkatkan pengetahuan dan keterampilan
4. Darin E. Hartley [Hartley, 2001] : *E-Learning* merupakan suatu jenis belajar mengajar yang memungkinkan tersampainya bahan ajar ke siswa dengan menggunakan media internet, intranet atau media jaringan komputer lain.
5. *Glossary of E-Learning Terms* [Glossary, 2001] : *E-Learning* merupakan sistem pendidikan yang menggunakan aplikasi elektronik untuk mendukung belajar mengajar dengan media Internet, jaringan komputer, maupun komputer *stand alone*

Sementara itu pembelajaran *online* adalah bagian dari pembelajaran berbasis teknologi yang memanfaatkan sumber daya internet. Lebih khusus lagi Rosenberg (2001) mendefinisikan *e-learning* sebagai pemanfaatan teknologi internet untuk mendistribusikan materi pembelajaran, sehingga siswa dapat mengakses dari mana saja.

Kaitan antara berbagai istilah yang berkaitan dengan *e-learning* dan pembelajaran jarak jauh dapat diilustrasikan dalam gambar di bawah



[Gambar 2.1] Distance Learning (Surjono, 2006)

Implementasi sistem *e-learning* yang ada sekarang ini sangat bervariasi, namun semua itu didasarkan atas suatu prinsip atau konsep bahwa *e-learning* dimaksudkan sebagai upaya pendistribusian materi pembelajaran melalui media elektronik atau internet sehingga peserta didik dapat mengaksesnya kapan saja dari seluruh penjuru dunia. Ciri pembelajaran dengan *e-learning* adalah terciptanya lingkungan belajar yang *flexible* dan *distributed*.

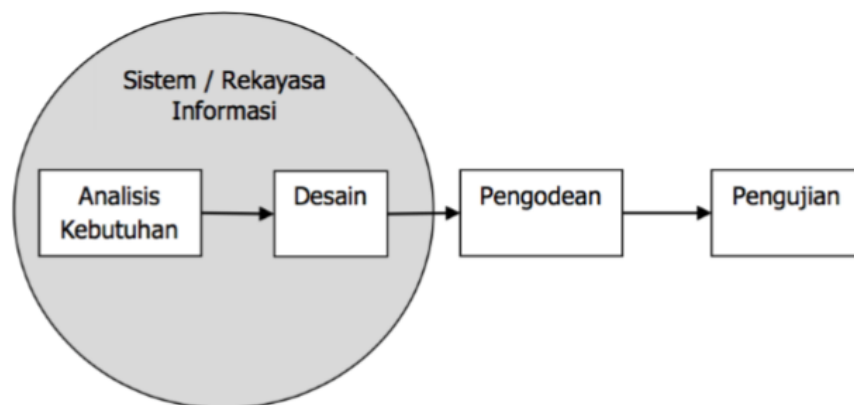
Dalam merancang sistem *e-learning* perlu mempertimbangkan dua hal, yakni peserta didik yang menjadi target dan hasil pembelajaran yang diharapkan. Pemahaman atas peserta didik sangatlah penting, yakni antara lain adalah harapan dan tujuan mereka dalam mengikuti *e-learning*, kecepatan dalam mengakses internet atau jaringan, keterbatasan *bandwidth*, biaya untuk akses internet, serta latar belakang pengetahuan yang menyangkut kesiapan dalam mengikuti pembelajaran. Pemahaman atas hasil pembelajaran diperlukan untuk menentukan cakupan materi, kerangka penilaian hasil belajar, serta pengetahuan awal.

### 2.2.5 Waterfall

*Waterfall* adalah metodologi yang digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi. Siklus hidup sistem itu sendiri merupakan metodologi, tetapi polanya lebih dipengaruhi oleh kebutuhan untuk mengembangkan sistem yang lebih cepat. Pengembangan sistem yang lebih cepat



dapat dicapai dengan peningkatan siklus hidup dan penggunaan peralatan pengembangan berbasis komputer. Tahapan *waterfall* dapat dilihat pada gambar 2.2



[Gambar 2.2] Ilustrasi Model Waterfall (Rosa dan Shlahudin, 2013)

#### A. Analisis

kebutuhan Menganalisis dan mendokumentasikan spesifikasi kebutuhan perangkat lunak agar perangkat lunak sesuai dengan apa yang dibutuhkan pengguna.

#### B. Desain

Pada tahap ini spesifikasi kebutuhan yang telah dianalisis diterjemahkan ke representasi desain seperti struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean agar mudah diimplementasikan ke tahap selanjutnya. Bahasa pemodelan sistem yang digunakan dalam penelitian ini adalah UML ( *Unified Modelling Language* ).

#### C. Pengodean

Pada tahap pengodean atau implementasi ini, desain yang telah dibuat diterjemahkan menjadi perangkat lunak. Perangkat lunak dibantu dengan penyedia data dan bahasa pemrograman.

#### D. Pengujian

Perangkat lunak yang telah selesai dikembangkan diuji untuk mengurangi kesalahan yang mengakibatkan perangkat lunak tidak sesuai dengan kebutuhan pengguna.

### 2.2.6 Object Oriented Programming

*Object Oriented Programming* adalah sebuah paradigma pemrograman yang berorientasikan kepada *object*. Semua data dan fungsi dalam paradigma ini dibungkus

dengan *class-class* atau *object-object*. Dalam pemrograman berbasis objek, kita diminta untuk memahami sebuah masalah dan memodelkan masalah tersebut menjadi sebuah *class* dan kemudian *class* akan diinstansiasi menjadi sebuah *object* pada saat *runtime*. Setiap *class/object* dalam pemrograman berbasis *object* dapat saling berinteraksi dan berkomunikasi satu sama lain untuk mendukung sebuah solusi dari suatu masalah.

Kelebihan OOP dibandingkan dengan *procedural* antara lain:

- Lebih terstruktur dan mudah dibaca
- *Class-Class* dapat digunakan kembali pada *project* yang lain (*reuse*).
- Pemetaan masalah jadi lebih mudah sehingga mudah untuk membuat solusinya.
- Pembatasan akses terhadap suatu fungsi dapat meningkatkan keamanan *code*.
- Interaksi antara *code* lebih terasa.

Adapun bahasa pemrograman yang mendukung OOP antara lain:

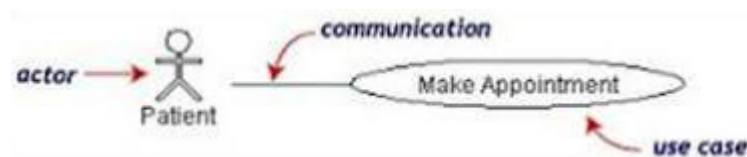
- PHP
- Java
- C++
- Python
- .Net
- Ruby
- Go

### 2.2.7 *Unified Modelling Language (UML)*

*Unified Modelling Language* adalah sebuah bahasa pemodelan dapat berupa *pseudo - code*, *actual code*, gambar, diagram atau bagian panjang deskripsi yang membantu untuk menjabarkan sebuah sistem (Lambang Probo Sumirat, 2017). *Unified Modelling Language* adalah keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP). UML mempunyai banyak tipe diagram yang dibagi menjadi dua kategori, yaitu kategori diagram yang mempresentasikan struktur informasi dan yang lain mempresentasikan behaviour (sifat).

### 1. Use case Diagram

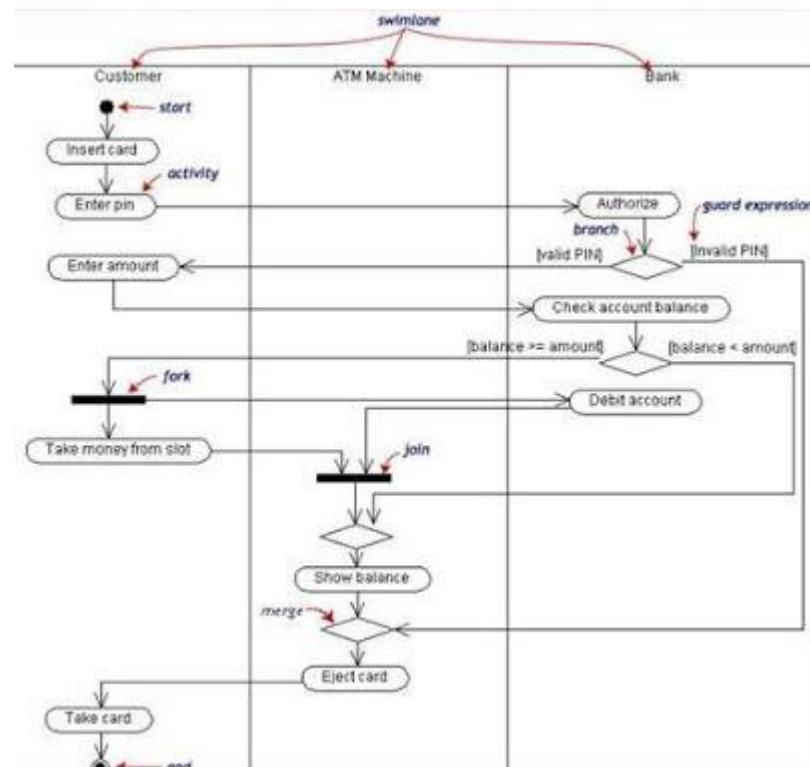
Sebuah *use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase *requirements*, *model use case* menggambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari input user dan respon-respon sistem. Setiap *use case* menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya. Selama pembuatan model *use case* secara *parallel* juga harus ditetapkan obyek-obyek yang terlibat dalam setiap *use case*. *Use case diagram* adalah abstraksi dari interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.



[Gambar 2.3] Use case diagram (jejaring, 2019)

### 2. Activity Diagram

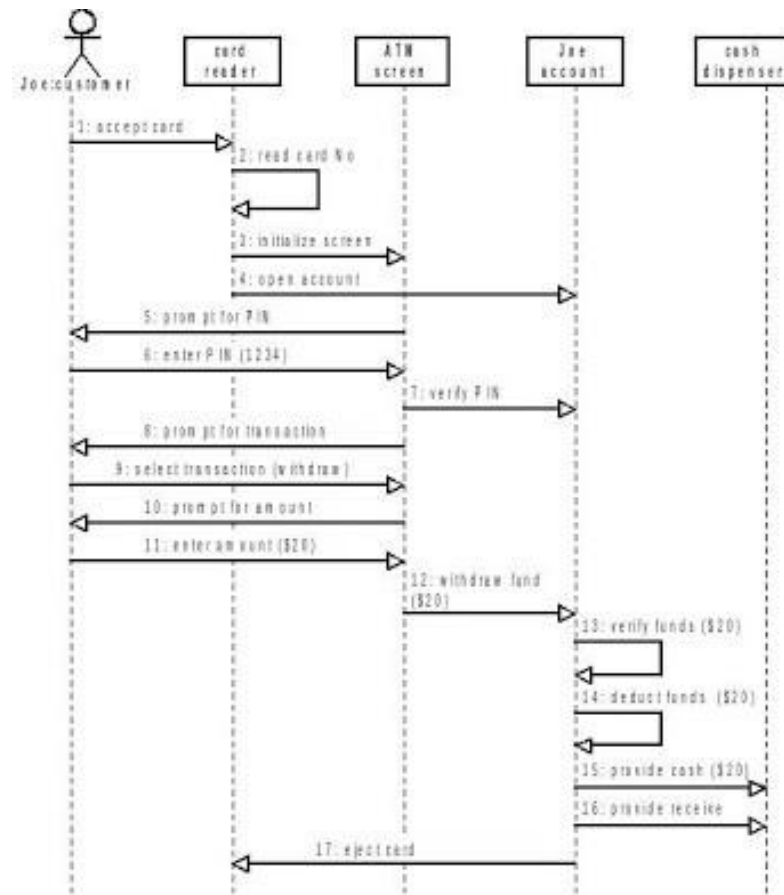
*Activity diagram* menyediakan analisis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. *Activity diagram* dapat digunakan untuk alur kerja model, *use case individual*, atau logika keputusan yang terkandung dalam *metode individual*<sup>3</sup>. *Activity diagram* juga menyediakan pendekatan untuk proses pemodelan paralel. *Activity diagram* lebih lanjut .



[Gambar 2.4] Activity Diagram (Jejaring, 2019)

### 3. Sequence Diagram

*Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. Bersifat dinamis. Diagram urutan adalah interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu. *Sequence diagram* menekankan penyusunan berbasis waktu untuk kegiatan yang dilakukan dengan satu set dari objek yang berkolaborasi. *Sequence diagram* sangat berguna dalam membantu analisis, memahami spesifikasi *real-time* dan menggunakan kasus yang rumit. Diagram ini dapat digunakan untuk mendeskripsikan baik secara fisik dan logis interaksi antara objek.



[Gambar 2.5] Contoh Sequence diagram (jejaring, 2019)

*Sequence diagram* diatas digambarkan contoh *use case* investasi perdagangan. Pada diagram tersebut obyek yang berinteraksi adalah *user*, *userinterface* sistem, dan *interface* terhadap sistem eksternal. Pada diagram tersebut terlihat aliran secara umum,yakni :

1. *User* memilih *account investment*.
2. Kemudian, sistem akan mengirimkan pesan pada sistem *investor* untuk melakukan *query* harga saham dari investasi pada *account user*.
3. Sistem akan menampilkan harga saham pada *account investasi user*.
4. *User* memilih investasi dan jumlah saham yang akan dijual.
5. Sistem akan mengirimkan pesan kepada sistem *investor* untuk menyampaikan permintaan untuk menjual saham yang telah ditentukan oleh *user*.

#### 4. Class Diagram

*Class* adalah dekripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat

memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan system berorientasi objek. *Class diagram* berfungsi untuk menjelaskan tipe dari *object* sistem dan hubungannya dengan *object* yang lain. *Object* adalah nilai tertentu dari setiap *attribute* kelas *entity*. Pada penggambaran *class diagram* ada dikenal dengan kelas analisis yaitu kelas ber-*stereotype*. Tapi yang biasanya dipakai adalah *class diagram* tanpa *stereotype*.

### 2.2.8 Framework

Ralph E. Johsson, Ketua *UIUC patterns/Software Architecture Group* dan koordinator program proyek senior di *Department of Computer Science* pada *University of Illinois*, menyatakan bahwa *framework* adalah desain yang reuseable dan biasanya dinyatakan sebagai satu set abstraksi kelas yang mengatur bagaimana kelas saling terhubung. Perancangan pada *framework* dibuat sedemikian rupa sehingga sebagian atau seluruh *software* dapat digunakan kembali. Contohnya: seorang *designer* membuat suatu UI (*user interface*) yang hanya dapat digunakan pada bagian tertentu suatu aplikasi. Desain tersebut tidak dapat digunakan kembali oleh bagian lainnya dari aplikasi tersebut, ini tentunya kurang efektif bila menggunakan suatu *framework*, contohnya *MacApp* atau *Macintosh Application Framework* maka dapat menciptakan desain untuk keseluruhan aplikasi sehingga meningkatkan efisiensi kerja.

Menurut Pressman (2005), *framework* adalah kerangka kode yang dapat disempurnakan dengan kelas-kelas yang spesifik atau dengan fungsi yang telah dirancang untuk mengatasi masalah yang dihadapi. Disimpulkan bahwa *framework* biasanya bersifat *object oriented* dan merupakan suatu desain sistem yang dapat digunakan kembali. Tujuannya untuk mengurangi pembuatan kembali kode yang sama sehingga Programmer dapat lebih fokus mengerjakan bagian lainnya.

### 2.2.13 Django

Django adalah sebuah *web framework* yang *open source* dan berbasis *Python*. *Web framework* adalah sebuah kerangka kerja yang dapat mempermudah pengembang *software* khususnya *website*. Django menggunakan konsep *Model Template and View* (MTV). *Model* adalah layer yang digunakan untuk berinteraksi dengan *database*. *Template* adalah *layar* yang digunakan untuk menangani masalah tampilan seperti XML, HTML dan lainnya. *View* adalah *layer* yang menghubungkan layer model dan *template* yang di dalamnya berisikan logika pengolahan data dari model dan menampilkannya di *template*. Sejak penyebaran pola MVC ke dalam pengembangan web, *Python* telah menyediakannya beberapa *web framework*, seperti Django, TurboGears, dan Zope. Django memiliki kelebihan di antara *web framework* lainnya yaitu (Hourieh, 2008).

#### a. *Tight Integration between Components*

Django menyediakan seperangkat komponen yang terintegrasi. semua komponen telah dikembangkan oleh tim django sendiri. Django awalnya dikembangkan sebagai *framework in-house* untuk mengelola serangkaian berita yang berorientasi pada situs web. Kemudian kodenya dirilis di Internet dan tim django melanjutkannya pengembangan menggunakan model *Open Source*. Oleh sebab itu, komponen django dirancang untuk *integration*, *reusability* dan *speed* sejak awal.

#### b. *Object-Relational Mapper (ORM, O/RM, and O/R mapping tool)*

Komponen database django, *Object-Relational Mapper* (ORM), menyediakan jembatan antara model data dan *database engine*. Ini mendukung satu set besar dalam sistem *database*, dan beralih dari satu mesin ke mesin lainnya dalam permasalahan perubahan sebuah konfigurasi *file*. Hal ini memberi fleksibilitas besar bagi pengembang jika menginginkan keputusan dari satu mesin *database* ke *database* lainnya.

#### c. *Clean URL Design*

Sistem URL di django sangat fleksibel dan *powerful*. Ini memungkinkan dalam mendefinisikan pola untuk URL dalam aplikasi dan menentukan fungsi *python* untuk menangani setiap pola. Hal ini memungkinkan pengembang membuat URL yang ramah pengguna dan mesin pencari.

#### d. *Authomatic Admin Interface*

Django hadir dengan antarmuka administrasi yang siap digunakan. Antarmuka ini membuat pengelolaan data aplikasi dengan mudah. Hal ini juga sangat *fleksibel* dan dapat disesuaikan.

e. *Advanced Development Environment*

Django menyediakan lingkungan pengembangan yang sangat bagus. Django hadir dengan *web server* yang ringan untuk pengembangan dan pengujian. Saat mode debugging diaktifkan, django memberikan pesan kesalahan yang sangat teliti dan terperinci dengan lebih banyak lagi informasi *debug*. Semua ini membuat proses isolasi dan memperbaiki *bug* dengan sangat mudah.

f. Mendukung Multibahasa

Django mendukung situs *web* multi bahasa melalui built-in internasionalisasi sistem. Ini bisa sangat berharga bagi mereka yang bekerja di situs *web* dengan lebih dari satu bahasa. Sistem membuat penerjemahan antarmuka menjadi tugas yang sangat sederhana.

Fitur standar yang diharapkan dari *framework* semuanya tersedia di django, seperti (Hourieh, 2008):

1. mesin *template* dan penyaringan *teks* dengan sintaks sederhana namun bisa diperluas;
2. form generasi terbaru dan API validasi;
3. sistem otentikasi yang dapat diperluas;
4. sistem *caching* untuk mempercepat kinerja aplikasi;
5. *feed framework* untuk menghasilkan RSS feeds.

Meskipun django tidak menyediakan perpustakaan *JavaScript* untuk mempermudah kerja sama dengan Ajax tetapi, django menyediakan seperangkat komponen yang terintegrasi dan matang, dengan dokumentasi yang bagus, di <http://www.djangoproject.com/documentation/>, berkat komunitas pengembang dan penggunanya yang besar (Hourieh, 2008).

### 2.2.8 Python

*Python* merupakan bahasa pemrograman tingkat tinggi yang dewasa ini telah menjadi standar dalam dunia komputasi ilmiah. *Python* merupakan bahasa pemrograman *open source multi- platform* yang dapat digunakan pada berbagai macam sistem operasi (*Windows*, *Linux*, dan *MacOS*), selain itu *python* juga merupakan bahasa pemrograman yang fleksibel dan mudah untuk dipelajari. Program



yang ditulis dalam *python* umumnya lebih mudah dibaca dan jauh lebih ringkas dibandingkan penulisan program dalam bahasa *C* atau *Fortran*. *Python* juga memiliki modul standar yang menyediakan sejumlah besar fungsi dan algoritma, untuk menyelesaikan pekerjaan seperti mengurai data teks, memanipulasi dan menemukan *file* dalam *disk*, membaca/menuliskan *file* terkompresi, dan mengunduh data dari *web server*.

Bahasa pemrograman *python* dalam banyak hal berbeda dengan bahasa pemrograman prosedural, seperti *C*; *C++*; dan *Fortran*. Dalam *Fortran*, *C*, dan *C++*, *file source code* harus dikompilasi ke dalam bentuk *executable file* sebelum dijalankan. Pada *python*, tidak terdapat langkah kompilasi, sebagai gantinya *source code* ditafsirkan secara langsung baris demi baris. Keunggulan utama dari suatu bahasa pemrograman terinterpretasi seperti *python* adalah tidak membutuhkan pendeklarasian variabel, sehingga lebih fleksibel dalam penggunaannya. Namun, terdapat kelemahan yang mencolok, yaitu program-program numerik yang dijalankan pada *Python* lebih lambat jika dibandingkan dengan bahasa pemrograman terkompilasi, *Python* memiliki banyak fungsi-fungsi sederhana yang dapat menjalankan hal-hal yang umumnya dikerjakan dengan subroutine rumit dalam *C* atau *Fortran* (Sandy, 2017).

Tidak perlu khawatir akan *operating system* yang digunakan, karena bahasa pemrograman ini dapat berjalan di berbagai *operating system* antara lain *UNIX*, *Macintosh* dan *DOS Machine*. Bahasa pemrograman *Python* memiliki dokumentasi yang lengkap (Swaroop, 2005).

Menurut Swaroop (2005), berikut adalah fitur-fitur dari bahasa pemrograman *python*.

a. Simple

*Python* merupakan bahasa pemrograman yang simple dan minimalis. Membaca dan menulis dengan bahasa pemrograman *python* hampir mirip dengan membaca dan menulis dengan menggunakan bahasa inggris. Ini adalah salah satu kekuatan *python* yang membuat *python* mudah dipahami dan diimplementasikan. Sehingga pengembang aplikasi dapat berkonsentrasi pada solusi dari masalah yang dihadapi daripada memahami bahasa *python* itu

sendiri.

b. Mudah dipahami

Seperti yang sudah disebutkan sebelumnya, *python* merupakan bahasa pemrograman yang mudah dipelajari. *python* memiliki sintaks yang sederhana.

c. Gratis dan *Open source*

*Python* merupakan salah satu contoh dari FLOSS (*Free/Libre and Open source Software*). Sederhananya, pengembang dapat dengan bebas mendistribusikan salinan dari software ini, membaca *source code*-nya, mengubah isi dari *source code* dan menggunakannya untuk software yang baru atau yang sudah ada. Latar belakang FLOSS adalah komunitas yang saling berbagi pengetahuan. Oleh karena itu *python* merupakan bahasa pemrograman yang baik, karena terus dikembangkan oleh komunitas, baik yang menggunakannya atau hanya ingin mencobanya. *python* termasuk kedalam kelompok bahasa pemrograman tingkat tinggi. Dengan menggunakan *python*, pengembang tidak perlu repot-repot mengetahui cara mengelola memori untuk program yang dibuat seperti yang dilakukan oleh pemrograman tingkat rendah.

d. Portabel

Karena *python* bersifat open source, *python* telah dibuat untuk dapat bekerja (porting) di berbagai platform. Semua program *python* yang dibuat dapat bekerja pada setiap platform tanpa harus melakukan perubahan. Pengembang dapat menjalankan *python* pada platform GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS / 2, Amiga, AROS, AS / 400, BeOS, OS / 390, z / OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE dan PocketPC. Bahkan pengembang dapat menggunakan platform seperti Kivy untuk membuat game untuk komputer, iPhone, iPad, dan Android.

### 2.2.9 Visual Studio Code

*Visual Studio Code* adalah *software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan built-in dukungan untuk *JavaScript*, naskah dan *Node.js* dan memiliki *array* beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C # , *Python*, dan PHP. Hal ini didasarkan sekitar *Github* ini *Elektron*, yang merupakan versi *cross-platform* dari Atom komponen kode editing, berdasarkan *JavaScript* dan HTML5. Editor ini adalah fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja

dengan teknologi *cloud* yang terbuka *Microsoft*. *Visual Studio Code* menggunakan *open source NET* perkakas untuk memberikan dukungan untuk ASP.NET C # kode, membangun alat pengembang *Omnisharp* NET dan *compiler Roslyn*. Antarmuka yang mudah untuk bekerja dengan, karena didasarkan pada gaya *explorer* umum, dengan panel di sebelah kiri, yang menunjukkan semua *file* dan *folder* Anda memiliki akses ke panel editor di sebelah kanan, yang menunjukkan isi dari *file* yang telah dibuka. Dalam hal ini, editor telah dikembangkan dengan baik, dan menyenangkan pada mata. Ia juga memiliki fungsi yang baik, dengan *intellisense* dan *autocomplete* bekerja dengan baik untuk JSON, CSS, HTML, {kurang}, dan *Node.js*.

*Visual Studio Code* telah dirancang untuk bekerja dengan alat-alat yang ada, dan *Microsoft* menyediakan dokumentasi untuk membantu pengembang bersama, dengan bantuan untuk bekerja dengan ASP.NET 5, *Node.js*, dan *Microsoft* naskah, serta alat-alat yang dapat digunakan untuk membantu membangun dan mengelola aplikasi *Node.js*. *Visual Studio Code* benar-benar sedang ditargetkan pada pengembang *JavaScript* yang ingin alat pengembangannya lengkap untuk *scripting server-side* mereka dan yang mungkin ingin usaha dari *Node.js* untuk kerangka berbasis NET. *Visual Studio Code*, adalah belum *solid*, lintas *platform* kode Editor ringan, yang dapat digunakan oleh siapa saja untuk membangun aplikasi untuk Web.

#### **2.2.10 Xampp Server**

Dalam pembangunan sebuah *website* pastinya setiap *programmer* memerlukan bantuan *web server* untuk mengkoneksikan *file-file website* ke basis data. Beberapa *web server* yang sering digunakan diantaranya: *Apache Web Server*, *Sun Java System Web Server*, *Xampp Server*, *Wamp server*, *Xitami Web Server*, dan sebagainya. Dalam hal ini, penulis menggunakan *Xampp Server* dalam membangun web tersebut.

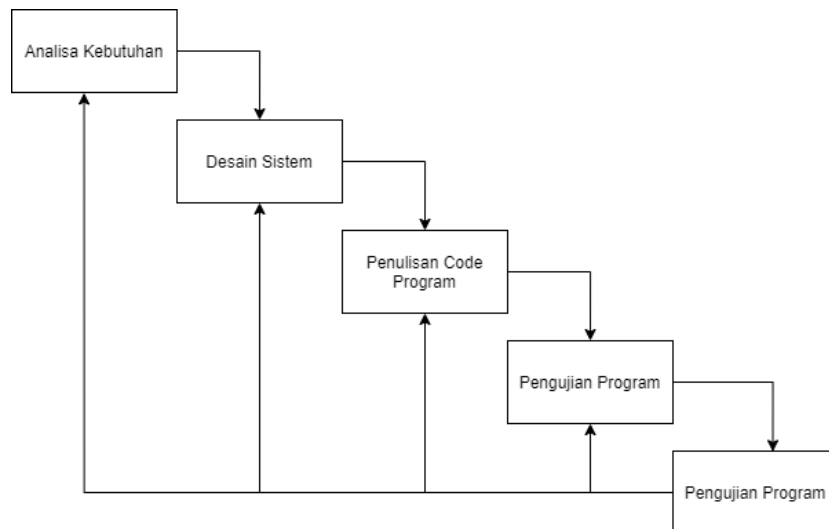
Menurut Winpec Solution (2010:1) “*XAMPP* merupakan suatu paket instalasi *Apache*, *PHP*, dan *MySQL*”. Dengan aplikasi ini, anda dapat langsung melakukan instalasi *Apache*, *PHP*, dan *MySQL* sekaligus Aplikasi *XAMPP* ini dapat diperoleh cukup dengan melakukan *download*.

## BAB III

### METODOLOGI PENELITIAN

#### 1.1 Metode Perancangan

Pada bab ini akan dibahas metode penelitian yang akan peneliti gunakan. Dalam melakukan pengembangan sistem aplikasi “SDku Online”, peneliti menggunakan analisis *metode waterfall* atau sering disebut juga model air terjun yang dapat dilihat pada gambar 3.1 di bawah ini.



[Gambar 3.1] Diagram Waterfall

Dari diagram *waterfall* diatas dapat dilihat bahwa pembahasan pada bab 3 akan dimulai dari menganalisis permasalahan yang ada dan mengumpulkan informasi yang menunjang dalam proses pengerjaan.

#### 1.2 Studi Literatur

Studi literatur berfungsi untuk memahami teori yang akan di gunakan sebagai landasan ilmiah dalam penelitian yang akan dilakukan. Adapun teori pendukung penelitian ini telah disebutkan pada bab sebelumnya yaitu pada dasar teori. pengumpulan data dengan menghimpun dan menganalisis dokumen-dokumen, baik dokumen tertulis, gambar maupun elektronik. Cara pengumpulan data yang akan dilakukan penulis adalah dengan studi literatur pada buku-buku yang terkait, jurnal dan penelitian yang sudah dilakukan berkaitan dengan penelitian yang akan dilakukan

### 1.3 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan untuk mengumpulkan segala informasi mengenai *sistem*/perangkat lunak yang akan dikembangkan secara spesifik. Analisis kebutuhan meliputi analisis kebutuhan fungsionalitas, analisis kebutuhan perangkat keras, dan analisis kebutuhan perangkat lunak. Analisis kebutuhan diperlukan untuk mendapatkan kebutuhan spesifik yang diperlukan sesuai dengan kebutuhan pengguna dan *stakeholder*

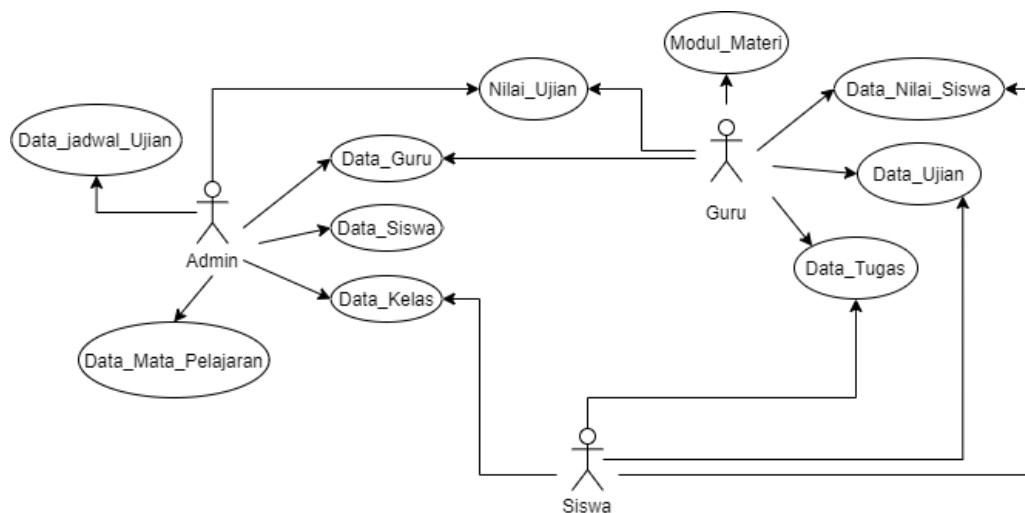
Alat dan bahan yang digunakan untuk menunjang pengerjaan penelitian ini adalah sebagai berikut:

- a. Laptop dengan spesifikasi:
  1. Laptop : *ASUS VivoBook Max X441B*
  2. Sistem operasi : *Windows 10 Pro 64-bit (10.0, Build 18362)*
  3. *Processor* : *AMD A9-9420 RADEON R5, 5 COMPUTE CORES 2C+3G*
  4. *Memory* : *4096MB RAM*
- b. Perangkat lunak:
  1. *Webserver* : *XAMPP v1.8.2-6*
  2. *Code editor* : *Visual Studio Code*
  3. *Browser* : *Google Chrome v.80.0.3987.132*
  4. Bahasa pemrograman : *Python 3*
  5. *Framework* : *Django v3.0.7*
  6. *Software pendukung* :
    - a) *Draw.io v12.9.14*
    - b) *Visual Paradigm v.16.1*
    - c) *Adobe Photoshop Cs6 v13.0.0.0*

### 1.4 Perancangan Sistem

#### 1.4.1 Use case Diagram

*Use Case Diagram* (Gambar 3.2) adalah pemodelan untuk menggambarkan behavior atau kelakuan sistem yang akan dibuat. *Use case diagram* menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara sederhana, *diagram use case* digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.



[Gambar 3.2] Use case aplikasi SDku Online

Pada gambar diatas, terdapat 3 actor yaitu admin, guru dan siswa, yang dimana setiap actor memiliki aktivitas yang berbeda-beda. Aktivitas dari masing-masing aktor dapat dilihat pada table berikut:

| Aktor | Aktivitas   |
|-------|---|
| Admin | Admin dapat melakukan <i>insert, update, delete</i> untuk data guru           |
|       | Admin dapat melakukan <i>insert, update, delete</i> untuk data siswa          |
|       | Admin dapat melakukan <i>insert, update, delete</i> untuk data kelas          |
|       | Admin dapat melakukan <i>insert, update, delete</i> untuk data mata pelajaran |
|       | Admin dapat melakukan <i>insert, update, delete</i> untuk data jadwal ujian   |
|       | Admin dapat melihat nilai ujian   |

[Tabel 3.1] Aktivitas pada aktor Admin

| Aktor | Aktivitas   |
|-------|---|
| Guru  | Guru dapat melakukan <i>insert, update, delete</i> untuk data guru  |
|       | Guru dapat melakukan <i>insert, update, delete</i> untuk data ujian |
|       | Guru dapat melakukan <i>insert, update, delete</i> untuk modul      |

|  |  |
|--|--|
|  | materi   |
|  | Guru dapat melakukan <i>insert</i> , <i>update</i> , <i>delete</i> untuk data nilai siswa (khusus untuk nilai tugas) |
|  | Guru dapat melihat nilai ujian   |

[Tabel 3.2] Aktivitas pada aktor guru

| Aktor | Aktivitas  |
|-------|--|
| Siswa | Siswa dapat melakukan <i>insert</i> untuk data ujian |
|       | Siswa dapat melihat data nilai siswa                 |
|       | Siswa dapat melihat data tugas                       |
|       | Siswa dapat melihat data kelas                       |

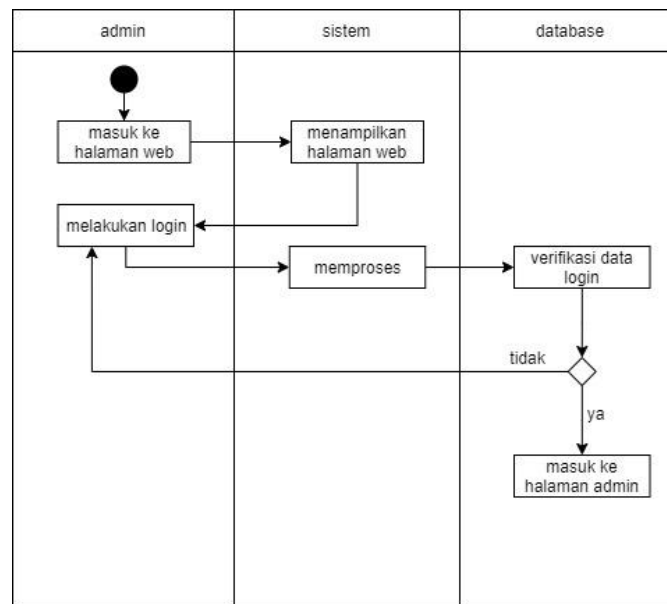
[Tabel 3.3] Aktivitas pada aktor siswa

### 1.4.2 Activity Diagram

*Activity Diagram* dibuat untuk menggambarkan proses dan urutan aktivitas, diagram ini menunjukkan langkah-langkah dalam proses kerja sistem yang akan dibuat. Adapun Proses kerja sistem adalah sebagai berikut:

#### 1.4.2.1 Activity Login Sebagai Admin

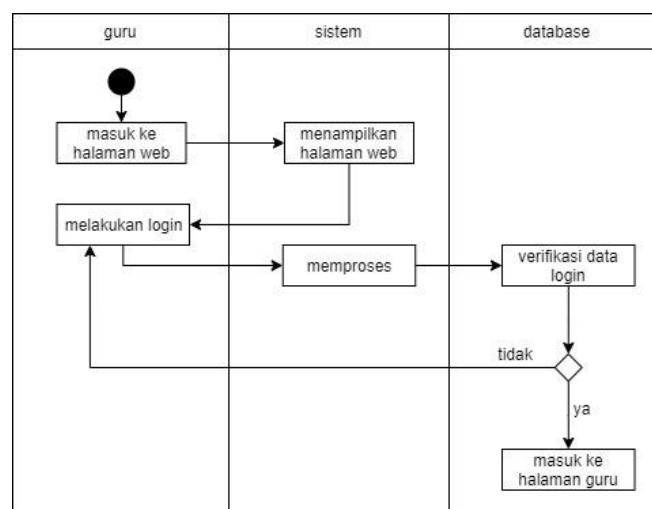
Aktivitas *login* sebagai admin (*user*) melibatkan admin, sistem dan *database*. Saat *user* menjalankan aplikasi, sistem akan menampilkan menu *login* untuk admin, kemudian *user* memasukkan *username* dan *password* agar bisa masuk pada sistem, setelah *user* memasukkan *username* dan *password*, sistem akan melakukan validasi atau verifikasi dari *database* yang diinputkan oleh *user*, jika data yang di inputkan benar maka sistem akan mengarahkan pada halaman berikutnya (halaman admin), dan jika data yang di inputkan salah, maka sistem akan mengarahkan kembali ke halaman *login* untuk mengulang penginputan data. Untuk skema diagramnya bisa dilihat pada gambar 3.2



[Gambar 3.3] Aktivitas login sebagai admin

#### 1.4.2.2 Activity Login Sebagai Guru

Aktivitas *login* sebagai guru melibatkan guru (*user*), sistem dan *database*. Saat *user* menjalankan aplikasi, sistem akan menampilkan menu *login* untuk guru, kemudian *user* memasukkan *username* dan *password* agar bias masuk pada sistem, setelah *user* memasukkan *username* dan *password*, sistem akan melakukan validasi atau verifikasi dari data yang diinputkan oleh *user*, jika data yang di inputkan benar maka sistem akan mengarahkan pada halaman berikutnya (halaman guru), dan jika data yang di inputkan salah, maka sistem akan mengarahkan kembali ke halaman *login* untuk mengulang penginputan data. Untuk skema diagramnya bisa dilihat pada gambar 3.3

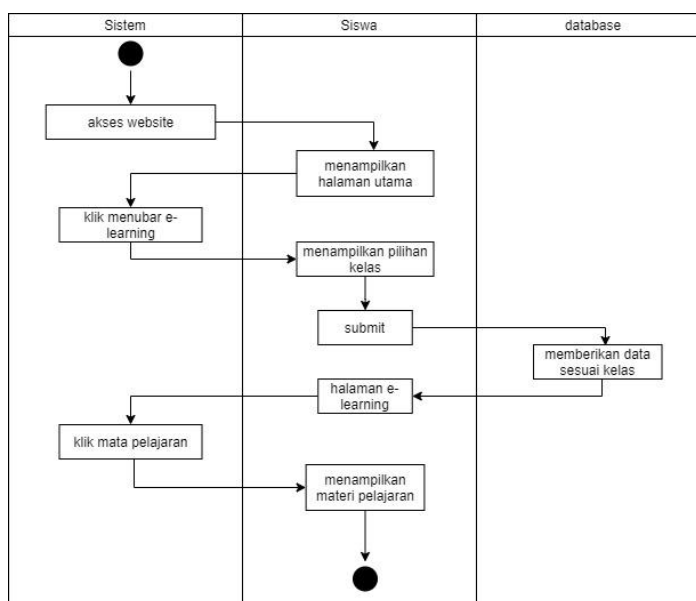


[Gambar 3.4] Aktivitas login sebagai guru



#### 1.4.2.3 Activity E-learning

Aktivitas *e-learning* melibatkan *user public* dan sistem. Saat *user* menjalankan *website*, maka sistem akan menampilkan halaman utama *website*, pada halaman utama, terdapat beberapa menu bar, ketika ingin mengakses *e-learning*, maka *user* harus mengklik menu *e-learning*, setelah mengklik menu *e-learning* maka sistem akan mengarahkan ke halaman *e-learning* yang dimana pada halaman ini terdapat beberapa item seperti menentukan kelas untuk menampilkan materi sesuai kelas yang dipilih, setelah *user* menentukan kelas maka tahap selanjutnya *user* harus mengklik *submit* untuk melanjutkan. Setelah melakukan *submit*, database akan memproses inputan data sesuai data kelas yang di inputkan, kemudian sistem akan mengarahkan ke halaman *e-learning* kelas yang telah di pilih pada halaman. Pada halaman ini terdapat *list* mata pelajaran yang dimana jika kita mengklik salah satu dari mata pelajaran tersebut akan menampilkan materi belajar yang berupa *video learning*. Untuk skema diagramnya bisa dilihat pada gambar 3.4

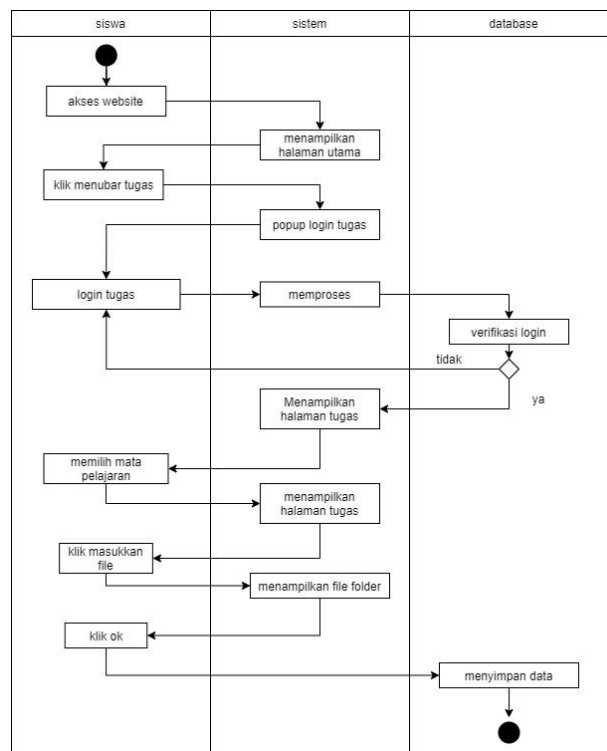


[Gambar 3.5] Aktivitas *e-learning*

#### 1.4.2.4 Activity Upload Tugas

Aktivitas *upload* tugas melibatkan siswa dan sistem. Ketika siswa akan mengirim tugas, maka siswa harus masuk ke sistem terlebih dahulu dengan melakukan *login*, ketika akan masuk ke halaman tugas siswa harus mengubah opsi *login* pada *selection bar* dan memilih “siswa tugas” agar bisa terhubung ke halaman tugas. Setelah mengganti opsi pada *selection bar* tahap selanjutnya adalah

memasukkan *username* dan *password* yang kemudian sistem akan memproses dan database akan melakukan validasi login, jika data yang dimasukan benar maka sistem akan menampilkan halaman tugas setelah berhasil masuk ke halaman *website*, Pada halaman tugas ini, sistem akan menampilkan mata pelajaran yang dimana siswa harus memilih mata salah satu dari mata pelajaran tersebut untuk melakukan pengiriman tugas, setelah memilih mata pelajaran sistem akan menampilkan halaman berikutnya. Untuk pengiriman tugas siswa harus memasukkan file pada menu yang telah disediakan dan kemudian menekan button “Ok” untuk mengirim tugas.

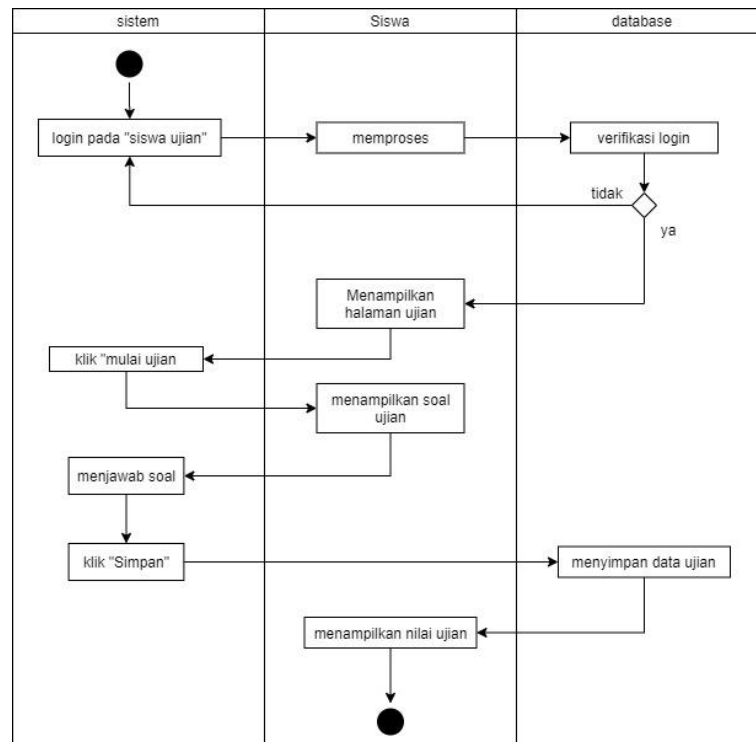


[Gambar 3.6] Aktivitas upload tugas

#### 1.4.2.5 Activity Login Ujian

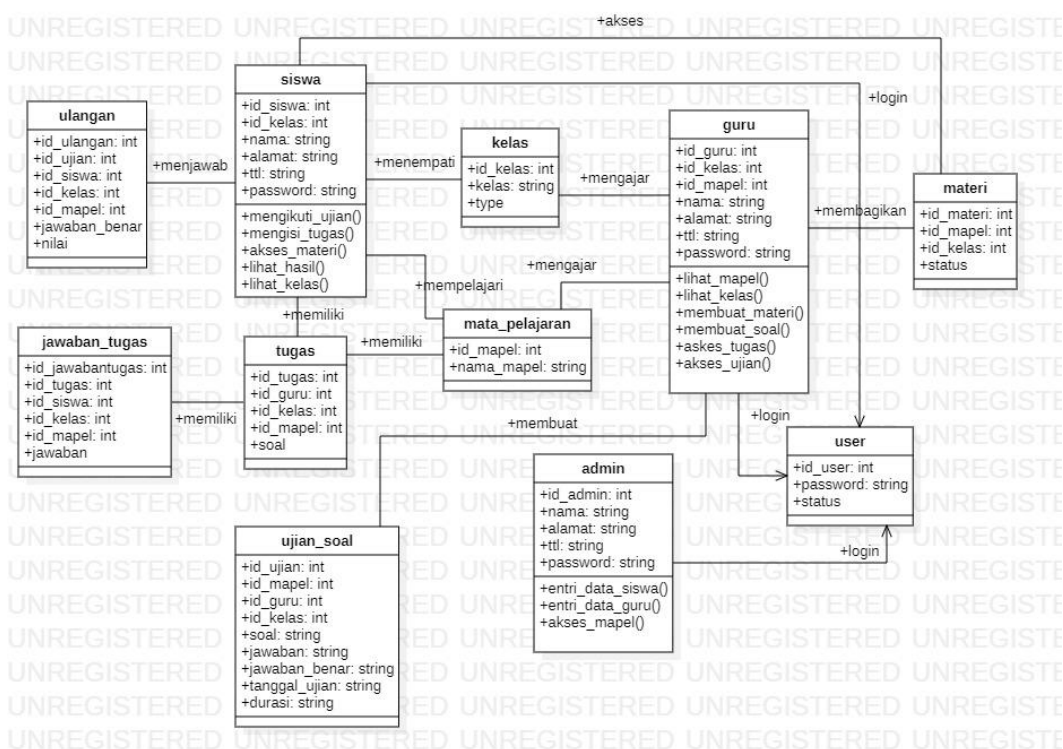
Aktivitas ujian melibatkan siswa sebagai *user*, sistem dan *database*, untuk mengakses halaman ujian *user* harus mengklik menubar ujian yang ada pada halaman utama, kemudian akan muncul *popup* untuk melakukan *login*. Setelah berhasil *login*, *database* akan melakukan verifikasi data, jika data benar, maka *user* akan di arahkan ke halaman ujian. Pada halaman ujian terdapat 2 kondisi, yaitu kondisi sebelum dan ketika mulai ujian, yang memedakan adalah ketika *user* mengaksesnya sebelum jadwal ujian maka tampilan yang akan muncul adalah tampilan yang berupa *countdown* atau penghitung mundur. Kondisi lainnya adalah ketika *user login* setelah jam/jadwal maka tampilan yang akan dimunculkan oleh sistem adalah berupa halaman yang menyatakan siap mulai yang ditandai dengan button mulai yang bisa di

klik. Ketika berhasil mengakses soal, *user* akan mengerjakan soal dan mengklik “simpan” jika *user* telah selesai mengerjakan soal, maka sistem akan secara otomatis menyimpan data ujian kemudian sistem akan memberikan respon berupa tampilan nilai akhir dan *grade* yang diperoleh oleh *user*. Untuk selengkapnya bisa melihat skema pada gambar 3.7



[Gambar 3.7] Aktivitas ujian

### 1.4.3 Class Diagram



[Gambar 3.8] Class Diagram Aplikasi SDku Online

Perancangan pada *Class Diagram* [Gambar 3.8] berguna untuk memberikan gambaran sistem berdasarkan alur relasi antar *entity* dalam tabel. Pada rancangan ini hanya beberapa tabel saja yang dapat melakukan relasi. Sehingga penulis hanya memberikan bagian inti dari tabel yang dapat melakukan relasi. [Gambar 3.8] merupakan gambaran perancangan sistem berdasarkan relasi antar entitas. Dimana terdapat beberapa kardinalitas yang terbentuk antar entitas yang ada. Pembagian kardinalitas juga disajikan pada [Tabel 3.4]

| Entitas 1 | Hubungan | Entitas 2      |
|-----------|----------|----------------|
| siswa     | m:1      | kelas          |
| siswa     | 1:m      | mata pelajaran |
| siswa     | 1:m      | tugas          |
| siswa     | 1:1      | ulangan        |
| siswa     | 1:m      | materi         |
| guru      | 1:1      | mata pelajaran |
| guru      | 1:1      | ujian soal     |
| guru      | 1:m      | materi         |

|       |     |                |
|-------|-----|----------------|
| guru  | 1:  | kelas          |
| guru  | 1:1 | user           |
| tugas | m:1 | mata pelajaran |
| tugas | 1:1 | jawaban tugas  |
| user  | 1:1 | admin          |

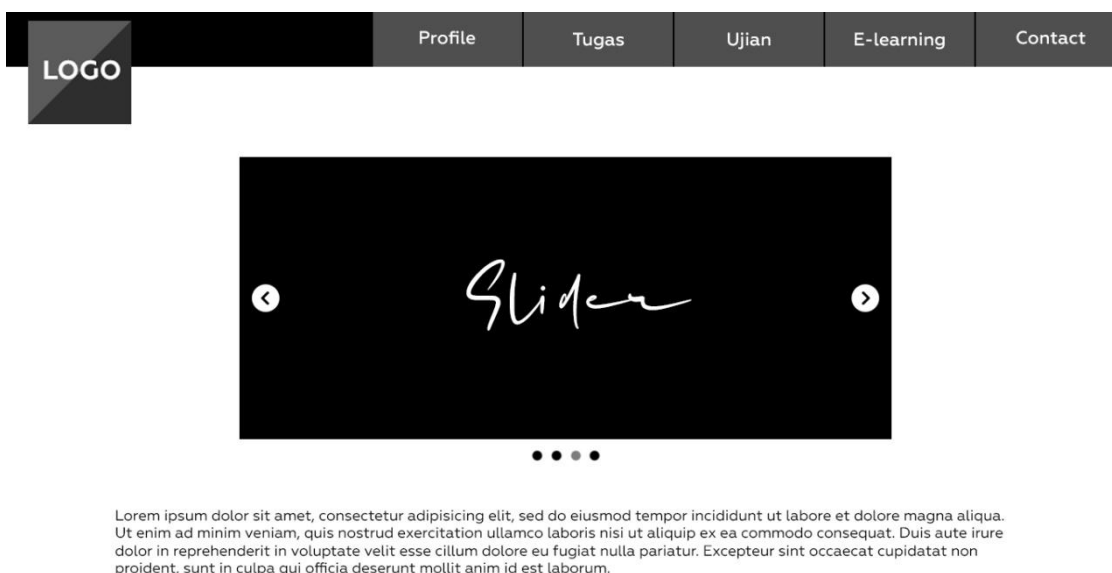
[Tabel 3.4] Kardinalitas antar entitas

#### 1.4.4 Perancangan Antar Muka

Pada tahapan ini akan dijelaskan tentang perancangan menu sistem dan perancangan antar muka (*interface*) sesuai dengan kebutuhan sistem. Antar muka yang dirancang dalam aplikasi ini adalah sebagai berikut

##### a. Perancangan Tampilan Halaman Utama

Halaman utama adalah halaman yang menampilkan pengenalan atau *welcome screen*. Halaman ini muncul pertama kali saat *website* di akses oleh pengguna. Pada halaman ini terdapat *logo website*, menubar untuk mengakses *profile*, tugas, ujian, *e-learning*, dan *contact*. Perancangan halaman utama disajikan dalam [Gambar 3.9]

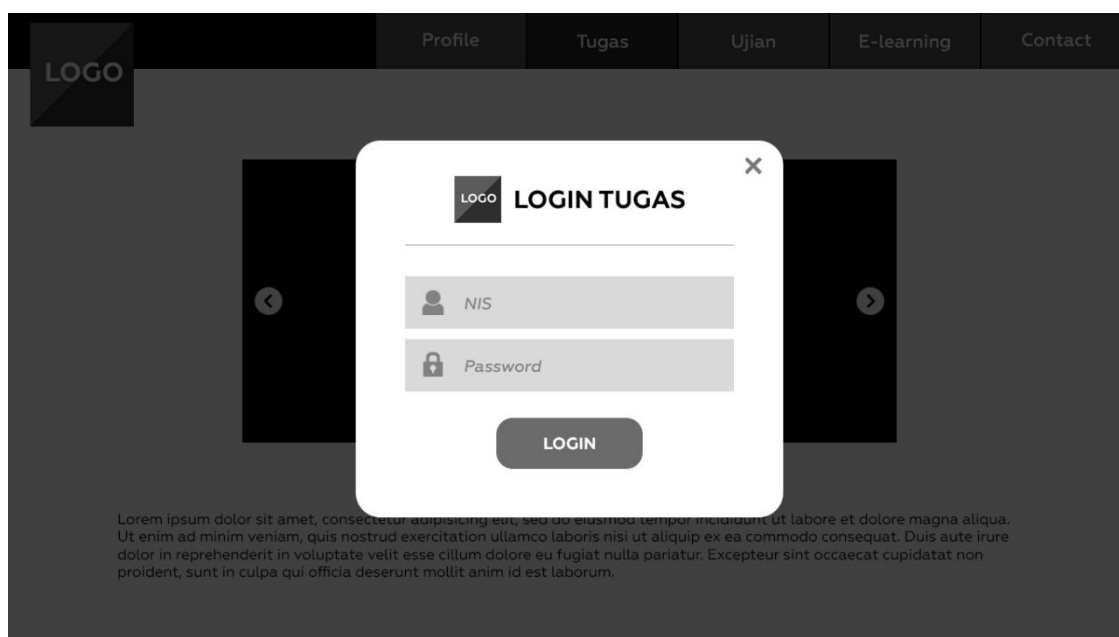


[Gambar 3.9] Rancangan halaman utama

##### b. Rancangan Tampilan Halaman Login Tugas

Halaman tugas merupakan halaman yang memfasilitasi siswa dalam pengumpulan tugas. Pada halaman ini terdapat beberapa opsi mata pelajaran yang bisa di pilih oleh


siswa sesuai dengan tugas yang akan dikumpulkan. Untuk mengakses halaman tugas, *user* harus mengklik tugas pada *menubar* dan akan muncul *pop up* berupa himbauan untuk melakukan login terlebih dahulu, *pop up login* disajikan pada [Gambar 3.8]. Setelah berhasil masuk pada halaman tugas, terdapat *menubar* yang berisi berbagai jenis mata pelajaran siswa, untuk mengakses tugas, siswa harus mengklik mata pelajaran sesuai tugas yang akan dikumpulkan, ketika siswa telah memilih salah satu mata pelajaran maka sistem akan mengarahkan ke halaman berikutnya yang menampilkan *text* soal tugas dan menu selection untuk mengunggah *file* tugas. Perancangan halaman *login* disajikan pada [Gambar 3.10] dan [Gambar 3.11]



[Gambar 3.10] Perancangan halaman login tugas

**SELAMAT DATANG**

Nama : Lalu Septian Wahyu Adipratama  
 Kelas : VIa  
 Nis : 35xxxxx



Logout

|            |                  |     |     |                |       |
|------------|------------------|-----|-----|----------------|-------|
| Matematika | Bahasa Indonesia | IPA | IPS | Bahasa Inggris | Agama |
|------------|------------------|-----|-----|----------------|-------|


"Text Soal"

1. ....

2. ....

3. ....

4. ....

 File...

Kirim

[Gambar 3.11] Perancangan halaman tugas

#### c. Perancangan Halaman Ujian

Pada halaman ujian sistem memiliki 2 kondisi/keadaan ketika di akses. Keadaan pertama adalah ketika user mengakses sebelum jam/jadwal ujian dimulai maka sistem akan menampilkan *countdown* atau penghitung waktu mundur, keadaan sistem pada kondisi pertama disajikan dalam [Gambar 3.12]. Keadaan kedua adalah ketika siswa mengakses sesuai jam/jadwal ujian sistem akan menampilkan halaman sudah bisa di akses dengan *button menu* yang bisa di klik, keadaan ini disajikan pada [Gambar 3.13]. Ketika siswa telah berhasil masuk pada halaman ujian sistem akan menampilkan soal-soal yang bisa di kerjakan, tampilan halaman ini disajikan pada [Gambar 3.14]



[Gambar 3.12] Kondisi pada keadaan pertama



[Gambar 3.13] Kondisi pada keadaan kedua



## SELAMAT DATANG

Nama : Lalu Septian Wahyu Adipratama  
 Kelas : VIa  
 Nis : 35xxxxx  
 Ujian : Matematika

Logout

Selamat Mengerjakan

1. Hasil dari  $9 \times 50 \div 30$  adalah...

☐ 5  
☒ 15  
☐ 20  
☐ 8

2. Soal matematika: Hasil dari  $172 - 152$  adalah....

☐ 5  
☒ 64  
☐ 43  
☐ 18

3. Hasil dari  $70 - (-25)$  adalah....

☐ -15  
☒ 95  
☐ 83  
☐ 44

234856

JamMenitDetik

[Gambar 3.14] Halaman ujian siswa

#### d. Perancangan Halaman *E-learning*

Halaman *e-learning* adalah halaman yang disediakan untuk edukasi siswa, siswa bisa mengakses materi mata pelajaran yang disediakan dalam bentuk video dari youtube yang telah di susun dalam sebuah *playlist* sesuai dengan materi pelajaran yang ada. Tampilan halaman *e-learning* disajikan dalam [Gambar 3.15] dan [Gambar 3.16]

LOGO

Profile

Tugas

Ujian

E-learning

Contact

◀
▶

Slider

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

▼ Kelas

Submit

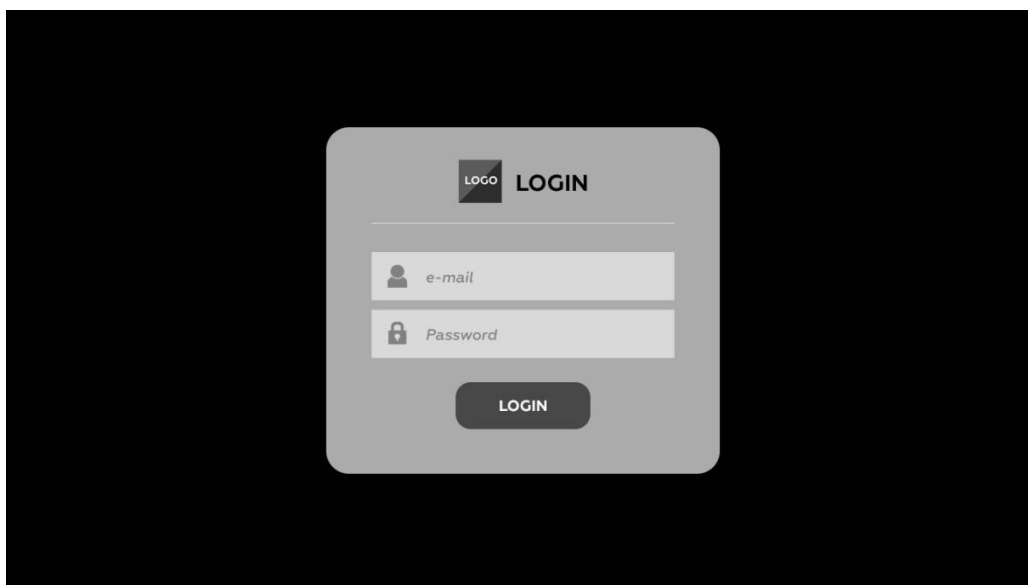
[Gambar 3.15] Tampilan halaman *E-learning*



[Gambar 3.16] Tampilan playlist video mata pelajaran

e. Perancangan *Login Admin* dan Guru

Halaman *login admin* dan guru adalah halaman yang disediakan untuk *admin* dan guru, halaman ini digunakan untuk masuk kedalam sistem. Tampilan halaman *login* disajikan dalam [Gambar 3.17]

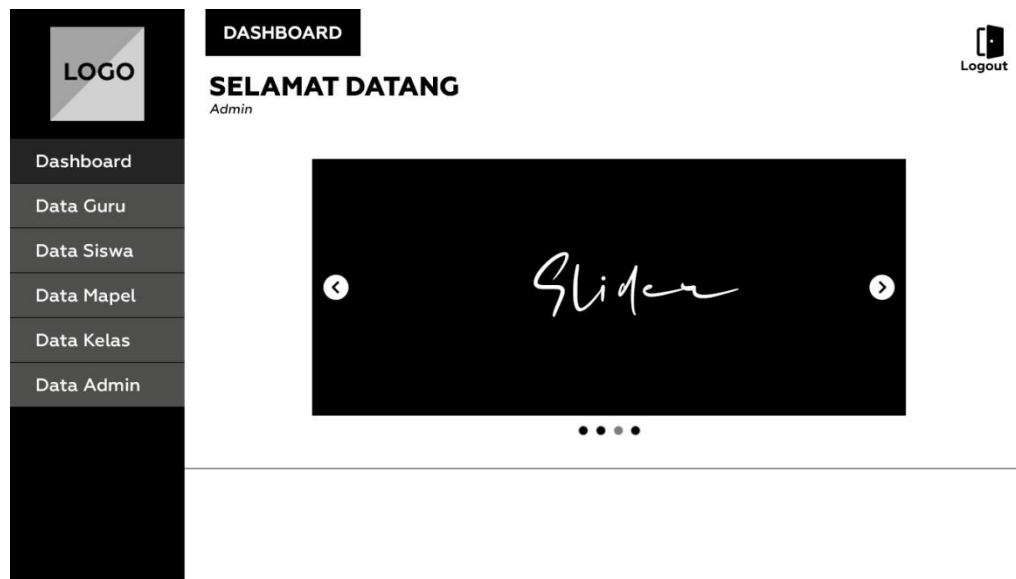


[Gambar 3.17] Tampilan login admin dan guru

f. Perancangan Halaman *Admin*

Halaman *admin* adalah halaman yang disediakan untuk *admin* dalam melakukan aktivitas seperti melakukan penginputan dan pngupdatean data guru, data siswa, data

mata pelajaran, data kelas, data *admin*. Tampilan halaman admin dapat dilihat pada [Gambar 3.16]



[Gambar 3.18] Tampilan halaman admin

#### g. Perancangan Halaman Guru

Halaman guru adalah halaman yang disediakan untuk guru dalam melakukan berbagai aktivitas, pada halaman guru terdapat beberapa menu yang biasa diakses seperti *profile*, tugas, ujian, rekap nilai, materi. Tampilan halaman guru dapat dilihat pada [Gambar 3.17]



[Gambar 3.19] Tampilan halaman guru

## DAFTAR PUSTAKA

Fadillah, umi. 2015. Rancang Bangun *Website* dan *E-Learning* di TPQ Al-Fadhillah [skripsi]. Surakarta: Universitas Muhammadiyah Surakarta

Ikhsanudin, Muhamad Surya. 2016. Belajar Santai OOP PHP. Jakarta: leanpub

Isbaniah, Fathiyah. 2020. *Pedoman Pencegahan Dan Pengendalian Coronavirus Disease (Covid-19)*. Jakarta Selatan: Kementrian Kesehatan RI Direktorat Jendral Pencegahan dan Pengendalian Penyakit (P2P)

Jejaring, 2019. Penjelasan Rinci *Use Case Diagram*. (diakses pada website jejaring.com pada 20 juni 2020)

Kristiyaningsih , Iswan . 2011. Pembuatan Aplikasi Pembelajaran *Oline(E-learning)* di MTs Negeri Pelupuh Berbasis Web [skripsi]. Surakarta: Universitas Muhammadiyah Surakarta

Muammar, Abdullah. 2018. Pengembangan *E-learning* Berbasis Web Di Jurusan Pendidikan Biologi UIN Raden Intan Lampung [skripsi]. Lampung: UIN Raden Intan Lampung

Nakayama M, Yamamoto H, dan S. R. 2007. *The Impact of Learner Characteristics on Learning Performance in Hybrid Courses among Japanese Students*. Elektronik Journal ELearning, Vol.5(3).1

Novitasari, Dian. 2010. Pembuatan *E-learning* Fisika SMA Kelas X Berbasis Web dengan Fasilitas Upload untuk Penambahan Mata Pelajaran [skripsi]. Surakarta: Universitas Muhammadiyah Surakarta

Putri, Marlina Kusuma. 2011 Implementasi *E-Learning* pada SMA Negeri 2 Surakarta Menggunakan PHP dan Mysql [skripsi thesis]. Surakarta: Universitas Muhammadiyah Surakarta.

Sumirat, Lambang Probo. 2017. UML (*Unified Modeling Language*). (diakses pada website blog.unitomo.ac.id pada tanggal 20 juni 2020)

Tommy. 2019. Pengertian Informasi Menurut Para Ahli. ( diakses pada website kotakpintar.com pada tanggal 18 juni 2020)