

IMPLEMENTASI ALGORITMA *FISHER YATES SHUFFLE*

DALAM PEMBUATAN UJIAN *ONLINE* BERBASIS WEB

(Studi Kasus : Fakultas Teknologi Informasi Universitas Bale Bandung)

Rustiyana, S.T., M.T., Nurul Imamah, S.T., M.T.¹, Muhamad Hanif

Ridwannulloh²

1. Dosen Teknik Informatika, Fakultas Teknologi Informasi, Universitas Bale Bandung
2. Mahasiswa Teknik Informatika, Fakultas Teknologi Informasi, Universitas Bale Bandung

ABSTRACT

Testing the learning process is needed to evaluate the learning process that has been implemented. Exams are usually carried out at the end of a certain period or at the end of the learning process and are also given by educators, which are usually of a manual nature, from the preparation of the exam to the examination process and recapitulation of exam results. But in the examination process there is often cheating among the examinees, one of which is cheating on each other cheating among the examinees, therefore exam questions are also usually made in packages so that the questions provided are also different.

In an application the use of algorithms is commonly used, one of them is Fisher Yates Shuffle algorithm in the use of online-based exams for the randomization process of exam questions. The fisher yates shuffle algorithm can produce a random permutation sequentially so that questions that have arisen will not appear again in the same session. This online exam application is a system to facilitate the process of evaluating learning outcomes, the main feature in the system is randomization of questions using the Fisher-Yates Shuffle algorithm so that the questions that appear on each exam will be different.

The conclusion of this whole thesis report is the online test application using the Fisher Yates Shuffle algorithm is a system designed to facilitate the process of evaluating learning outcomes, reducing the level of cheating among examinees by randomizing questions that are already available so that each test implementation every student gets an arrangement questions that differ from one another, making the exam run according to a predetermined schedule, and the results of the test results can be directly obtained in real time, also the application is made online or online so that it can be accessed anytime and anywhere.

Keywords: Online Examination, Algorithm, Fisher-Yates shuffle, Random

ABSTRAK

Pengujian proses pembelajaran sangat dibutuhkan untuk mengevaluasi proses pembelajaran yang sudah dilaksanakan. Ujian biasanya dilakukan diakhir periode tertentu atau diakhir proses pembelajaran juga diberikan oleh pendidik dengan yang biasanya bersifat manual, mulai dari pembuatan ujian tersebut hingga proses pemeriksaan dan rekap hasil ujian. Namun dalam proses ujian sering kali terjadi kecurangan antar peserta ujian, salah satunya adalah saling contek menyontek antara peserta ujian, oleh karena itu soal ujian juga biasanya dibuat perpaket agar soal yang disediakan juga berbeda beda.

Dalam sebuah aplikasi penggunaan algoritma sudah biasa digunakan, salah satunya Algoritma fisher-yates *shuffle* dalam penggunaan ujian berbasis *online* untuk proses pengacakan soal ujian. Algoritma fisher yates *shuffle* dapat menghasilkan suatu permutasi acak secara berurut sehingga pertanyaan yang telah muncul tidak akan muncul lagi di sesi yang sama. Aplikasi ujian *online* ini merupakan sebuah sistem untuk memudahkan proses evaluasi hasil pembelajaran, fitur utama yang ada didalam sistem tersebut adalah pengacakan soal menggunakan algoritma fisher-yates *shuffle* sehingga soal-soal yang muncul pada setiap pelaksanaan ujian akan berbeda satu sama lain.

Kesimpulan dari keseluruhan laporan skripsi ini adalah aplikasi ujian *online* menggunakan algoritma fisher-yates *shuffle* ini adalah sistem yang dirancang untuk mempermudah proses evaluasi hasil pembelajaran, mengurangi tingkatan kecurangan antar peserta ujian dengan cara mengacak soal-soal yang sudah tersedia sehingga setiap pelaksanaan ujian setiap mahasiswa mendapatkan susunan soal yang saling berbeda satu sama lain, membuat jalannya ujian sesuai dengan jadwal yang sudah ditentukan, dan hasil nilai dari pelaksanaan ujian dapat langsung didapatkan secara *real time*, juga aplikasi dibuat secara daring atau *online* sehingga dapat di-akses kapanpun dan dimanapun.

Kata Kunci :Ujian Online, Algoritma , Fisher-Yates *shuffle*, Acak

I. PENDAHULUAN

1.1 Latar Belakang

Setiap proses pembelajaran pasti mempunyai cara untuk mengevaluasi dan mengukur keberhasilan pembelajaran yang sudah dilakukan, biasanya proses evaluasi tersebut dilaksanakan diakhir periode tertentu atau diakhir sebuah materi pembelajaran dalam bentuk ujian baik secara manual

maupun secara *online*. Didalam ujian secara manual, untuk mengantisipasi terjadinya kecurangan antara peserta ujian biasanya menggunakan cara membuat paket soal sehingga soal yang muncul biasanya berbeda satu sama lain.

Pada setiap sistem atau aplikasi penggunaan algoritma sudah biasa digunakan. Dalam ujian *online* untuk melakukan tahapan seperti

pengacakan soal pasti membutuhkan algoritma salah satunya adalah penggunaan *algoritma fisher yates shuffle*, algoritma ini menghasilkan suatu permutasi acak secara berurut sehingga pertanyaan yang muncul tidak akan muncul lagi disesi yang sama.

Setelah melalui proses identifikasi di Fakultas Teknologi Informasi Universitas Bale Bandung penulis menemukan beberapa temuan permasalahan antara lain, proses ujian masih menggunakan metode manual, Sebagian besar soal ujian hanya dibuat dalam satu paket sehingga setiap mahasiswa mendapatkan soal ujian yang sama, penyampaian hasil ujian yang lambat (tidak dilaksanakan secara *real time*).

Sebelumnya telah dilakukan beberapa penelitian yang menerapkan konsep *Algoritma Fisher Yates Shuffle*, seperti jurnal yang berjudul “Implementasi Algoritma Fisher-Yates Dan Fuzzy Tsukamoto Dalam Game Kuis Tebak Nada Sunda Berbasis Android” oleh Haditama Imam, Slamet Cepy, dan Rahman Fauzy Deny (2016), dan juga “Aplikasi Alarm Weker Berbasis Android Dengan Algoritma Fisher

Yates *Shuffle* Untuk Mengacak Pertanyaan” oleh Banyumanis Ulumuddin Ihya, Fitriana Devi (2017).

Berdasarkan latar belakang yang telah dibahas diatas, penulis ingin mengangkat judul penelitian “Implementasi Algoritma Fisher Yates *Shuffle* Dalam Pembuatan Ujian *Online* Berbasis Web (Studi Kasus : Fakultas Teknologi Informasi Universitas Bale Bandung).”

1.2 Rumusan Masalah

Adapun rumusan masalah yakni sebagai berikut:

1. Bagaimana merancang dan membangun aplikasi ujian *online* berbasis web di FTI UNIBBA?
2. Bagaimana mengimplementasikan algoritma fisher yates *shuffle* pada aplikasi ujian *online* berbasis web di FTI UNIBBA?

1.3 Batasan Masalah

Adapun batasan masalah yakni sebagai berikut:

1. Aplikasi ini dibuat berbasis *Web*
2. Aplikasi ini hanya dapat menampilkan hasil evaluasi dalam bentuk nilai

3. Aplikasi ini melakukan pengacakan hanya pada susunan soal
4. Jenis ujian *online* dibatasi hanya dalam bentuk pilihan ganda

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah:

1. Terancang dan terbangunnya aplikasi ujian *online* berbasis web di FTI UNIBBA
2. Terimplementasikan-nya algoritma fisher yates *shuffle* pada aplikasi ujian *online* berbasis web di FTI UNIBBA.

1.5 Metodologi Penelitian

1.5.1 Metode Pengumpulan Data

Adapun teknik untuk pengumpulan data adalah sebagai berikut:

1. Observasi (Pengamatan)

Merupakan metode pengumpulan data dengan cara melaksanakan tinjauan secara langsung ke tempat penelitian atau objek yang diteliti dengan bertujuan mendapatkan data yang bersifat nyata dan meyakinkan. Oleh karena itu penulis melaksanakan

pengamatan secara langsung di Fakultas Teknologi Informasi Universitas Bale Bandung.

2. Interview (Wawancara)

Merupakan metode pengumpulan data dengan cara melaksanakan tanya jawab atau dialog secara langsung dengan pihak-pihak yang terkait dengan penelitian yang dilakukan. Dalam hal ini penulis melakukan tanya jawab kepada perwakilan Mahasiswa dan perwakilan akademik di Fakultas Teknologi Informasi Universitas Bale Bandung.

3. Studi Pustaka

Merupakan metode pengumpulan data yang bersifat teoritis. Penulis melakukan data tersebut dengan cara membaca, mengkaji dan mempelajari jurnal, buku, ataupun referensi lain yang masih berhubungan dengan masalah yang dibahas.

1.5.2 Metode Waterfall

Pada penelitian ini metode yang digunakan adalah metode Model Waterfall (*sequential linear*). Model

waterfall (*sequential linear*) atau sering juga disebut model alur hidup klasik (*classic life cycle*). Model waterfall menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut .

II. TINJAUAN PUSTAKA

2.1 Landasan Teori

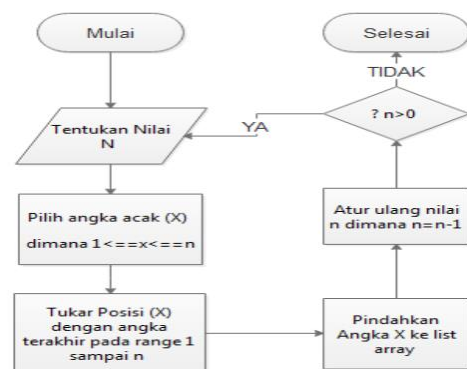
Dalam melakukan penelitian, penulis juga mempelajari penelitian terdahulu sebagai acuan adalah sebagai berikut:

1. Haditama Imam, Slamet Cepy, dan Rahman Fauzy Deny (2016) yang berjudul “Implementasi Algoritma Fisher-Yates dan Fuzzy Tsukamoto Dalam Game Kuis Tebak Nada Sunda Berbasis Android”.
2. Ekojono, Irawati Ayu Dyah, Affandi Lugman, dan Rahmanto Nur Anugrah (2017) yang berjudul “Penerapan Algoritma Fisher-Yates pada Pengacakan Soal Game Aritmatika”.
3. Banyumanis Ulumudin Ihya, Fitriana Devi (2017) yang berjudul “Aplikasi Alarm Weker Berbasis Android Dengan Algoritma Fisher Yates *Shuffle* Untuk Mengacak Pertanyaan”.

2.2 Dasar Teori

2.2.1 Algoritma Fisher-Yates Shuffle

Fisher-Yates *shuffle* (diambil dari nama Ronald Fisher dan Frank Yates) atau juga dikenal dengan nama Knuth *shuffle* (diambil dari nama Donald Knuth). Algoritma ini menghasilkan suatu permutasi acak secara berurut sehingga pertanyaan yang telah muncul tidak akan muncul lagi di sesi yang sama. (Haditama Imam, Slamet Cepy, Rahma Fauzy Deny, 2016:52)



Gambar 1: *Flowchart* algoritma fisher-yates *shuffle*

Metode dasar algoritma fisher-yates *shuffle* untuk mendapatkan suatu permutasi acak adalah sebagai berikut:

1. Tentukan jumlah array yang akan di acak (N),

2. Cari nilai acak (X) dimana X lebih dari sama dengan 0 dan X kurang dari N ($0 \leq X < N$),
3. Tukar isi dari array (X) dengan isi dari array (N-1),
4. Kurangi nilai (N) dengan angka 1
5. Jika nilai (N) masih lebih besar dari nol (0) maka kembali ke proses nomor 2 hingga nilai N tidak lebih besar dari nol (0).

2.2.2 Pemrograman Web

Pemrograman web (*web programming*) terdiri dari kata pemrograman dan web. Pemrograman sendiri dapat diartikan sebagai proses atau cara pembuatan program menggunakan bahasa pemrograman. Adapun bahasa pemrograman merupakan bahasa yang digunakan untuk memberikan instruksi kepada komputer sehingga komputer dapat memproses data dan menampilkan informasi sesuai yang dikehendakan oleh pemrogram. Dengan demikian pemrograman web dapat diartikan sebagai kegiatan pembuatan program atau aplikasi berbasis web menggunakan bahasa pemrograman tertentu sehingga dapat memproses data dan menghasilkan informasi sesuai yang dikehendaki

oleh pemilik *website*. (Abdulloh Rohi, 2018:2)

Pada pemrograman web bahasa yang sering digunakan biasanya : HTML (*Hyper Text Markup Language*), CSS (*Cascading Style Sheets*), PHP (*Hypertext Preprocessor*), SQL, dan JavaScript.

2.2.3 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah di oleh atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. (Salahudin M, S A Rosa, 2018:43)

2.2.4 Web Server

Web server adalah suatu program komputer yang mempunyai tanggung jawab atau tugas menerima permintaan HTTP dari *computer client*, yang dikenal dengan nama web browser dan melayani mereka dengan menyediakan respon HTTP berupa konten data, biasanya berupa halaman web yang terdiri dari dokumen HTML dan objek terkait seperti gambar dan lain-lain.

Ada beberapa jenis *software* untuk membangun web server lokal atau *localhost* yang suport sistem windows diantaranya adalah Wampserver, Appserv, XAMPP, PHP Triad atau Vertigo.(Madcoms, 2016)

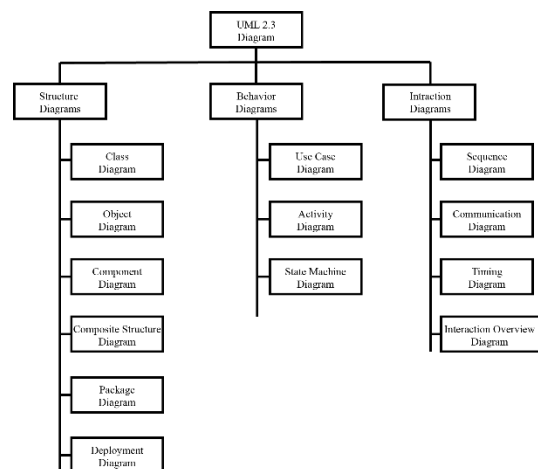
2.2.5 UML (*Unified Modelling Language*)

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *data flow diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural, kemudian juga ada *state transition diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya

kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

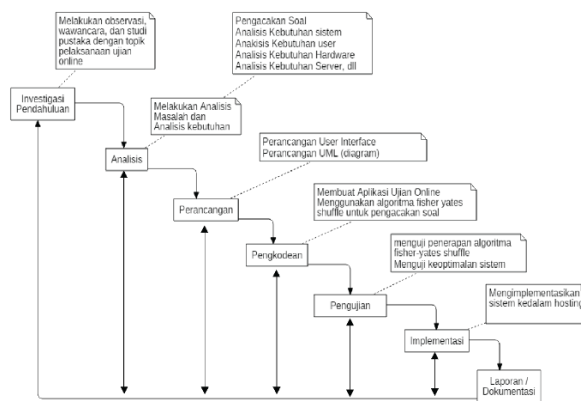
Dalam UML ada beberapa jenis diagram, pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam tiga kategori titik pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar di bawah.(Salahudin M, S A Rosa, 2018:140)



Gambar 2: Diagram UML

III. METODOLOGI PENELITIAN

Metodologi penelitian ini berisi langkah-langkah dalam penelitian ini agar terstruktur dengan baik. Dengan sistematika ini dapat dipahami dan diikuti oleh pihak lain. Penelitian untuk merancang sistem diperoleh dari pengamatan data-data yang ada. Adapun langkah - langkah yang dilakukan untuk mencapai tujuan dari penelitian ini, seperti gambar dibawah ini:



Gambar 3: Kerangka pemikiran

3.1 Investigasi Pendahuluan

Pada tahapan ini dilakukan dengan cara observasi langsung di Fakultas Teknologi Informasi Universitas Bale Bandung juga wawancara kepada beberapa staf akademik di Fakultas Teknologi Informasi Universitas Bale Bandung.

Pada bagian ini dilakukan investigasi terhadap proses ujian yang dilakukan

3.2 Analisis

Pada tahapan analisis dalam penelitian ini merupakan lanjutan dari tahapan investigasi pendahuluan, yaitu menganalisis hasil dari observasi, wawancara, dan studi pustaka yang sudah dilakukan pada tahapan investigasi pendahuluan. Analisis yang dilakukan antara lain analisis masalah, analisis instrument penelitian, analisis perangkat keras, analisis pengguna, dan analisis kebutuhan sistem. Dalam penelitian ini narasumber yang diwawancarai adalah beberapa staf akademik di Fakultas Teknologi Informasi (FTI) Universitas Bale Bandung (UNIBBA) antara lain : Dekan Fakultas Teknologi Informasi, Kepala Prodi Informatika (IF), Kepala Prodi Sistem Informasi (SI).

3.3 Perancangan

Setelah melakukan analisis maka tahapan berikutnya adalah melakukan perancangan aplikasi. Pada tahapan perancangan aplikasi ini penulis membuat perancangan antara lain: Perancangan UML (*usecase*

diagram, class diagram, activity diagram, dan sequence diagram).

3.4 Coding / Pengkodean

Setelah tahapan perancangan maka tahapan selanjutnya adalah tahapan pembuatan aplikasi. Pada tahapan ini hasil dari perancangan yang dibuat diubah menjadi sebuah aplikasi.

3.5 Pengujian

Pada tahapan pengujian penelitian ini dilakukan dengan cara “black box”. Sehingga dapat memastikan bahwa penerapan algoritma fisher-yates *shuffle* tersebut berhasil diimplementasikan atau tidak pada aplikasi ujian *online* ini untuk pengacakan pada soal ujian.

3.6 Implementasi

Setelah pengacakan soal ujian menggunakan algoritma fisher-yates *shuffle* pada aplikasi ujian *online* berhasil dilakukan. Pada tahapan ini dilakukan penerapan aplikasi yang telah dibuat kedalam server hosting.

3.7 Laporan / Dokumentasi

Pada tahapan ini semua tahapan-tahapan penelitian yang telah dilakukan dituliskan dalam sebuah laporan penelitian seperti tahapan

investigasi pendahuluan, analisis masalah dan analisis kebutuhan, perancangan, pengkodean, pengujian, implementasi dan dokumentasi. Sehingga penelitian ini dapat menjadi bahan referensi untuk penelitian-penelitian selanjutnya.

IV. ANALISIS DAN PERANCANGAN

4.1 Analisis

4.1.1 Analisis Masalah

Berdasarkan hasil dari observasi, wawancara, dan studi pustaka yang telah dilakukan terhadap sistem ujian di FTI UNIBBA maka permasalahan yang diperoleh adalah sebagai berikut:

- a. Pada sistem ujian di FTI UNIBBA masih menggunakan cara yang manual, yaitu ujian dengan menggunakan kertas.
- b. Soal ujian yang disediakan biasanya hanya satu paket, sehingga setiap mahasiswa mendapatkan soal ujian yang sama.
- c. Penyampaian hasil ujian di FTI UNIBBA tidak dilakukan secara *real-time*.

4.1.2 Analisis Kebutuhan

Instrumen – instrumen yang dibutuhkan pada penelitian ini terdiri dari perangkat keras, perangkat lunak, kebutuhan internet, dan hasil dari proses pengumpulan data. Berikut penjabaran dari instrumen – instrumen yang dibutuhkan:

1. Kebutuhan Perangkat Keras

Penulis menggunakan instrumen perangkat keras untuk dijadikan penunjang dalam penelitian ini, antara lain:

Tipe Laptop	Lenovo V110
Prosesor	AMD A9-9420 RADEON R5
RAM	4GB
Hard disk	500GB

Tabel 1: Instrumen Kebutuhan Perangkat Keras Laptop

2. Kebutuhan Perangkat Lunak

Penulis menggunakan beberapa perangkat lunak untuk dijadikan penunjang dalam penelitian ini terutama dalam perancangan juga pembuatan aplikasi, perangkat lunak tersebut antara lain: Sublime Text, XAMPP, Web Browser, Star UML, Mockplus.

3. Kebutuhan Pengguna

Dalam aplikasi yang dibuat dalam penelitian ini penulis membutuhkan 3 (tiga) tipe user yang dapat mengakses aplikasi tersebut, antara lain: User Admin, User Mahasiswa, dan User Dosen.

4. Kebutuhan Server

Dalam penelitian ini penulis menggunakan server hosting dengan spesifikasi sebagai berikut:

Spesifikasi	Deskripsi
Memory	2GB
RAM	512MB
ServerAPI	LiteSpeed V7.6
OS	Linux
PHP Version	PHP v.7.2.31

Tabel 2: Server Hosting

5. Kebutuhan Aplikasi

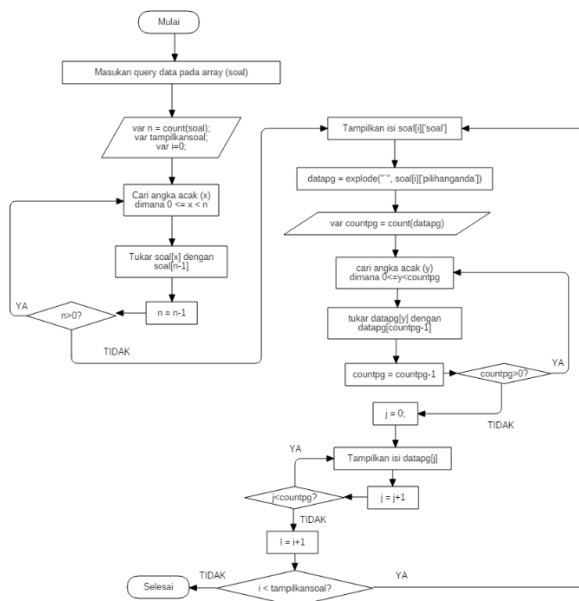
Penulis menggunakan proses pengumpulan data untuk dapat menganalisis kebutuhan dari aplikasi yang akan dibuat ini, salah satunya adalah wawancara. Penulis melaksanakan wawancara penelitian terhadap 4 (empat) narasumber yang bertugas di bagian akademik di Fakultas Teknologi Informasi Universitas Bale Bandung secara *offline* juga *online*.

Dari hasil wawancara bersama ke 4 (empat) narasumber tersebut, didapatkan hasil analisis kebutuhan aplikasi antara lain:

- 1) Semua user harus melakukan *login* untuk dapat mengakses aplikasi,
- 2) Semua soal ujian menggunakan model pilihan ganda dengan hanya 1 jawaban,
- 3) Pelaksanaan ujian dilaksanakan sesuai dengan jadwal yang telah ditetapkan,
- 4) Dilakukannya pengacakan pada susunan soal ujian,
- 5) Dilakukannya pengacakan pada pilihan ganda setiap soal,
- 6) Hasil ujian dapat diketahui secara *real-time*,
- 7) Adanya proses perbaikan nilai untuk mahasiswa yang mendapatkan hasil kurang memuaskan,
- 8) Mahasiswa hanya dapat mengikuti dan melihat hasil ujiannya sendiri,
- 9) Dosen dapat mencetak hasil ujian mahasiswa yang mengikuti ujian matakuliahnya,
- 10) Admin dapat membuat dan menghapus user mahasiswa,
- 11) Admin dapat mengedit sebagian data dari user mahasiswa,
- 12) Admin dapat membuat dan menghapus user dosen,
- 13) Admin dapat mengedit Sebagian data dari user dosen,
- 14) Admin dapat melihat hasil ujian,
- 15) Admin dapat mencetak transkrip – transkrip hasil ujian,
- 16) User dosen dapat menambah, mengubah, dan menghapus soal ujian (apabila status ujian masih belum terkunci),
- 17) Apabila status ujian sudah “dikunci” maka soal ujian tidak dapat diubah maupun dihapus,
- 18) User dosen diwajibkan mengisi ketentuan / petunjuk pengerjaan pada ujian *online*,
- 19) User dosen dan user mahasiswa dapat mengedit Sebagian *profile* pribadinya masing – masing,
- 20) User dosen diberikan hak untuk membuat ujian dalam konteks *quiz* atau ujian harian,
- 21) Disediakan kode untuk dapat mengikuti *quiz* atau ujian harian,
- 22) Hasil ujian tidak dapat diubah,

- 23) Ujian akan berlangsung sesuai dengan tanggal dan waktu yang sudah ditentukan,
- 24) Hasil ujian ditampilkan dalam bentuk angka.

Agar sistem dapat melakukan pengacakan pada setiap susunan soal ujian juga pada pilihan ganda setiap soal, penulis membuat *flowchart* pengimplementasian algoritma fisher-yates *shuffle*. Berikut adalah *flowchart* pengacakan pada soal ujian online menggunakan algoritma fisher-yates *shuffle*:



Gambar 4: *Flowchart* algoritma fisher-yates dalam ujian online

Flowchart diatas akan dijelaskan melalui contoh kasus

berikut, diketahui array soal adalah sebagai berikut:

Array

```
(
  [0] => Array
    (
      [soal] => 'soal0'
      [jawaban] => 'Jawabannya0'
      [PilihanGanda] =>
        '0`Iya0`Jawabannya0`Masa0`Oh0'
    )

  [1] => Array
    (
      [soal] => 'soal1'
      [jawaban] => 'Jawabannya1'
      [PilihanGanda] =>
        '1`Iya1`Jawabannya1`Masa1`Oh1'
    )

  [2] => Array
    (
      [soal] => 'soal2'
      [jawaban] => 'Jawabannya2'
      [PilihanGanda] =>
        '2`Iya2`Jawabannya2`Masa2`Oh2'
    )

  [3] => Array
    (
      [soal] => 'soal3'
      [jawaban] => 'Jawabannya3'
      [PilihanGanda] =>
        '3`Iya3`Jawabannya3`Masa3`Oh3'
    )

  [4] => Array
    (
      [soal] => 'soal4'
      [jawaban] => 'Jawabannya4'
      [PilihanGanda] =>
        '4`Iya4`Jawabannya4`Masa4`Oh4'
    )
);
```

maka proses pengacakannya sebagai berikut :

- a. soal = {[0],[1],[2],[3],[4]}
- b. Tentukan nilai n, n =
count(soal), n = 5
- c. Cari angka acak (x) dimana $0 > x > n$, Misalkan x = 3,
- d. Tukar isi soal[x] dengan isi
soal[n-1]

Maka array soal menjadi =

```
Array
(
    [0] => Array
        (
            [soal] => 'soal0'
            [jawaban] =>
                'Jawabannya0'
            [PilihanGanda] =>
                '0`Iya0`Jawabannya0`Masa0`Oh0'
        )

```

```
[1] => Array
(
    [soal] => 'soal1'
    [jawaban] =>
        'Jawabannya1'
    [PilihanGanda] =>
        '1`Iya1`Jawabannya1`Masa1`Oh1'
    ,
)

```

```
[2] => Array
(
    [soal] => 'soal2'
    [jawaban] =>
        'Jawabannya2'
    [PilihanGanda] =>
        '2`Iya2`Jawabannya2`Masa2`Oh2'
    ,
)

```

```
[3] => Array
(
    [soal] => 'soal4'
```

```
[jawaban] =>
    'Jawabannya4'
    [PilihanGanda] =>
        '4`Iya4`Jawabannya4`Masa4`Oh4'
    ,
)

```

```
[4] => Array
(
    [soal] => 'soal3'
    [jawaban] =>
        'Jawabannya3'
    [PilihanGanda] =>
        '3`Iya3`Jawabannya3`Masa3`Oh3'
    ,
)
);

```

e. n = n-1, n = 5-1, n = 4

f. Jika n masih memenuhi syarat
dari (n>0) maka kembali
lakukan tahapan (c).

```
n = 4,
misal x = 0,
Maka array soal menjadi =

Array
(
    [0] => Array
        (
            [soal] => 'soal4'
            [jawaban] =>
                'Jawabannya4'
            [PilihanGanda] =>
                '4`Iya4`Jawabannya4`Masa4`Oh4'
        )

```

```
[1] => Array
(
    [soal] => 'soal1'
    [jawaban] =>
        'Jawabannya1'
```

```

        [PilihanGanda] =>
        '1`Iya1`Jawabannya1`Masa1`Oh
        1'
    )

    [2] => Array
    (
        [soal] => 'soal2'
        [jawaban] =>
        'Jawabannya2'
        [PilihanGanda] =>
        '2`Iya2`Jawabannya2`Masa2`Oh
        2'
    )

    [3] => Array
    (
        [soal] => 'soal0'
        [jawaban] =>
        'Jawabannya0'
        [PilihanGanda] =>
        '0`Iya0`Jawabannya0`Masa0`Oh
        0'
    )

    [4] => Array
    (
        [soal] => 'soal3'
        [jawaban] =>
        'Jawabannya3'
        [PilihanGanda] =>
        '3`Iya3`Jawabannya3`Masa3`Oh
        3'
    )
);
n = n-1, n=3,
n>0? = YA

```

- g. Lakukan kembali tahapan seperti diatas sampai nilai n tidak lebih besar dari 0 ($n \leq 0$). Misalkan hasil akhir dari pengacakan adalah sebagai berikut:

```

Array
(

```

```

[0] => Array
(
    [soal] => 'soal2'
    [jawaban] =>
    'Jawabannya2'
    [PilihanGanda] =>
    '2`Iya2`Jawabannya2`Masa2`Oh
    2'
)

[1] => Array
(
    [soal] => 'soal1'
    [jawaban] =>
    'Jawabannya1'
    [PilihanGanda] =>
    '1`Iya1`Jawabannya1`Masa1`Oh
    1'
)

[2] => Array
(
    [soal] => 'soal4'
    [jawaban] =>
    'Jawabannya4'
    [PilihanGanda] =>
    '4`Iya4`Jawabannya4`Masa4`Oh
    4'
)

[3] => Array
(
    [soal] => 'soal0'
    [jawaban] =>
    'Jawabannya0'
    [PilihanGanda] =>
    '0`Iya0`Jawabannya0`Masa0`Oh
    0'
)

[4] => Array
(
    [soal] => 'soal3'
    [jawaban] =>
    'Jawabannya3'
    [PilihanGanda] =>
    '3`Iya3`Jawabannya3`Masa3`Oh
    3'
)

```

```

    )
);

```

- h. Setelah selesai melakukan pengacakan pada susunan soal maka dilakukan perulangan untuk menampilkan setiap soal menggunakan metode for;

```

Misalkan : $stampilkansoal = 3;
for ($i=0; $i < $stampilkansoal;
    $i++) {

```

- i. Tampilkan pada array soal[kolom ke-i][key = soal].

```

Misalkan soal[0]['soal'];
echo $soal[$i]['soal'];
maka array soal yang
ditampilkan adalah =
[0] => Array
(
    [soal] => 'soal2'
    [jawaban] =>
    'Jawabannya2'
    [PilihanGanda] =>
    '2`Iya2`Jawabannya2`Masa2`
    Oh2'
)

```

- j. Untuk dapat melakukan pengacakan pada pilihan ganda, dalam perulangan diatas lakukan pemecahan string pilihan ganda pada setiap soal kedalam bentuk array menggunakan *library* explode(), misal \$i=0.

```

$datapg = Explode("",
$soal[$i][PilihanGanda])

```

Maka array variabel datapg =
 ['2', 'Iya2', 'Jawabannya2',
 'Masa2', 'Oh2'];

- k. Tentukan nilai variabel countpg (jumlah array) yang ada pada variabel datapg.

```

$countpg = count(datapg)

```

- l. Cari angka acak (y) dimana 0 >= y > countpg, Misalkan y = 2,

- m. Tukarkan datapg[y] dengan datapg[countpg-1]

- n. countpg = countpg-1, countpg = 5-1, countpg = 4

- o. jika countpg masih memenuhi syarat (countpg>0) maka lakukan kembali tahapan (l), sampai nilai countpg sudah tidak memenuhi syarat (countpg>0).

- p. Setelah selesai melakukan pengacakan pada susunan pilihan ganda maka dilakukan perulangan untuk menampilkan setiap pilihan ganda menggunakan metode for atau foreach;

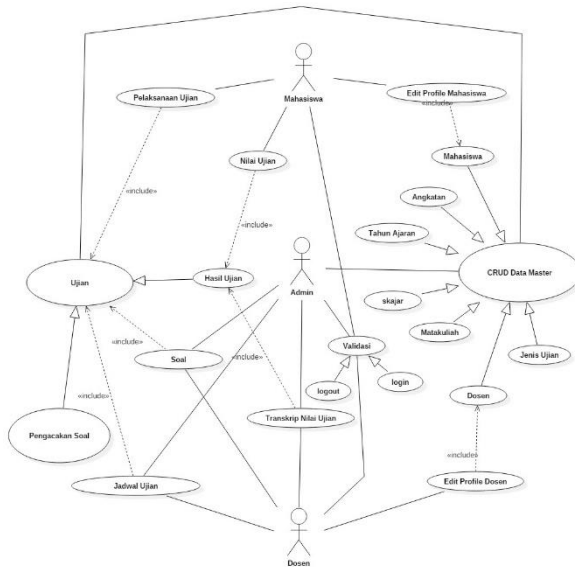
4.2 Perancangan

Dalam penelitian ini penulis membuat perancangan antara lain: Perancangan *Usecase diagram* ujian

online, Class diagram ujian online,
Activity diagram ujian online,
Sequence diagram ujian online.

4.2.1 Diagram UML

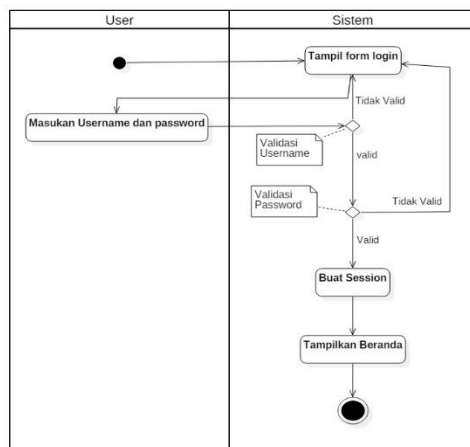
1. Usecase Diagram Ujian Online



Gambar 5: Usecase diagram ujian online

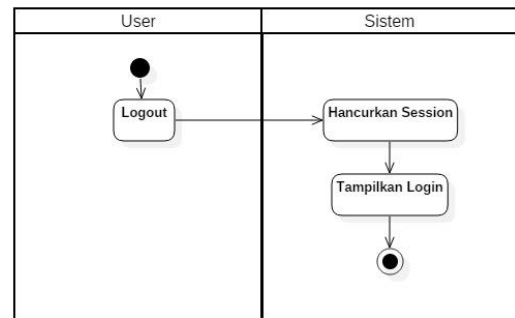
2. Activity Diagram

Activity Diagram Login



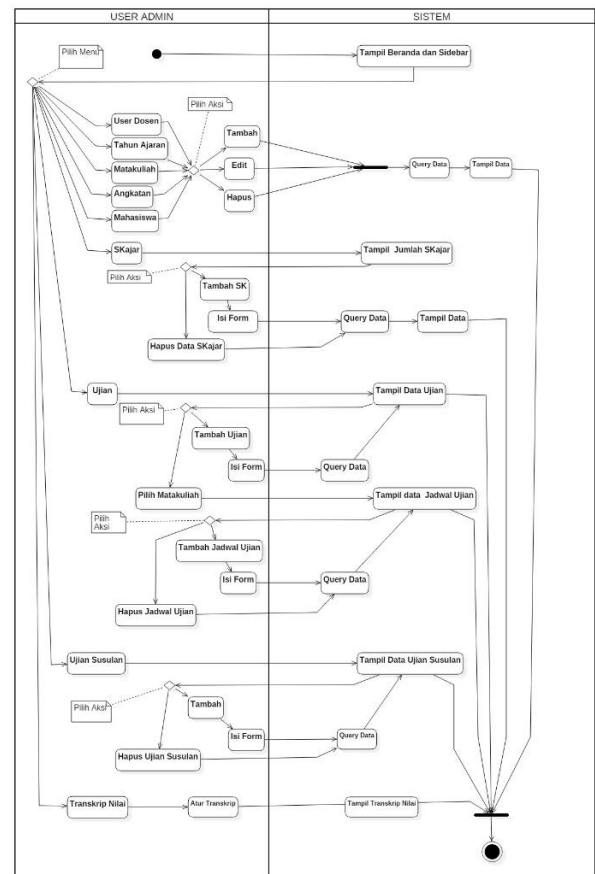
Gambar 6: Activity Diagram Login

Activity Diagram Logout



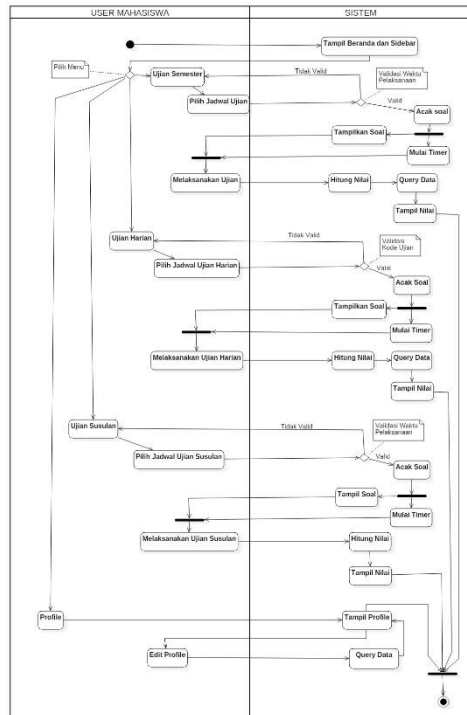
Gambar 7: Activity diagram logout

Activity Diagram User Admin



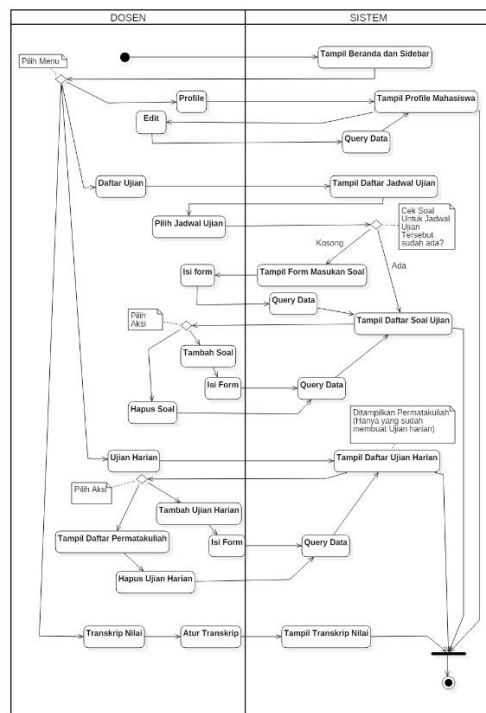
Gambar 8: Activity diagram user admin

Activity Diagram User Mahasiswa



Gambar 9: Activity diagram user mahasiswa

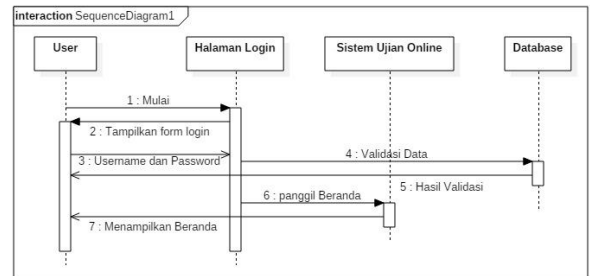
Activity Diagram User Dosen



Gambar 10: Activity diagram user dosen

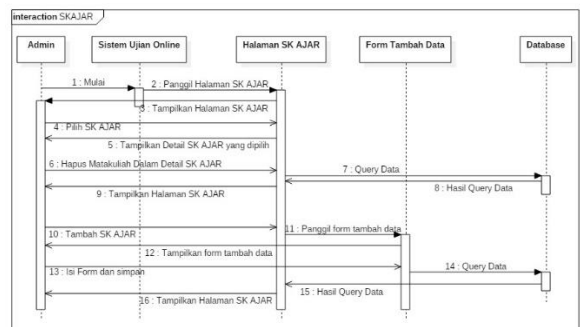
3. Sequence Diagram

Sequence Diagram Login



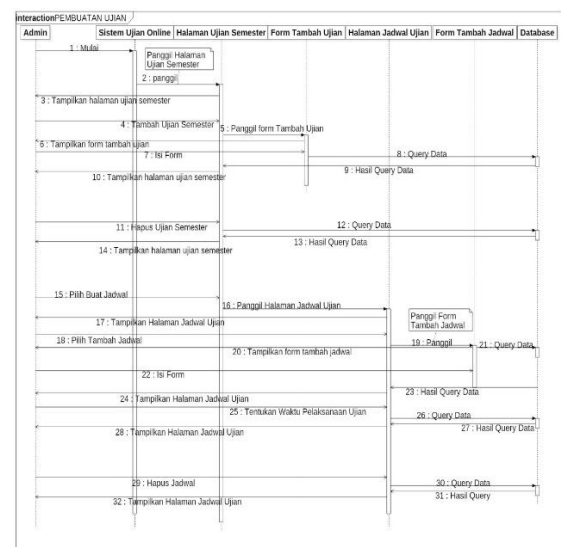
Gambar 11: Sequence diagram login

Sequence Diagram Skajar

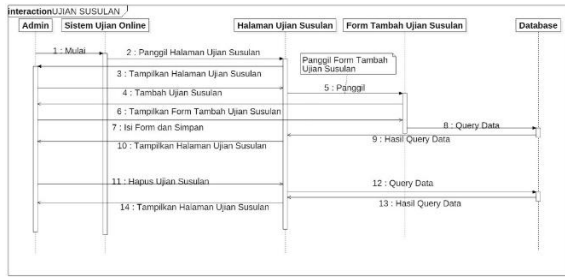


Gambar 12: Sequence diagram skajar

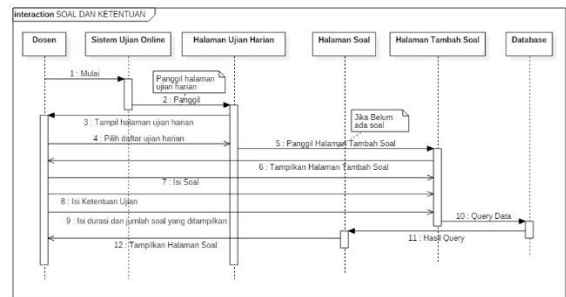
Sequence diagram Pembuatan Ujian



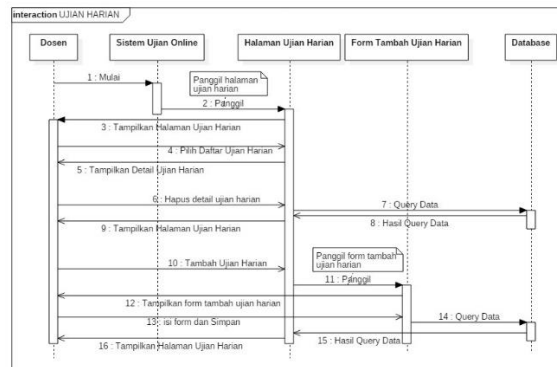
Gambar 13: Sequence diagram pembuatan ujian semester



Gambar 14: *Sequence diagram* pembuatan ujian susulan

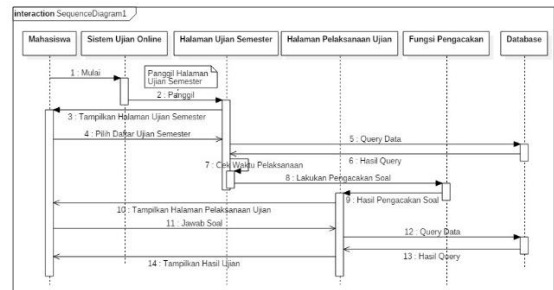


Gambar 17: *Sequence diagram* Pembuatan soal ujian harian



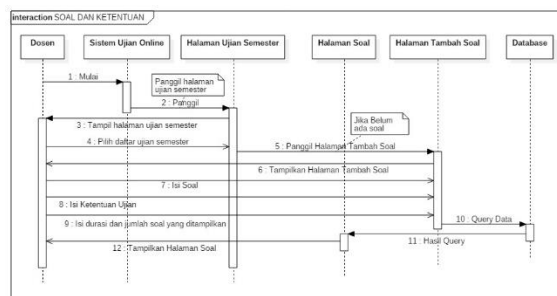
Gambar 15: *Sequence diagram* pembuatan ujian harian

Sequence Diagram Pelaksanaan Ujian

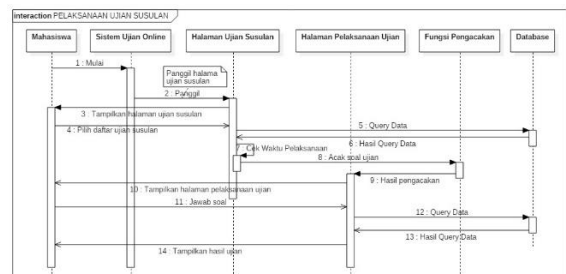


Gambar 18: *Sequence diagram* pelaksanaan ujian semester

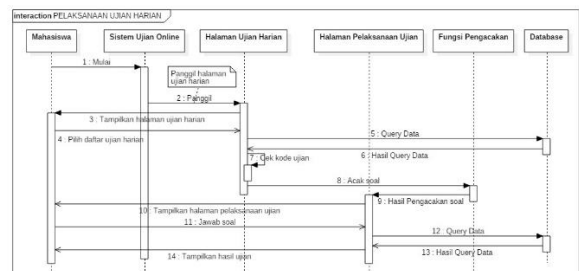
Sequence Diagram Pembuatan Soal Ujian



Gambar 16: *Sequence diagram* pembuatan soal ujian semester

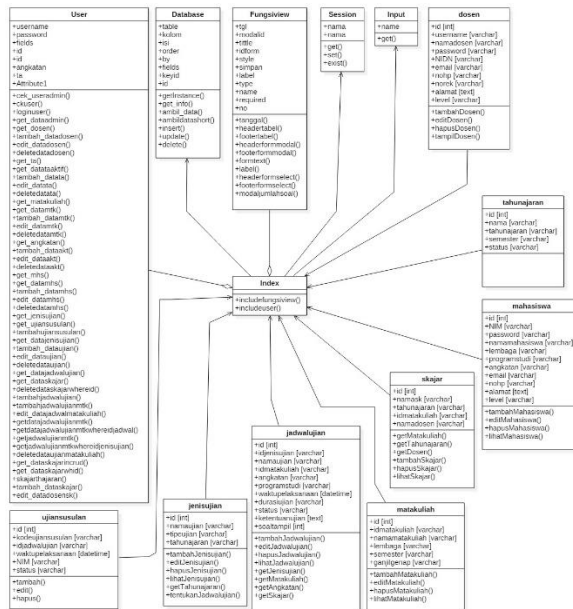


Gambar 19: *Sequence diagram* pelaksanaan ujian susulan



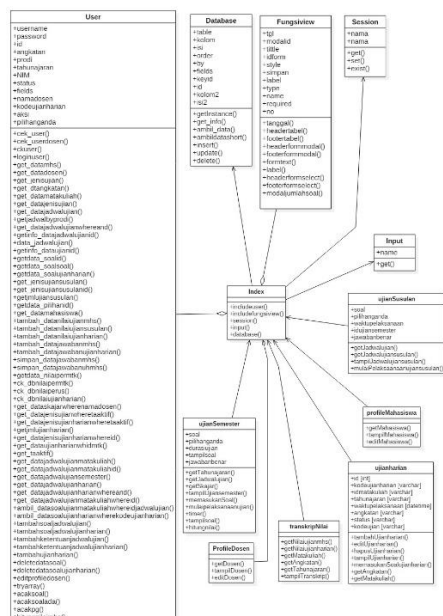
Gambar 20: *Sequence diagram* pelaksanaan ujian harian

Class Diagram Pengelolaan Data Master



Gambar 21: *Class diagram* pengelolaan data master

Class Diagram Pengelolaan Ujian



Gambar 22: *Class diagram*
pengelolaan ujian

Pada *Class diagram* diatas menunjukkan bahwa *class* user dan *class* fungsiview memiliki Sebagian besar *method* dan berada didalam *class index*, juga *index* yang meminta *class-class* lain untuk menjalankan *method – method* nya. Sehingga semua *method* berpusat pada *index*.

V. IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Implementasi merupakan tahapan pembuatan aplikasi setelah tahapan analisis dan perancangan selesai dilalui. Hasil dari analisis dan perancangan tersebut diimplementasikan kedalam bahasa pemrograman.

5.1.1 Implementasi Algoritma Fisher-Yates Shuffle

Pengimplementasian

algoritma fisher-yates *shuffle* pada sistem ujian *online* ini dilakukan pada bagian pengacakan soal ujian beserta dengan pilihan ganda setiap soal. Untuk mengimplementasikan algoritma fisher-yates *shuffle* pada sistem, penulis menggunakan bahasa pemrograman PHP. Berikut adalah pengkodean implementasi algoritma

fisher-yates *shuffle* pada sistem ujian *online*:

```
public function acaksoal($aksi,$idjadwalujian){
    if ($aksi == 'ujianharian') {
        // jika variabel aksi = 'ujianharian' maka lakukan query pada tabel soalujianharian
        // untuk mengambil soal ujian harian yang sudah dimasukan untuk ujian yang akan dilakukan
        $datasoal = $this->koneksi->ambil_data('soalujianharian', 'kodeujianharian', $idjadwalujian);
    } else {
        // jika variabel aksi = 'ujiansemester' maka lakukan query pada tabel soal
        // untuk mengambil soal ujian yang sudah dimasukan untuk ujian yang akan dilakukan
        $datasoal = $this->koneksi->ambil_data('soal', 'idjadwalujian', $idjadwalujian);
    }
    // hitung jumlah data yang ada pada variabel datasoal dan inisialisasikan ke variabel count
    $count = count($datasoal);
    // jika variabel count lebih besar dari 0
    if ($count > 0) {
        while ($count > 0) {
            $dataacak = $count - 1;
            // dapatkan hasil acak angka dimana hasil acak harus diantara 0 dan nilai variabel dataacak
            $hasilacak = rand(0,$dataacak);
            // setelah mendapatkan angka pengacakan // lakukan pertukaran antara
            $penyimpanansementara = $datasoal[$hasilacak]; // isi array datasoal[hasilacak] dengan
            $datasoal[$hasilacak] = $datasoal[$dataacak]; // isi array datasoal[dataacak]
            $datasoal[$dataacak] = $penyimpanansementara; // menggunakan bantuan variabel baru
            $count--;
        }
        // variabel datasoal setelah dilakukan pengacakan
        return $datasoal;
    } else {
        return $datasoal;
    }
}
```

Gambar 23: Implementasi pengkodean algoritma fisher-yates *shuffle* pada pengacakan susunan soal

Gambar diatas adalah fungsi acak soal dimana hasil query data yang sudah didapatkan dimasukan kedalam variabel datasoal (array), setelah data ditampungkan maka sistem menghitung jumlah data yang ada pada variabel datasoal. Apabila jumlah data pada variabel datasoal lebih besar dari nol (0) maka nilai jumlah datasoal dikurangi satu (1) dan ditampungkan kedalam variabel dataacak.

Setelah proses pengurangan selesai maka dapatkan nilai untuk variabel hasilacak dimana nilai tersebut harus diantara atau sama dengan 0 dan nilai dari variabel dataacak menggunakan fungsi rand. Setelah didapatkan nilai variabel hasilacak maka lakukan pertukaran

antara isi array datasoal[hasilacak] dengan isi array datasoal[dataacak] menggunakan bantuan variabel baru, setelah pertukaran isi array selesai maka nilai jumlah variabel datasoal kembali dikurangi satu (1). Apabila setelah proses pengurangan nilai variabel datasoal masih lebih besar daripada 0 maka Kembali ulangi proses pengacakan sampai nilai variabel datasoal tersebut menjadi nol (0) dan didapatkan data terbaru hasil pengacakan pada variabel datasoal.

```
public function acakpg($pilihganda){
    // split string berdasarkan simbol
    $datapg = explode(" ", $pilihganda);
    // hitung jumlah array var datapg
    $countpg = count($datapg);
    $acakpg = $countpg - 1;
    // jika jumlah array var datapg lebih besar dari 0
    while ($countpg > 0) {
        // dapatkan nilai var hasilacakpg dimana harus diantara atau samadengan 0 dan nilai var acakpg
        $hasilacakpg = rand(0,$acakpg);
        // Setelah mendapatkan angka pengacakan // lakukan pertukaran
        $simpansementara = $datapg[$hasilacakpg]; // antara isi array datapg[hasilacakpg]
        $datapg[$hasilacakpg] = $datapg[$acakpg]; // dengan isi array datapg[acakpg]
        $datapg[$acakpg] = $simpansementara; // menggunakan bantuan variabel baru
        $countpg--;
    }
    return $datapg;
}
```

Gambar 24: Implementasi pengkodean algoritma fisher-yates *shuffle* pada pengacakan pilihan ganda

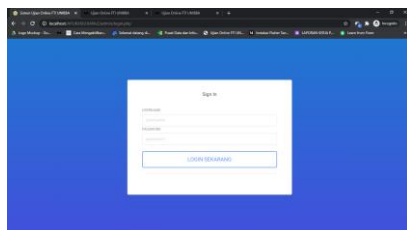
Gambar diatas adalah fungsi acakpg, dimana dikarenakan isi dari variabel pilihganda setiap soal dibuat dalam bentuk string dengan pemisah simbol “ ” maka dilakukan pemisahan pada string variabel pilihganda dengan pemisah simbol “ ” dan ditampungkan kedalam array datapg. Setelah string selesai dipisahkan hitung jumlah data pada array datapg, setelah didapatkan lakukan pengurangan jumlah array

datapg dengan satu (1) dan tampungkan hasil pengurangan kedalam variabel acakpg.

Setelah proses pengurangan selesai maka dapatkan nilai untuk variabel hasilacakpg dimana nilai tersebut harus diantara atau sama dengan 0 dan nilai dari variabel acakpg menggunakan fungsi rand. Setelah didapatkan nilai variabel hasilacakpg maka lakukan pertukaran antara isi array datapg[hasilacakpg] dengan isi array datapg[acakpg] menggunakan bantuan variabel baru, setelah pertukaran isi array selesai maka nilai jumlah variabel datapg kembali dikurangi satu (1). Apabila setelah proses pengurangan nilai variabel datapg masih lebih besar daripada 0 maka Kembali ulangi proses pengacakan sampai nilai variabel datapg tersebut menjadi nol (0) dan didapatkan data terbaru hasil pengacakan pada variabel datapg.

5.2 Tampilan Sistem

1. Halaman *Login*



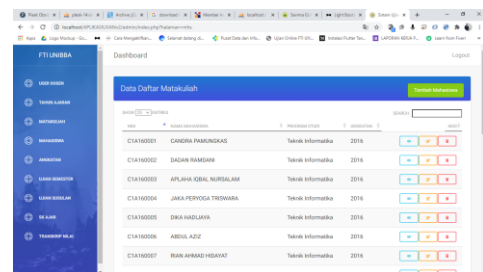
Gambar 25: Tampilan Halaman *Login*

2. Halaman Utama



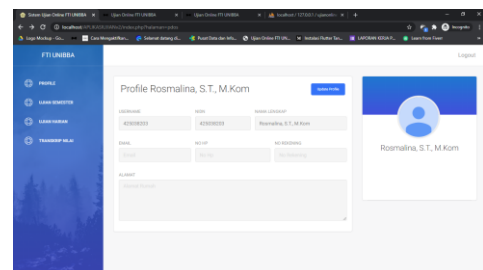
Gambar 26: Tampilan halaman utama

3. Halaman Pengelolaan Data Master



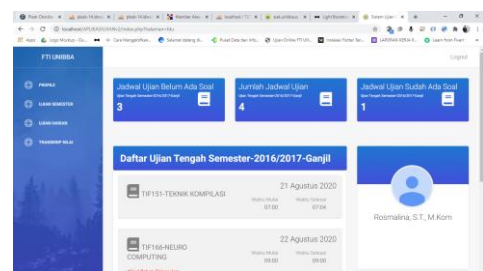
Gambar 27: Tampilan halaman pengelolaan data master

4. Halaman Profil



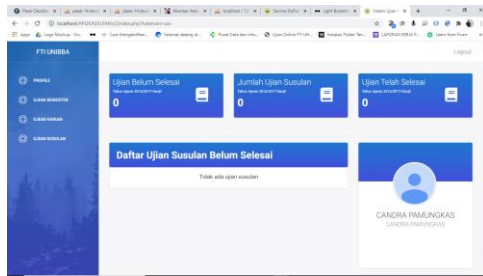
Gambar 28: Tampilan Halaman Profil

5. Halaman Ujian Semester



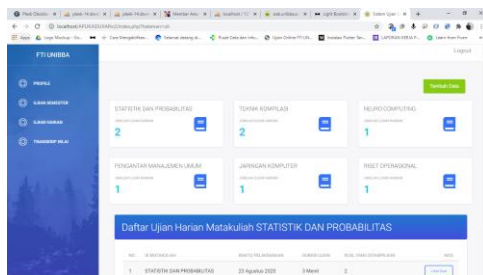
Gambar 29: Tampilan halaman ujian semester

6. Halaman Ujian Susulan

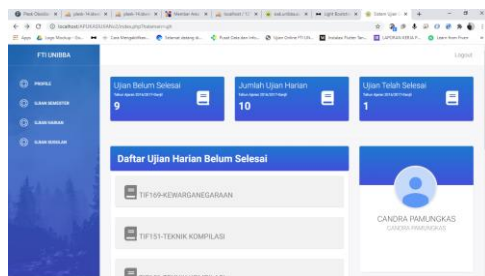


Gambar 30: Tampilan halaman ujian susulan

7. Halaman Ujian Harian

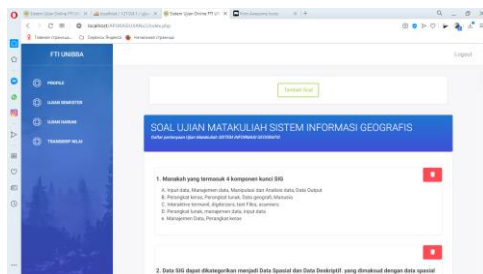


Gambar 31: Tampilan halaman ujian harian dosen

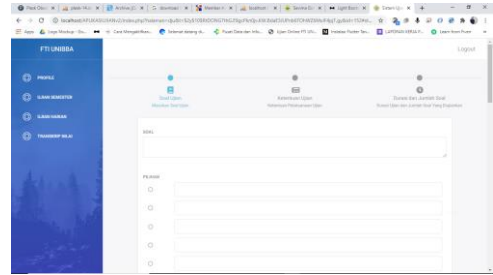


Gambar 32: Tampilan halaman ujian harian mahasiswa

8. Halaman Soal

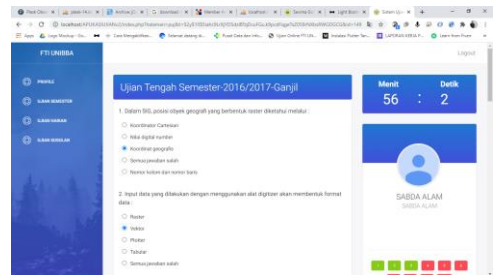


Gambar 33: Tampilan halaman soal

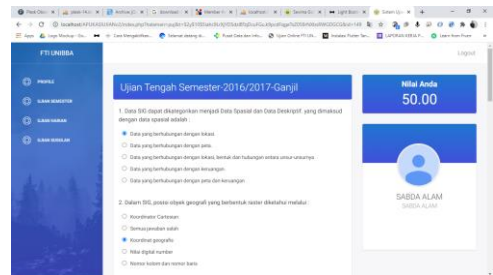


Gambar 34: Tampilan halaman tambah soal

9. Halaman Pelaksanaan Ujian



Gambar 35: Tampilan halaman pelaksanaan ujian



Gambar 36: Tampilan halaman selesai pelaksanaan ujian

5.3 Pengujian

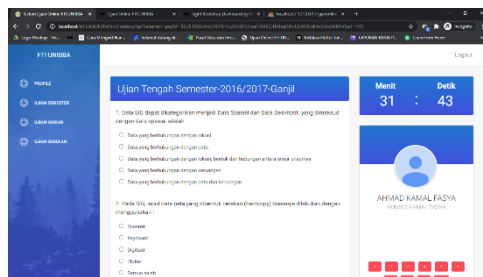
5.2.1 Pengujian Algoritma Fisher-Yates Shuffle

Proses pengujian implementasi algoritma fisher-yates shuffle pada pengacakan ujian online dilakukan dengan batasan pengujian pada satu ujian dengan jumlah soal yang disediakan adalah 15 soal dan jumlah soal yang akan ditampilkan

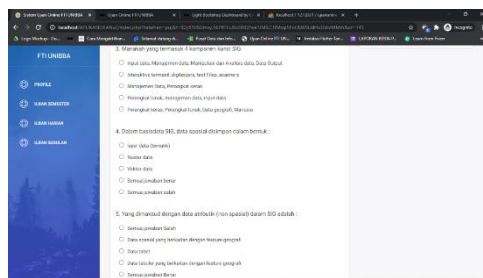
setiap mahasiswa hanya 10 soal. Juga jumlah mahasiswa yang melaksanakan pengujian juga dibatasi dengan berjumlah 3 orang. Berikut adalah proses pengujian:

1. Pengujian Pertama

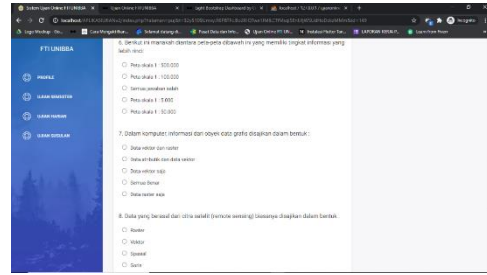
Pada proses pengujian pertama ini, user mahasiswa yang akan melakukan ujian adalah user dengan NIM C1A160013 dengan nama mahasiswa AHMAD KAMAL FASYA, dengan tampilan pelaksanaan ujian sebagai berikut:



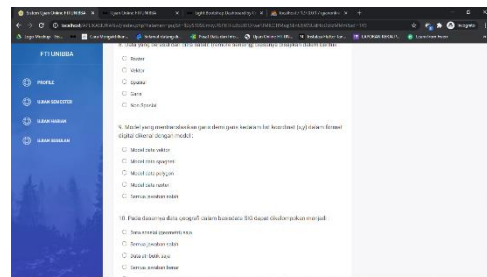
Gambar 37: Tampilan pelaksanaan ujian C1A160013-page1



Gambar 38: Tampilan pelaksanaan ujian C1A160013-page2



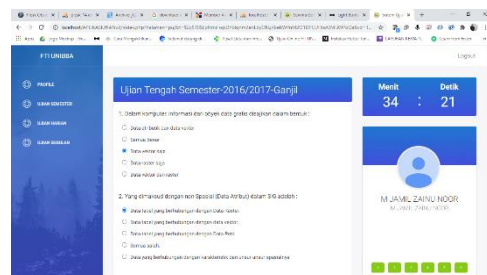
Gambar 39: Tampilan pelaksanaan ujian C1A160013-page3



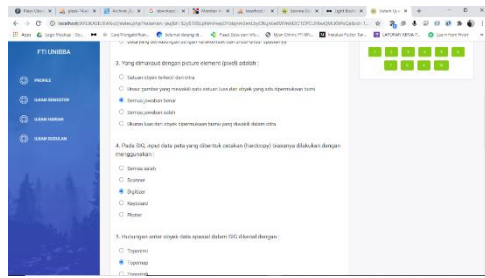
Gambar 40: Tampilan pelaksanaan ujian C1A160013-page4

2. Pengujian Kedua

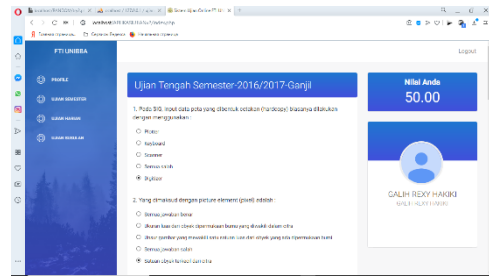
Pada proses pengujian kedua ini, user mahasiswa yang akan melakukan ujian adalah user dengan NIM C1A160029 dengan nama mahasiswa M JAMIL ZAINU NOOR, dengan tampilan pelaksanaan ujian sebagai berikut:



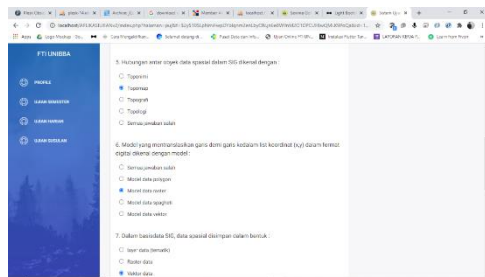
Gambar 41: Tampilan pelaksanaan ujian C1A160029-page1



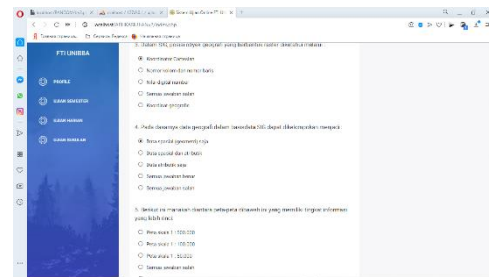
Gambar 42: Tampilan pelaksanaan ujian C1A160029-page2



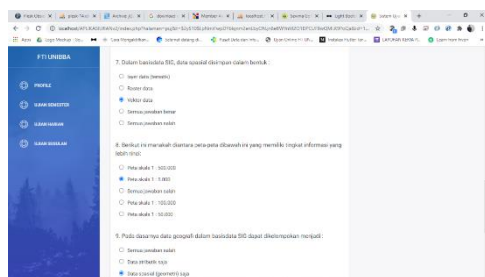
Gambar 45: Tampilan pelaksanaan ujian C1A160009-page1



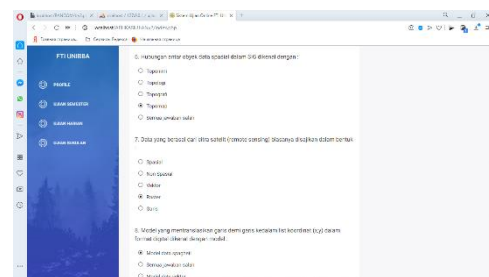
Gambar 43: Tampilan pelaksanaan ujian C1A160029-page3



Gambar 46: Tampilan pelaksanaan ujian C1A160009-page2



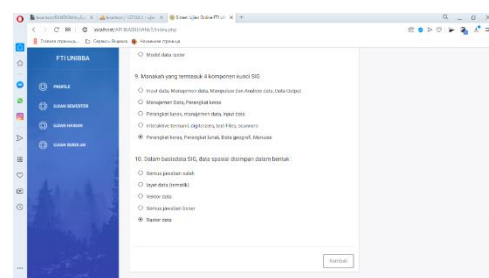
Gambar 44: Tampilan pelaksanaan ujian C1A160029-page4



Gambar 47: Tampilan pelaksanaan ujian C1A160009-page3

3. Pengujian Ketiga

Pada proses pengujian ketiga ini, user mahasiswa yang akan melakukan ujian adalah user dengan NIM C1A160009 dengan nama mahasiswa GALIH REXY HAKIKI, dengan tampilan pelaksanaan ujian sebagai berikut:



Gambar 48: Tampilan pelaksanaan ujian C1A160009-page4

Dari hasil pengujian yang telah dilakukan, masing masing mahasiswa mendapatkan susunan soal dan susunan pilihan ganda yang

saling berbeda satu sama lain. Adapun terjadinya kesamaan susunan soal ujian pada setiap mahasiswa mempunyai persentase yang kecil.

Selain melakukan pengujian diatas, penulis juga melakukan beberapa pengujian – pengujian lainnya, antara lain:

Jumlah User	Hasil
3	Berhasil
5	Berhasil
15	Berhasil

Tabel 3: Tabel uji pengacakan soal pada ujian *online*

VI. KESIMPULAN DAN SARAN

Berdasarkan hasil dari penelitian yang telah dilakukan ini, penulis mencoba mengambil kesimpulan dan mengajukan beberapa saran-saran yang berhubungan dengan pembahasan yang telah dipaparkan pada bab – bab sebelumnya.

6.1 Kesimpulan

1. Algoritma fisher-yates *shuffle* dapat diimplementasikan kedalam proses pengacakan soal pada aplikasi ujian *online*, untuk

mendukung pelaksanaan ujian agar kemungkinan setiap mahasiswa mendapatkan susunan soal yang sama semakin kecil.

2. Pada aplikasi ujian *online* ini, pelaksanaan ujian *online* dilakukan berdasarkan jadwal yang sudah ditentukan, tetapi juga disediakan pelaksanaan ujian susulan.
3. Pada aplikasi ujian *online* ini, nilai hasil ujian dapat diketahui secara *real-time* (langsung).

6.2 Saran

Adapun saran yang dapat penulis berikan, untuk pengembangan aplikasi ujian *online* ini agar lebih baik adalah sebagai berikut :

1. Perlu adanya filterisasi – filterisasi untuk siapa saja yang layak untuk mengikuti pelaksanaan ujian, dan siapa saja yang tidak dapat mengikuti pelaksanaan ujian.
2. Perlu dikembangkannya tipe soal ujian, sehingga soal yang dapat dimasukan tidak hanya pilihan ganda dengan satu jawaban yang benar saja, tetapi bisa juga memasukan soal pilihan ganda

dengan banyak jawaban yang benar.

3. Kembangkan atau tingkatkan UX sehingga dapat lebih memudahkan *user* aplikasi ujian *online* ini dalam melaksanakan ujian.
4. Untuk mempermudah dosen dalam memasukan soal ujian, sediakan fitur untuk *import* data, sehingga dosen hanya perlu mengimpor data soal untuk memasukan soal kedalam sistem.

DAFTAR PUSTAKA

- Abdulloh Rohi, 2017. *Amazing project aplikasi ujian online full ajax*. Tegal, Jawa Tengah : CV. ASFA Solution, Software Development, IT & Publishing.
- Abdulloh Rohi., 2018. *7 in 1 pemrograman web untuk pemula*. Jakarta : PT Elex Media Komputindo
- Andy, "Pengertian XAMPP Lengkap dengan Fungsi dan Cara Instalasi", qwords, dipublikasikan pada 19/09/2019, tersedia <https://qwords.com/blog/pengertian-xampp/>, diakses (10/03/2020 jam 09:07)
- Banyumanis Ulumuddin Ihya, Fitriana Devi (2017) "Aplikasi Alarm Weker Berbasis Android Dengan Algoritma Fisher Yates Shuffle Untuk Mengacak Pertanyaan", Informatika, Universitas Mercubuana, Januari 2017.
- Bolung, M., Tampangela, H, R, K., (2017) "Analisa penggunaan metodologi pengembangan perangkat lunak", Teknik Informatika, Universitas Amikom Yogyakarta, Juni 2017.
- Ekojono, Irawati Ayu Dyah, Affandi Lugman, Rahmanto Nur Anugrah, (2017) "Penerapan Algoritma Fisher Yates Pada Pengacakan Soal Game Aritmatika", Teknologi Informasi, Politeknik Negri Malang, 2017.
- Haditama Imam, Slamet Cepy, Rafma Fauzy Deny (2016) "Implementasi Algoritma Fisher Yates dan Fuzzy Tsukamoto Dalam Games Kuis Tebak Nada Sunda Berbasis Android", Teknik Informatika, Universitas Islam Negri Sunan Gunung Djati Bandung, Juni 2016.
- Hidayatullah Priyanto, Kawisatra Khairul Jauhari., 2016. *Pemrograman web edisi revisi*. Bandung : Informatika.
- Kaban Roberto, "Design Website Responsive Dengan Bootstrap", tersedia <https://osf.io/preprints/inarxiv/mfw39/>, Medan, April 2017.
- Madcoms., 2016 *Pemrograman php dan mysql untuk pemula*. Madiun : CV Andi Offset
- Nugraha Ryan, Exridores Edo, Sopryadi Hendri, "Penerapan Algoritma Fisher Yates Pada Aplikasi The Lost Insect Untuk Pengenalan Jenis Serangga Berbasis Unity 3D", Program

Studi Informatika, STMIK Global
Informatika MDP, Palembang.

Pengertian Sublime Text,
hasantarmizi.blogspot,
dipublikasikan pada 26/04/2017,
tersedia
[https://hasantarmizi.blogspot.com/](https://hasantarmizi.blogspot.com/2017/04/pengertian-sublime-text.html)
[2017/04/pengertian-sublime-](https://hasantarmizi.blogspot.com/2017/04/pengertian-sublime-text.html)
[text.html](https://hasantarmizi.blogspot.com/2017/04/pengertian-sublime-text.html), diakses pada
(10/03/2020 jam 02:14)

Pengertian Web Browser, Fungsi,
Manfaat dan Contohnya,
rejekinomplok.net, tersedia
<https://rejekinomplok.net/pengerti>

[an-web-browser/](#), diakses pada
(10/03/2020 jam 08:57)

Romadhoni Firmansyah, 9 Software
Desain Gratis untuk Membuat
Mockup Website, dipublikasikan
pada 18/10/2019 tersedia
[https://www.jagoanhosting.com/bl](https://www.jagoanhosting.com/blog/software-mockup/)
[og/software-mockup/](https://www.jagoanhosting.com/blog/software-mockup/), diakses
pada (15/06/2020 jam 10:18)

Salahudin M, S A Rosa,. 2018.
Rekayasa perangkat lunak
terstruktur dan berorientasi objek.
Bandung : Informatika.