

Dokumentasi Program Web Kalkulator Perkiraan Nilai Akhir untuk Universitas Terbuka

Yoeru (Yoel Wiyadi)
26 Januari 2025

Abstract

The UT Final Score Estimation Calculator is a web-based application designed to assist students in estimating their final course grades. This system calculates the final score using exam and tutorial scores, following the weight distribution set by Universitas Terbuka. By using this calculator, students can easily estimate their grades without the need for complex manual calculations. The application is developed using HTML, CSS, and JavaScript and is deployed via GitHub Pages for broad accessibility.

Keywords: Universitas Terbuka, Grade Estimation, Automatic Calculation, Web-Based Application.

Abstrak

Kalkulator Perkiraan Nilai Akhir Universitas Terbuka adalah sebuah aplikasi berbasis web yang dirancang untuk membantu mahasiswa dalam memperkirakan nilai akhir suatu mata kuliah. Sistem ini menggunakan data nilai ujian dan nilai tutorial untuk menghitung nilai akhir berdasarkan porsi yang telah ditetapkan oleh Universitas Terbuka. Dengan adanya kalkulator ini, mahasiswa dapat dengan mudah mengetahui estimasi nilai mereka tanpa harus melakukan perhitungan manual yang sering kali membingungkan. Pengembangan aplikasi ini dilakukan menggunakan *HTML*, *CSS*, dan *JavaScript*, serta di-deploy melalui GitHub Pages agar dapat diakses secara luas.

Kata Kunci: Universitas Terbuka, Prediksi Nilai, Perhitungan Otomatis, Aplikasi Berbasis Web.

1. Pendahuluan

1.1. Latar Belakang

Universitas Terbuka (UT) merupakan salah satu Perguruan Tinggi Negeri (PTN) ke-45 yang menerapkan sistem pendidikan yang terbuka dan jarak jauh. Dalam sistem belajarnya, mahasiswa akan mengikuti masa tutorial untuk belajar dan masa ujian yang nantinya akan menjadi penentu nilai akhir pada mata kuliah yang dipilihnya.

Informasi mengenai bocoran dan rumus menghitung nilai telah tersebar ke media sosial, terutama kepada komunitas internal mahasiswa. Akan tetapi, informasi tersebut memicu kebingungan yang disebabkan karena informasi rumus perhitungan lain yang berbeda-beda.

Oleh karena itu, program *tools* kalkulator perkiraan nilai akhir dibuat dengan berbentuk *website* agar dapat digunakan secara *online* dan lebih mudah.

1.2. Tujuan

Penulis sekaligus *developer* berinisiatif membuat program *website* ini dengan tujuan untuk membantu mahasiswa Universitas Terbuka mengetahui perkiraan/estimasi nilai akhir yang akan diperoleh pada mata kuliah yang ditempuhnya.

2. Tinjauan Pustaka

2.1. Universitas Terbuka dan Sistem Pembelajarannya

Universitas Terbuka merupakan Perguruan Tinggi Negeri ke-45 yang menerapkan sistem pembelajaran secara *online*, jarak jauh, dan mandiri. Bentuk

pembelajaran terbuka menerima semua kalangan yang ingin menempuh pendidikan kuliah tanpa/minim adanya batasan (ict_diyah. Juni 30).

Mahasiswa yang terdaftar akan mengikuti serangkaian sistem belajar, yang dapat disederhanakan menjadi dua fase berikut.

2.1.1. Fase Tutorial/Fase Belajar

Di fase tutorial/fase belajar, mahasiswa akan mengikuti pembelajaran dari jenis tutorial dan mata kuliah yang dipilihnya. Adapun jenis tutorial yang dapat dipilih di antaranya:

- a. Tutorial Webinar (Tuweb) dengan pembelajaran yang dilakukan secara *online* dan langsung melalui *video conference*. Mahasiswa dan dosen dapat berinteraksi secara langsung dengan jarak jauh.
- b. Tutorial *Online* (Tuton) dengan pembelajaran yang dilakukan secara *online* dan tidak langsung melalui kelas *online*. Mahasiswa dan dosen hanya bisa berinteraksi layaknya sedang berpesan di forum *online*.
- c. Tutorial Tatap Muka (TTM) dengan pembelajaran yang dilakukan di tempat secara langsung layaknya belajar di kelas saat sekolah.
- d. Tugas Mata Kuliah (TMK) dengan pembelajaran yang lebih mengandalkan tugas dan nilai daripada mengikuti pembelajaran seperti biasanya.

Seluruh kegiatan belajar yang telah diselesaikan akan dihitung sebagai nilai khusus untuk fase tutorial.

2.1.2. Fase Ujian

Setelah fase tutorial dituntaskan, mahasiswa akan mengikuti ujian yang dilaksanakan dengan tempat yang beragam, antara di kampus UT, menumpang di kampus lain, menumpang di sekolah terdekat, atau di tempat lain.

Dalam pelaksanaannya, UT sudah menetapkan jenis pelaksanaan ujian berdasarkan mata kuliah yang ada, tetapi mahasiswa dapat melakukan sedikit negosiasi untuk mengganti jenis pelaksanaan ujiannya. Adapun jenis pelaksanaan ujiannya terdiri atas:

- a. Ujian Tatap Muka (UTM) yang dilaksanakan di tempat secara langsung, dengan lembar soal dan jawaban yang telah disediakan. Mahasiswa harus datang ke tempat ujian.
- b. Ujian *Online* (UO) yang dilaksanakan di tempat secara langsung, dengan komputer yang telah disediakan untuk mengerjakan ujian secara *online*. Akan tetapi, mahasiswa juga harus datang ke tempat ujian.
- c. *Take Home Exam (THE)* yang memberikan kebebasan mahasiswa untuk mengerjakan ujian di rumah dengan men-*download* soal beserta *template* lembar jawaban dari aplikasi ujian, lalu meng-*upload* lembar jawaban yang sudah diisi.
- d. Ujian *Online Remote Proctoring (UORP)* yang memberikan kebebasan mahasiswa untuk mengerjakan ujian di rumah, tetapi ujian dilaksanakan dengan *video conference* untuk memantau ujiannya.

Untuk bentuk soal ujian terdapat dua jenis yang menjadi intinya, yaitu bentuk pilihan ganda dan *essay*.

2.1.3. Porsi Nilai untuk Penentuan Nilai Akhir

Nilai akhir suatu mata kuliah ditentukan dari perolehan nilai ujiannya. Nilai ujian dapat ditunjang/dibantu dengan nilai akumulasi di fase tutorial, seperti yang dijelaskan pada tabel berikut.

Tabel 2.1.3.1. Porsi Nilai Akhir untuk Setiap Jenis Tutorial

Jenis Tutorial	Porsi Nilai Ujian	Porsi Nilai Tutorial
Tutorial Webinar	50%	50%
Tutorial Tatap Muka	50%	50%
Tutorial <i>Online</i>	70%	30%
Tugas Mata Kuliah	80%	20%

Sumber: Kontribusi Nilai Tuweb, TMK, Tuton, dan TMK
Universitas Terbuka

Nilai akhir suatu mata kuliah dihitung melalui perolehan nilai ujian sesuai porsinya dan ditambah dengan nilai tutorial sesuai porsinya. Adapun rumus perhitungannya ialah sebagai berikut.

$$\text{Nilai Akhir} = (\text{Nilai Ujian} \times PNU) + (\text{Nilai Tutorial} \times PNT)$$

Dengan keterangan PNU sebagai Porsi Nilai Ujian dan PNT sebagai Porsi Nilai Tutorial. Akan tetapi, rumus tersebut tidak akan berlaku bila terjadi salah satu syarat/kondisi berikut.

- Bila nilai ujian lebih besar dari nilai tutorial, maka nilai ujianlah yang diambil sebagai nilai akhir suatu mata kuliah.
- Bila perolehan nilai ujian kurang dari 30%, maka nilai tutorial tidak bisa digunakan untuk membantu nilai akhir, dan langsung dianggap sebagai tidak lulus.

2.2. *Website Development*

Website Development mengacu pada kegiatan untuk membuat, mengembangkan, atau membangun *website* dengan menerapkan aspek desain, *coding*, pemrograman, dan publikasi *website*-nya (GeeksforGeeks. 2025). Bahasa

program dan aplikasi yang dimanfaatkan dalam program ini beberapa di antaranya adalah sebagai berikut.

2.2.1. *HTML, CSS, dan JavaScript*

Hyper Text Markup Language (HTML) merupakan bahasa *markup* yang digunakan untuk membuat suatu rangka, fondasi, dan susunan isi halaman suatu *website*. *Cascading Style Sheet (CSS)* adalah bahasa yang mendukung *HTML* untuk mengatur tampilan dan dekorasi dari suatu *website*. *JavaScript* merupakan bahasa program yang digunakan untuk merancang program *website*, membuat *website* yang interaktif, dan dapat dibuatkan aksi sesuai kebutuhan (Ubah, K. 2021).

2.2.2. *GitHub dan GitHub Pages*

GitHub merupakan sarana yang digunakan untuk menyimpan proyek pemrograman, serta dapat menjadi wadah untuk mengerjakan proyek bersama dengan tim/keompok dalam pengembangan programnya. Adapun GitHub juga menyediakan layanan GitHub Pages untuk *men-deploy website* statik dan mempublikasikan *website* secara *online* agar dapat diakses oleh semua orang.

3. Pengembangan

3.1. *Bahasa dan Tools yang Digunakan*

Program *website* ini dikembangkan dengan berbagai bantuan *tools* yang telah tersedia secara gratis dan bebas untuk digunakan. Bahasa penunjang tampilan yang digunakan dalam program *website* ini adalah *HTML* dan *CSS* saja, tanpa mengandalkan *framework-framework* lain karena program yang hanya berfungsi secara sederhana.

Program *website* ini menggunakan bahasa pemrograman *JavaScript* yang difungsikan untuk membaca data yang di-*input*-kan oleh pengguna, melakukan operasi perhitungan matematika, serta menampilkan hasil hitungannya ke dalam *website* ini.

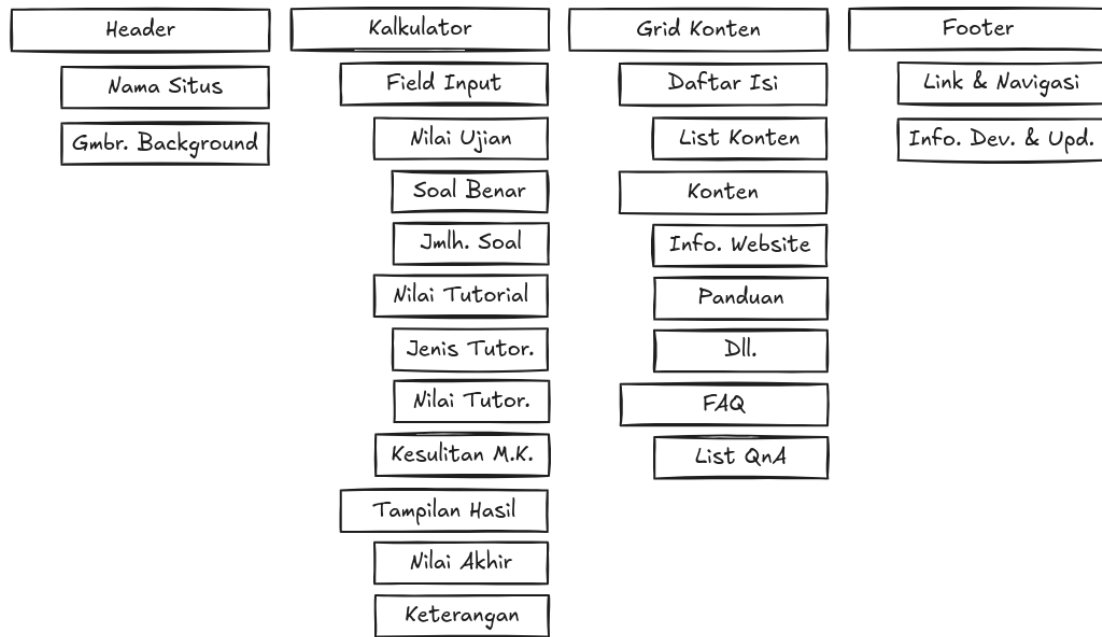
Adapun *tools* yang dipakai untuk menulis semua kode program *website* ini adalah dengan aplikasi Visual Studio Code. Program *website* yang sudah jadi kemudian diuji di perangkat lokal, lalu setelah pengujian menunjukkan hasil yang layak, seluruh kode program di-*upload* dan disimpan di GitHub, serta di-*deploy* dan dipublikasikan secara *online* melalui GitHub Pages.

Setelah *website* di-*deploy*, pengembangan tetap dilanjutkan secara bertahap dengan cara membuat *branch* GitHub baru untuk menuliskan hal-hal yang perlu di-*update*, kemudian menerapkan *update* tersebut dengan menggabungkan *branch* tersebut ke *branch* utama.

3.2. Tampilan Website (*User Interface*) dan Format Kodenya

3.2.1. Markup/Struktur

Pada mulanya, *website* dibentuk dengan struktur/*markup* yang terdiri dari *header*, kalkulator itu sendiri, *grid* konten, dan *footer*, seperti yang disajikan dalam gambar berikut.



Gambar 3.2.1.1. Struktur/Markup Website
Sumber: Dokumen Pribadi

Kemudian kode *markup* HTML dengan bentuk *set-up* awal dituliskan dengan bentuk kode sebagai berikut.

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>UT's Score Calculator</title>
  <link rel="icon" href="./images/favicon/calculator.svg"
type="image/x-icon">
  <!-- File CSS -->
  <link rel="stylesheet" href="./style.css">
  <!-- Font Online -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?
family=Kaushan+Script&family=Radio+Canada:ital,wght@0,300..700;1,300..
700&display=swap" rel="stylesheet">
</head>
<body>
```



```

...
</body>
</html>

```

Kode *markup HTML* untuk *set-up body* yang memuat berbagai bagian *website* dan keseluruhan isinya ditulis dengan bentuk kode sebagai berikut.

```

<body>
  <header>
    <h1 class="title">Kalkulator Nilai Akhir</h1>
    <p class="second-title">Universitas Terbuka</p>
  </header>

  <main>
    <section class="score-information-card">
      ...
    </section>

    <!-- Main Content -->
    <div class="grid-content">

      <!-- Table of Content -->
      <aside class="table-of-content">
        ...
      </aside>

      <!-- Konten -->
      <section class="content">
        ...
      </section>

      <!-- FAQ -->
      <aside class="sidebar-right-content">
        ...
      </aside>
    </div>
  </main>

  <div class="footer-image"></div>
  <footer>
    ...
  </footer>

```

```
</body>
```

Berdasarkan bentuk kode *markup* yang telah ter-*setup* seperti kode di atas, *markup* juga dibuat kembali untuk bagian-bagian yang lebih spesifik dan mendalam, dimulai dari *header* hingga *footer*-nya. *Markup HTML* untuk bagian *header* dituliskan dengan bentuk kode berikut.

```
<body>
  <header>
    <h1 class="title">Kalkulator Nilai Akhir</h1>
    <p class="second-title">Universitas Terbuka</p>
    <!-- gambar background header dimuat dengan CSS -->
  </header>
  ...
</body>
```

Selanjutnya *markup HTML* untuk bagian kalkulator yang menjadi program utama akan memuat *field input*, garis pemisah, dan *output* hasil. Kode dituliskan dengan bentuk kode berikut.

```
<main>
  <!-- Kalkulator -->
  <section class="score-information-card">

    <!-- Field Input -->
    <section>
      <p>Hasil Pengerjaan Ujian</p>
      <div class="flex">
        <div class="field-nilai-benar">
          <p>Benar</p>
          <input type="number" id="nilai-benar" class="nilai-benar">
        </div>
        <p class="slash-sign">/</p>
        <div class="field-jumlah-soal">
          <p>Soal</p>
          <input type="number" id="jumlah-soal" class="jumlah-soal"
placeholder="50">
        </div>
        <div class="field-nilai-ujian-sementara">
```

```

        <p id="nilai-ujian-sementara"></p>
    </div>
</div>

<p>Hasil Nilai Masa Tutorial</p>
<div class="field-jenis-tutorial">
    <p>Jenis Tutorial</p>
    <div class="flex">
        <select id="jenis-tutorial">
            <option value="online">Online (TUTON)</option>
            <option value="webinar">Webinar (TUWEB)</option>
            <option value="tatap-muka">Tatap Muka (TTM)</option>
            <option value="penugasan">Penugasan (TMK)</option>
        </select>
        <p id="info-tunjangan-nilai"></p>
    </div>
</div>
<div class="field-nilai-akumulasi">
    <p>Nilai Akumulasi</p>
    <input type="number" id="nilai-akumulasi" class="nilai-
akumulasi">
</div>

<p>Mata Kuliah</p>
<div class="field-tingkat-kesulitan">
    <p>Tingkat Kesulitan</p>
    <select id="tingkat-kesulitan">
        <option value="mudah">Mudah</option>
        <option value="sedang">Sedang</option>
        <option value="sulit">Sulit</option>
    </select>
</div>
</section>

<!-- Separator Line -->
<div class="line-separator">
    <div class="line"></div>
</div>

<!-- Score Result -->
<section class="hasil-hitungan">
    <p>Perkiraan Nilai Akhirmu Adalah</p>

```

```

    <div class="flex-centered">
      <div class="field-nilai-akhir">
        <p id="nilai-akhir"></p>
      </div>
      <div class="field-keterangan">
        <p id="keterangan"></p>
      </div>
    </div>
    <p id="text-penyemangat" class="text-penyemangat"></p>
  </section>
</section>
...
</main>

```

Berikutnya *markup HTML* untuk bagian *grid* konten yang memuat daftar isi, bagian konten, dan bagian *FAQ* dituliskan dengan bentuk kode berikut.

```

<main>
  ...
  <!-- Main Content -->
  <div class="grid-content">

    <!-- Table of Content -->
    <aside class="table-of-content">
      <div>
        <h2>Daftar Isi</h2>
        <a href="#judul-dan-subjudul-konten">
          <div class="title-list">
            <p>Judul dan Subjudul Konten</p>
          </div>
        </a>
        ...
      </div>
    </aside>

    <!-- Konten -->
    <section class="content">
      <h2 id="subjudul-satu">Subjudul Satu</h2>
      <p>Paragraf dan isinya...</p>
      ...
    </section>

```

```

<!-- FAQ -->
<aside class="sidebar-right-content">
  <div>
    <h2>FAQ</h2>
    <div class="faq-list">
      <div class="title-list">
        <p>Apa yang Tidak Bisa Dilakukan di Tools Ini?</p>
      </div>
      ...
    </div>
  </div>
</aside>
</div>
</main>

```

Terakhir *markup HTML* untuk bagian *footer* dituliskan dengan bentuk kode sebagai berikut.

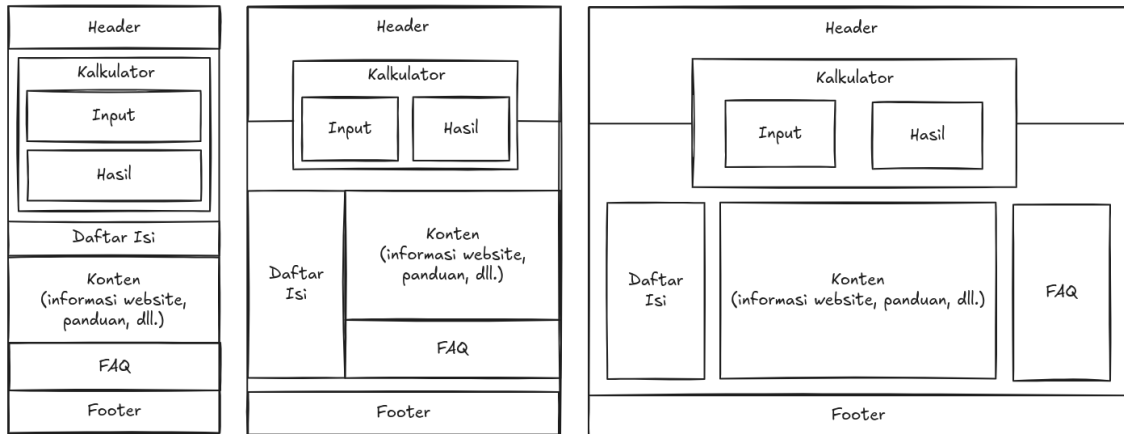
```

<div class="footer-image"></div>
<footer>
  <p>&copy; 2025 - Kalkulator Nilai Akhir UT</p>
  <div class="button-group">
    <a href="link-navigasi">
      <button><svg>...</svg>Link Navigasi</button>
    </a>
    ...
  </div>
  <p>Developed by Yoeru - Version 2.0 - Last Update 13 Jan 2025</p>
</footer>

```

3.2.2. Transisi Tampilan untuk Ukuran Layar yang Berbeda

Website ini memiliki tampilan yang berbeda untuk setiap ukuran layar pada perangkat yang dipakai, yaitu untuk tampilan *mobile*, tampilan tablet, dan tampilan *desktop*.



Gambar 3.2.2.1. *Layout Website* di Berbagai Ukuran Layar

Sumber: Dokumen Pribadi

Transisi *layout* tampilan untuk masing-masing ukuran layar diatur dengan menggunakan *CSS*. Adapun kode *CSS* yang digunakan sebagai setelan *layout* bawaan sekaligus *layout mobile* adalah sebagai berikut.

```
/* Root Settings */
* {
  position: relative;
  margin: 0px;
  padding: 0px;
  text-decoration: none;
  text-decoration-color: currentColor;
  border-radius: 0%;
  scroll-behavior: smooth;
  color-scheme: light dark;

  font-family: "Radio Canada", serif;
  font-optical-sizing: auto;
  font-weight: 400;
  font-style: normal;
  font-size: 1rem;
  font-variation-settings:
    "wdth" 100;
}
:root {
  --black-half: rgb(20, 20, 20);
  --black-page: rgb(40, 40, 40);
  --semi-grey: rgb(100, 100, 100);
  --white-page: rgb(240, 240, 240);
}
```

```

/* Color Pallete */
--marian-blue: #223E7Cff;
--vista-blue: #8697DCff;
--slate-blue: #5A56ADff;
--marian-blue-2: #3F4B94ff;
--marian-blue-3: #3C3B80ff;
}
body {
  background-color: light-dark(unset, var(--black-half));
  color: light-dark(unset, var(--white-page));
}

h1, h2, h3, h4, h5 {
  font-family: "Kaushan Script", serif;
  font-weight: 400;
  font-style: normal;
  font-size: 2rem;
}

/* HEADER */
header {
  text-align: center;
  height: 90px;
  margin-bottom: 10px;
  padding: 30px 0px 0px 0px;
  background-image: url(./images/1680x1050.png);
  background-position: center;
  background-size: cover;
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.4);
}

/* SCORE FIELD */
main {
  display: grid;
  place-items: center;
}

.score-information-card {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  padding: 0% 5% 0% 5%;
}

```

```

    background-color: light-dark(rgba(255, 255, 255, 0.8), rgba(20, 20,
20, 0.8));
    border-width: 1px;
    border-style: solid;
    border-color: light-dark(black, grey);
}

.score-information-card > section {
    width: 100%;
}

.flex > div > p {
    text-align: center;
}

.flex {
    display: flex;
    align-items: center;
    width: 100%;
}

.flex-centered {
    display: flex;
    justify-content: center;
}

/* Field Input */
.score-information-card > section > p {
    font-family: "Kaushan Script", serif;
    font-size: 1.8rem;
    margin: 1.5rem 0px 1rem 0px;
}

select option {
    background-color: light-dark(white, black);
}

input, select {
    height: 40px;
    background-color: transparent;
}

.nilai-benar, .jumlah-soal {

```



```

    width: 70px;
    text-align: center;
    line-height: 50px;
}

.slash-sign {
    margin: 1.15rem 0.5rem 0px 0.5rem;
}

.field-nilai-akumulasi {
    margin-top: 1rem;
}

#nilai-ujian-sementara{
    margin: 1.15rem 0px 0px 1rem;
}

#info-tunjangan-nilai {
    margin-left: 1rem;
}

#nilai-akumulasi,
#jenis-tutorial,
#tingkat-kesulitan {
    text-align: center;
    width: 170px;
    background-color: transparent;
}

.line-separator {
    display: none;
}

.line-separator .line {
    height: 300px;
    border-right-width: 1px;
    border-right-style: solid;
    border-right-color: light-dark(black, grey);
}

/* Field Hasil */
#nilai-akhir,
#keterangan {

```

```

    font-family: "Kaushan Script", serif;
    font-size: 5rem;
    text-align: center;
    margin: 0px 1rem 0px 1rem;
}

.hasil-hitungan p {
    text-align: center;
}

/* Content */
.grid-content {
    display: grid;
    grid-template-columns: 1fr;
}
.grid-single-content {
    display: grid;
    grid-template-columns: min(100%, 700px);
    justify-content: center;
}

.content {
    padding: 20px;
}

/* Table of Content */
.table-of-content,
.sidebar-right-content {
    display: flex;
    justify-content: center;
}

.table-of-content p,
.sidebar-right-content p {
    line-height: 1.5rem;
}

.table-of-content > div,
.sidebar-right-content > div {
    padding: 20px;
    width: 100%;
}

```

```

.table-of-content .title-list {
  border-left-width: 5px;
  border-left-style: solid;
  border-left-color: light-dark(var(--marian-blue), var(--vista-blue));
  padding: 10px;
  margin: 10px 0px 10px 10px;
  color: light-dark(var(--marian-blue), var(--vista-blue));
  transition: 0.7s;
  background-color: transparent;
}
.table-of-content .title-list:hover {
  border-left-width: 20px;
  border-left-color: light-dark(var(--black-half), var(--white-page));
  color: light-dark(var(--black-half), var(--white-page));
  transition: 0.2s;
}
.table-of-content .title-list:active {
  background-color: light-dark(var(--black-half), var(--white-page));
  color: light-dark(white, black);
  transition: 10ms;
  transform: translateX(10px);
}
.table-of-content .indent {
  margin-left: 30px;
}

/* Table */
.table {
  display: flex;
  justify-content: center;
  margin-bottom: 1rem;
}
.table table {
  border-collapse: collapse;
  text-align: center;
  display: block;
  margin-left: auto;
  margin-right: auto;
}
.table table thead {

```

```

    border: 1px solid;
    border-color: light-dark(black, var(--white-page));
}
.table table th,
.table table td {
    padding: 8px;
    border: 1px dashed;
    border-color: light-dark(black, var(--white-page));
}

.content img {
    margin: auto;
    display: block;
    margin: 1rem auto;
}

.content h2 {
    margin: 2rem 0px 1rem 0px;
}
.content h3 {
    font-size: 1.5rem;
    margin: 0.5rem 0rem 0.5rem 0rem;
}

.content p {
    line-height: 1.5rem;
    font-style: italic;
    margin-bottom: 0.7rem;
}
.content .note {
    color: grey;
}

.content ol,
.content ul {
    margin-left: 2rem;
}
.content ol li,
.content ul li {
    padding-left: 0.7rem;
    line-height: 1.5rem;
    font-style: italic;

```

```

    margin-bottom: 1rem;
}
.content ol li:last-child,
.content ul li:last-child {
    margin-bottom: 0px;
}

.content .video-gif,
.content video {
    width: 100%;
}

/* FAQ Sidebar */
.sidebar-right-content .faq-list {
    margin: 20px 0px;
}
.sidebar-right-content .faq-list:first-of-type {
    margin-top: 7px;
}
.sidebar-right-content .faq-list:last-of-type {
    margin-bottom: 7px;
}

.sidebar-right-content .title-list {
    border: 1px solid light-dark(var(--marian-blue-3), var(--vista-blue));
    color: light-dark(var(--marian-blue-3), var(--vista-blue));
    padding: 12px;
}

.sidebar-right-content .answer {
    border: 1px solid light-dark(var(--marian-blue-3), var(--vista-blue));
    border-top: none;
    padding: 12px;
    color: light-dark(var(--semi-grey), grey);
}

.sidebar-right-content .answer p {
    margin-bottom: 1rem;
}

```

```

.sidebar-right-content .answer p:last-of-type {
  margin-bottom: 0px;
}

/* Footer */
.footer-image {
  background-image: url(./images/city-footer.png);
  background-size: contain;
  background-position-y: bottom;
  background-repeat: repeat-x;
  margin-bottom: -5px;
  height: 20vh;
  width: 100%;
  opacity: 0.96;
}

footer {
  text-align: center;
  padding: 10px;
  background-color: light-dark(var(--black-half), rgb(10, 10, 10));
  color: white;
}

footer .button-group {
  display: flex;
  flex-direction: column;
  justify-content: center;
  margin: 10px 0px 10px 0px;
}

footer button {
  background-color: transparent;
  border: 1px solid white;
  color: white;
  padding: 10px;
  margin: 5px 10px;
  transition: 200ms;
  cursor: pointer;
  width: calc(100% - 20px);
}

footer button:hover {

```

```

    background-color: white;
    color: black;
    transition: 200ms;
}
footer button:active {
    transition: 0ms;
    transform: scale(1.05);
}

footer button svg {
    margin-right: 5px;
    margin-bottom: -2px;
}

```

Ketika *website* diakses dengan ukuran layar tablet, kode CSS yang akan dijalankan adalah sebagai berikut.

```

/* RESPONSIVE IN TABLET */
@media screen and (min-width: 600px) {
    footer .button-group {
        flex-direction: row;
    }
    footer button {
        width: unset;
    }
}

@media screen and (min-width: 800px) {
    header {
        height: 50vh;
        margin-bottom: -30vh;
        padding-top: 5vh;
    }

    .score-information-card {
        flex-direction: row;
        align-items: center;
        box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.4);
        backdrop-filter: blur(20px);
        width: min(880px, 80%);
    }
}

```

```
.score-information-card section {
  padding: 2%;
}

.score-information-card > section > p:first-child {
  margin-top: 0px;
}

.line-separator {
  display: flex;
  align-items: center;
}

.grid-content {
  grid-template-columns: 300px 1fr;
}

.table-of-content {
  padding: 40px 20px;
}

.table-of-content > div {
  position: sticky;
  top: 20px;
  height: fit-content;
  max-height: calc(100vh - 80px);
  max-width: 350px;
  overflow-y: scroll;
}

.sidebar-right-content {
  grid-column: 2;
}

.sidebar-right-content > div {
  position: sticky;
  top: 20px;
  height: fit-content;
  max-height: calc(100vh - 80px);
  overflow-y: scroll;
}
```



```

@media screen and (min-height: 500px) {
  header h1 {
    font-size: 3rem;
  }
}

@media screen and (min-height: 1000px) {
  header {
    height: 500px;
    margin-bottom: -350px;
  }
}
}

```

Demikian juga ketika *website* diakses dengan ukuran layar *desktop*. Adapun kode CSS yang akan dijalankan adalah sebagai berikut.

```

/* RESPONSIVE IN DESKTOP */
@media screen and (min-width: 1000px) {
  .grid-content {
    grid-template-columns: 1fr min(700px, 50%) 1fr;
  }

  .sidebar-right-content {
    grid-column: unset;
    padding: 40px 20px;
  }

  .sidebar-right-content > div {
    max-width: 350px;
  }
}

```

3.3. Kode Program *JavaScript*

Dalam program *website* ini, *JavaScript* dipakai untuk membaca data berupa *input* angka dan *input* berbentuk *list option* yang diberikan oleh pemakainya, kemudian diolah dan dihitung untuk memperoleh nilai akhir, lalu

menampilkannya ke dalam *website* secara langsung. Adapun kode *JavaScript*-nya disajikan secara utuh seperti berikut.

```
// Event listener untuk input dan perubahan pada HTML
document.getElementById('nilai-benar').addEventListener('input',
calculateFinalScore);
document.getElementById('jumlah-soal').addEventListener('input',
calculateFinalScore);
document.getElementById('nilai-akumulasi').addEventListener('input',
calculateFinalScore);
document.getElementById('jenis-tutorial').addEventListener('change',
calculateFinalScore);
document.getElementById('tingkat-
kesulitan').addEventListener('change', calculateFinalScore);

function calculateFinalScore() {
    // Membaca data yang diinputkan
    const nilaiBenar = parseInt(document.getElementById('nilai-
benar').value) || 0;
    const jumlahSoal = parseInt(document.getElementById('jumlah-
soal').value) || 0;
    const nilaiAkumulasi = parseInt(document.getElementById('nilai-
akumulasi').value) || 0;
    const jenisTutorial = document.getElementById('jenis-
tutorial').value;
    const tingkatKesulitan = document.getElementById('tingkat-
kesulitan').value;

    // Menghitung nilai ujian sementara
    let nilaiUjian = 0;
    if (jumlahSoal > 0) {
        nilaiUjian = Math.round((nilaiBenar / jumlahSoal) * 100);
        document.getElementById('nilai-ujian-sementara').innerText =
'Nilainya ' + nilaiUjian;
    } else {
        document.getElementById('nilai-ujian-sementara').innerText = '';
    }

    // Menentukan persentase nilai ujian dan nilai tutorial berdasarkan
jenis tutorial
    let persentaseNilaiUjian = 0;
    let persentaseNilaiTutorial = 0;
```

```

let infoTunjanganNilai = '';

switch (jenisTutorial) {
  case 'online':
    persentaseNilaiUjian = 70;
    persentaseNilaiTutorial = 30;
    infoTunjanganNilai = '30% Tutorial & 70% Ujian';
    break;
  case 'webinar':
  case 'tatap-muka':
    persentaseNilaiUjian = 50;
    persentaseNilaiTutorial = 50;
    infoTunjanganNilai = '50% Tutorial & 50% Ujian';
    break;
  case 'penugasan':
    persentaseNilaiUjian = 80;
    persentaseNilaiTutorial = 20;
    infoTunjanganNilai = '20% Tutorial & 80% Ujian';
    break;
}

// Menampilkan informasi tunjangan nilai pada elemen HTML
document.getElementById('info-tunjangan-nilai').innerText =
infoTunjanganNilai;

// Menghitung nilai akhir berdasarkan nilai ujian dan nilai
akumulasi
let nilaiAkhir = 0;
if (nilaiUjian < 30) {
  nilaiAkhir = nilaiUjian;
} else if (nilaiUjian > nilaiAkumulasi) {
  nilaiAkhir = nilaiUjian;
} else {
  nilaiAkhir = Math.round((nilaiUjian * persentaseNilaiUjian / 100)
+ (nilaiAkumulasi * persentaseNilaiTutorial / 100));
}

// Menampilkan nilai akhir pada elemen HTML
document.getElementById('nilai-akhir').innerText = nilaiAkhir;

// Menentukan keterangan berdasarkan tingkat kesulitan dan nilai
akhir

```

```

let keterangan = '';
switch (tingkatKesulitan) {
  case 'mudah':
    keterangan = getKeterangan(nilaiAkhir, [80, 75, 70, 60, 55, 50,
40, 0]);
    break;
  case 'sedang':
    keterangan = getKeterangan(nilaiAkhir, [75, 70, 65, 55, 50, 45,
35, 0]);
    break;
  case 'sulit':
    keterangan = getKeterangan(nilaiAkhir, [70, 65, 60, 50, 45, 40,
30, 0]);
    break;
}

// Menampilkan keterangan pada elemen HTML
document.getElementById('keterangan').innerText = keterangan;

// Menentukan teks penyemangat berdasarkan keterangan
let textPenyemangat = '';
switch (keterangan) {
  case 'A':
  case 'A-':
    textPenyemangat = 'Well Played!!';
    break;
  case 'B':
  case 'B-':
    textPenyemangat = 'Good Job!';
    break;
  case 'C':
  case 'C-':
    textPenyemangat = 'Keep Improving!';
    break;
  case 'D':
    textPenyemangat = 'You can do better!';
    break;
  case 'E':
    textPenyemangat = 'Don\'t give up!';
    break;
}

```

```

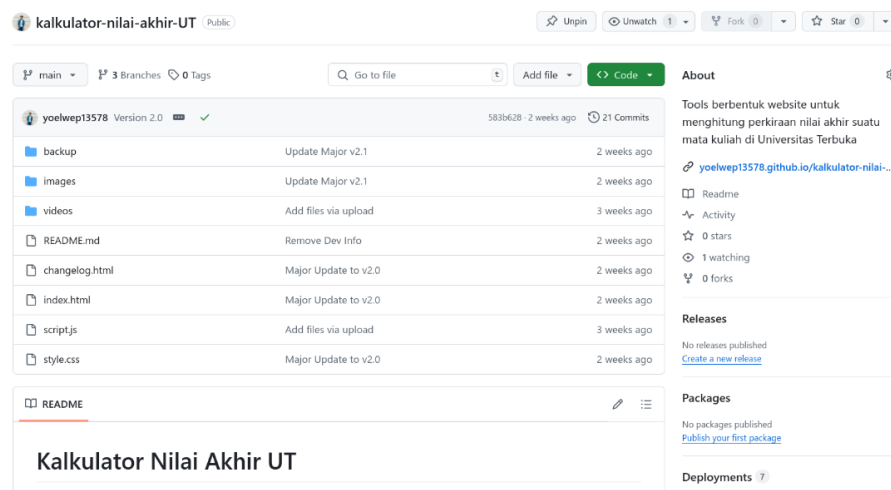
// Menampilkan teks penyemangat pada elemen HTML
document.getElementById('text-penyemangat').innerText =
textPenyemangat;
}

// Fungsi untuk menentukan keterangan berdasarkan nilai dan batas
nilai
function getKeterangan(nilai, thresholds) {
  if (nilai ≥ thresholds[0]) return 'A';
  if (nilai ≥ thresholds[1]) return 'A-';
  if (nilai ≥ thresholds[2]) return 'B';
  if (nilai ≥ thresholds[3]) return 'B-';
  if (nilai ≥ thresholds[4]) return 'C';
  if (nilai ≥ thresholds[5]) return 'C-';
  if (nilai ≥ thresholds[6]) return 'D';
  return 'E';
}

```

3.4. Publikasi Secara *Online (Deployment)*

Seluruh kode program yang menjadi bagian besar dari proyek ini disimpan ke *repository* GitHub dengan tujuan untuk memudahkan pengerjaan dalam menerapkan *update* baru untuk ke depannya.



Gambar 3.4.1. *Repository* “kalkulator-nilai-akhir-UT” di GitHub
Sumber: Dokumen Pribadi

Setelah di-upload ke repository GitHub, repository tersebut di-deploy dengan mengaktifkan GitHub Pages. Sekarang website tersebut dapat dikunjungi dan dapat digunakan oleh semua orang melalui link <https://yoelwep13578.github.io/kalkulator-nilai-akhir-UT>.

4. Hasil dan Pembahasan

Program website kalkulator perkiraan nilai akhir untuk Universitas Terbuka sudah siap digunakan. Adapun tampilan website-nya tampak seperti gambar yang disajikan berikut, dengan catatan bahwa tampilan dapat berubah seiring adanya *update* yang akan diberikan di masa mendatang.



Gambar 4.1. Tampilan Website di Versi Mobile dan Tablet
(Sebelah Kiri untuk Versi Mobile dan Sebelah Kanan untuk Versi Tablet)
Sumber: Dokumen Pribadi



Gambar 4.2. Tampilan Website di Versi Desktop
Sumber: Dokumen Pribadi

4.1. Penggunaan

Program *website* ini dapat digunakan untuk menghitung perkiraan nilai akhir berdasarkan hasil pengerjaan ujian, jenis tutorial, nilai tutorial, serta tingkat kesulitan dari suatu mata kuliah.

Penggunaan program *website* ini lebih efektif jika nilai akhir yang ingin diketahui berasal dari mata kuliah dengan ujian pilihan ganda. Bila bentuk ujiannya adalah *essay* atau selain dari pilihan ganda, maka *input* bagian pengerjaan ujian bisa diisi dengan jumlah soal sebanyak 100, lalu mengisi *input* bagian jawaban benar dengan mengira-ngira (bisa juga diisi dengan jawaban benar 30 untuk batas paling minimal). Nilai ujian yang lebih jelas akan muncul setelah mengisi jumlah jawaban benar dan jumlah soal.

Setelah mengisi *input* bagian pengerjaan ujian, langkah selanjutnya adalah mengisi *input* bagian nilai masa tutorial dengan jenis tutorial yang dipilih dan nilai tutorial yang diperoleh. Bila nilai tutorial tidak diberi tahu kepada mahasiswanya, maka nilai tutorial bisa diisi dengan mengira-ngira berdasarkan perasaan akan nilai yang pantas untuk didapatnya. Informasi mengenai porsi nilai akan muncul setelah memilih opsi jenis tutorial yang dipilih.

Tahapan terakhir adalah mengisi *input* bagian tingkat kesulitan mata kuliah. Adapun tingkat kesulitan mata kuliah ditentukan dari pihak UT dengan mengacu pada nilai rata-rata dari semua peserta ujian di semester itu. Semakin kecil perolehan nilai ujiannya, maka mata kuliah tersebut dianggap semakin sulit dan standar nilainya menurun. Berdasarkan hal tersebut, pengisian *input* tingkat kesulitan mata kuliah bisa ditentukan dengan mengira-ngira apakah mata kuliah tersebut dianggap sulit, sedang, atau mudah baginya.

Perkiraan nilai akhir secara otomatis akan muncul di bawah (untuk versi *mobile*) atau muncul di sebelah kanan (untuk versi tablet & *desktop*) dengan nilai angka, predikat, serta kalimat keterangan sebagai penyemangat.

Hasil Pengerjaan Ujian

Benar: 30 / Soal: 100 Nilainya 30

Hasil Nilai Masa Tutorial

Jenis Tutorial: Online (TUTON) 30% Tutorial & 70% Ujian

Nilai Akumulasi: 95

Mata Kuliah

Tingkat Kesulitan: Sulit

Perkiraan Nilai Akhirmu Adalah

50 B-

Good Job!

Gambar 4.3. Contoh Penggunaan Kalkulator Perkiraan Nilai Akhir UT
Sumber: Dokumen Pribadi

4.2. Cara Kerja

4.2.1. Membaca Data yang Di-input-kan

Saat program *website* ini digunakan, semua data *input* yang diisi akan dibaca dan dideteksi segala perubahan nilainya.

The diagram illustrates the input area of a program. It contains several input fields arranged in a list-like structure, separated by horizontal lines. The fields are: 'Benar', 'Soal', 'Nilai Ujian', 'Jenis Tutorial', 'Ket. Porsi Nilai', 'Nilai Tutorial', and 'Kesulitan MatKul'. To the right of these fields, a text box contains the note: 'Semua input akan dideteksi dan dibaca oleh programnya untuk dilanjut ke proses perhitungan'.

Gambar 4.2.1.1. Area *Input* yang akan Dibaca dan Dideteksi Perubahan Nilainya
Sumber: Dokumen Pribadi

Adapun kode *JavaScript* yang disajikan berikut ini bertugas untuk mendeteksi dan membaca *input* dari bagian pengerjaan ujian, pengerjaan tutorial, hingga tingkat kesulitan mata kuliahnya.

```
// Event listener untuk input dan perubahan data pada HTML
document.getElementById('nilai-benar').addEventListener('input',
calculateFinalScore);
document.getElementById('jumlah-soal').addEventListener('input',
calculateFinalScore);
document.getElementById('nilai-akumulasi').addEventListener('input',
calculateFinalScore);
document.getElementById('jenis-tutorial').addEventListener('change',
calculateFinalScore);
document.getElementById('tingkat-
kesulitan').addEventListener('change', calculateFinalScore);

function calculateFinalScore() {
  // Membaca data yang diinputkan
  const nilaiBenar = parseInt(document.getElementById('nilai-
benar').value) || 0;
```

```

    const jumlahSoal = parseInt(document.getElementById('jumlah-
soal').value) || 0;
    const nilaiAkumulasi = parseInt(document.getElementById('nilai-
akumulasi').value) || 0;
    const jenisTutorial = document.getElementById('jenis-
tutorial').value;
    const tingkatKesulitan = document.getElementById('tingkat-
kesulitan').value;
    ...
}

```

Nilai yang berada di *input* tersebut nantinya akan diolah ke proses perhitungan dan menjadi penentu atas keterangan yang akan ditampilkan di *website*-nya.

4.2.2. Menghitung & Menampilkan Nilai Ujian

Program akan menghitung nilai ujian dengan menggunakan angka yang dimasukkan pada bagian *input* jawaban benar dan jumlah soal. Adapun rumus yang dipakai untuk memperolehnya adalah sebagai berikut.

$$\text{Nilai Ujian} = \frac{\text{Jawaban Benar}}{\text{Jumlah Soal}} \times 100$$

Angka 100 tersebut dimaksudkan agar nilai maksimalnya dapat mencapai 100, atau dapat disebut sebagai nilai berskala 0–100. Perhitungan dilakukan dengan menerapkan rumus tersebut dan langsung menampilkannya di sebelah kanannya dengan menggunakan kode *JavaScript* berikut.

```

function calculateFinalScore() {
    ...
    // Menghitung nilai ujian sementara
    let nilaiUjian = 0;
    if (jumlahSoal > 0) {
        nilaiUjian = Math.round((nilaiBenar / jumlahSoal) * 100);
        document.getElementById('nilai-ujian-sementara').innerText =
'Nilainya ' + nilaiUjian;
    } else {
        document.getElementById('nilai-ujian-sementara').innerText = '';
    }
}

```

```

}
...
}

```



Gambar 4.2.2.1. Posisi Nilai Ujian yang akan Ditampilkan
Sumber: Dokumen Pribadi

4.2.3. Menampilkan Keterangan Porsi Nilai pada Semua Jenis Tutorial

Kedua, program akan menampilkan keterangan porsi nilai berdasarkan jenis tutorial yang dipilih & dimasukkan pada *input* bagian jenis tutorial. Keterangan tersebut mengacu pada porsi nilai dan kontribusi tutorial yang telah disajikan pada Tabel 2.1.3.1. Keterangan tersebut akan ditampilkan di sebelah kanannya.



Gambar 4.2.2.2. Posisi Keterangan Porsi Nilai yang akan Ditampilkan
Sumber: Dokumen Pribadi

Berikut adalah bagian kode *JavaScript* yang bertugas untuk menampilkan keterangan porsi nilai untuk setiap jenis tutorial.

```
function calculateFinalScore() {  
    ...  
    // Menentukan persentase nilai ujian dan nilai tutorial berdasarkan  
    jenis tutorial  
    ...  
    let infoTunjanganNilai = '';  
  
    switch (jenisTutorial) {  
        case 'online':  
            ...  
            infoTunjanganNilai = '30% Tutorial & 70% Ujian';  
            break;  
        case 'webinar':  
        case 'tatap-muka':  
            ...  
            infoTunjanganNilai = '50% Tutorial & 50% Ujian';  
            break;  
        case 'penugasan':  
            ...  
            infoTunjanganNilai = '20% Tutorial & 80% Ujian';  
            break;  
    }  
}
```

```

}

// Menampilkan informasi tunjangan nilai pada elemen HTML
document.getElementById('info-tunjangan-nilai').innerText =
infoTunjanganNilai;

...
}

```

4.2.4. Menghitung Nilai Ujian & Nilai Tutorial dengan Porsinya

Seperti yang telah dijelaskan sebelumnya bahwa nilai akhir memerlukan nilai ujian dan nilai tutorial, serta nilai tersebut sudah ditentukan porsinya oleh jenis tutorial yang dipilih. Program ini akan menghitung nilai ujian dengan porsinya, dan akan disebut sebagai Nilai Ujian Terolah. Adapun rumus perhitungannya sebagai berikut.

$$\boxed{\text{Nilai Ujian}} = \text{Nilai Ujian} \times \text{Porsi Nilai Ujian (\%)}$$

Tidak lupa juga bahwa program ini juga akan menghitung nilai tutorial dengan porsinya, dan akan disebut sebagai Nilai Tutorial Terolah. Adapun rumus perhitungannya sebagai berikut.

$$\boxed{\text{Nilai Tutorial}} = \text{Nilai Tutorial} \times \text{Porsi Nilai Tutorial (\%)}$$

Dalam kenyataannya, rumus tersebut cukup sulit diterapkan dalam bahasa pemrograman, sebab lambang persen akan dianggap sebagai *modulo* (sisanya). Pada dasarnya, persen memiliki arti sebagai “per seratus”. Oleh sebab itu, rumus yang lebih mudah untuk diterapkan dalam pemrograman di antaranya adalah sebagai berikut.

$$\boxed{\text{Nilai Ujian}} = \frac{\text{Nilai Ujian} \times \text{Porsi Nilai Ujian (tanpa tanda \%)}}{100}$$

$$\boxed{\text{Nilai Tutorial}} = \frac{\text{Nilai Tutorial} \times \text{Porsi Nilai Tutorial (tanpa tanda \%)}}{100}$$

Adapun bagian kode *JavaScript* yang bertugas untuk menentukan porsi nilai tutorial dan porsi nilai ujian telah disajikan sebagai berikut.

```
function calculateFinalScore() {  
    ...  
    // Menentukan persentase nilai ujian dan nilai tutorial berdasarkan  
    jenis tutorial  
    let persentaseNilaiUjian = 0;  
    let persentaseNilaiTutorial = 0;  
    let infoTunjanganNilai = '';  
  
    switch (jenisTutorial) {  
        case 'online':  
            persentaseNilaiUjian = 70;  
            persentaseNilaiTutorial = 30;  
            infoTunjanganNilai = '30% Tutorial & 70% Ujian';  
            break;  
        case 'webinar':  
        case 'tatap-muka':  
            persentaseNilaiUjian = 50;  
            persentaseNilaiTutorial = 50;  
            infoTunjanganNilai = '50% Tutorial & 50% Ujian';  
            break;  
        case 'penugasan':  
            persentaseNilaiUjian = 80;  
            persentaseNilaiTutorial = 20;  
            infoTunjanganNilai = '20% Tutorial & 80% Ujian';  
            break;  
    }  
    ...  
}
```

4.2.5. Menghitung & Menampilkan Perkiraan Nilai Akhir

Program ini akan menghitung perkiraan nilai akhir dengan menjumlahkan nilai ujian terolah dan nilai tutorial terolah dengan rumus perhitungan sebagai berikut.

$$\text{Nilai Akhir} = \left[\frac{\text{Nilai Ujian} \times \text{Porsi Nilai Ujian}}{100} \right] + \left[\frac{\text{Nilai Tutor.} \times \text{Porsi Nilai Tutor.}}{100} \right]$$

$$\text{Nilai Akhir} = \boxed{\text{Nilai Ujian}} + \boxed{\text{Nilai Tutorial}}$$

Akan tetapi, terdapat syarat yang menyebabkan rumus tersebut tidak berlaku dan hanya menggunakan nilai ujian sebagai nilai akhirnya. Pertama, saat nilai ujian lebih tinggi/sama dengan nilai tutorial; Kedua, saat nilai ujian sudah tidak tertolong oleh nilai tutorial karena nilainya kurang dari 30 (jawaban benar < 30%).

$if (\text{Nilai Ujian} \geq \text{Nilai Tutorial}) \implies \text{Nilai Akhir} = \text{Nilai Ujian}$

$if (\text{Nilai Ujian} < 30) \implies \text{Nilai Akhir} = \text{Nilai Ujian}$

Rumus dan syarat tersebut akan diterapkan dalam program, kemudian menampilkan hasilnya ke *website*. Bagian kode *JavaScript* yang menangani hal tersebut akan disajikan seperti berikut.

```
function calculateFinalScore() {
    ...
    // Menampilkan informasi tunjangan nilai pada elemen HTML
    document.getElementById('info-tunjangan-nilai').innerText =
    infoTunjanganNilai;

    // Menghitung nilai akhir berdasarkan nilai ujian dan nilai
    akumulasi
    let nilaiAkhir = 0;
    if (nilaiUjian < 30) {
        nilaiAkhir = nilaiUjian;
    } else if (nilaiUjian > nilaiAkumulasi) {
        nilaiAkhir = nilaiUjian;
    } else {
        nilaiAkhir = Math.round((nilaiUjian * persentaseNilaiUjian / 100)
+ (nilaiAkumulasi * persentaseNilaiTutorial / 100));
    }

    // Menampilkan nilai akhir pada elemen HTML
    document.getElementById('nilai-akhir').innerText = nilaiAkhir;
    ...
}
```



Gambar 4.2.5.1. Posisi Nilai Akhir yang Ditampilkan
Sumber: Dokumen Pribadi

4.2.6. Menampilkan Predikat dan Keterangan

Bagian sebelumnya telah disinggung bahwa perolehan nilai akhir juga bisa dipengaruhi oleh tingkat kesulitan mata kuliahnya. Program akan mencocokkan nilai akhir yang berbentuk angka dengan nilai huruf/predikat, seperti pada tabel berikut.

Tabel 4.2.6. Standar Nilai Setiap Tingkat Kesulitan

	Mudah	Sedang	Sulit
A	80–100	75–100	70–100
A-	75–79	70–74	65–69
B	70–74	65–69	60–64
B-	60–69	55–64	50–59
C	55–59	50–54	45–49
C-	50–54	45–49	40–44
D	40–49	35–44	30–39
E	0–39	0–34	0–29

Sumber: Cak Ipho

Adapun kode *JavaScript* yang bertugas menentukan predikat dan keterangan berdasarkan nilai akhirnya disajikan sebagai berikut.

```
function calculateFinalScore() {  
    ...  
    // Menentukan keterangan berdasarkan tingkat kesulitan dan nilai  
akhir  
    let keterangan = '';  
    switch (tingkatKesulitan) {  
        case 'mudah':  
            keterangan = getKeterangan(nilaiAkhir, [80, 75, 70, 60, 55, 50,  
40, 0]);  
            break;  
        case 'sedang':  
            keterangan = getKeterangan(nilaiAkhir, [75, 70, 65, 55, 50, 45,  
35, 0]);  
            break;  
        case 'sulit':  
            keterangan = getKeterangan(nilaiAkhir, [70, 65, 60, 50, 45, 40,  
30, 0]);  
            break;  
    }  
  
    // Menampilkan keterangan pada elemen HTML  
    document.getElementById('keterangan').innerText = keterangan;  
  
    // Menentukan teks penyemangat berdasarkan keterangan  
    let textPenyemangat = '';  
    switch (keterangan) {  
        case 'A':  
        case 'A-':  
            textPenyemangat = 'Well Played!!';  
            break;  
        case 'B':  
        case 'B-':  
            textPenyemangat = 'Good Job!';  
            break;  
        case 'C':  
        case 'C-':  
            textPenyemangat = 'Keep Improving!';  
    }  
}
```

```

        break;
    case 'D':
        textPenyemangat = 'You can do better!';
        break;
    case 'E':
        textPenyemangat = 'Don\'t give up!';
        break;
}

// Menampilkan teks penyemangat pada elemen HTML
document.getElementById('text-penyemangat').innerText =
textPenyemangat;
}

// Fungsi untuk menentukan keterangan berdasarkan nilai dan batas
nilai
function getKeterangan(nilai, thresholds) {
    if (nilai ≥ thresholds[0]) return 'A';
    if (nilai ≥ thresholds[1]) return 'A-';
    if (nilai ≥ thresholds[2]) return 'B';
    if (nilai ≥ thresholds[3]) return 'B-';
    if (nilai ≥ thresholds[4]) return 'C';
    if (nilai ≥ thresholds[5]) return 'C-';
    if (nilai ≥ thresholds[6]) return 'D';
    return 'E';
}

```



Gambar 4.2.6.1. Posisi Predikat dan Keterangan yang Ditampilkan
Sumber: Dokumen Pribadi

4.3. *Disclaimer*

Program *website* ini ditujukan untuk membantu mahasiswa mengetahui perkiraan nilai akhir yang akan diperoleh di suatu mata kuliahnya. Nilai akhir yang telah ditampilkan di sini hanya bersifat sebagai perkiraan, serta kebijakan-kebijakan tersebut dapat berubah sewaktu-waktu.

Keefektifan program *website* ini dapat berkurang apabila: 1) Nilai tidak ditampilkan/tidak ditunjukkan kepada mahasiswa, baik nilai ujiannya ataupun nilai tutorialnya; 2) Jika bentuk soal ujiannya selain umum/pilihan ganda; dan 3) Pengguna telah lupa dengan nilai yang diperoleh sebelumnya (terlalu banyak data yang diisi secara mengira-ngira).

Terkait dengan cara menentukan tingkat kesulitan mata kuliah, hal tersebut telah ditentukan oleh pihak UT dengan mengacu pada perolehan nilai rata-rata peserta ujian di semester itu.

5. Kesimpulan

Kalkulator Perkiraan Nilai Akhir untuk Universitas Terbuka telah berhasil dikembangkan dan dapat digunakan oleh mahasiswa untuk memperkirakan nilai akhir mata kuliah mereka secara mudah dan cepat. Dengan menerapkan perhitungan yang sesuai dengan porsi nilai yang berlaku, alat ini memberikan estimasi yang cukup akurat, terutama untuk mata kuliah dengan ujian berbentuk pilihan ganda. Meskipun begitu, hasil yang diberikan tetap bersifat perkiraan dan tidak dapat menggantikan nilai resmi dari Universitas Terbuka. Untuk ke depannya, pengembangan terus dilanjutkan dengan memberi sejumlah *update* untuk peningkatan *user interface*, dan fitur lain yang akan mendatang.

Referensi

- Cak Ipho. (2022, Desember 9). *Tutorial Kilat Cara Hitung Nilai Akhir Mahasiswa Universitas Terbuka UT: Nilai Ujian Tutorial IPK* [Video]. YouTube. <https://youtu.be/kbwpJ0nJqDs?si=EZh3bLRSnMtH8XYS>
- Cak Ipho. (2022, Desember 15). *Tutorial Kilat Cara Konversi Nilai Huruf dari Angka di Universitas Terbuka UT Kategori Akhir IP IPK* [Video]. YouTube. https://youtu.be/WupTCXF-FIY?si=po3Kn39tW_cnUT0Q
- Ubah, K. (2021, Agustus 10). *Learn Web Development Basics – HTML, CSS, and JavaScript Explained for Beginners*. freeCodeCamp. Diakses 26 Januari 2025 melalui <https://www.freecodecamp.org/news/html-css-and-javascript-explained-for-beginners/>
- GeeksforGeeks. (2025, Januari 13). *Web Development*. Diakses 26 Januari 2025 melalui <https://www.geeksforgeeks.org/web-development/>
- ict_diyah. (Juni 30). *Mengenal Universitas Terbuka, Solusi Bagi Mereka yang Terhalang Jarak, Waktu, dan Usia*. Diakses 26 Januari 2025 melalui <https://samarinda.ut.ac.id/blog/mengenal-universitas-terbuka-solusi-bagi-mereka-yang-terhalang-jarak-waktu-dan-usia>

Universitas Terbuka Bengkulu. (n.d.). *Sistem Ujian UT*. Diakses 26 Januari 2025 melalui <https://bengkulu.ut.ac.id/sistem-ujian>