

# **Rencana Pembelajaran Semester Terintegrasi untuk Program Studi Sistem Informasi STMIK Tazkia: Manajemen Proyek, Analisa dan Perancangan Sistem, dan Pengembangan Aplikasi Web**

## **BAB I: Pendahuluan dan Justifikasi Filosofis**

### **1.1 Latar Belakang dan Urgensi Pembelajaran Terintegrasi**

Perkembangan industri digital yang pesat menuntut institusi pendidikan tinggi untuk merefleksikan dan merekonstruksi kurikulum agar relevan dengan kebutuhan dunia kerja. Program Studi Sistem Informasi di STMIK Tazkia telah mengidentifikasi beberapa profil lulusan kunci, termasuk System Analyst, Software Developer, dan UI/UX Designer.<sup>1</sup> Peran-peran ini bukanlah entitas yang berdiri sendiri; sebaliknya, mereka sangat terintegrasi dalam siklus hidup pengembangan perangkat lunak (Software Development Life Cycle atau SDLC) yang berkelanjutan dan saling bergantung.<sup>2</sup> Seorang

System Analyst harus memahami kendala dan proses pengembangan yang akan dihadapi oleh Software Developer, sementara Software Developer harus mampu menerjemahkan rancangan sistem yang dibuat oleh analis menjadi kode yang fungsional. Seluruh proses ini harus dikelola secara efektif dan efisien oleh seorang manajer proyek.

Pola pembelajaran yang memisahkan mata kuliah seperti Manajemen Proyek, Analisa dan Perancangan Sistem (APS), dan Pengembangan Aplikasi Web ke dalam sesi-sesi yang terisolasi (siloed) berisiko menciptakan lulusan yang memiliki pemahaman parsial. Mahasiswa mungkin ahli dalam satu aspek (misalnya, membuat kode) tetapi tidak memahami bagaimana keputusan teknis mereka memengaruhi jadwal atau anggaran proyek, atau bagaimana kebutuhan bisnis yang dianalisis pada fase awal diterjemahkan menjadi arsitektur perangkat lunak. Pembelajaran terfragmentasi semacam ini tidak optimal dalam membentuk lulusan yang memiliki pemahaman holistik tentang SDLC. Oleh karena itu, diperlukan sebuah pendekatan kurikulum yang menyatukan ketiga kompetensi ini ke dalam satu pengalaman pembelajaran yang kohesif, realistis, dan berorientasi pada hasil. Pendekatan ini secara langsung mereplikasi dinamika tim pengembangan perangkat lunak di dunia nyata, di mana setiap anggota harus memahami dan berkolaborasi dalam setiap fase proyek, mulai dari inisiasi

hingga penutupan.

## **1.2 Landasan Hukum dan Kebijakan: Penjaminan Mutu dan Inovasi Kurikulum**

Rancangan Rencana Pembelajaran Semester (RPS) ini didasarkan pada landasan hukum terbaru yang memberikan fleksibilitas dan otonomi yang lebih besar bagi perguruan tinggi untuk berinovasi. Peraturan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi Nomor 53 Tahun 2023 tentang Penjaminan Mutu Pendidikan Tinggi telah mencabut peraturan sebelumnya, termasuk Permendikbud No. 5 Tahun 2020 dan Permendikbud No. 3 Tahun 2020. Pergeseran fokus dari "standar nasional" yang kaku menjadi "penjaminan mutu" yang ditetapkan oleh perguruan tinggi sendiri menandai sebuah perubahan paradigma. Regulasi ini memberikan keleluasaan bagi STMIK Tazkia untuk merancang kurikulum yang tidak lagi terpaku pada format 16 sesi di dalam kelas, asalkan institusi dapat menunjukkan dan menjamin bahwa capaian pembelajaran (CPMK) dan kualitas lulusan tetap terpenuhi.

Fleksibilitas ini sejalan dengan semangat program Merdeka Belajar Kampus Merdeka (MBKM), yang memberikan hak kepada mahasiswa untuk mengambil pembelajaran di luar program studi mereka hingga 40 SKS, atau setara dengan dua semester.<sup>3</sup> Meskipun rancangan ini berada di dalam program studi, filosofi MBKM tentang pembelajaran yang adaptif, berbasis minat, dan berorientasi pada pengalaman nyata menjadi inspirasi utama. Dengan memanfaatkan keleluasaan regulasi ini, STMIK Tazkia dapat mengimplementasikan model pembelajaran yang lebih inovatif dan tidak konvensional, yang memungkinkan mahasiswa untuk mengalami seluruh siklus pengembangan perangkat lunak secara terintegrasi. Hal ini juga memungkinkan perguruan tinggi untuk secara proaktif menyesuaikan kurikulumnya dengan dinamika industri tanpa harus menunggu perubahan regulasi yang preskriptif, selama ada sistem penjaminan mutu yang kuat untuk mengevaluasi dan memvalidasi hasilnya.

## **BAB II: Konsep Metode Pembelajaran Terintegrasi (Integrated Project-Based Learning)**

### **2.1 Filosofi Model Pembelajaran: PjBL sebagai Jembatan Teori-Praktik**

Model pembelajaran yang diusulkan untuk RPS terintegrasi ini adalah Integrated Project-Based Learning (PjBL). PjBL dipilih karena kemampuannya untuk mengarahkan mahasiswa pada pemikiran kreatif dan penyelesaian masalah. Model ini telah terbukti secara signifikan meningkatkan hasil belajar mahasiswa dibandingkan metode pembelajaran langsung yang bersifat teoretis. Dengan PjBL, mahasiswa tidak hanya mempelajari teori secara pasif, tetapi secara aktif mengaplikasikan pengetahuan tersebut untuk membangun sebuah produk nyata.

Dalam konteks mata kuliah Manajemen Proyek, Analisa dan Perancangan Sistem, dan Pengembangan Aplikasi Web, PjBL menciptakan sebuah narasi pembelajaran yang realistis.

Mahasiswa akan menjalankan sebuah proyek tunggal yang berfungsi sebagai benang merah yang menyatukan ketiga mata kuliah tersebut. Proyek ini akan dimulai dengan identifikasi masalah (fase APS), dilanjutkan dengan perencanaan dan manajemen (fase Manajemen Proyek), dan diakhiri dengan implementasi dan pengujian (fase Pengembangan Aplikasi Web). Melalui pendekatan ini, mahasiswa tidak hanya mendapatkan wawasan teoretis, tetapi juga pengalaman praktis dalam mengelola, merancang, dan mengembangkan sebuah sistem informasi berbasis web dari awal hingga akhir, yang merupakan keterampilan esensial bagi profesional di industri.

## 2.2 Sinergi Tiga Matakuliah dalam Satu Siklus Proyek

Siklus hidup pengembangan perangkat lunak (SDLC) berfungsi sebagai mekanisme pemandu yang ideal untuk mengintegrasikan ketiga mata kuliah ini. Mata kuliah tidak lagi diajarkan secara terpisah, melainkan sebagai fase-fase yang berurutan dalam sebuah proyek tunggal.

- **Manajemen Proyek** berperan sebagai kerangka kerja yang membingkai seluruh kegiatan. Mahasiswa akan belajar tentang tahapan manajemen proyek, mulai dari inisiasi, perencanaan, eksekusi, pengawasan, hingga penutupan. Kompetensi ini akan diaplikasikan untuk mengatur waktu, anggaran, dan sumber daya proyek pengembangan aplikasi web yang sedang mereka kerjakan.
- **Analisa dan Perancangan Sistem (APS)** bertindak sebagai fase perancangan yang menghasilkan blueprint sistem. Mahasiswa akan menerapkan konsep-konsep APS untuk menganalisis kebutuhan fungsional dan non-fungsional, serta merancang model sistem menggunakan alat seperti Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), dan Unified Modelling Language (UML).<sup>4</sup> Hasil dari fase ini akan menjadi acuan teknis untuk fase selanjutnya.
- **Pengembangan Aplikasi Web** adalah fase implementasi yang mengubah blueprint dari fase APS menjadi produk nyata. Mahasiswa akan menggunakan teknologi HTML, CSS, JavaScript, dan PHP serta memanfaatkan framework untuk membangun aplikasi web yang fungsional dan sesuai dengan rancangan.

Hubungan kausal antara ketiga mata kuliah ini sangat erat. Keputusan yang diambil pada fase analisis dan perancangan akan langsung memengaruhi kompleksitas dan durasi fase pengembangan, yang pada gilirannya akan berdampak pada jadwal dan biaya yang diatur dalam manajemen proyek. Model terintegrasi ini secara otomatis mendorong mahasiswa untuk melihat proyek dari perspektif 360 derajat, di mana setiap fase memengaruhi yang lain, seperti halnya di lingkungan kerja profesional.

Untuk memastikan transparansi dan akuntabilitas akademik, matrik di bawah ini mendokumentasikan bagaimana capaian pembelajaran dari masing-masing mata kuliah konsolidasi menjadi satu set capaian pembelajaran terintegrasi.

**Tabel 1: Matriks Capaian Pembelajaran Terintegrasi**

Capaian Pembelajaran Lulusan (CPL)	CPMK Mata Kuliah Asli	CPMK Terintegrasi (RPS)					
Mampu menganalisis, merancang, dan mengimplementasikan sistem informasi.	<b>Manajemen Proyek:</b> <ul style="list-style-type: none"> <li>- Mampu menjelaskan konsep manajemen proyek TI. <sup>2</sup></li> <li>- Mampu membuat penjadwalan proyek (</li> </ul>	WBS, Gantt Chart). <sup>8</sup>	<b>APS:</b> <ul style="list-style-type: none"> <li>- Mampu memahami konsep dasar sistem informasi. <sup>6</sup></li> <li>- Mampu menganalisis dan merancang sistem berbasis prosedural dan objek. <sup>6</sup></li> <li>- Mampu merancang model perancangan (</li> </ul>	DFD, ERD, UML). <sup>6</sup>	<b>Pengembangan Web:</b> <ul style="list-style-type: none"> <li>- Mampu menguasai konsep dan keahlian bahasa pemrograman web.</li> <li>- Mampu menerapkan dasar algoritma untuk perancangan aplikasi web.</li> </ul>	1. <b>Manajemen Proyek Holistik:</b> Mampu menerapkan prinsip manajemen proyek (Initiating, Planning, Executing, Monitoring, Closing) untuk mengelola seluruh siklus pengembangan aplikasi web. <sup>9</sup> 2.	<b>Analisis dan Perancangan Solusi:</b> Mampu menganalisis kebutuhan fungsional dan non-fungsional serta merancang blueprint sistem informasi berbasis web menggunakan model perancangan yang relevan. 3. <b>Pengembangan dan</b>

							<b>Implementasi Berbasis Rancangan:</b> Mampu mengimplementasikan rancangan sistem menjadi aplikasi web fungsional yang menggunakan teknologi web terkini.
Mampu bekerja secara mandiri dan dalam tim untuk menyelesaikan masalah kompleks.	<b>Manajemen Proyek:</b> - Mampu mengelola tim proyek. <sup>2</sup>	<b>Pengembangan Web:</b> - Mampu terlibat dalam pengembangan profesi berkelanjutan.	<b>4. Kolaborasi dan Komunikasi Efektif:</b> Mampu bekerja secara kolaboratif dalam tim, mengelola konflik,				

			dan berkomunikasi secara efektif untuk mencapai tujuan proyek.
Mampu bertanggung jawab atas pekerjaan di bidang keahliannya.	<b>Pengembangan Web:</b> - Mampu menunjukkan sikap bertanggung jawab atas pekerjaan di bidang keahliannya. <sup>10</sup>	<b>5. Tanggung Jawab dan Etika Profesional:</b> Mampu menunjukkan etos kerja yang bertanggung jawab dan etika profesional dalam setiap fase proyek.	

### 2.3 Konversi Beban Belajar SKS Berdasarkan Regulasi Terbaru

Rancangan pembelajaran ini didukung penuh oleh Peraturan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi Nomor 53 Tahun 2023, yang mengubah definisi dan perhitungan Satuan

Kredit Semester (SKS). Aturan lama menetapkan beban belajar 1 SKS secara rinci ke dalam sesi tatap muka (50 menit per minggu), penugasan terstruktur (60 menit per minggu), dan kegiatan mandiri (60 menit per minggu).<sup>11</sup>

Dalam peraturan terbaru, definisi ini diubah secara fundamental. Beban belajar 1 SKS kini setara dengan **45 jam kegiatan per semester**.<sup>11</sup> Perubahan ini memberikan otonomi dan fleksibilitas yang jauh lebih besar bagi perguruan tinggi untuk menentukan cara pemenuhan beban belajar tersebut.<sup>11</sup> Beban belajar dapat dipenuhi melalui berbagai bentuk kegiatan, seperti kuliah, responsi, tutorial, praktikum, penelitian, perancangan, pengembangan, atau proyek.<sup>11</sup> Dengan demikian, model pembelajaran

Integrated Project-Based Learning yang diusulkan ini tidak lagi terikat pada format 16 sesi tatap muka di dalam kelas. Sebaliknya, dosen dapat memfasilitasi pembelajaran melalui kegiatan yang berorientasi pada proyek di luar kelas, seperti bimbingan proyek mingguan, studi kasus, atau pengembangan aplikasi, yang semuanya terakumulasi untuk memenuhi total jam SKS yang ditentukan per semester. Ini sejalan dengan tujuan regulasi baru untuk menciptakan pengalaman belajar yang lebih adaptif, inovatif, dan relevan dengan dunia kerja.<sup>12</sup>

## **BAB III: Teknis Pelaksanaan: Alur Proyek dan Metode Kolaboratif**

### **3.1 Pendekatan Proyek Hibrida: Menggabungkan Struktur dan Fleksibilitas**

Rancangan RPS ini mengadopsi pendekatan Hybrid Agile-Waterfall. Pendekatan ini adalah solusi pragmatis yang mengakui keterbatasan model tradisional Waterfall yang linear dan kaku, yang sering kali kurang fleksibel untuk proyek IT yang dinamis.<sup>13</sup> Namun, pendekatan

Agile murni, meskipun fleksibel, mungkin terlalu bebas untuk lingkungan akademik yang membutuhkan struktur jelas untuk pengawasan dan penilaian.

Model hibrida ini akan menggunakan kerangka Waterfall sebagai struktur tingkat tinggi yang membagi proyek ke dalam fase-fase yang berurutan (Analisis, Desain, Implementasi). Kerangka ini memastikan bahwa mahasiswa memiliki deliverables yang jelas di setiap fase, yang sangat penting untuk penilaian dan dokumentasi.<sup>14</sup> Di saat yang sama, prinsip-prinsip

Agile, seperti iterasi dan siklus pengembangan yang lebih pendek (sprint), akan diterapkan dalam setiap fase, terutama saat coding dan pengujian. Pendekatan ini memberikan manfaat dari struktur Waterfall untuk perencanaan dan dokumentasi proyek yang komprehensif, sementara pada saat yang sama memungkinkan tim mahasiswa untuk beradaptasi dengan perubahan kebutuhan atau tantangan teknis, mencerminkan praktik terbaik di industri modern.

## 3.2 Alur Fase Proyek dan Deliverables

Proyek semester ini akan dibagi menjadi lima fase utama yang mencerminkan siklus SDLC. Mahasiswa akan bekerja dalam tim untuk menyelesaikan setiap fase dan menghasilkan deliverables spesifik yang akan menjadi dasar penilaian. Pembelajaran akan berlangsung secara fleksibel di luar 16 sesi kelas tradisional, di mana dosen berperan sebagai fasilitator dan mentor, bukan sebagai pengajar di depan kelas.

### Fase 1: Inisiasi & Perencanaan Proyek (Manajemen Proyek & Analisa Sistem)

- **Aktivitas:** Tim mahasiswa akan memulai dengan mendefinisikan tujuan sistem, mengidentifikasi masalah, dan menganalisis persyaratan awal. Mereka akan menyusun struktur tim, peran, dan tanggung jawab.
- **Deliverables:** Project Charter yang mendefinisikan ruang lingkup, Work Breakdown Structure (WBS) yang memecah pekerjaan kompleks menjadi bagian-bagian yang lebih kecil<sup>9</sup>, dan Project Plan yang mencakup jadwal kasar dan alokasi sumber daya.

### Fase 2: Analisis & Perancangan Sistem (Analisa dan Perancangan Sistem)

- **Aktivitas:** Berdasarkan kebutuhan yang teridentifikasi, tim akan merancang blueprint sistem. Ini termasuk pemodelan proses bisnis, perancangan model basis data (ERD), dan perancangan antarmuka pengguna (UI/UX).<sup>6</sup>
- **Deliverables:** Dokumen Analisis Sistem yang berisi Use Case Diagram, DFD atau UML, dan Desain Antarmuka yang telah divalidasi.

### Fase 3: Pengembangan & Implementasi Aplikasi Web (Pengembangan Aplikasi Web)

- **Aktivitas:** Tim akan mulai coding berdasarkan desain yang telah dibuat pada fase sebelumnya.<sup>2</sup> Mereka akan mengimplementasikan front-end (menggunakan HTML, CSS, JavaScript) dan back-end (menggunakan PHP dan MySQL sebagai basis data), serta mengintegrasikannya.
- **Deliverables:** Source Code aplikasi web yang berfungsi, terstruktur, dan memiliki dokumentasi teknis yang memadai.

### Fase 4: Pengujian & Validasi (Pengembangan Aplikasi Web & Manajemen Proyek)

- **Aktivitas:** Tim akan melakukan serangkaian pengujian untuk memastikan aplikasi berfungsi sesuai persyaratan. Jenis pengujian yang dilakukan meliputi unit testing, integration testing, validation testing, dan compatibility testing di berbagai peramban.
- **Deliverables:** Dokumen Uji Sistem yang berisi test case dan laporan hasil pengujian.

### Fase 5: Evaluasi & Penutupan Proyek (Manajemen Proyek)

- **Aktivitas:** Fase ini menandai akhir dari proyek. Tim akan menyiapkan Laporan Akhir yang komprehensif, melakukan presentasi akhir untuk mendemonstrasikan aplikasi, dan mengevaluasi kinerja tim secara keseluruhan.



- **Deliverables:** Laporan Akhir Proyek, presentasi dan demonstrasi aplikasi.

**Tabel 2: Alur Fase Proyek dan Deliverables**

Fase Proyek	Aktivitas Utama	CPMK Terkait	Deliverables
<b>Fase 1:</b> Inisiasi & Perencanaan	Identifikasi masalah, analisis kebutuhan awal, perencanaan WBS dan jadwal.	CPMK 1 (Manajemen Proyek Holistik), CPMK 4 (Kolaborasi).	Project Charter, Work Breakdown Structure (WBS), Project Plan.
<b>Fase 2:</b> Analisis & Perancangan	Pengumpulan kebutuhan, pemodelan sistem, perancangan UI/UX.	CPMK 2 (Analisis dan Perancangan Solusi), CPMK 5 (Tanggung Jawab).	Dokumen Analisis Sistem (Use Case, DFD/UML), Desain Antarmuka.
<b>Fase 3:</b> Pengembangan & Implementasi	Coding front-end dan back-end, integrasi basis data, penerapan framework.	CPMK 3 (Pengembangan dan Implementasi), CPMK 5 (Tanggung Jawab).	Source Code aplikasi web, Dokumentasi teknis.
<b>Fase 4:</b> Pengujian & Validasi	Pengujian unit, integrasi, validasi, dan kompatibilitas aplikasi.	CPMK 3 (Pengembangan dan Implementasi).	Dokumen Uji Sistem (Test Case, Test Report).
<b>Fase 5:</b> Evaluasi & Penutupan	Presentasi, demonstrasi aplikasi, evaluasi kinerja tim, penyusunan Laporan Akhir.	CPMK 1 (Manajemen Proyek Holistik), CPMK 4 (Kolaborasi), CPMK 5 (Tanggung Jawab).	Laporan Akhir Proyek, Presentasi dan demonstrasi.

### 3.3 Jadwal Proyek Semesteran dan Deliverable Mingguan

Jadwal ini berfungsi sebagai peta jalan proyek yang terperinci. Setiap minggu, tim mahasiswa harus menunjukkan kemajuan yang terukur, yang akan diverifikasi melalui agenda monitoring mingguan dengan dosen. Jadwal ini tidak mengikat pada sesi kelas tatap muka, melainkan pada pencapaian deliverable secara mandiri.

**Tabel 3: Jadwal Proyek Semesteran dan Deliverable Mingguan**

Minggu Ke-	Fase Proyek	Agenda Monitoring Mingguan	Deliverable Mahasiswa
<b>1-2</b>	<b>Fase 1: Inisiasi Proyek</b>	Presentasi ide proyek dan kasus bisnis. Dosen memberikan umpan balik dan persetujuan awal.	<b>Minggu 2:</b> Project Proposal dan Business Case awal.
<b>3-4</b>	<b>Fase 1: Perencanaan Proyek</b>	Diskusi mendalam tentang Work Breakdown Structure (WBS) dan jadwal. Dosen meninjau Project Plan.	<b>Minggu 4:</b> Project Charter dan Project Plan (WBS dan Gantt Chart). <sup>8</sup>
<b>5-6</b>	<b>Fase 2: Analisis Kebutuhan</b>	Review Use Case Diagram, DFD atau UML, dan Functional Requirements.	<b>Minggu 6:</b> Dokumen Analisis Sistem yang mencakup semua model analisis.
<b>7-8</b>	<b>Fase 2: Perancangan Sistem</b>	Review Database Design (ERD) dan User Interface (UI/UX) mockups.	<b>Minggu 8:</b> Dokumen Perancangan Sistem yang mencakup ERD dan Desain Antarmuka.

9-10	<b>Fase 3: Pengembangan (Front-End)</b>	Demosi awal prototipe front-end yang fungsional.	<b>Minggu 10:</b> Source Code front-end yang telah selesai.
11-12	<b>Fase 3: Pengembangan (Back-End &amp; Integrasi)</b>	Demosi fungsionalitas back-end (logika bisnis dan integrasi basis data) serta integrasi front-end.	<b>Minggu 12:</b> Source Code aplikasi web yang terintegrasi (minimal prototipe berfungsi).
13-14	<b>Fase 4: Pengujian &amp; Validasi</b>	Presentasi Test Case dan hasil pengujian (unit, integrasi, validasi, kompatibilitas).	<b>Minggu 14:</b> Dokumen Uji Sistem dan laporan bug.
15-16	<b>Fase 5: Penutupan Proyek</b>	Presentasi final dan demonstrasi aplikasi. Diskusi evaluasi proyek.	<b>Minggu 15:</b> Final Project Report. <b>Minggu 16:</b> Final Presentation dan Application Demonstration.

### 3.4 Panduan Teknis Pengumpulan Deliverable (Git dan GitHub/GitLab)

Untuk mendukung kolaborasi tim yang efektif dan pengawasan proyek yang fleksibel, setiap tim mahasiswa diwajibkan menggunakan sistem kontrol versi terdistribusi (Distributed Version Control System) seperti Git, dengan repositori yang di-hosting di platform gratis seperti GitHub atau GitLab. Penggunaan platform ini berfungsi sebagai **sumber kebenaran tunggal** (single source of truth) untuk semua artefak proyek, mulai dari dokumen hingga kode program.<sup>16</sup>

#### Panduan Penggunaan Git untuk Tim Mahasiswa:

1. **Struktur Repositori:** Tim harus membuat satu repositori proyek (project repository) untuk seluruh deliverable semester. Repositori ini akan menjadi tempat utama bagi semua anggota tim untuk menyimpan dan memperbarui pekerjaan mereka.
2. **Manajemen Cabang (Branching):** Repositori harus memiliki satu cabang utama (main atau master) yang hanya digunakan untuk menyimpan versi proyek yang stabil atau yang sudah disetujui. Setiap anggota tim harus membuat cabang baru (feature branch) untuk

setiap tugas atau fitur yang mereka kerjakan. Hal ini memastikan bahwa perubahan yang dilakukan oleh satu anggota tidak langsung mengganggu pekerjaan anggota lainnya di cabang utama.<sup>17</sup>

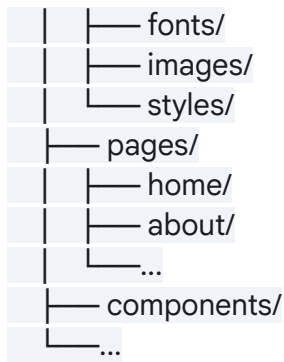
### 3. Alur Kerja Sederhana:

- **Kloning Repositori:** Setiap anggota tim mengkloning repositori ke komputer lokal mereka menggunakan perintah git clone.
- **Buat Cabang Baru:** Sebelum memulai tugas, buat cabang baru dengan nama yang jelas (misalnya, fitur-login atau analisis-kebutuhan) menggunakan git checkout -b <nama-cabang>.
- **Kerjakan Tugas & Simpan Perubahan:** Lakukan pekerjaan (menulis kode atau dokumen). Setelah selesai, simpan perubahan secara lokal (git add. dan git commit -m "Pesan commit yang deskriptif"). Penting untuk melakukan commit secara teratur.<sup>18</sup>
- **Kirim ke Repositori Pusat:** Kirim perubahan dari cabang lokal ke repositori pusat dengan git push origin <nama-cabang>.
- **Ajukan Pull Request:** Setelah tugas selesai dan siap untuk diintegrasikan, ajukan Pull Request (PR) ke cabang utama. Ini adalah mekanisme untuk meminta anggota tim lain meninjau kode sebelum digabungkan.<sup>17</sup>
- **Gabungkan Perubahan:** Setelah PR disetujui oleh anggota tim, gabungkan (merge) perubahan ke cabang utama (main). Setelah itu, cabang fitur bisa dihapus untuk menjaga kerapian repositori.<sup>17</sup>

### Struktur Folder Repositori yang Disarankan:

Untuk memastikan navigasi yang mudah dan kolaborasi yang efisien, tim harus mengikuti struktur folder yang konsisten.<sup>19</sup>

```
.
├── .gitignore
├── README.md
├── docs/
│   ├── deliverable-minggu-4/
│   │   ├── project-charter.pdf
│   │   └── project-plan.docx
│   ├── deliverable-minggu-6/
│   │   └── dokumen-analisis-sistem.pdf
│   ├── deliverable-minggu-8/
│   │   └── dokumen-perancangan-sistem.pdf
│   ├── deliverable-minggu-14/
│   │   └── dokumen-uji-sistem.pdf
│   └── final-project-report.pdf
├── src/
└── assets/
```



### Penjelasan Struktur Folder:

- **\*\*README.md\*\***: Ini adalah file pertama yang dilihat orang ketika membuka repositori. Gunakan file ini untuk menjelaskan tujuan proyek, petunjuk cara menjalankan aplikasi, dan informasi kontak tim. File ini mendukung format Markdown untuk deskripsi yang jelas dan rapi.<sup>16</sup>
- **\*\*docs/\*\***: Folder ini khusus untuk menyimpan semua dokumen proyek, termasuk laporan, presentasi, dan deliverable non-kode lainnya.<sup>21</sup> Setiap deliverable mingguan sebaiknya ditempatkan dalam sub-folder tersendiri (misalnya, deliverable-minggu-4) untuk mempermudah pengawasan dan penilaian.
- **\*\*src/\*\***: Folder ini adalah tempat semua kode sumber aplikasi berada.
  - **assets/**: Berisi semua aset statis seperti gambar, font, dan stylesheet.<sup>19</sup>
  - **pages/** atau **views/**: Berisi kode untuk setiap halaman atau tampilan utama aplikasi (misalnya, halaman utama, halaman profil).<sup>20</sup>
  - **components/**: Berisi komponen-komponen UI yang bisa digunakan kembali (misalnya, tombol, header, footer).<sup>19</sup>

### Aturan Penilaian Berbasis Repositori:

Dosen akan melakukan monitoring mingguan dengan meninjau langsung repositori tim. Kepatuhan terhadap panduan ini akan menjadi salah satu kriteria penilaian. Dosen akan mengukur kemajuan proyek (Project Plan vs. commit history), meninjau kualitas kode, dan memeriksa kelengkapan dokumen yang diunggah. Deliverable yang terlambat atau tidak sesuai dengan struktur yang telah ditetapkan akan memengaruhi penilaian pada komponen kinerja tim.

## BAB IV: Sistem Penilaian Komprehensif: Rubrik dan Kriteria Objektif

## 4.1 Filosofi Penilaian Holistik: Proses dan Produk

Sistem penilaian untuk RPS terintegrasi ini dirancang untuk mengevaluasi tidak hanya kualitas produk akhir, tetapi juga kualitas proses kerja yang dijalani oleh mahasiswa. Ini adalah respons terhadap tantangan dalam penilaian proyek, di mana penting untuk menghindari subjektivitas dan memastikan bahwa kriteria penilaian jelas dan terukur. Dengan mengukur proses dan produk secara terpisah, penilaian dapat memberikan gambaran yang lebih akurat tentang kompetensi yang dikuasai mahasiswa, yang mencakup keterampilan teknis, manajerial, dan kolaboratif.

Penilaian akan menggunakan rubrik komprehensif sebagai instrumen utama. Rubrik ini akan mendefinisikan kriteria dan indikator keberhasilan yang spesifik untuk setiap deliverable proyek. Hal yang paling penting dari rubrik ini adalah kemampuannya untuk menilai konsistensi dan keterkaitan antara fase-fase proyek. Sebagai contoh, kualitas source code (deliverable dari fase Pengembangan) akan dinilai tidak hanya dari segi fungsionalitasnya, tetapi juga dari kesesuaiannya dengan Dokumen Analisis Sistem (deliverable dari fase Perancangan). Inkonsistensi antara deliverable dari satu fase ke fase berikutnya akan memengaruhi nilai di kedua komponen penilaian, sehingga secara langsung memvalidasi pentingnya integrasi kurikulum.

## 4.2 Komponen dan Bobot Penilaian

Untuk mencerminkan bahwa proyek ini menggabungkan tiga mata kuliah yang berbeda, sistem penilaian akan memisahkan bobot untuk setiap mata kuliah secara individual. Setiap mata kuliah akan memiliki bobot penilaian 100% yang dihitung dari kumpulan deliverable dan aspek terkait. Proporsi ini memastikan bahwa mahasiswa mendapatkan nilai yang adil untuk setiap mata kuliah yang diajarkan, dan juga memastikan bahwa mereka menginvestasikan upaya yang setara dalam perencanaan, perancangan, dan pengembangan.

### A. Bobot Penilaian Mata Kuliah: Manajemen Proyek (Total 100%)

- **Deliverables Perencanaan Proyek (40%):** Penilaian didasarkan pada kualitas Project Charter (15%) dan kelengkapan Project Plan yang mencakup Work Breakdown Structure (WBS) dan jadwal yang realistis (25%). Kriteria evaluasi akan mencakup efektivitas manajemen waktu dan risiko yang diidentifikasi.
- **Dokumentasi dan Laporan Akhir (40%):** Penilaian berfokus pada kelengkapan dan kualitas Laporan Akhir Proyek yang merangkum seluruh proses proyek, dari inisiasi hingga penutupan. Laporan harus mendokumentasikan pencapaian, masalah yang dihadapi, dan pembelajaran yang didapat.
- **Kinerja Tim & Presentasi Akhir (20%):** Penilaian mencakup aspek kolaborasi dan komunikasi dalam tim (10%) serta kemampuan untuk mempresentasikan deliverable manajemen proyek (10%) di hadapan penguji.

## B. Bobot Penilaian Mata Kuliah: Analisa dan Perancangan Sistem (Total 100%)

- **Deliverables Analisis Kebutuhan (50%):** Penilaian didasarkan pada kualitas Dokumen Analisis Sistem yang berisi Use Case Diagram, Functional Requirements, dan Non-Functional Requirements. Kriteria ini akan mengukur kemampuan mahasiswa dalam menerjemahkan kebutuhan bisnis ke dalam solusi teknis yang terstruktur.
- **Deliverables Perancangan Sistem (40%):** Penilaian berfokus pada kualitas Dokumen Perancangan Sistem, termasuk model data (ERD) dan perancangan antarmuka pengguna (UI/UX).
- **Kinerja Tim & Presentasi Akhir (10%):** Penilaian mencakup kontribusi individu dan kemampuan untuk mempresentasikan serta mempertahankan deliverable analisis dan perancangan di hadapan penguji.

## C. Bobot Penilaian Mata Kuliah: Pengembangan Aplikasi Web (Total 100%)

- **Kualitas Source Code (40%):** Penilaian berfokus pada struktur dan keterbacaan kode, penggunaan praktik terbaik (best practices), serta dokumentasi teknis yang menyertainya.
- **Fungsionalitas & Pengujian Aplikasi (40%):** Penilaian didasarkan pada fungsionalitas aplikasi yang dihasilkan dan kelengkapan Dokumen Uji Sistem (Test Case dan laporan hasil pengujian validasi, integrasi, dan kompatibilitas).
- **Kinerja Tim & Presentasi Akhir (20%):** Penilaian mencakup kemampuan kolaborasi tim dalam fase coding (10%) dan demonstrasi aplikasi yang berfungsi penuh di hadapan penguji (10%).

## 4.3 Rubrik Penilaian Komprehensif

Berikut adalah contoh rubrik penilaian yang akan digunakan, yang dirancang untuk mengukur capaian pembelajaran secara objektif di setiap fase proyek.

**Tabel 4: Rubrik Penilaian Mata Kuliah Manajemen Proyek (Total 100%)**

Aspek Penilaian	Kriteria	Indikator Keberhasilan (Skala 1-4)	Bobot
<b>Deliverables Perencanaan Proyek</b>	Kelengkapan Project Charter dan Business Case	1: Dokumen tidak lengkap, tidak terstruktur. 2: Dokumen cukup lengkap, namun kurang terstruktur. 3: Dokumen lengkap dan	15%

		terstruktur. 4: Dokumen sangat lengkap, terstruktur, dan realistis.	
	Efektivitas WBS dan Project Plan	1: WBS tidak dibuat atau tidak relevan. 2: WBS dibuat, tetapi tidak memecah pekerjaan secara memadai. 3: WBS memecah pekerjaan dengan baik, tetapi alokasi tugas kurang jelas. 4: WBS memecah pekerjaan secara detail, logis, dan alokasi tugas jelas.	25%
<b>Dokumentasi &amp; Laporan Akhir</b>	Kelengkapan Final Project Report	1: Laporan tidak lengkap. 2: Laporan cukup lengkap. 3: Laporan lengkap, terstruktur, dan berisi refleksi. 4: Laporan sangat komprehensif dan mendalam.	40%
<b>Kinerja Tim &amp; Presentasi Akhir</b>	Kolaborasi & Komunikasi Tim	1: Tidak ada kolaborasi atau komunikasi antar anggota. 2: Kolaborasi terbatas, sering terjadi miskomunikasi. 3: Kolaborasi dan	10%



		komunikasi berjalan baik. 4: Kolaborasi sangat efektif, mampu menyelesaikan konflik dengan baik.	
	Kualitas Presentasi Manajemen Proyek	1: Presentasi tidak terstruktur. 2: Presentasi cukup memadai. 3: Presentasi jelas dan terstruktur. 4: Presentasi sangat persuasif, meyakinkan, dan profesional.	10%

**Tabel 5: Rubrik Penilaian Mata Kuliah Analisa dan Perancangan Sistem (Total 100%)**

Aspek Penilaian	Kriteria	Indikator Keberhasilan (Skala 1-4)	Bobot
<b>Deliverables Analisis Kebutuhan</b>	Kualitas Analisis Kebutuhan	1: Kebutuhan tidak teridentifikasi. 2: Kebutuhan teridentifikasi sebagian, ada inkonsistensi. 3: Kebutuhan teridentifikasi dengan jelas, konsisten. 4: Kebutuhan teridentifikasi secara komprehensif dan divalidasi.	50%

<b>Deliverables Perancangan Sistem</b>	Kualitas Model Perancangan (ERD, DFD/UML)	1: Model tidak dibuat atau tidak relevan. 2: Model dibuat, tetapi tidak sesuai dengan kebutuhan. 3: Model dibuat dengan benar dan sesuai dengan kebutuhan. 4: Model dibuat dengan benar, terstruktur, dan inovatif.	40%
<b>Kinerja Tim &amp; Presentasi Akhir</b>	Kualitas Presentasi Analisis dan Perancangan	1: Presentasi tidak terstruktur. 2: Presentasi cukup memadai. 3: Presentasi jelas, logis, dan dapat dipertahankan. 4: Presentasi sangat persuasif dan menunjukkan pemahaman mendalam.	10%

**Tabel 6: Rubrik Penilaian Mata Kuliah Pengembangan Aplikasi Web (Total 100%)**

Aspek Penilaian	Kriteria	Indikator Keberhasilan (Skala 1-4)	Bobot
<b>Kualitas Source Code</b>	Struktur, Keterbacaan, dan Dokumentasi Kode	1: Source code tidak terstruktur dan sulit dibaca. 2: Source code cukup terstruktur, namun tidak menggunakan	40%

		<p>praktik terbaik. 3: Source code terstruktur, bersih, dan mengikuti praktik terbaik. 4: Source code sangat efisien, mudah di-maintain, dan didokumentasikan dengan baik.</p>	
<b>Fungsionalitas &amp; Pengujian Aplikasi</b>	Fungsionalitas Aplikasi	<p>1: Aplikasi tidak berfungsi atau banyak bug. 2: Aplikasi berfungsi sebagian, masih ada bug besar. 3: Aplikasi berfungsi sesuai dengan kebutuhan, hanya ada bug minor. 4: Aplikasi berfungsi penuh, stabil, dan sesuai dengan semua kebutuhan.</p>	25%
	Kelengkapan Dokumen Uji Sistem	<p>1: Dokumen tidak dibuat. 2: Dokumen dibuat, namun tidak lengkap. 3: Dokumen lengkap dan berisi test case yang memadai. 4: Dokumen sangat komprehensif, mencakup laporan bug dan solusi.</p>	15%
<b>Kinerja Tim &amp; Presentasi Akhir</b>	Kualitas Demonstrasi	<p>1: Demonstrasi gagal atau tidak</p>	10%

	Aplikasi	menunjukkan fungsionalitas utama. 2: Demonstrasi berhasil, namun ada masalah teknis minor. 3: Demonstrasi berjalan lancar, semua fitur utama berfungsi. 4: Demonstrasi sangat profesional, menampilkan semua fitur dengan mulus.	
	Kualitas README.md & Struktur Folder	1: README.md kosong atau struktur folder tidak konsisten. 2: README.md dasar, struktur folder belum rapi. 3: README.md jelas, struktur folder rapi dan konsisten. 4: README.md komprehensif, struktur folder modular dan efisien.	10%

## BAB V: Kesimpulan dan Rekomendasi Lanjutan

### 5.1 Ringkasan Model Pembelajaran Inovatif

Rancangan RPS terintegrasi ini merupakan respons strategis STMIK Tazkia terhadap tuntutan industri digital dan fleksibilitas regulasi pendidikan tinggi terbaru. Dengan mengadopsi model Integrated Project-Based Learning dan kerangka Hybrid Agile-Waterfall, RPS ini menyatukan

tiga mata kuliah krusial—Manajemen Proyek, Analisa dan Perancangan Sistem, dan Pengembangan Aplikasi Web—ke dalam satu siklus proyek semester yang utuh dan kohesif. Model ini tidak terpaku pada 16 sesi tradisional, melainkan berfokus pada hasil nyata berupa aplikasi web fungsional yang dikembangkan melalui proses yang terstruktur dan terukur. Pendekatan ini secara langsung mereplikasi dinamika kerja di industri, sehingga mahasiswa tidak hanya menguasai teori, tetapi juga memiliki pengalaman praktik yang relevan.

## 5.2 Rekomendasi Implementasi Strategis

Untuk memastikan keberhasilan implementasi model pembelajaran inovatif ini di STMIK Tazkia, beberapa langkah strategis direkomendasikan:

1. **Pengembangan Kapasitas Dosen:** Dosen yang akan mengampu RPS ini harus diberi pelatihan intensif mengenai metodologi Project-Based Learning dan pendekatan Hybrid Agile. Tim dosen harus dibentuk secara kolaboratif, di mana setiap anggota bertindak sebagai ahli di bidangnya (manajemen, analisis, atau pengembangan) untuk membimbing mahasiswa secara holistik.
2. **Penyusunan Repositori Kasus Nyata:** Untuk memastikan pembelajaran berbasis kasus yang efektif, program studi harus membangun repositori kasus nyata dari kebutuhan internal kampus atau bekerja sama dengan mitra industri. Kasus-kasus ini harus bervariasi dan menantang, mencerminkan permasalahan riil di dunia kerja.
3. **Pemanfaatan Platform Digital:** Implementasi RPS ini akan membutuhkan dukungan teknologi untuk kolaborasi dan pengawasan proyek yang tidak terbatas pada ruang kelas. Pemanfaatan sistem manajemen pembelajaran (LMS) atau platform project management seperti Odoo-ERP<sup>9</sup> akan sangat membantu tim dosen dan mahasiswa dalam melacak kemajuan proyek, mengelola deliverables, dan memfasilitasi komunikasi yang efisien.
4. **Mekanisme Evaluasi Berkelanjutan:** Setelah implementasi, program studi perlu menetapkan mekanisme evaluasi berkelanjutan untuk mengukur efektivitas RPS ini. Umpan balik dari mahasiswa, dosen, dan pihak industri harus dikumpulkan untuk terus menyempurnakan model ini di semester-semester berikutnya, memastikan bahwa kurikulum tetap relevan dan menghasilkan lulusan yang kompeten dan siap bersaing.

### Works cited

1. (PDF) SISTEM INFORMASI BERBASIS WEB UNTUK PENILAIAN ..., accessed August 13, 2025, [https://www.researchgate.net/publication/371419236\\_SISTEM\\_INFORMASI\\_BERBASIS\\_WEB\\_UNTUK\\_PENILAIAN\\_TUGAS\\_AKHIR\\_BERBASIS\\_RUBRIK](https://www.researchgate.net/publication/371419236_SISTEM_INFORMASI_BERBASIS_WEB_UNTUK_PENILAIAN_TUGAS_AKHIR_BERBASIS_RUBRIK)
2. Pengembangan Aplikasi Pembelajaran Sistem Informasi Sekolah ..., accessed August 13, 2025, <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/9525/4299>
3. PANDUAN MERDEKA BELAJAR KAMPUS MERDEKA - FBP UNAS, accessed August 13, 2025, <https://fbp.unas.ac.id/wp-content/uploads/2024/01/BUKU-PANDUAN-MBKM-UNA>

[S.pdf](#)

4. RPS matakuliah Manajemen Proyek Sistem Informasi | | Universitas Negeri Surabaya, accessed August 13, 2025,  
<https://sindig.unesa.ac.id/rps-pdf/s1-sistem-informasi/manajemen-proyek-sistem-informasi.pdf>
5. Course: PENGEMBANGAN APLIKASI BERBASIS WEB | LMS-SPADA INDONESIA, accessed August 13, 2025,  
<https://lmsspada.kemdiktisaintek.go.id/course/view.php?id=3641>
6. RPS - Analisa Perancangan Sistem (APS).docx - Fikom UMI, accessed August 13, 2025,  
<https://fikom.umi.ac.id/wp-content/uploads/2024/02/RPS-Analisa-Perancangan-Sistem-APS.docx.pdf>
7. POLITEKNIK NEGERI MEDAN JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK, accessed August 13, 2025,  
<https://trpl.polmed.ac.id/wp-content/uploads/sites/21/2022/08/RPS-Analisis-Perancangan-Sistem.pdf>
8. (PDF) SOFTWARE DEVELOPMENT PROJECT MANAGEMENT ..., accessed August 13, 2025,  
[https://www.researchgate.net/publication/373607080\\_SOFTWARE\\_DEVELOPMENT\\_PROJECT\\_MANAGEMENT\\_BASED\\_ON\\_WORK\\_BREAKDOWN\\_STRUCTURE\\_AND\\_ODOO\\_ERP](https://www.researchgate.net/publication/373607080_SOFTWARE_DEVELOPMENT_PROJECT_MANAGEMENT_BASED_ON_WORK_BREAKDOWN_STRUCTURE_AND_ODOO_ERP)
9. Program Studi Sistem Informasi Buku Pedoman ... - FTI UAJY, accessed August 13, 2025,  
<https://fti.uajy.ac.id/sisteminformasi/wp-content/uploads/2024/02/Pedoman-Skripsi-2024.pdf>
10. Alur & Perangkat Pengembangan Aplikasi Web | PDF - Scribd, accessed August 13, 2025,  
<https://id.scribd.com/presentation/431971045/Alur-Perangkat-Pengembangan-Aplikasi-Web>
11. PERHITUNGAN BARU SKS DAN IPK - Politeknik Pariwisata Prima Internasional, accessed August 17, 2025,  
<https://poltekparprima.ac.id/perhitungan-baru-sks-dan-ipk/>
12. jdih.kemdikbud.go.id SALINAN PERATURAN MENTERI PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI REPUBLIK INDONESIA NOMOR 53 TAHUN 20 - Silemkerma, accessed August 17, 2025,  
[https://silemkerma.kemdiktisaintek.go.id/assets/panduan/peraturanperundangan/Permen\\_53\\_2023.pdf](https://silemkerma.kemdiktisaintek.go.id/assets/panduan/peraturanperundangan/Permen_53_2023.pdf)
13. Pengembangan Sistem Aplikasi Manajemen Proyek menggunakan Kanban Framework, accessed August 13, 2025,  
<https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6533>
14. Sistem Informasi (S1) - STMIK Tazkia, accessed August 13, 2025,  
<https://stmik.tazkia.ac.id/program-studi/sistem-informasi>
15. Permendikbudriset No. 53 Tahun 2023 - Peraturan BPK, accessed August 13, 2025,

- <https://peraturan.bpk.go.id/Details/265158/permendikbudriset-no-53-tahun-2023>
16. Best practices for Projects - GitHub Docs, accessed August 17, 2025,  
<https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/best-practices-for-projects>
  17. GitHub Best Practices, accessed August 17, 2025,  
<https://widdowquinn.github.io/github-best-practice/>
  18. Using Git and GitHub for Teams by Michelle Mannering - GitKraken, accessed August 17, 2025, <https://www.gitkraken.com/gitkon/github-for-teams>
  19. Folder structure for modern web application | by Michael Gichuhi - Medium, accessed August 17, 2025,  
<https://medium.com/@gmikekamau/folder-structure-for-modern-web-application-6dc0bd1b0e5d>
  20. How to Structure Files and Folder in your Project? - GeeksforGeeks, accessed August 17, 2025,  
<https://www.geeksforgeeks.org/javascript/file-and-folder-organization-best-practices-for-web-development/>
  21. Project Documentation for the students also | PPTX - SlideShare, accessed August 17, 2025,  
<https://www.slideshare.net/sairaazeem3/project-documentation-for-the-students-also>
  22. How to Create Project Documentation with Examples? - GeeksforGeeks, accessed August 17, 2025,  
<https://www.geeksforgeeks.org/business-studies/how-to-create-project-documentation-with-examples/>