RENCANA PEMBELAJARAN *IMMERSION PROGRAM*Logo

Description automatically generated

UNIVERSITAS PIGNATELLI TRIPUTRA

Informasi Mahasiswa

| Nama |  |
| --- | --- |
| NIM |  |
| Email |  |
| Telp./HP |  |
| Alamat |  |
| Program Studi |  |
| Fakultas |  |
| Semester |  |

Informasi Perusahaan/Komunitas/Lembaga

| Nama |  |
| --- | --- |
| Alamat |  |
| Nama Pengawas/Mentor |  |
| Posisi/Jabatan |  |
| Email |  |
| Nomor Telp. Kantor |  |
| Nomor HP Pengawas/Mentor |  |

Informasi *Immersion Program*

| Skema *Immersion*\* |  |
| --- | --- |
| Tanggal Mulai |  |
| Tanggal Selesai |  |
| Jam Operasional |  |
| Posisi Pekerjaan |  |

\*pilih salah satu

Rencana Belajar

Mahasiswa akan berpengalaman dalam satu atau lebih bidang kompetensi berikut :

| **No** | **Mata Kuliah\*\*\*** | **CPMK** | **Aktivitas Magang** | **Masukan Mitra/Tambahan Aktivitas/Evaluasi** |
| --- | --- | --- | --- | --- |
| 1 | *Professional Soft Skills* | Mampu menginternalisasi sikap rendah hati, serta berkomitmen untuk terus memperbaiki diri demi peningkatan kualitas diri dan profesi. | 1. Melakukan diskusi konstruktif dengan tim (front-end, QA, PM). 2. Menerima dan menerapkan feedback mentor terkait kualitas kode & kerja sama. 3. Menyusun catatan evaluasi diri mingguan (refleksi kelebihan & kekurangan pribadi). 4. Menunjukkan sikap disiplin waktu, tanggung jawab, serta menjaga integritas data. 5. Aktif berbagi pengetahuan kecil (misalnya best practice penggunaan Git) dengan tim. |  |
| 2 | *Software Engineering Environment* | Mampu memahami konsep teoritis bidang pengetahuan rekayasa perangkat lunak yang relevan dengan kebutuhan industri dan masyarakat | 1. Mengkaji teori arsitektur perangkat lunak (monolith vs microservices) dan menghubungkannya dengan proyek yang sedang dikerjakan. 2. Menjelaskan kembali kebutuhan fungsional & non-fungsional dalam konteks industri mitra. 3. Mendeskripsikan perbedaan database relasional vs non-relasional berdasarkan literatur dan kebutuhan aplikasi. 4. Menganalisis teori API (REST, GraphQL) serta standar keamanannya (JWT, OAuth) dari sisi konsep. 5. Mengaitkan design pattern (MVC, Repository, Singleton, Factory) dengan skenario kasus yang ditemui di proyek. |  |
| Mampu menganalisis informasi/data untuk pengambilan keputusan | 1. Memahami teori teknik analisis data sistem (misalnya analisis log, analisis performansi API, analisis query database). 2. Menjelaskan konsep metrics evaluasi sistem (response time, throughput, error rate, resource utilization). 3. Menggunakan pengetahuan tentang optimasi basis data (indexing, normalisasi, denormalisasi) untuk memberikan rekomendasi berbasis konsep. 4. Mendeskripsikan perbedaan strategi deployment (vertical scaling vs horizontal scaling) untuk mendukung pengambilan keputusan teknis. 5. Menyusun laporan analisis data berbasis teori untuk mendukung keputusan proyek. |  |
| 3 | *Software Engineering Solution* | Mampu mengimplementasikan proses rekayasa perangkat lunak secara efektif dan efisien | 1. Menggunakan Git & Git Flow untuk versioning kolaboratif. 2. Membuat branch, pull request, code review di GitHub/GitLab. 3. Menulis unit testing & integration testing untuk fungsi back-end. 4. Menggunakan tools CI/CD (misal Jenkins, GitHub Actions) untuk deployment otomatis. 5. Menulis dokumentasi teknis |  |
| Mampu mengidetifikasi permasalahan computing serta memberikan solusi berbasis perangkat lunak | 1. Menganalisis bottleneck query & memberikan optimasi database (indexing, caching). 2. Memperbaiki bug & error handling. 3. Menyusun solusi untuk integrasi jika terhubung dengan layanan eksternal 4. Mengoptimalkan waktu response API dan konsumsi memori server. 5. Menyusun rencana mitigasi error (retry mechanism, logging system, exception handling). |  |
| 4 | *Software Engineering Practices* | Mampu menerapkan metode komputasi, rekayasa, dan prinsip pengembangan perangkat lunak dalam kegiatan praktik kerja di lingkungan industri | 1. Membuat arsitektur modul back-end sesuai standar proyek. 2. Mengembangkan API untuk modul inti (autentikasi, user management, dll). 3. Menerapkan security practices seperti akses kontrol, dll. |  |
|  | Mampu merencanakan dan menyelesaikan tugas pengembangan perangkat lunak sesuai batasan proyek dengan pendekatan rekayasa perangkat lunak secara sistematis | 1. Menyusun rencana sprint/timeline pengembangan back-end. 2. Membuat task breakdown (backlog) untuk fitur back-end. 3. Menyelesaikan modul back-end sesuai estimasi waktu & dokumentasi teknis. 4. Melaporkan progres ke project manager/mentor setiap minggu. |  |
|  | Mampu mengimplementasikan solusi perangkat lunak secara sistematis dan terukur sesuai kebutuhan industri mitra | 1. Melakukan deployment back-end ke staging & production server. 2. Mengintegrasikan back-end dengan front-end & service eksternal. 3. Melakukan stress testing & performance monitoring pada sistem back-end. 4. Membuat laporan akhir (arsitektur sistem, dokumentasi API, manual deployment). 5. Menyajikan demo sistem kepada mitra industri sebagai deliverable akhir. |  |

\*\*\*Program studi WAJIB menyebutkan Mata kuliah, CPMK, Rencana Pembelajaran untuk pelaksanaan Immersion Program sesuai yang tercantum dalam dokumen kurikulum.

Hasil Kerja

Mahasiswa akan menyerahkan dokumen berikut yang direncanakan untuk mencapai hasil pembelajaran:

• Laporan bulanan

• Hasil (Dalam proses penempatan, bagian ini harus diisi oleh Dosen Pembimbing dengan jelas untuk menunjukkan bahwa proses Skema Immersion Program telah selesai dan hasil pembelajaran diperoleh dengan menghasilkan hasil ini):

o Laporan Akhir

Disetujui oleh:

| Mahasiswa | Mentor/Supervisor | Dosen Pembimbing Program |
| --- | --- | --- |

| Proposed by | Dikoreksi oleh : | Diterima oleh |
| --- | --- | --- |
| Penanggung Jawab Immersion Program | Kaprodi S1 Rekayasa Perangkat Lunak | Drs. Dylmoon Hidayat, M.S., M.A., Ph.D.  Dekan Fakultas Sains dan Teknologi |
| Neil Semuel Rupidara, S.E., [M.Sc](http://m.sc)., Ph.D.  Wakil Rektor I |