**FR.IA.02. TPD - TUGAS PRAKTIK DEMONSTRASI**

| Skema Sertifikasi  (KKNI/Okupasi/Klaster) | Judul | : | PEMROGRAM (PROGRAMMER) |
| --- | --- | --- | --- |
| Nomor | : | SS-001-TI.LSP.Poliban.01 |
| TUK | | : | Sewaktu / Tempat Kerja / Mandiri\* |
| Nama Asesor | | : | Rahimi Fitri, S.Kom., M.Kom. |
| Nama Asesi | | : | Muhammad Hasan Akbar |
| Tanggal | | : | 02 Juni 2024 |

\*Coret yang tidak perlu

**A. Petunjuk**

1. Baca dan pelajari setiap instruksi kerja di bawah ini dengan cermat sebelum melaksanakan praktek

2. Klarifikasi kepada asesor kompetensi apabila ada hal-hal yang belum jelas

3. Laksanakan pekerjaan sesuai dengan urutan proses yang sudah ditetapkan

4. Seluruh proses kerja mengacu kepada SOP/WI yang dipersyaratkan (Jika Ada)

**B. Skenario Tugas Praktik Demonstrasi**

| **Kelompok**  **Pekerjaan 1**  **Melakukan**  **Perancangan** | **No.** | **Kode Unit** | **Judul Unit** |
| --- | --- | --- | --- |
| 1. | J.620100.009.01 | Menggunakan Spesifikasi Program |

**Skenario Tugas Praktik Demonstrasi :**

| **B.1 Pendahuluan**  Anda sebagai **PROGRAMMER** diminta untuk membuat sebuah aplikasi untuk “Pengelolaan Surat Masuk dan Surat Keluar”. Aplikasi tersebut untuk mengatasi permasalahan pengarsipan surat menyurat yang tidak terorganisir dengan baik. Permasalahan tersebut dapat dilihat sebagai berikut :  ● Penomoran Nomor Surat Keluar yang duplikat  ● Salinan fisik Surat Masuk dan Surat Keluar terkadang hilang  ● Perihal surat terdapat perbedaan pada isi surat  ● Waktu dan tempat pada surat yang keliru  ● Penerima surat terdapat perbedaan dengan isi surat  ● Pencarian surat dengan membuka stopmap / bundle  **B.2 Aktifitas Pengelolaan Sistem yang Dibuat**  Aktifitas yang berkaitan dengan pengelolaan program Surat Masuk dan Surat Keluar adalah sebagai berikut 1. Terdapat menu “Surat Masuk”, “Surat Keluar”, dan “*Logout*” pada awal tampilan 2. Terdapat algoritma pengamanan program berupa *login* sistem, ketika pertama kali melakukan akses program  3. Pengelolaan Surat Masuk dan Surat Keluar hanya memiliki 1 akun, yaitu administrator 4. **Unit Penerbitan Surat** dapat melalui :  ● Institusi,  ● Jurusan Teknik Sipil,  ● Jurusan Teknik Mesin, |
| --- |

| ● Jurusan Teknik Elektro,  ● Jurusan Akuntansi,  ● Jurusan Administrasi Bisnis.  0. **Pengesahan surat** dapat melalui :  ● Direktur,  ● Ketua Jurusan Teknik Sipil,  ● Ketua Jurusan Teknik Mesin,  ● Ketua Jurusan Teknik Elektro,  ● Ketua Jurusan Akuntansi,  ● dan Ketua Jurusan Administrasi Bisnis.  0. Surat Masuk dan Surat Keluar **wajib tersimpan** disistem berupa :  ● Nomor Surat,  ● Pengirim Surat,  ● Waktu Surat,  ● Lampiran Surat,  ● Perihal Surat,  ● Penerima Surat,  ● Isi Surat,  ● Unit Penerbit,  ● Tempat Surat,  ● Nama yang Mengesahkan,  ● Nama Tembusan.  0. Setiap **Surat Masuk** yang diterbitkan dan **Surat Keluar** yang diterima tersimpan disistem dan dapat dikelola dengan fitur pembaharuan data dan penghapusan data.  0. Terdapat fitur **Pecarian** data berdasarkan nomor surat pada  0. Terdapat fitur **Cetak** pada surat masuk dan surat keluar |
| --- |
| **INSTRUKSI**  Berdasarkan skenario tersebut, maka tentukanlah Spesifikasi Perangkat Lunak (SKPL) yang akan dibangun.  **Perlengkapan dan Peralatan :** 1. Alat Bantu Spesifikasi Perangkat Lunak (SKPL) Sederhana **Durasi Waktu** : 120 Menit |

| **Alat Bantu Spesifikasi Perangkat Lunak (SKPL) Sederhana** | | |
| --- | --- | --- |
| 1 | Nama Proyek | : Aplikasi Surat Menyurat Sederhana |
| 2 | Basis Proyek\* | : **Web** / Desktop / Mobile |
| 3 | Metode Pengembangan Sistem\* | : Waterfall / Prototipe / Agile / Lainnya |
| 4 | Identifikasi Dasar |  |
| 4.a | Framework\* | : **Native** / Codeigniter / Laravel / Lainnya |
| 4.b | Jenis *Database Management System (DBMS)\** | : **MySQL** / PostgreSQL / NoSQL / Lainnya |
| 4.c | Engine Database\* | : InnoDB / MyIsam / Lainnya |
| 4.d | Aplikasi Web Server\* | : WAMPP / **XAMPP** / Laragon / Bitnami / Lainnya |
| 4.e | Aplikasi Code Editor\* | : **Visual Studio Code** / Sublime / Netbeans / Android Studio / Lainnya |
| 5 | Kebutuhan Fungsional | : Menambah Surat Masuk |
|  |  | : Menambah Surat Keluar |
|  |  | : Mengedit Surat Masuk |
|  |  | : Mengedit Surat Keluar |
|  |  | : Hapus Surat Masuk |
|  |  | : Hapus Surat Keluar |
|  |  | : Cetak Surat Masuk |
|  |  | : Cetak Surat Masuk |
|  |  | : Menampilkan Surat Masuk |
|  |  | : Menampilkan Surat Keluar |
|  |  | : Pencarian |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
| 6 | Kebutuhan Non-Fungsional | : Login |
|  |  | : Logout |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |
|  |  | : |

\*Diberikan tanda lingkaran

| 7. Usecase |
| --- |
|  |

| **Kelompok**  **Pekerjaan 2**  **Menulis Kode Sumber** | **No.** | **Kode Unit** | **Judul Unit** |
| --- | --- | --- | --- |
| 1. | J.620100.010.01 | Menulis kode dengan prinsip sesuai guidelines dan best practices |
| 2. | J.620100.017.02 | Mengimplementasikan Pemrograman Terstruktur |
| 3. | J.620100.018.02 | Mengimplementasikan Pemrograman Berorientasi Objek |
| 4. | J.620100.019.02 | Menggunakan library atau komponen pre existing |
| 5. | J.620100.021.02 | Menerapkan akses basis data |
| 6. | J.620100.023.02 | Membuat Dokumen Kode Program |
| 7. | J.620100.025.02 | Melakukan Debugging |

**Skenario Tugas Praktik Demonstrasi :**

| Pembuatan aplikasi dibuat sesuai dengan kenyamanan **PROGRAMMER**. Aplikasi tersebut perlu dibangun sesuai dengan SKPL sebelumnya. Silahkan ikuti ketentuan berikut dalam pembuatan aplikasinya : 1. Buatlah **workspace** sesuai dengan format yang ditentukan, yaitu : **NamaAsesi\_NamaTUK\_TglUjian** 2. Buatlah **basisdata** sesuai dengan format ditentukan yaitu : **dbNamaTuk\_NamaAsesi** 3. Buatlah **tabel** sesuai dengan persyaratan studi kasus dan perencanaan yang telah disusun 4. Buatlah **class atau prosedur koneksi** agar aplikasi bisa mengakses basis data  5. Buatlah ***interface*** sesuai dengan SKPL yang dibuat (Boleh mengikuti Panduan *Interface* yang disediakan) 6. Jika asesi sudah menyelesaikan pembuatan aplikasi, asesi menjalankan aplikasi yang dibuat dan menunjukkannya kepada asesor  7. **Simpanlah** setiap tampilan hasil eksekusi/running berupa screen shoot/print screen dan disimpan pada folder yang sama dengan seluruh pekerjaan dengan nama tampilan1, tampilan2 dst., selanjutnya edit dan susun kedalam editor aplikasi pengolah kata simpan dengan nama: **NamaAnda\_NamaTUK\_TglUjian.docx**  8. Simpanlah dan export **basisdata** yang telah selesai anda gunakan dengan nama **dbNamaTuk\_NamaAsesi** dengan format ekstensi \*.sql (bagi SQL-based)  9. Asesi meng-kompres folder workspace project beserta dokumennya dan selanjutnya menggunggah ke server atau elearning sesuai arahan asesor |
| --- |
| **INSTRUKSI**  Berdasarkan skenario tersebut, maka bangunlah aplikasinya berdasarkan SKPL yang dibuat pada tahapan sebelumnya.  **Perlengkapan dan Peralatan :** 1. Panduan *Interface*  2. SOP *Coding Guidelines*  **Durasi Waktu** : 360 Menit |

| **Panduan *Interface*** |
| --- |
| **Bukti Data Fisik**    Gambar 1. Contoh Surat  **Layout Tampilan**    Gambar 2. Tampilan Login |

| Gambar 3. Tampilan Dashboard / Home  Gambar 4. Tampilan Surat Masuk |
| --- |

| Gambar 5. Tampilan Surat Keluar  Gambar 6. Tampilan Input Data |
| --- |

| **SOP Coding Guidelines** |
| --- |
| Standar dan Pedoman Pengkodean :  Organisasi pengembangan perangkat lunak yang baik ingin pemrogramnya mempertahankan gaya pengkodean yang terdefinisi dengan baik dan terstandar yang disebut standar pengkodean. Sangat penting bagi pemrogram untuk menjaga standar pengkodean. Jika tidak, kode tersebut akan ditolak selama peninjauan kode. Tujuan utama dari fase pengkodean adalah membuat kode dari dokumen desain yang disiapkan setelah fase desain melalui bahasa tingkat tinggi dan kemudian menguji unit kode ini. Lima pilar kualitas kode pada SOP ini yaitu Keterbacaan, Pemeliharaan, Efisiensi, Keandalan, dan Dapat digunakan kembali. Beberapa standar pengkodean diberikan di bawah ini:  1. Penggunaan global yang terbatas :  Menjelaskan tentang tipe data mana yang dapat dinyatakan global dan data mana yang tidak. 2. Header standar untuk modul yang berbeda:  Untuk pemahaman dan pemeliharaan kode yang lebih baik, header modul yang berbeda harus mengikuti beberapa format dan informasi standar.  ● Nama modul,  ● Tanggal pembuatan modul,  ● Penulis modul,  ● Sejarah modifikasi,  ● Sinopsis modul tentang apa yang dilakukan modul,  ● Berbagai fungsi didukung dalam modul beserta parameter input-outputnya,  ● Variabel global diakses atau dimodifikasi oleh modul  3. Konvensi penamaan untuk variabel lokal, variabel global, konstanta dan fungsi: ● Nama variabel yang bermakna dan mudah dipahami membantu siapa pun memahami alasan penggunaannya,  ● Variabel lokal harus diberi nama menggunakan huruf camel yang dimulai dengan huruf kecil (misalnya localData),  ● sedangkan nama variabel Global atau *class* harus dimulai dengan huruf kapital (misalnya GlobalData),  ● Nama konstanta sebaiknya dibentuk menggunakan huruf kapital saja (misal CONSDATA), ● Sebaiknya hindari penggunaan angka pada nama variabel,  ● Nama fungsi harus ditulis dalam huruf unta yang diawali dengan huruf kecil,  ● Nama fungsi harus menjelaskan alasan penggunaan fungsi tersebut secara jelas dan singkat 4. Indentasi:  Indentasi yang tepat sangat penting untuk meningkatkan keterbacaan kode. Agar kode dapat dibaca, pemrogram harus menggunakan spasi dengan benar.  ● Harus ada spasi setelah pemberian koma di antara dua argumen fungsi.  ● Setiap blok yang disarangkan harus diberi indentasi dan spasi yang benar.  ● Indentasi yang tepat harus ada di awal dan di akhir setiap blok dalam program. ● Semua kurung kurawal harus dimulai dari baris baru dan kode setelah akhir kurung kurawal juga dimulai dari baris baru.  5. Nilai pengembalian kesalahan dan konvensi penanganan pengecualian: Semua fungsi yang mengalami kondisi kesalahan harus mengembalikan 0 atau 1 untuk menyederhanakan proses debug. |

| **Kelompok**  **Pekerjaan 3**  **Melakukan**  **Pengujian**  **Perangkat Lunak** | **No.** | **Kode Unit** | **Judul Unit** |
| --- | --- | --- | --- |
| 1. | J.620100.033.02 | Melaksanakan pengujian unit program |

**Skenario Tugas Praktik Demonstrasi :**

| Diperlukan pengujian berupa blackbox testing berdasarkan kebutuhan fungsional yang dibuat. Blackbox biasanya digunakan sebagai pengecekan akhir pada aplikasi yang dibangun dan memastikan setiap fitur berjalan dengan baik. |
| --- |
| **INSTRUKSI**  Buatlah black box testing berdasarkan kebutuhan fungsional dari SKPL yang dibangun  **Perlengkapan dan Peralatan :** 1. Alat Bantu *Black box testing*  **Durasi Waktu** : 60 Menit |

| **Alat Bantu Black box testing** | | | |
| --- | --- | --- | --- |
| **No** | **Kebutuhan Fungsional** | **Keterangan\*** | |
| **Berhasil** | **Tidak Berhasil** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\* Berikan tanda centang (√) pada kolom keterangan

| **ASESI :** | | |
| --- | --- | --- |
| Nama | : |  |
| Tanda tangan  dan Tanggal | : |  |
| **ASESOR :** | | |
| Nama | : |  |
| No. Reg | : |  |
| Tanda tangan  dan Tanggal | : |  |

**PENYUSUN DAN VALIDATOR**

| **STATUS** | **NO** | **NAMA** | **NOMOR MET** | **TANDA**  **TANGAN**  **DAN**  **TANGGAL** |
| --- | --- | --- | --- | --- |
| PENYUSUN | 1 |  |  |  |
| 2 |  |  |  |
| VALIDATOR | 1 |  |  |  |
| 2 |  |  |  |