

FR.IA.02. TUGAS PRAKTIK DEMONSTRASI

| | | | | | |
|--|-------|---|-----------------------------|--|--|
| Skema Sertifikasi/ Okupasi Nasional | Judul | : | JUNIOR WEB DEVELOPER | | |
| | Nomor | : | FR.SKEMA-02-07 | | |
| TUK | : | <input type="checkbox"/> Sewaktu <input type="checkbox"/> Tempat Kerja <input type="checkbox"/> Mandiri | | | |
| Nama Asesor | : | | | | |
| Nama Asesi | : | | | | |
| Tanggal | : | | | | |

☒ yang perlu

A. Petunjuk

1. Baca dan pelajari setiap instruksi kerja di bawah ini dengan cermat sebelum melaksanakan praktek
2. Klarifikasi kepada asesor 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/IK yang dipersyaratkan
5. Waktu pengerjaan yang disediakan: **180 menit**.

B. Alat dan Bahan

- a. Perangkat komputer dengan spesifikasi minimum
 - CPU Intel Pentium 4 dengan kecepatan 3 GHz atau sederajat
 - Sistem operasi: Microsoft Windows 7, Ubuntu 18.04.3 LTS atau yang sederajat
 - Memori 4 GB
 - Hardisk memiliki tempat kosong 250 GB
 - Resolusi layar 1024 x 768 pixel
- b. *Software tools*:
 - XAMPP (PHP development environment)
 - Text Editor (Sublime Text3/Atom/VS Code/dll.)
 - Diagramming Tools (Visio/Visual Paradigm/Lucid Chart/dll.)
- c. Alat tulis

C. Skenario

| | | | |
|-----------------|------------|---|--|
| Unit Kompetensi | Kode Unit | : | J.620100.005.02 |
| | Judul Unit | : | Mengimplementasikan <i>User Interface</i> |
| | Kode Unit | : | J.620100.010.01 |
| | Judul Unit | : | Menerapkan Perintah Eksekusi Bahasa Pemrograman Berbasis Teks, Grafik, dan Multimedia |
| | Kode Unit | : | J.620100.015.01 |
| | Judul Unit | : | Menyusun Fungsi, File atau Sumber Daya Pemrograman yang Lain dalam Organisasi yang Rapi |
| | Kode Unit | : | J.620100.016.01 |
| | Judul Unit | : | Menulis Kode dengan Prinsip Sesuai <i>Guidelines</i> dan <i>Best Practices</i> |
| | Kode Unit | : | J.620100.017.02 |
| | Judul Unit | : | Mengimplementasikan Pemrograman Terstruktur |
| | Kode Unit | : | J.620100.019.02 |
| | Judul Unit | : | Menggunakan <i>Library</i> atau Komponen <i>Pre- Existing</i> |

Anda seorang yang berprofesi sebagai Web Developer yang diminta oleh perusahaan untuk mengembangkan system pendaftaran beasiswa di sebuah kampus secara online. Dimana ketentuan kampus tersebut mengajukan beasiswa dengan melihat Index Prestasi Kumulatif (IPK) terakhir yang didapat. Adapun petunjuk pengerjaannya sebagai berikut:

1. Pada halaman utama terdiri dari menu yang diminta:

- a. Jenis pilihan beasiswa dan ketentuan syarat beasiswa minimal 2 pilihan misal beasiswa akademik, non akademik, dsb
- b. Daftar Beasiswa sesuai ketentuan (pada butir 2 dibawah ini)
- c. View Hasil beasiswa yg di daftarkan

2. Mahasiswa dapat mendaftarkan diri melalui form dengan ketentuan dalam pengisian form sesuai logika dibawah ini:

Adapun asumsi dan batasan yang perlu diperhatikan dalam menyelesaikan program ini yaitu:

1. Masukkan nama
2. Masukkan Email dimana program bisa mengecek format email
3. Masukkan nomor HP dengan format number saja
4. Semester saat ini bisa dipilih (hanya antara angka 1 sampai 8 yang bisa dipilih) dikarenakan untuk S1 hanya 8 Semester
5. IPK tidak diinput tetapi diasumsikan dapat dari system secara otomatis muncul. Ketika klik registrasi beasiswa, oleh karena itu asumsinya gunakan variable konstanta untuk mendeklarasikan IPK secara otomatis terpanggil misal asumsikan $IPK = 3,4$ atau $IPK = 2,9$
6. Apabila IPK dibawah 3 maka tidak bisa melanjutkan pilihan beasiswa dan elemen pilihan beasiswa, upload berkas serta simpan tidak aktif.
7. Apabila IPK diatas 3 maka secara otomatis kursor berada dipilihan beasiswa.
8. Silahkan upload berkas syarat (dengan asumsi upload apa saja misal pdf/jpg/ zip)
9. Ketika diklik daftar maka akan tampil tampilan isian di Menu hasil, tambahkan elemen "status_ajuan" ketika di daftar untuk menyimpan dengan keterangan isi yaitu "belum di verifikasi"
10. Tampilan hasil silahkan buat dengan ketentuan semua elemen yg diform pendaftaran tampil dan tambahan elemen status ajuan.

Note: Silakan tambahkan asumsi lainnya bila diperlukan tetapi tidak mengurangi ketentuan yang ada diatas!

Tugas-tugas:

Berdasarkan skenario di atas, lakukanlah langkah-langkah kerja berikut:

1. Mengimplementasikan *User Interface*

| No. | Langkah Kerja | Instruksi Kerja |
|-----|--|---|
| 1 | Mengidentifikasi rancangan <i>user interface</i> | 1.1 Identifikasi rancangan <i>user interface</i> sesuai kebutuhan 1.2 Identifikasi komponen <i>user interface dialog</i> sesuai konteks rancangan proses 1.3 Jelaskan urutan dari akses komponen <i>user interface dialog</i> 1.4 Buat simulasi (<i>mock-up</i>) dari aplikasi yang akan dikembangkan |
| 2 | Melakukan implementasi rancangan <i>user interface</i> | 2.1 Tetapkan menu program sesuai dengan rancangan program 2.2 Atur penempatan <i>user interface dialog</i> secara sekuensial 2.3 Sesuaikan setting aktif-pasif komponen <i>user interface dialog</i> dengan urutan alur proses 2.4 Tentukan bentuk <i>style</i> dari komponen <i>user interface</i> 2.5 Jadikan penerapan simulasi suatu proses yang sesungguhnya |

2. Menerapkan Perintah Eksekusi Bahasa Pemrograman Berbasis Teks, Grafik, dan Multimedia

| No. | Langkah Kerja | Instruksi Kerja |
|-----|--|---|
| 1 | Mengidentifikasi mekanisme <i>running</i> atau eksekusi <i>source code</i> | 1.1 Identifikasi cara dan <i>tools</i> untuk mengeksekusi <i>source code</i> 1.2 Identifikasi parameter untuk mengeksekusi <i>source code</i> 1.3 Identifikasi peletakan <i>source code</i> sehingga bisa dieksekusi dengan benar |
| 2 | Mengeksekusi <i>source code</i> | 2.1 Eksekusi <i>source code</i> sesuai dengan mekanisme eksekusi <i>source code</i> dari <i>tools</i> pemrograman yang digunakan 2.2 Identifikasi perbedaan antara <i>running</i> , <i>debugging</i> , atau membuat <i>executable file</i> |
| 3 | Mengidentifikasi hasil eksekusi | 3.1 Eksekusi <i>source code</i> sesuai skenario yang direncanakan 3.2 Identifikasi permasalahan bila eksekusi <i>source code</i> gagal/tidak berhasil |

3. Menyusun Fungsi, File atau Sumber Daya Pemrograman yang Lain dalam Organisasi yang Rapi

| No. | Langkah Kerja | Instruksi Kerja |
|-----|--|---|
| 1 | Mengelola sumber daya pemrograman sesuai karakter | 1.1 Buat nama file, fungsi, variabel, konstanta, dan sumber daya pemrograman lain sesuai konteks 1.2 Lengkapi setiap fungsi/prosedur/program dengan penulisan komentar di awal mengenai deskripsi fungsi/prosedur/program tersebut; <i>initial state</i> dan <i>final state</i> ; <i>author</i> (pembuat); versi dan/atau tanggal 1.3 Lengkapi badan <i>source code</i> dengan komentar/keterangan yang cukup, yang memberikan penjelasan atas baris-baris intruksi |
| 2 | Mengorganisasikan sumber daya pemrograman sesuai konteks | 2.1 Susun <i>folder</i> dan <i>sub-sub folder</i> sesuai konteks dan isinya 2.2 Buat File "readme" yang mengandung penjelasan mengenai struktur/hirarki folder serta penjelasan mengenai sumber daya pemrograman |

4. Menulis Kode dengan Prinsip Sesuai *Guidelines* dan *Best Practices*

| No. | Langkah Kerja | Instruksi Kerja |
|-----|---|---|
| 1 | Menerapkan coding-guidelines dan best practices dalam penulisan program (kode sumber) | 1.1 Tulis kode sumber mengikuti <i>coding-guidelines</i> dan <i>best practices</i> 1.2 Buat Struktur program yang sesuai dengan konsep paradigmanya 1.3 Tangani Galat/error |
| 2 | Menggunakan ukuran performansi dalam menuliskan kode sumber | 2.1 Hitung efisiensi penggunaan <i>resources</i> oleh kode 2.2 Implementasikan kemudahan interaksi sesuai standar yang berlaku |

5. Mengimplementasikan Pemrograman Terstruktur

| No. | Langkah Kerja | Instruksi Kerja |
|-----|--|--|
| 1 | Menggunakan tipe data dan <i>control program</i> | 1.1 Tentukan Tipe data yang sesuai standar 1.2 Gunakan Syntax program yang dikuasai sesuai standar 1.3 Gunakan Struktur kontrol program yang dikuasai sesuai standar |
| 2 | Membuat program sederhana | 2.1 Buat program baca tulis untuk memasukkan data dari keyboard dan menampilkan ke layar monitor termasuk variasinya sesuai standar masukan/keluaran 2.2 Gunakan Struktur kontrol percabangan dan pengulangan dalam membuat program |
| 3. | Membuat program menggunakan prosedur dan fungsi | 3.1 Buat program dengan menggunakan prosedur dibuat sesuai aturan penulisan program 3.2 Buat program dengan menggunakan fungsi sesuai aturan penulisan program 3.3 Buat Program dengan menggunakan prosedur dan fungsi secara bersamaan sesuai aturan penulisan program 3.4 Berikan keterangan untuk setiap prosedur dan fungsi |
| 4. | Membuat program menggunakan array | 4.1 Tentukan dimensi <i>array</i> 4.2 Tentukan tipe data <i>array</i> 4.3 Tentukan panjang <i>array</i> 4.4 Gunakan pengurutan <i>array</i> |
| 5. | Membuat program untuk akses file | 5.1 Buat program untuk menulis data dalam media penyimpan 5.2 Buat program untuk membaca data dari media penyimpan |
| 6. | Mengkompilasi Program | 6.1 Koreksi kesalahan program 6.2 Bebaskan kesalahan <i>syntax</i> dalam program |

6. Menggunakan *Library* atau Komponen *Pre-Existing*

| No. | Langkah Kerja | Instruksi Kerja |
|-----|---|---|
| 1 | Melakukan pemilihan unit-unit <i>reuse</i> yang potensial | 1.1 Identifikasi <i>class unit-unit reuse</i> (dari aplikasi lain) yang sesuai 1.2 Hitung keuntungan efisiensi dari pemanfaatan komponen <i>reuse</i> 1.3 Tidak melanggar Lisensi, Hak cipta dan hak paten dalam pemanfaatan komponen <i>reuse</i> tersebut |
| 2 | Melakukan integrasi <i>library</i> atau komponen <i>pre-existing</i> dengan <i>source code</i> yang ada | 2.1 Identifikasi ketergantungan antar unit 2.2 Hindari penggunaan komponen yang sudah <i>obsolete</i> 2.3 Hubungkan program dengan <i>library</i> |
| 3. | Melakukan pembaharuan <i>library</i> atau komponen <i>pre-existing</i> yang digunakan | 3.1 Identifikasi cara-cara pembaharuan <i>library</i> atau komponen <i>pre-existing</i> 3.2 Lakukan pembaharuan <i>library</i> atau komponen <i>pre-existing</i> |

-- oOo --