|  |  |
| --- | --- |
| **FR.06.B.** | **LEMBAR JAWABAN PERTANYAAN TERTULIS ESAI** |

|  |  |  |  |
| --- | --- | --- | --- |
| Skema Sertifikasi Okupasi Nasional | Judul | : | **Pengembang Web Pratama *(Junior Web Developer)*** |
| Nomor | : | S-2019-016 |
| TUK | | : | Sewaktu/~~Tempat Kerja/Mandiri~~\* |
| Nama Asesor | | : |  |
| Nama Asesi | | : | Muhammad Rillo Nugraha |
| Tanggal | | : | 29 May 2023 |
| Waktu | | : | 30 menit |

\*Coret yang tidak perlu

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Kompetensi | Kode Unit | : | J.620100.005.02 |
| Judul Unit | : | Mengimplementasikan *user interface* |

|  |  |  |
| --- | --- | --- |
| **Jawaban:** | **Rekomendasi** | |
| **K** | **BK** |
| * **Memahami kebutuhan pengguna:** Langkah pertama adalah memahami kebutuhan dan harapan pengguna. Melakukan penelitian pengguna dan pengumpulan data untuk memahami siapa pengguna target, apa tujuan mereka, dan bagaimana mereka akan menggunakan sistem atau aplikasi yang akan dirancang. * **Penentuan tujuan desain:** Berdasarkan pemahaman yang diperoleh tentang pengguna, menentukan tujuan desain yang spesifik. Tujuan ini harus mempertimbangkan keinginan dan kebutuhan pengguna serta mengikuti prinsip-prinsip desain yang baik, seperti kegunaan, konsistensi, kejelasan, dan efisiensi. * **Membuat sketsa kasar (sketching):** Membuat sketsa kasar adalah langkah awal dalam menggambarkan struktur antarmuka pengguna. Ini melibatkan pembuatan sketsa tangan atau menggunakan perangkat lunak desain sederhana untuk mengilustrasikan tata letak elemen-elemen antarmuka dan aliran interaksi secara kasar. * **Pembuatan wireframe:** Setelah sketsa kasar disetujui, membuat wireframe yang lebih rinci. Wireframe adalah representasi visual yang menggambarkan tata letak elemen-elemen antarmuka tanpa desain visual yang terlalu detail. Ini membantu dalam menentukan hierarki informasi, penempatan elemen, dan alur navigasi. * **Desain visual:** Setelah wireframe disetujui, langkah selanjutnya adalah mendesain tampilan visual antarmuka pengguna. Ini mencakup pemilihan warna, tipografi, ikon, dan grafis lainnya yang sesuai dengan identitas merek dan tujuan desain yang telah ditetapkan. * **Implementasi:** Setelah desain antarmuka pengguna selesai, mempersiapkan file dan sumber daya desain yang dibutuhkan untuk mengimplementasikannya dalam pengembangan aplikasi atau sistem. Kerja sama dengan tim pengembang untuk memastikan desain yang diimplementasikan sesuai dengan | ☐ | ☐ |

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Kompetensi | Kode Unit | : | J.620100.010.01 |
| Judul Unit | : | Menerapkan perintah eksekusi bahasa pemrograman  berbasis teks, grafik, dan multimedia |

|  |  |  |
| --- | --- | --- |
| **Jawaban:** | **Rekomendasi** | |
| **K** | **BK** |
| Langkah yang akan saya ambil jika adanya terjadi kegagalan pada source pada saat eksekusi, yaitu :   1. Memeriksa kesalahan seperti beberapa warning error ketika eksekusi program 2. Memeriksa kembali sintaks jika ada kekurangan yang ada seperti : kurangnya ada kurung kurawal atau penulisan variabel yang salah 3. Secara logika kita dapat menganalisis bagaimana alur program ini dapat dieksekusi dari perintah/function yang ada agar dapet berjalan dengan baik 4. Penambahan Debugging Tools, sebagai daya guna plugin tambahan yang dapat memeriksa dan melacak perbaikan masalah dari text editor yang menyediakan fitur tambahan tersebut | ☐ | ☐ |

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Kompetensi | Kode Unit | : | J.620100.015.01 |
| Judul Unit | : | Menyusun fungsi, file atau sumber daya pemrograman yang  lain dalam organisasi yang rapi |

|  |  |  |
| --- | --- | --- |
| **Jawaban:** | **Rekomendasi** | |
| **K** | **BK** |
| 1. Menggunakan penamaan yang jelas dan deskriptif: Memberikan nama yang mudah dimengerti dan deskriptif untuk variabel, fungsi, kelas, dan metode dalam source code.  2. Membuat komentar yang informatif: Menyertakan komentar yang menjelaskan tujuan, logika, atau alur program yang kompleks. Komentar ini membantu developer lain memahami maksud dari setiap bagian source code dan memudahkan pemeliharaan dan pengembangan di masa mendatang.  3. Menggunakan pemisahan logika: Memisahkan logika bisnis, tampilan, dan akses data ke dalam file atau modul yang terpisah.  4. Menghindari kode yang berlebihan atau kompleks: Berusaha untuk menulis kode yang sederhana dan langsung ke inti permasalahan. Menghindari pengulangan atau duplikasi kode yang tidak perlu dan menggunakan konsep pemrograman yang sesuai untuk memecahkan masalah. | ☐ | ☐ |

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Kompetensi | Kode Unit | : | J.620100.016.01 |
| Judul Unit | : | Menulis kode dengan prinsip sesuai *guidelines* dan *best*  *practices* |

|  |  |  |
| --- | --- | --- |
| **Jawaban:** | **Rekomendasi** | |
| **K** | **BK** |
| Saya akan mencari kesalahan tersebut melalui keterangan baris line yang di tampilan dari web browser semisalkan adanya,  udentified variabel atau parse error, warning error, fatal error, dan lain sebagainya  kita dapat mengecek kembali pada text editor memerika kesalahannya tersebut. | ☐ | ☐ |

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Kompetensi | Kode Unit | : | J.620100.017.02 |
| Judul Unit | : | Mengimplementasikan pemrograman terstruktur |

|  |  |  |
| --- | --- | --- |
| **Jawaban:** | **Rekomendasi** | |
| **K** | **BK** |
| Untuk pemrograman php kita dapat memperbaharuinya file.txt dengan klik new file pada project php yang telah di buat lalu pilih icon new file > lalu buatlah file baru tersebut dengan nama Readme dan di sambungkan dengan .txt sehinnga menjadi ( readme.txt ), maka readme akan muncul pada htdocs program tersimpan pada php | ☐ | ☐ |

|  |  |  |  |
| --- | --- | --- | --- |
| Unit Kompetensi | Kode Unit | : | J.620100.019.02 |
| Judul Unit | : | Menggunakan *library* atau komponen *pre-existing* |

|  |  |  |
| --- | --- | --- |
| **Jawaban:** | **Rekomendasi** | |
| **K** | **BK** |
| Identifikasi library yang perlu diperbarui: Tinjau proyek Anda dan identifikasi library yang perlu diperbarui. Biasanya, Anda akan melihat versi library yang lebih baru telah dirilis atau ada masalah keamanan atau bug yang telah diperbaiki dalam versi terbaru.  Tinjau dokumentasi pembaruan: Baca dokumentasi library untuk melihat apa yang berubah dalam versi terbaru. Periksa catatan rilis dan daftar perubahan untuk mengetahui fitur baru, perbaikan bug, atau kemungkinan perubahan yang dapat mempengaruhi kode Anda.  Periksa kompatibilitas: Pastikan bahwa versi library yang baru kompatibel dengan proyek Anda dan tidak akan menyebabkan konflik atau masalah dengan bagian lain dari kode Anda atau library lain yang digunakan. Periksa persyaratan sistem, dependensi, atau perubahan API yang perlu diperhatikan.  Buat cadangan (backup) kode: Sebelum memperbarui library, selalu buat salinan cadangan kode proyek Anda. Jika terjadi masalah selama pembaruan, Anda dapat dengan mudah kembali ke versi sebelumnya. | ☐ | ☐ |

|  |  |  |
| --- | --- | --- |
| **Nama** | **Asesi: Muhammad Rillo Nugraha** | **Asesor:** |
| **Tanda Tangan dan**  **Tanggal** | **Muhammad Rillo Nugraha**  **29 may 2023** |  |