|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **UNIT KOMPETENSI** | **KODE UNIT** | **JAM PELATIHAN** | | |
| PENGE-TAHUAN | KETERAMPILAN | JUMLAH |
| **I.** | **KELOMPOK UNIT KOMPETENSI KEAHLIAN BIDANG MANAJEMEN** |  |  |  |  |
|  | * 1. Menjaga Etika | TIK.PR01. 016.01 | 3 | 4,5 | 7,5 |
|  | **Jumlah I** |  | **3** | **4,5** | **7,5** |
| **II.** | **KELOMPOK UNIT KOMPETENSI INTI** |  |  |  |  |
|  | 2.1. Membuat algoritma pemrograman |  |  |  |  |
| 2.2. Menerapkan pemrograman berorientasi objek |  |  |  |  |
| 2.1 | TIK.PR02. 023.01 | 8 | 8 | 16 |
| 2.3. Membuat dokumen kode program |  |  |  |  |
| **Jumlah II** |  | **8** | **8** | **16** |
| **III.** | **KELOMPOK UNIT KOMPETENSI SPESIALISASI** |  |  |  |  |
|  | 3.1. Menerapkan pemrograman untuk mengakses database |  |  |  |  |
|  | 3.1 Membuat program untuk mengakses perangkat keras | TIK.PR08. 012.01 | 9 | 11,5 | 20,5 |
|  | **Jumlah III** |  | **9** | **11,5** | **20,5** |
|  | **Jumlah I s/d III** |  | **20** | **24** | **44** |
| **IV.** | **PELATIHAN DI TEMPAT KERJA (OJT)** |  |  |  |  |
|  | * 1. Membuat Aplikasi Game Sederhana Berbasis Java | - | - | 5 | 5 |
|  | **Jumlah IV** |  |  | **5** | **5** |
| **V.** | **EVALUASI PROGRAM PELATIHAN** |  |  |  |  |
|  | * 1. Evaluasi pelaksanaan pelatihan | - | - | 2 | 2 |
|  | **Jumlah V** |  |  | **2** | **2** |
|  | **Jumlah I s/d V** |  | **20** | **31** | **51** |

1. **Unit Kompetensi : Membuat Algoritma Pemograman Dasar**

**Kode Unit : TIK.PR02.001.01**

**Durasi Pembelajaran : 13 Jam@60 menit**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Pengetahuan** | **Keterampilan** | **Sikap Kerja** | **Pengetahu-an** | **Kete-rampi-lan** |
| 01 Menjelaskan varian dan invarian | 1.1 Tipe data, variable, konstanta dan parameter yang berlaku umum pada pemrograman dijelaskan.   * 1. Tipe data baru dan tipe data yang ada dibuat dan dijelaksan pemanfaatannya.   2. Pemakaian nama pada tipe data, variable, konstanta dan parameter dijelaskan | * Mengetahui jenis-jenis tipe data umum. * Mengetahui tipe data yang dapat digabung dengan m*modifier*. * Mengetahui pengertian dan jenis-jenis konstanta * Mengetahui pengertian dan jenis-jenis variabel. * Mengetahui pengertian dan jenis-jenis operator relasi. * Mengetahui pengertian dan jenis-jenis *Expression* (ekspresi) | * Mampu menjelaskan tipe data, Variabel, konstanta dan parameter. * Mampu membuat dan menjelaskan tipe data baru dan tipe data yang ada. * Mampu menjelaskan nama pada tipe data, variabel, konstanta dan parameter. | * Memahami dan mengerti tentang penggunaan berbagai macam komponen yang menunjang dalam pemrograman * Mengerti perbedaan tipe data baru dan tipe data yang dibuat | 1 | 1 |
| 02 Membuat alur logika pemrograman | 2.1. Algoritma dengan skema program sekuensial dibuat  2.2. Algoritma dengan skema program percabangan dibuat.  2.3. Algoritma dengan skema program pengulangan dibuat. | * Mengetahui dan mengerti struktur program C. * Mengetahui fungsi dari instruksi *for*. * Mengetahui fungsi Instruksi *while*. * Mengetahui fungsi instruksi *do while* * Mengetahui fungsi instruksi *break and continue*. * Mengetauhi fungsi instruksi *if*. * Mengetahui fungsi instruksi *else clause*. * Mengetahui fungsi instruksi *switch*. | * Mampu membuat algoritma program sekuensial. * Mampu membuat algoritma program percabangan. * Mampu membuat algoritma program pengulangan. | * + Melakukan pembuatan alur program   + Mengerti dan memahami proses alur program | 1,5 | 2 |
| 03.Menerapkan pengelolaan *array* | 3.1 Algoritma dibuat dengan*array* atau dimensi.  3.2. Algoritma pencarian data dan algoritma pengurutan data (*sorting*) dibuat dengan mengguna-kan *array*. | * Mengetahui pengertian dan jenis-jenis *sorting*. * Mengetahui dan mengerti struktur program dan algoritma jenis-jenis *sorting*. | * Mampu membuat algoritma dengan *array* atau dimensi. * Mampu membuat algoritma pencarian data dan algoritma pengurutan (*sorting*) dengan menggunakan *array*. | * + Mengerti penggunaan array   + Mengerti dan mendengar-kan bagaimana melakukan sorting | 1 | 1 |
| 04 Memelihara pekerjaan secara praktis | 4.1 Algoritma program penulisan data dan pembacaan data ke dalam media penyimpanan data berupa hard disk, floppy disk atau lainya dibuat dan ditujukan.  4.2. Algoritma program pembacaan dan penulisan *file* dibuat secara sekuensial dan indeks. | * Mengetahui dan mengerti bagaimana memelihara pekerjaan secara praktis | * Mampu membuat dan menunjukan algoritma program penulisan data dan pembacaan data kedalam media penyimpanan data berupa hardisk, floppy disk atau lainnya. * Mampu membuat algoritma program pembacaan dan penulisan *file* secara sekuensial dan indeks. | * Mengerti dan mendengar-kan berbagai cara dalam penulisan program * Mengerti dan melakukan pembacaan program secara sekuensial dan indeks | 2 | 2 |

* 1. **Unit Kompetensi : Membuat Algoritma Pemograman Lanjut**

**Kode Unit : TIK.PR02.002.01**

**Durasi Pembelajaran : 9 Jam@60 menit**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Pengetahuan** | **Keterampilan** | **Sikap Kerja** | **Pengetahu-an** | **Kete-rampi-lan** |
| 01.Menggunakan *array* multi dimensi | * 1. Algoritma *array* multi dimensi dibuat dan dijelaskan.   2. Matriks 2D, 3D, dibuat dalam bentuk penjumlahan, pengurangan, perkalian, dsb.   3. Algoritma pembacaan, pengisian, pengubahan dan penghapusan data dalam *array* multi dimensi dibuat dan ditunjukkan. | * Mengetahui konsep tentang *array* multi dimensi. * Mengetahui algoritma pengoperasian data dalam *array* multi dimensi. | * Dapat menjelaskan penggunaan *array* multi dimensi. * Dapat membuat algoritma dengan menggunakan *array* multi dimensi. * Dapat membuat kode program dengan menggunakan *array* multi dimensi dan operasinya. | * Membuat kode program menggunakan *array* multi dimensi dan algoritma pengoperasiannya dengan tepat sehingga data yang dihasilkan benar. | 1 | 1,5 |
| 02.Menggunakan prosedur dan fungsi | 2.1 Algoritma program dibuat dengan prosedur.   * 1. Algoritma program dibuat dengan fungsi.   2. Algoritma program dibuat dengan prosedur dan fungsi. | * Mengetahui konsep tentang prosedur dan fungsi. * Mengetahui cara membuat prosedur dan fungsi. * Mengetahui cara membuat prosedur dan fungsi yang digunakan secara bersamaan dalam satu algoritma pemrograman. | * Dapat membuat algoritma pemrograman menggunakan prosedur dan diterapkan ke dalam bahasa pemrograman. * Dapat membuat algoritma pemrograman menggunakan fungsi dan diterapkan ke dalam bahasa pemrograman. * Dapat membuat algoritma pemrograman menggunakan prosedur dan fungsi serta diterapkan ke dalam bahasa pemrograman. | * Mengidentifikasi modul program yang dapat dibuat dengan menggunakan prosedur dan fungsi. * Memanggil dan menggunakan prosedur dan fungsi yang telah dibuat. | 1,5 | 2 |
| 03.Menggunakan *library* pemrograman grafik | 3.1 Algoritma program dibuat dengan *library* berbasis grafik.   1. Algoritma pemrograman dengan grafik (gambar sederhana) berbentuk 2D dan 3D dibuat.    1. Algoritma pemrograman animasi dibuat dengan grafik (gambar sederhana). | * + Mengetahui konsep mengenai *library* berbasis grafik.   + Mengetahui cara membuat grafik dalam algoritma pemrograman.   + Mengetahui cara membuat animasi dengan menggunakan grafik dalam algoritma pemrograman. | * Dapat membuat algoritma yang menggunakan *library* berbasis grafik. * Dapat membuat algoritma menggunakan *array* 2D dan 3D dengan memanfaatkan pemrograman grafik sederhana. * Dapat membuat algoritma berbasiskan grafik untuk membuat animasi sederhana. | * + Taat aturan dalam menggunakan *library* sehingga tidak bersifat *try* dan *error*. | 1,5 | 1,5 |

* 1. **Unit Kompetensi : Mengembangkan User Interface**

**Kode Unit : TIK.PR02.007.01**

**Durasi Pembelajaran : 9 jam**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGE-TAHUAN** | **KETERAM-PILAN** |
| 01 Menerapkan hasil rancangan *user interface* | I.1 Bentuk rancangan *user inteface*, alur proses, data/tabel diidentifikasi.  I.2 Struktur menu dan *user interface dialog* diaplikasikan.  I.3 Komponen *user interface dialog* diidentifikasikan.  I.4 Sekuensial dari akses komponen *user interface dialog* diaplikasikan. | * Mengetahui alur proses, data/tabel yang dibutuhkan untuk merancang *user interface* sesuai dengan spesifikasi pekerjaan. * Mengetahui fungsi, jenis menu, dan kegunaan struktur menu dan *user interface dialog* dalam perancangan *user interface*. * Mengetahui komponen-komponen *user interface dialog*. * Mengetahui sekuensial dari akses komponen *user interface dialog*. | * Dapat menggambarkan dan menjelaskan rancangan *user interface,* alur proses, data/tabel sesuai dengan spesifikasi pekerjaan. * Dapat menggunakan menu dan *user interface dialog*, jika mouse tidak berfungsi. * Dapat menentukan komponen-komponen yang digunakan untuk menyusun suatu *user interface*. * Dapat menunjukkan sekuensial dari akses komponen *user interface dialog*. | * Merancang *user interface*. | 1 | 2 |
| 02 Mengimplementasi-kan rancangan *user interface* | 2.1 Menu program sesuai dengan rancangan program di implementasikan.  2.2 Penempatan *user interface dialog* diatur sesuai sekuensial dari alur proses di implementasikan.  2.3 *Setting* aktif-pasif komponen *user interface* *dialog* disesuaikan dengan urutan alur proses.  2.4 Penentuan atau bentuk *style* dari komponen *user interface* disesuaikan dengan kebutuhan pengguna. | * Mengetahui menu-menu program dan fungsinya yang terdapat pada rancangan *user interface*. * Dapat menjelaskan sekuensial alur proses dalam penempatan *user interface dialog*. * Mengetahui cara dan perbedaan *setting* aktif-pasif komponen *user interface dialog* sesuai urutan alur proses. * Mengetahui bentuk-bentuk *style* dan hubungannya dengan komponen *user interface*. | * Dapat mengimplementasikan menu program sesuai dengan rancangan program. * Dapat mengatur penempatan *user interface dialog* sesuai sekuensial alur proses. * Dapat menyesuaikan *setting* aktif-pasif komponen *user interface dialog* sesuai urutan alur proses. * Dapat menentukan bentuk *style* dari komponen *user interface* sesuai dengan kebutuhan pengguna. | * Mengidentifikasi rancangan *user interface*. | 1,5 | 2 |
| 03 Menguji *user interface* | 3.1 Kode program dikompilasi dan dijalankan.  3.2 Menu dialog dijalankan dan diuji sesuai dengan spesifikasi program.  3.3 *User interface dialog* dijalankan dan diuji sesuai dengan spesifikasi program.  3.4 Hasil pengujian direkam ke dalam catatan pengujian. | * + Mengetahui cara menjalankan dan mengkompilasi kode program.   + Mengetahui cara menguji menu dialog apakah sudah sesuai dengan spesifikasi program.   + Mengetahui cara menguji *user interface dialog* apakah sudah sesuai dengan spesifikasi program.   + Dapat menjelaskan hasil pengujian yang terdapat di catatan pengujian. | * Dapat mengkompilasi dan menjalankan kode program. * Dapat menguji dan menjalankan menu dialog sesuai dengan spesifikasi program. * Dapat menjalankan dan menguji *user interface dialog* sesuai dengan spesifikasi program. * Dapat merekam hasil pengujian ke dalam catatan pengujian. | * + Menyocokkan *user interface* dengan alur proses kebutuhan pelanggan. | 1 | 1,5 |

**Unit Kompetensi : Membuat Dokumen Kode Program**

**Kode Unit : TIK.PR02.019.01**

**Durasi Pembelajaran : 17 Jam@60 menit**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Pengetahuan** | **Keterampilan** | **Sikap Kerja** | **Pengetahu-an** | **Kete-rampi-lan** |
| 1. Mengidentifikasi kode program | * 1. Modul program berdasarkan fungsi, prosedur dan data di identifikasi.   2. Parameter data yang menjadi masukan dan keluaran pada fungsi atau prosedur di ketahui.   3. Data dan algoritma pada deskripsi fungsi dan prosedur di jelaskan. | * Mengetahui program berdasarkan fungsi, prosedur dan data. * Mengetahui parameter data yang menjadi masukan dan keluaran pada fungsi atau prosedur. * Mengenal data dan algoritma. | * Mampu mengidentifikasi modul program berdasarkan fungsi, prosedur dan data. * Mengidentifikasi parameter pada fungsi dan prosedur. * Mampu mengenal data dan algoritma pada fungsi dan prosedur. * Bertanya dan mendengarkan secara aktif dilakukan untuk menambah informasi. | * Membuat modul program sesuai dengan fungsi, prosedur dan data. * Menjelaskan data dan algoritma. | 2 | 1 |
| 1. Membuat dokumentasi modul program | * 1. Dokumentasi modul di buat sesuai dengan identitas untuk memudahkan pelacakan.   2. Nama fungsi dan prosedur yang ada di dalam modul di jelaskan pada dokumentasi modul. | * Mengetahui cara mendokumentasikan modul. * Mengetahui nama fungsi dan prosedur dan dijelaskan pada dokumentasi modul. | * Mampu melakukan dokumentasi modul program. | * Mendokumentasik-an modul program. * Menjelaskan nama fungsi dan prosedur pada modul. | 2 | 2 |
| 1. Membuat dokumentasi fungsi atau prosedur program | * 1. Dokumentasi fungsi dan prosedur sesuai dengan identitas fungsi atau prosedur di buat.   2. Parameter data masukan dan keluaran di jelaskan pada dokumen.   3. Data dan tipe data yang ada fungsi atau prosedur di jelaskan pada dokumen. | * Mengetahui dokumentasi fungsi dan prosedur berdasarkan identitas fungsi atau prosedur. | * Mampu melakukan dokumentasi fungsi atau prosedur program berdasarkan identitas fungsi atau prosedur. | * Membuat dokumentasi fungsi dan prosedur program. * Menjelaskan data masukan dan keluaran. * Menjelaskan tipe data dan data. | 2 | 3 |
| 1. Mengevaluasi dokumen kode program | * 1. Dokumen di revisi berdasarkan standar dokumen.   2. Dokumen di koreksi untuk menjamin kesempurnaan teknis dan tata bahasa.   3. Dokumen di lengkapi untuk memenuhi kebutuhan dokumen teknis.   4. Dokumen di lengkapi untuk memenuhi kebutuhan dokumen teknis. | * Mengetahui cara melakukan revisi dokumen berdasarkan standar dokumen. * Mengetahui cara melakukan koreksi dan melengkapi dokumen sesuai dengan kebutuhan teknis. | * Mampu melakukan evaluasi dokumen kode program. * Mampu melengkapi dan mengoreksi dokumen. | * Melakukan dokumentasi untuk setiap evaluasi kode program. * Melakukan revisi berdasarkan standar | 2 | 3 |

* 1. **Unit Kompetensi : Mengoperasikan Aplikasi Basis Data**

**Kode Unit : TIK.PR02.020.01**

**Durasi Pembelajaran : 16 jam**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGE-TAHUAN** | **KETERA-MPILAN** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 01. Menjelaskan menu aplikasi basis data | * 1. Menu-menu yang tersedia beserta *shortcut*nya ditunjukkan.   2. Fitur-fitur pengolahan *table* digunakan dan diaplikasikan.   3. Fitur-fitur *view* digunakan dan diaplikasikan.   4. Fitur-fitur pengolahan *form* digunakan dan diaplikasikan.   5. Fitur-fitur pengolahan *report* digunakan dan diaplikasikan.   6. Fitur-fitur pengolahan modul digunakan dan diaplikasikan. | * Memahami konsep aplikasi basis data secara umum * Memahami karakteristik aplikasi basis data yang digunakan * Menunjukkan menu-menu dan *shortcut* pada aplikasi basis data. * Memahami dan memanfaatkan fitur pengolahan tabel * Memahami dan memanfaatkan fungsi fitur *view* * Memahami dan memanfaatkan fungsi fitur pengolahan *form* * Memahami dan memanfaatkan fungsi fitur pengolahan *report* * Memahami dan memanfaatkan fungsi fitur pengolahan modul | * Mengeksplorasi karakteristik aplikasi basis data yang digunakan * Mempraktekkan penggunaan fitur-fitur pengolahan tabe, view, form, report, dan pengolahan modul pada aplikasi basis data yang digunakan. | * Memanfaatkan *help* atau referensi lain untuk mengenali karakteristik aplikasi basis data yang digunakan      * Memperhatikan petunjuk manual dalam menggunakan menu dan *shortcut* yang tersedia * Menggunakan pentunjuk dalam pengolahan tabel, *view, form,report* dan modul dengan benar. | 1 | 2 |

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGE-TAHUAN** | **KETERA-MPILAN** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 02. Membuat tabel | * 1. Tabel dibuat dengan pengguna-an *wizard. Wizard table* merupakan fitur aplikasi untuk membantu membuat tabel dengan pemanfaatan fasilitas yang tersedia.   2.2 Tabel dibuat dengan penggunaan fitur pembuat tabel. Fitur pembuat tabel tersedia kolom nama *field*, tipe data dan deskripsi *field.*   * 1. Tabel dibuat dengan penggunaan bahasa data deskripsi (*SQL*)   2. Nama *field*, tipe data dan deskripsi *field* telah ditulis pada kolom.   3. Tabel telah diberi nama dan disimpan sesuai dengan format standar aplikasi yang digunakan. | * Mengetahui konsep penggunaan *wizard*. * Mengetahui teknis pengelolaan (mengisi data pada tabel, menghapus, dan mengubah isi) tabel melalui *wizard* * Mengetahui teknis pengelolaan tabel menggunakan *syntaks* DDL dan DML dalam standar *database* *relational* (SQL), yang digunakan untuk membuat tabel baru, memasukkan data ke dalam tabel, mengubah atau menghapus data pada tabel. | * Membuat tabel menggunakan *wizard* * Membuat tabel menggunakan bahasa SQL jenis *Data Definition Language* (DDL). * Mempraktekkan pengelolaan tabel (*Insert, Update* dan *Delete* data) menggunakan *wizard* * Mempraktekkan pengelolaan tabel (*Insert, Update* dan *Delete* data) menggunakan *Data Manipulation Langauage* (DML) | * Menerapkan *wizard* tabel untuk membantu dalam pembuatan tabel * Taat aturan dalam menerapkan sintak DDL dan DML di lingkungan perangkat lunak basis data. | 2 | 2 |
| 03. Membuat *table* *view (query)* | 3.1 *View* dibuat dengan *wizard view* yang ada, *wizard view* merupakan fitur aplikasi untuk panduan dalam pembuatan *view* dari *table* yang ada dengan pemanfaatan fasilitas yang telah disediakan  3.2 Fitur-fitur *view* dipahami dan diaplikasikan.   * 1. *Field* telah dipilih sesuai dengan tabelnya.   2. *View* telah diberi nama dan disimpan sesuai dengan format standar aplikasi yang digunakan | * Memahami manfaat dan konsekuensi penggunaan *view.* * Membuat *view* menggunakan *wizard* maupunmenggunakanSQL | * Menentukan *field-field* dari tebel yang ada untuk pembuatan *view* * Membuat *view* menggunakan *wizard*  dan SQL * Memberi nama pada *view* yang telah dibuat dengan tepat. | * Menganalisis manfaat dan konsekuensi penggunaan *view* * Mencermati *field*-*field* pada tabel yang akan diakses untuk pembuatan *view* * Mempertimbangkan nama dengan cermat untuk setiap *view* yang dibuat | 1 | 1,5 |
| 04. Membuat *form* basis data | 4.1 *Form* dibuat dengan penggunaan *wizard*. *Wizard form* merupakan fitur aplikasi untuk panduan dalam pembuatan *report* dengan pemanfaatan fasilitas yang tersedia.  4.2 Fitur-fitur *form* diidentifikasi dan diaplikasikan. *Field-field table/view* diatur pada *form*.  4.3 *Form* diberi nama dan disimpan. | * Memahami sekilas konsep penggunaan form dan konsep komunikasi *visual* melalui *form* (pemberian nama, *layout* , tata warna, pemilihan *toolbox*, *typografi* dan penentuan informasi yaang akan ”tampil” pada *form*) * Mengenali fitur-fitur untuk pembuatan *form* pada aplikasi basis data yang digunakan * Membuat form menggunakan *wizard*. * Memberi nama form dengan tepat | * Mengeksplorasi fitur-fitur pembuatan *form* pada aplikasi basis data yang digunakan * Mempraktekkan pembuatan form menggunakan fitur-fitur pembuatan *form* melalui *wizard* * Mengakses *field*-*field* pada table/*view* yang ada untuk membuat pada *form*. * Menyimpan dan memberi nama *form* | * Mempelajari sekilas aspek-aspek komunikasi visual yang diperlukan dalam pembuatan form, agar diperoleh form yang komunikatif * Dengan cermat memanfaatkan fitur-fitur untuk pembuatan *form* pada aplikasi basis data yang digunakan * Memberikan nama dengan tepat pada setiap informasi yang ”tampil” pada form. * Mencermati pemberian nama form agar mudah dikenali | 1 | 2,5 |
| 05. Membuat modul basis data | 5.1 Modul dibuat dengan penggunaan fitur pembuat modul. Pada fitur pembuat modul tersedia editor untuk pengaturan/penulisan kode program dari bentuk *form/report* yang dirancang.  5.2 Fitur-fitur modul diidentifikasi dan diaplikasikan untuk penanangan *event* atau proses yang diperlukan oleh *form* atau *report*.  5.3 Modul diberi nama dan disimpan. | * Memahami konsep dan manfaat modul * Mengidentifikasi fitur-fitur pembuatan modul pada aplikasi yang digunakan. * Memahami teknis pembuatan modul * Memberi nama modul dengan tepat | * Membuat modul menggunakan fitur pembuat modul. * Membuat modul untuk penanganan *event* yang dikenakan pada *form,* *report* atau objek basis data yang lain * Memberi dan menyimpan nama modul | * Mempelajari manfaat pembuatan modul pada aplikasi basis data. * Dengan seksama mengeksplorasi fitur-fitur dan sintak pembuat modul pada aplikasi yang digunakan * Mencermati pemberian nama modul agar mudah dikenali | 1 | 2 |

1. **Unit Kompetensi : Menerapkan Basis Data**

**Kode Unit : TIK.PR02.021.01**

**Durasi Pembelajaran : 10,5 Jam@60 menit**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Pengetahuan** | **Keterampilan** | **Sikap Kerja** | **Pengetahu-an** | **Kete-rampi-lan** |
| 1. Mengguna-kan basis data | * 1. Pengelolaan data dapat dijelaskan dan diaplikasikan berdasarkan perancangan.   2. Basis data diaplikasikan sesuai dengan perancangan.   3. Sistem pengelolaan basis data (*DBMS – Database Management System*) diaplikasikan. | * Mengenal teknik-teknik dan langkah-langkah perancangan basis data. * Mengetahui cara mengaplikasikan basis data berdasarkan perancangan. * Mengetahui definisi, keuntungan dan kelemahan *DBMS (Database Management System).* * Mengetahui cara mengaplikasikan *DBMS (Database Management System)* | * Mampu melakukan dasar perancangan basis data. * Mampu mengaplikasikan basis data sesuai dengan perancangan. * Mampu mengaplikasikan sistem pengelolaan basis data. | * Memperhatikan penjelasan dari pelatih. * Melakukan estimasi ukuran *memory* data yang akan disimpan, sehingga dapat memilih perangkat basis data yang tepat. | 2 | 1 |
| 1. Menerapkan teknik pemodelan | 2.1 Pemodelan data sudah diidentifikasi sesuai kebutuhan.  2.2 Pemodelan data dianalisa sesuai dengan kebutuhan (spesifikasi). Diagram data sesuai spesifikasi yang telah ditentukan.  2.3 Data diidentifikasi sampai dengan *data dictionary* (kamus data). Diagram data menunjukkan relasi dari setiap data yang ada.  2.4 Pemodelan data disiapkan sehingga *tools* yang akan digunakan siap untuk mengimplementasikan pemodelan tersebut | * Mengetahui cara mengidentifikasi kebutuhan yang akan digunakan untuk pemodelan data. * Mengetahui cara mengidentifikasi dan membuat diagram data dan kamus data. * Mengetahui cara mengimplementasikan pemodelan data dengan *tools* yang ada. | * Mampu mengidentifikasi kebutuhan yang akan digunakan untuk pemodelan data. * Mampu mengidentifikasi dan membuat diagram dan kamus data. * Mampu mengidentifikasi *tools* yang sesuai untuk implementasi pemodelan data. * Mampu implementasi pemodelan data dengan *tools* yang digunakan. | * Melakukan perancanaan dan identifikasi komputer. * Mendengarkan dan bertanya secara aktif kepada pelatih. * Melakukan pemodelan data. * Dapat melakukan identifikasi data. * Menerapkan model data relational yang syarat dengan kemampuan dalam menjaga *integrity* dan eksistensi data. | 2 | 1.5 |
| 1. Mengintegrasikan konsep pemodelan data dengan *tools* | 3.1 Kebutuhan komputer ditentukan sesuai dengan *platform* basis data  3.2 *Platform* diidentifikasi atas dasar kapasitas dan kemampuan komputer dengan cakupan pada pertimbangan teknologi, bisnis dan kemampuan penyampaian informasi  3.3 Kebutuhan untuk *backup* dan *restore* tersedia dalam *DBMS*  3.4 Akses basis data dapat dikendalikan dan dimonitor menggunakan *tools* spesifik | * Mengetahui kebutuhan komputer yang sesuai dengan *platform* basis data. * Mengetahui mekanisme *backup* dan *restore.* * Mengetahui cara mengendalikan dan memonitor akses basis data. | * Mampu mengidentifikasi kebutuhan komputer yang sesuai dengan *platform* basis data. * Mampu melakukan *backup* dan *restore* pada aplikasi basis data. | * Melakukan estimasi *space/memory* data yang akan disimpan sehingga dapat menentukan *platform* basis data yang sesuai. * Menerapkan mekanisme *backup, restore* dan memonitor. | 2 | 2 |

1. **Unit Kompetensi : Mengoperasikan Bahasa Pemrograman Data Description (SQL) – Dasar**

**Kode Unit : TIK.PR03.001.01**

**Durasi Pembelajaran : 13 Jam@60 menit**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Pengetahuan** | **Keterampilan** | **Sikap Kerja** | **Pengetahu-an** | **Kete-rampi-lan** |
| 1. Mempersiapkan perangkat lunak aplikasi data deskripsi */ SQL* | | 1.1 Perangkat lunak aplikasi *SQL* telah diinstall dan dapat berjalan normal.   * 1. Manual pengguna (*user manual*) perangkat lunak aplikasi *SQL* sudah tersedia dan diidentifikasi.   2. Perangkat komputer sudah dinyalakan dengan sistem operasi dan persyaratan sesuai dengan petunjuk instalasi.   3. Perangkat lunak aplikasi *SQL* dijalankan | * Mengidentifikasi kebutuhan *hardware* dan *software* yang digunakan untuk meng-*install* *database* *SQL*. * Mengetahui langkah-langkah menginstall *database* *SQL*. * Mengetahui cara menggunakan petunjuk manual yang sudah tersedia dalam aplikasi *database* *SQL*. * Mengetahui cara-cara menggunakan program *database* *SQL*. | * Mampu mengidentifikasi kebutuhan *hardware* dan *software* yang digunakan untuk menginstall *database SQL.* * Mampu menginstall *database SQL.* * Mampu menggunakan petunjuk manual program *database SQL.* | * Memperhatikan tuntunan spesifikasi sistem operasi dan spesifikasi perangkat keras yang menjadi syarat instalasi perangkat lunak basis data. * Membeli cd perangkat lunak yang asli. * Menyalakan komputer sesuai dengan prosedur. | 2 | 1 |
| 1. Mengenali menu aplikasi *SQL* | | 2.1 Menu-menu yang tersedia beserta *shortcut*nya dikenali.  2.2 Fitur-fitur pengolahan *table, view, stored procedure, trigger* dapat diidentifikasi dan diaplikasikan seperti pembuatan baru (*new*), penyimpanan (*save*), pembukaan (*open*), penghapusan (*delete*). | * Mengetahui menu-menu dan *shortcut* pada aplikasi *database SQL*. * Mengetahui cara-cara mengidentifikasi fitur-fitur pengolahan *table, view, stored procedure* dan *trigger*. | * Mampu menggunakan aplikasi *database* beserta *shortcut*nya. * Mampu mengaplikasikan fitur-fitur pengolahan *table,view,trigger*. | * Mengenali dan mencoba menerapkan *shortcut* pada aplikasi basis data. * Bertanya dan mendengarkan secara aktif kepada pelatih. | 1 | 1 |
| 1. Membuat dan mengisi *table* | | 3.1 Tabel dibuat dan diatur dengan penggunaan perintah standar dalam *DDL*.  3.2. Tabel yang dibuat sudah memasukkan nama kolom/*field,* tipe data dan konstrainnya.  3.3. Pengisian data pada tabel dengan penggunaan perintah standar seperti *insert, update* dan *delete* disesuaikan dengan informasi dan kondisinya.  3.4. Tabel sudah disimpan, diberi nama dan terisi data.  3.5.*Index table* dibuat dan diatur dengan menggunakan perintah standar berupa *create index* dan *drop index.*  3.6.*Index* disimpan dan diberi nama.  3.7.*View table* dibuat dan diatur dengan penggunaan perintah standar berupa *create view* dan *drop view.*  3.8.*View* disimpan dan diberi nama. | * Mengetahui perintah standar dalam *DDL (Data Definition Language)*. konstrainnya * Mengetahui cara mengidentifikasi tipe data dan konstrain. * Mengetahui cara penyimpanan, pemberian nama dan mengisi tabel. * Mengetahui cara membuat, menyimpan dan memberi nama *index.* * Mengetahui cara penggunaan perintah *view.* | * Mampu menggunakan perintah standar dalam *DDL (Data Definition Language)* yaitu *insert, update* dan *delete.* * Mampu menganalisa dan menggunakan tipe data beserta dalam pembuatan tabel. * Mampu menyimpan, memberi nama dan mengisi kolom data pada tabel. * Mampu membuat *index* dan *view* | * Menerapkan penggunaan perintah standar dalam *DDL*. * Membuat tabel, memasukkan data, mengupdate data dan menghapus data sesuai dengan prosedur. * Latihan membuat *index* dan *view* sesuai dengan prosedur. | 2 | 2 |
| 1. Mengoperasikan *table* dan *view* | | 4.1 Data tabel ditampilkan dengan perintah *select*. Perintah *select* yang digunakan dapat berupa *select* dengan kondisinya seperti *distinct, where, order by, group by, having,* dsb.  4.2.Data tabel ditampilkan dengan perintah *select* yang terkait (*join*) dengan beberapa tabel yang lain. Perintah *select* yang digunakan dapat berupa select dengan kondisinya seperti *distinct, where, order by, group by, having*, dsb. | * Mengetahui penggunaan perintah dasar *select.* * Mengetahui cara penggunaan perintah *select* dengan berbagai kondisinya seperti *distinct, where, order by, group by, having,* dsb. * Mengetahui cara penggabungan tabel dengan perintah *join.* | * Mampu menggunakan perintah dasar *select.* * Mampu menggunakan perintah *select* dengan berbagai kondisinya seperti *distinct, where, order by, group by, having,* dsb. * Mampu menggunakan perintah *join* dan *select* dengan berbagai kondisinya seperti *distinct, where, order by,* dsb. | * Menerapkan perintah *select* dengan berbagai kondisi. * Menerapkan cara menggabungkan tabel sesuai dengan prosedur. | 2 | 2 |

**Unit Kompetensi : Mengoperasikan Bahasa Pemograman Berorientasi Objek (*Object Oriented*)**

**Kode Unit : TIK.PR.02.009.01**

**Durasi Pembelajaran : 13.5 jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Pengetahuan** | **Keterampilan** | **Sikap Kerja** | **Pengetahu-an** | **Kete-rampi-lan** |
| 01. Menggunakan tipe data dan kontrol program. | * 1. Tipe data dasar dan tipe data bentukan dijelaskan.   2. Tata cara penulisan program *Object Oriented* diidentifikasi dan diaplikasikan.   3. Kontrol program berupa urutan atau sekuensial program dibuat dan dikendalikan dalam kode program. | * Mengetahui dasar mengenai definisi dan tipe-tipe data dasar dan data bentukan. * Mengetahui tata cara penulisan program *Object Oriented*. * Mengetahui cara mengontrol program berdasarkan urutan/sekuensial program dibuat dalam kode program. | * Mampu menjelaskan definisi dan membedakan antara tipe data dasar dan data bentukan. * Dapat menuliskan program *Object Oriented* dengan benar. * Mampu mengontrol program berdasarkan urutan program dibuat dalam kode program. | * Mengerti tipe data dan cara mengontrol program. | 1 | 1,5 |
| 02 Membuat program *object oriented* dengan *class.* | 1. Program dengan penggunaan obyek atau *class* dibuat. 2. Properti *class* yang akan direalisasikan dalam bentuk prosedur/fungsi dibuat. 3. Data di dalam *class* dibuat mandiri. Artinya data di dalam *class* tidak dapat diakses secara langsung oleh *class* lain. Data hanya dapat diakses melalui property *class* (berupa fungsi/prosedur). | * + Mengetahui cara menggunakan obyek atau *class* dalam kode program.   + Mengetahui cara merealisasikan properti *class* dalam bentuk prosedur/fungsi.   + Mengetahui cara membuat data di dalam *class* yang tidak dapat diakses oleh *class* lain dan dapat diakses oleh *class* lain melalui fungsi/prosedur. | * Mampu membuat program dengan menggunakan obyek/*class*.   + Dapat merealisasikan properti *class* dalam bentuk prosedur/fungsi dalam kode program.   + Mampu membuat data di dalam *class* tidak dapat diakses secara langsung oleh *class* lain dan hanya dapat diakses melalui fungsi/prosedur. | * + Menguasai konsep pemrograman *object* *oriented* dan *class*. | 1,5 | 1,5 |
| 03 Membuat program *object oriented* dengan *inheritance*, *polimorphy*, *overloading*, dan *friends.* | * 1. *Inheritance* pada *class* diaplikasikan.  1. *Polimorphy* pada *class* diaplikasikan.    1. *Overloading* pada *class* diaplikasikan.    2. *Friends* pada *class* diaplikasikan. | * + Mengetahui cara menggunakan konsep *inheritance* pada *class*.   + Mengetahui cara menggunakan konsep *polimorphy* pada *class*.   + Mengetahui cara menggunakan konsep *overloading* pada *class*.   + Mengetahui cara menggunakan konsep *friends* pada *class*. | * Mampu menerapkan *inheritance* pada *class* dalam kode program. * Mampu menerapkan *polimorphy* pada *class* dalam kode program. * Mampu menerapkan *overloading* pada *class* dalam kode program. * Mampu menerapkan *friends* pada *class* dalam kode program. | * + Menguasai konsep *inheritance*, *polimorphy*, *overloading*, dan *friends*. | 1,5 | 2 |
| 04 Membuat program *object oriented* dengan *interface* dan paket. | * 1. *Interface class* program dibuat.   2. *Paket* dengan penggunaan *class* program dibuat. | * + Mengetahui cara pembuatan program menggunakan *interface class*.   + Mengetahui cara pembuatan program menggunakan paket. | * Mampu membuat *interface* *class* dalam kode program. * Mampu membuat *interface class* dalam kode program. | * + Mengerti cara membuat program *object oriented* dengan *interface* dan paket. | 1 | 1,5 |
| 05 Mengkompilasi dengan *object oriented*. | 5.1 Program dikompilasi dengan *setting* pada *editor* program dan kode program yang dilibatkan.  5.2 Kesalahan sintak dan logika dikoreksi dan dibetulkan. Program yang dibuat harus bebas dari *error* dan dapat dijalankan. | * + Mengetahui cara mengkompilasi program.   + Mengetahui kesalahan sintak dan logika dari hasil kompilasi. | * Dapat mengkompilasi program dengan *setting* pada *editor* program dan kode program. * Mampu mengecek adanya kesalahan sintak dan logika pada program dan membetulkannya. | * + Program berorientasi objek telah selesai dikerjakan. | 1 | 1 |

**Unit Kompetensi : Membuat Program Untuk Akses Basis Data**

**Kode Unit : TIK.PR.02.023.01**

**Durasi Pembelajaran : 16 jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Pengetahuan** | **Keterampilan** | **Sikap Kerja** | **Pengetahu-an** | **Kete-rampi-lan** |
| 01 Mengidentifikasi basis data | * 1. Kebutuhan sistem dipenuhi sesuai lingkupan dan fungsi dari basis data.   2. Bentuk struktur data, bentuk *query*, bentuk tampilan, bentuk laporan (cetak) disesuaikan dengan *table* yang telah dibuat.   3. Sekuriti dan metode akses pada basis data digunakan sesuai perancangan. | * Pengertian Sistem Managemen Basis Data * Keperluan Dasar membangun Basis Data * Bentuk-bentuk Struktur Data, Query, tampilan dan Laporan Cetak. * Prinsip dasar keamanan Basis Data. | * Mencari piranti lunak Basis Data sesuai kebutuhan. * Kemampuan untuk berpikir sesuai dengan konseptual Basis Data. | * Mengumpulkan informasi dengan rajin dan tekun * Aktif bertanya kepada pengajar. * Mengembangkan kemampuan berpikir analitis. | 2,5 | 2 |
| 02 Membuat program basis data | 2.1 Struktur data dibangun pada bahasa pemrograman sesuai dengan disain yang telah dibuat. Struktur data diimplementasikan dalam bentuk *table* dan *view*, mengikuti disain yang telah dibuat.  2.2 Basis data dinormalisasikan untuk penyederhanaan relasi *table* dan memperkokoh relasi *table*.  2.3 Query, tampilan dan laporan (cetak) dibangun menurut bahasa pemrograman yang sesuai dengan disain. Query dapat *diimplementasikan* dengan prosedur, fungsi maupun *trigger*. Tampilan dan laporan akan berisikan data agar *table* atau *view* ditampilkan | * Teori pembentukan tabel relasi * Teori Normalisasi * Prinsip – prinsip dasar pembangunan Basis Data dengan bahasa pemrograman *Query*. | * Membuat tabel relasi sesuai dengan kebutuhan. * Melakukan Normalisasi dari tabel yang ada untuk menghindari redudansi. * Membangun tabel yang telah dinormalisasi dengan bahasa *query*. | * Aktif Bertanya apabila ada yang tidak dimengerti. * Mengembangkan pola pikir konseptual. | 2 | 2 |
| 03 Membuat koneksi dan sekuriti basis data | 3.1 Pengguna dan *password* dibangun disesuaikan dengan disain yang dibuat. Pengguna dan password dapat diimplementasikan pada DBMS atau dalam bahasa pemrograman sehingga basis data dapat diakses.  3.2 *Privilage* setiap pengguna terhadap basis data dibangun disesuaikan dengan disain yang dibuat. *Privilage* pengguna terhadap basis data dapat diimplementasikan pada DBMS.  3.3 Sedangkan *privilege* untuk akses tampilan dan laporan diimplementasi-kan pada bahasa pemrograman. | * Prinsip-prinsip dasar pembangunan sistem keamanan pada Basis Data. * Prosedur pembangunan hak akses dari basis data. * Teori tampilan/view Basis Data. | * Memberikan *password* terhadap Basis data. * Membangun view/tampilan sesuai dengan keperluan masing-masing pengguna. * Memberikan hak akses tertentu bagi setiap pengguna *Database*. | * Mengembangkan kemampuan untuk berpikir kritis. * Aktif bertanya apabila ada yang tidak dimengerti. * Aktif menjawab pertanyaan. | 2 | 2 |
| 04 Menguji program basis data | 4.1 Data uji dan skenario uji disiapkan. Data uji sebagai data dimasukkan ke dalam sistem. Sedangkan skenario uji sebagai urutan/langkah pengujian.  4.2 Logika program basis data diperiksa dan dikoreksi. Program basis data dikompilasi dan dijalankan.Jika ditemukan kesalahan, kesalahan tersebut dibetulkan. Program basis data dapat dikompilasi dan dijalankan.  4.3 Data uji disimpan dan ditampilkan sesuai kebutuhan. Laporan dapat dicetak sesuai kebutuhan. Kesalahan yang ditemukan pada tampilan dan laporan diperiksa dan dikoreksi.  4.4 Program basis data sesuai kebutuhan yang telah ditentukan. Hasil pengujian harus sesuai terhadap kebutuhan/spesifikasi program. | * Prinsip dasar uji coba data. * Prosedur pemeriksaan program. * Prosedur penseleksian data sesuai kebutuhan. * Prinsip-prinsip penampilan informasi sesuai kebutuhan. | * Melakukan penginputan data. * Melakukan uji coba tabel relasi. * Melakukan pembenaran apabila ada kesalahan. * Menampilkan informasi/data sesuai dengan kebutuhan masing-masing user. | * Mengembangkan pola berpikir terstruktur. * Mengembangkan logika pemrograman. * Aktif bertanya apabila ada yang tidak dimengerti. | 1,5 | 2 |

**Unit Kompetensi : Membuat Program Aplikasi Berbasis Java**

**Kode Unit : TIK.PR.08.012.01**

**Durasi Pembelajaran : 20,5 jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **Pengetahuan** | **Keterampilan** | **Sikap Kerja** | **Pengetahu-an** | **Kete-rampil-an** |
| 01 Menjelaskan File I/O (*Input/Output*), tipe data dan *variabel* pada *java* | 1.1 Konsep dari *I/O* di *java* dan *File streams*  dijelaskan.  1.2 Berbagai tipe data dan sifat – sifatnya dijelaskan.   * 1. Konsep variabel pada *java* dijelaskan. | * Mengetahui penempatan *input* dan *output* * Mengenal *stream* standar * Mengetahui penggunaan *input* dan *output stream* * Mengetahui berbagai macam tipe data pada *java.* * Mengerti penggunaan tipe data. * Mampu mendeklarasikan *variabel* * Mengerti untuk memberikan nilai terhadap *variabel* | * Membaca dan menulis file di tempat kerja sehingga dapat dipahami dan ditampilkan. * Bertanya dan mendengarkan secara aktif dilakukan untuk menambah informasi. * Mampu membedakan fungsi – fungsi dari tipe data. * Mengidentifikasi komponen – komponen yang menunjang pembuatan program. | * Membaca dan mengaplikasikan file I/O, tipe data dan variabel java. * Menjaga ketertiban selama proses pelatihan. | 1 | 1 |
| 02 Menjelaskan *Exception Handling* | 2.1 Konsep dari *exception handling* serta pemodelan dari *Try, Throw, Catch* dan *Finally* dijelaskan. | * Mengetahui penjelasan tentang *exception.* * Mengetahui cara-cara melalui *exception* yang terdapat pada *java.* * Mengerti kelebihan dari *exception* * Mengerti cara penggunaan dari *try, throw, catch* dan *finally* pada *exception* di *java* | * Mampu mengidentifikasi *exception handling* pada pembuatan program *java*. * Mampu melakukan beberapa pemodelan yang menunjang pembuatan *exception handling*. | * Mengikuti pelatihan sesuai bahan yang telah diberikan dengan seksama * Memahami dan mengaplikasikannya ke dalam program java. | 1 | 2 |
| 03 Menerapkan *multi threading* | 3.1 *Thread class* dan sinkronisasinya digunakan.  3.2. Penggunaan *thread* tingkat lanjut ( *concurrent execution runnable interface, daemon threads, thread priority, thread groups*) dijelaskan. | * Mengerti penjelasan mengenai *multithreading* dan *thread.* * Mengetahui cara untuk menangani *thread* dengan menggunakan class *thread* dan melalui *runnable*. * Mengetahui daur hidup *thread* | * Mampu melakukan sinkronisasi terhadap *thread.* * Mampu melakukan penggunaan *thread* beserta komponen – komponennya yang menunjang. | * Membaca dan mengaplikasikan penggunaan thread dalam java. | 1,5 | 2 |
| 04 Menjelaskan *Network Programming.* | 4.1 Konsep dasar *TCP/IP* dan *java.net* *package* dijelaskan.  4.2. *Socket programming server side* dan *client side* dijelaskan. | * Mengetahui kegunaan dari TCP/IP dan java.net * Mengetahui layer – layer yang terdapat pada TCP/IP * Mengetahui perbedaan *server side* dan *client side.* * Mengetahui kelebihan dan kekurangan masing – masing *socket programming* tersebut. | * Mampu melakukan *network programming* pada *java*. * Mampu melakukan penggunaan dari *server side* dan *client side.* | * Membaca dan mengaplikasikan jaringan *TCP/IP* dan *java.net* * Mengaplikasikan penggunaan *socket programming.* | 1 | 2 |
| 1. Menerapkan *Object Oriented* | 5.1 Konsep dasar dari *object oriented* dan kemampuan untuk identifikasi *classes* dan *object* dijelaskan.  5.2 *Objects* dirancang sesuai kebutuhan.  5.3Penggunaan *constructors, Destructors, class methods, class extending, overriding methods, visibility modifiers, abstract classes* dan *interface* dijelaskan.   * 1. *Inheritance* dan *polymorphism* dijelaskan. | * Mengetahui konsep pemrograman *berorientasi object.* * Mengetahui cara penggunaan *Object* dan mengerti menciptakan *class.* * Mengetahui penggunaan *class* dalam program. * Mengetahui penggunaan dari beberapa fungsi yang menunjang dalam penerapan *object oriented.* | * Mampu melakukan identifikasi terhadap *object oriented.* * Mampu merancang *object* serta mengerti dalam penggunaan beberapa komponen yang menunjang *object oriented.* | * Membaca dan mengaplikasikan object oriented di dalam pemrograman java. * Membaca dan mengaplikasikan inherintance dan polymorphism. | 1,5 | 2 |
| 1. Menjelaskan file *I/O* (*Input/Output*), tipe data dan variabel pada *java.* | 6.1 Penggunaan dari *drawing graphic*, komponen *GUI* dan animasi dijelaskan. | * Mengetahui penggunaan dari *drawing graphic* * Mengetahui macam – macam komponen *GUI* dan animasi. * Mengetahui penggunaan *GUI* dan animasi pada *java.* | * Mampu menerapkan *drawing graphic*, *GUI,* dan animasi dalam pemrograman *java*. | * Membaca dan mengerti materi tersebut. * Mengaplikasikan *drawing graphic*, komponen *GUI* dan animasi ke dalam pemrograman *java*. | 1 | 1,5 |
| 1. Menetapkan konektivitas *database* | 7.1 Koneksi *database* dengan penggunaan *JDBC (Java Database Connectivity*) dilakukan.  7.2 Interoperabilitas dari *JDBC* dan *ODBC* dijelaskan. | * Mengetahui tahapan – tahapan koneksi *database*. * Mengetahui dalam pembuatan *database*. * Mengetahui pembuatan tabel dan mengerti dalam menampilkan tabel. * Mengerti dalam mengakses *database* via aplikasi java. | * Mampu melakukan koneksi *database* dalam pembuatan program *java.* | * Membaca dan melakukan koneksi *database*. * Mengaplikasikan penggunaan dari *JDBC* dan *ODBC*. | 2 | 1 |

**KEAHLIAN BIDANG MANAJEMEN**

* 1. **Unit Kompetensi : Menerapkan ketrampilan dalam pengelolaan komunikasi dasar**

**Kode Unit : TIK.PR01.011.01**

**Durasi Pembelajaran : 8 jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGE-TAHUAN** | **KETERAM-PILAN** |
| 01 Memberikan kontribusi pada perencanaan komunikasi | * 1. Persyaratan informasi yang relevan diidentifikasi dan kontribusi dibuat pada pengembangan seluruh kebutuhan informasi proyek.   2. Kontribusi dibuat dalam proses tim, pada pengembangan dan implementasi rencana pengelolaan komunikasi, kemudian ditetapkan jaringan komunikasi. | * Mengetahui persyaratan informasi yang relevan. * Mengetahui informasi tentang jaringan komunikasi. | * Mampu membuat kontribusi pada pengembangan seluruh kebutuhan informasi proyek. * Mampu mengembangkan dan melakukan implementasi rencana pengelolaan komunikasi. | * Mencatat persyaratan informasi yang relevan. | 1 | 1 |
| 02 Mengadakan aktivitas -aktivitas pengelolaan informasi | 1. Informasi dikelola (dikumpulkan, divalidasi, disimpan, didapatkan kembali, disaring dan diseminasi) seperti yang telah diatur dalam prosedur dan proses pengambilan keputusan. 2. Informasi dijaga dalam struktur yang disetujui sebagai jaminan bagi keamanan dan auditabilitas data. | * + Mengetahui prosedur dan proses pengambilan keputusan.   + Mengetahui tujuan informasi dikelola dan disetujui. | * Dapat mengelola informasi   + Dapat menjaga informasi dalam struktur yang disetujui. | * Mencatat prosedur dan proses pengambilan keputusan. * Mencatat informasi mana saja yang telah disetujui. | 1 | 1 |
| 03 Mengkomunikasi informasi proyek | 1. Komunikasi dalam proyek dengan klien dan *stakeholder* lainnya dilakukan dengan jaringan yang telah disepakati, proses-proses dan prosedur sebagai jaminan bagi kejelasan dari identifikasi sasaran.    1. Laporan ditulis dan disiarkan menurut otorisasi, atau diusulkan untuk disiarkan oleh otorisasi proyek yang lebih tinggi sesuai dengan proses dan prosedur. | * + Mengetahui proses-proses, tujuan dan prosedur mengkomunikasi-kan informasi.   + Mengetahui proses dan prosedur otorisasi. | * Mampu melakukan komunikasi dalam proyek dengan klien dan *stakeholder* lainnya. * Mampu menulis dan menyiarkan laporan menurut otorisasi yang disetujui. | * + Mencatat laporan yang disiarkan. | 1 | 1 |
| 04 Memberikan kontribusi untuk menetapkan penilaian hasil pengelolaan komunikasi. | 4.1 Hasil proyek ditinjau bagi penetapan efektivitas dari aktivitas pengelolaan komunikasi  4.2 Hal-hal pengelolaan komunikasi dan respon-respon dilaporkan pada otoritas proyek yang lebih tinggi untuk aplikasi proyek di masa depan. | * Mengetahui informasi tentang efektivitas dari aktivitas pengelolaan komunikasi. * Mengetahui informasi tentang aplikasi proyek masa depan. | * Mampu meninjau hasil proyek. * Dapat melaporkan hal-hal pengelolaan komunikasi dan respon-respon pada otoritas proyek yang lebih tinggi. | * Mencatat informasi tentang efektivitas dari aktivitas pengelolaan komunikasi. | 1 | 1 |

* 1. **Unit Kompetensi : Mengoperasikan bahasa pemograman data description (SQL) lanjut**

**Kode Unit : TIK.PR.03.002.01**

**Durasi Pembelajaran : 19 jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGE-TAHUAN** | **KETERAM-PILAN** |
| 01 Membuat dan mengoperasikan prosedur | * 1. Prosedur dibuat dan diatur dengan menggunakan perintah standar pada *DDL* dan *DML*.   2. Parameter prosedur diproses sesuai dengan kebutuhan di dalam prosedur.   3. Prosedur disimpan dan diberi nama.   4. Prosedur dapat dijalankan dan bekerja dengan benar. | * Teori Himpunan, relasi dan operasi yang terkait dengan relasi matematika * Konsep dan keutamaan *relational database* * Normalisasi skema relasi dalam *relational database* * *Integrity* dan *domain constraint* dalam implementasi basis data relasional * *DDL, DML* dan *DCL* dalam *Structured Query Language* * Pengelompokkan struktur proses *non parallel* dengan menggunakan *SQL*, terkait dengan deklarasi *variable, working with attributes, Assigning value, operasi Read/Write* ke peralatan I/O, *control statement dan iterative control* | * Memodelkan *e-r diagram* dan *skema relasional* *database* dengan memperhatikan normalisasi model data * Menerapkan *DDL* sintaks dengan mengimplementasikan *integrity* dan *domain constraint* * Menerapkan *DML* sintaks dengan mempraktekkan operasi *Insert, Update, Delete* dan *Select* * Menerapkan *DCL* sintaks dengan mempraktekkan *commit* dan *rollback* * Mempraktekkan mekanisme penjagaan *integrity data* dan *atomic transaction* melalui *stored procedures* | * Senantiasa merencanakan dan memodelkan struktur relasi antar tabel data dalam konteks *relational database* * *Data dictionary* disusun sebelum mengimplementasikan sebuah model data fisikal * Menggunakan *tools pemodelan data* yang relevan sehingga pemodelan data secara konseptual dan fisikal dapat dipermudah * Taat aturan akan Syntaks *DDL, DML* dan *DCL* beserta variasinya sehingga tidak bersifat *try and error* * Menelusuri/*trace* setiap *stored procedure* dalam menemukenali *bug/*kesalahan logika dalam sebuah implementasi *relational database* | 2 | 4 |
| 02 Membuat dan mengoperasikan prosedur dan fungsi | * 1. Fungsi dibuat dan diatur dengan menggunakan perintah standar pada *DDL* dan *DML*.   2. Parameter fungsi diproses sesuai dengan kebutuhan di dalam fungsi.   3. Fungsi disimpan dan diberi nama.   4. Fungsi dapat dijalankan dan bekerja dengan benar. | * Mekanisme dan penerapan *Stored procedure* dan *function* menggunakan perintah standar *DDL* dan *DML* * *Calling* dan *maintaining* sebuah fungsi serta *parameterisasi* | * Mengidentifikasi modul program yang dapat dibuat dalam rangka menjaga *integrity* dan *atomic transaction* * Mengidentifikasi *type parameter* dan jenis *passing parameter* yang dibutuhkan oleh *stored procedure* maupun *function* * Memanggil dan menggunakan *function*  yang telah dibuat | * Taat aturan akan Syntaks *DDL, DML* dan *DCL* beserta variasinya sehingga tidak bersifat *try and error* * Menelusuri/*trace* setiap *Function* dalam menemukenali *bug/*kesalahan logika dalam sebuah implementasi *relational database* | 2 | 4 |
| 03 Membuat dan mengoperasikan *trigger* | * 1. *Trigger* dibuat dan diatur dengan penggunaan perintah standar pada *DDL* dan *DML*.   2. *Trigger* disimpan dan diberi nama.   3. *Trigger* dapat dijalankan dan bekerja dengan benar. | * Mekanisme penjagaan *integrity data* dan *atomic transaction* melalui *trigger* * Jenis dan klasifikasi *trigger* * Activasi dan Reaksi *trigger* | * Mempraktekkan mekanisme penjagaan *integrity data* dan *atomic transaction* melalui *trigger* | * Senantiasa memodelkan syntaks *Query* yang akan diterapkan * Taat aturan akan Syntaks *DDL, DML* dan *DCL* beserta variasinya sehingga penerapannya tidak lagi bersifat *try and error* * Menelusuri/*trace* setiap *trigger* dalam menemukenali *bug/*kesalahan logika dalam sebuah implementasi *relational database* | 1 | 3 |
| 04 Mengoperasikan *Commit* dan *Rollback* data tabel | 4.1 Perintah *commit* digunakan untuk penyimpanan data ke dalam fisik *file*.   * 1. Perintah *rollback* untuk pembatalan perintah yang telah dikerjakan dan penyimpanan data ke dalam fisik *file* digagalkan. | * Sintaks *DCL* dalam *Structured Query Language* * Fungsi dan makna perintah *Commit* dan *Rollback* | * Menerapkan *DCL* sintaks dengan mempraktekkan *commit, rollback* serta pemberian *privillege* atau hak akses | * Taat aturan akan Syntaks *DDL, DML* dan *DCL* beserta variasinya sehingga penerapannya tidak lagi bersifat *try and error* * Membiasakan diri untuk melakukan *commit* jika transaksi sudah dianggap benar. | 1 | 2 |

* 1. **Unit Kompetensi : Membuat program untuk akses basis data**

**Kode Unit : TIK.PR.02.023.01**

**Durasi Pembelajaran : 19 jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGE-TAHUAN** | **KETERAM-PILAN** |
| 01 Mengidentifikasi basis data | * 1. Kebutuhan sistem dipenuhi sesuai lingkupan dan fungsi dari basis data.   2. Bentuk struktur data, bentuk *query*, bentuk tampilan, bentuk laporan (cetak) disesuaikan dengan table yang telah dibuat.   3. Sekuriti dan metode akses pada basis data digunakan sesuai perancangan. | * Pengertian Sistem Managemen Basis Data * Keperluan Dasar membangun Basis Data * Bentuk-bentuk Struktur Data, Query, tampilan dan Laporan Cetak. * Prinsip dasar keamanan Basis Data. | * Mencari piranti lunak Basis Data sesuai kebutuhan. * Kemampuan untuk berpikir sesuai dengan konseptual Basis Data. | * Mengumpulkan informasi dengan rajin dan tekun * Aktif bertanya kepada pengajar. * Mengembangkan kemampuan berpikir analitis. | 4 | 4 |
| 02 Membuat program basis data | 2.1 Struktur data dibangun pada bahasa pemrograman sesuai dengan disain yang telah dibuat. Struktur data diimplementasikan dalam bentuk table dan view, mengikuti disain yang telah dibuat.  2.2 Basis data dinormalisasikan untuk penyederhanaan relasi table dan memperkokoh relasi table.  2.3 Query, tampilan dan laporan (cetak) dibangun menurut bahasa pemrograman yang sesuai dengan disain. Query dapat diimplementasikan dengan prosedur, fungsi maupun trigger. Tampilan dan laporan akan berisikan data agar table atau view ditampilkan | * Teori pembentukan tabel relasi * Teori Normalisasi * Prinsip – prinsip dasar pembangunan Basis Data dengan bahasa pemrograman Query. | * Membuat tabel relasi sesuai dengan kebutuhan. * Melakukan Normalisasi dari tabel yang ada untuk menghindari redudansi. * Membangun tabel yang telah dinormalisasi dengan bahasa query. | * Aktif Bertanya apabila ada yang tidak dimengerti. * Mengembangkan pola pikir konseptual. |  |  |
| 03 Membuat koneksi dan sekuriti basis data | 3.1 Pengguna dan password dibangun disesuaikan dengan disain yang dibuat. Pengguna dan password dapat diimplementasikan pada DBMS atau dalam bahasa pemrograman sehingga basis data dapat diakses.  3.2 Privilage setiap pengguna terhadap basis data dibangun disesuaikan dengan disain yang dibuat. Privilage pengguna terhadap basis data dapat diimplementasikan pada DBMS.  3.3 Sedangkan privilege untuk akses tampilan dan laporan diimplementasikan pada bahasa pemrograman. | * Prinsip-prinsip dasar pembangunan sistem keamanan pada Basis Data. * Prosedur pembangunan hak akses dari basis data. * Teori tampilan/view Basis Data. | * Memberikan password terhadap Basis data. * Membangun view/tampilan sesuai dengan keperluan masing-masing pengguna. * Memberikan hak akses tertentu bagi setiap pengguna Data Base. | * Mengembangkan kemampuan untuk berpikir kritis. * Aktif bertanya apabila ada yang tidak dimengerti. * Aktif menjawab pertanyaan. |  |  |
| 04 Menguji program basis data | 4.1 Data uji dan skenario uji disiapkan. Data uji sebagai data dimasukkan ke dalam sistem. Sedangkan skenario uji sebagai urutan/langkah pengujian.  4.2 Logika program basis data diperiksa dan dikoreksi. Program basis data dikompilasi dan dijalankan.Jika ditemukan kesalahan, kesalahan tersebut dibetulkan. Program basis data dapat dikompilasi dan dijalankan.  4.3 Data uji disimpan dan ditampilkan sesuai kebutuhan. Laporan dapat dicetak sesuai kebutuhan. Kesalahan yang ditemukan pada tampilan dan laporan diperiksa dan dikoreksi.  4.4 Program basis data sesuai kebutuhan yang telah ditentukan. Hasil pengujian harus sesuai terhadap kebutuhan/spesifikasi program. | * Prinsip dasar uji coba data. * Prosedur pemeriksaan program. * Prosedur penseleksian data sesuai kebutuhan. * Prinsip-prinsip penampilan informasi sesuai kebutuhan. | * Melakukan penginputan data. * Melakukan uji coba tabel relasi. * Melakukan pembenaran apabila ada kesalahan. * Menampilkan informasi/data sesuai dengan kebutuhan masing-masing user. | * Mengembangkan pola berpikir terstruktur. * Mengembangkan logika pemrograman. * Aktif bertanya apabila ada yang tidak dimengerti. |  |  |

* 1. **Unit Kompetensi : Melakukan Debugging Program.**

**Kode Unit : TIK.PR02.014.01**

**Durasi Pembelajaran : 14,5 Jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGETAHU-AN** | **KETERAM-PILAN** |
| 01 Mempersiapkan kode program | 1. Kode program ditulis sesuai dengan spesifikasi. Penulisan kode program dilakukan pada editor yang disarankan oleh bahasa pemrograman (*compiler*). 2. Aplikasi bahasa pemrograman disiapkan. | * + - Mampu menjelaskan teknis penulisan program mengggunakan bahasa yang telah ditentukan, menggunakan editor yang disarankan * Mengetahui teknis instalasi bahasa pemrograman yang akan digunakan (jika belum ada) | * + Menuliskan program sesuai spesifikasi yang diberikan ke dalam bahasa pemrograman yang disarankan   + Menginstallasi bahasa pemrograman yang akan digunakan | * + Mentaati struktur program, sintaks, dan aturan-aturan lain terkait dengan bahasa pemrograman yang digunakan   + Mencermati apakah bahasa pemrograman yang akan digunakan telah terinstall dengan benar | 1 | 2 |
| 02 Melakukan *debugging* | 1. Kode program dikompilasi sesuai bahasa pemograman yang digunakan. 2. Hasil proses kompilasi diperiksa, analisa kesalahan yang ditemukan. 3. Jika tidak ada kesalahan kompilasi, proses *build/link* kode program dilakukan. 4. Hasil proses *build/link* diperiksa. Jika ada kesalahan, kesalahan proses *build/link* dianalisa. | * Mengetahui manfaat dan beberapa cara mengkompilasi program pada lingkungan (*environment*) yang digunakan * Memahami arti pesan kesalahan yang muncul (jika ada), dan dapat mendokumentasikan kesalahan tersebut * Memahami manfaat dan cara melakukan *link*/ *build* pada lingkungan / *environment* yang digunakan * Memahami fungsi proses *build/link*, teknis pelaksanaan, kapan proses ini dilakukan, serta arti pesan kesalahan yang muncul pada proses *link/build*. | * + Melakukan beberapa cara kompilasi program pada lingkungan (*environment*) yang digunakan   + Menganalisis hasil kompilasi   + Mendokumentasikan pesan kesalahan (jika ada)   + Menentukan perbaikan pada setiap kesalahan yang diinformasikan   + Melakukan proses *link/bulid*   + Melakukan analisis terhadap hasil proses *link/build* dan mencatat pesan kesalahan dari proses tersbut (jika ada) | * + Mencermati hasil proses kompilasi   + Dengan seksama memperhatik-an dan memahami setiap pesan kesalahan yang muncul pada proses kompilasi maupun proses *link/build* (jika ada)   + Mencermati hasil proses *link/build* | 1,5 | 2,5 |
| 03 Menganalisa kode kesalahan | 1. Kode kesalahan kompilasi dianalisa sesuai dengan informasi yang diberikan. 2. Kesalahan yang terjadi pada kode program dianalisa, sesuai dengan spesifikasi dan standar penulis / sintak pemrograman. 3. Kode kesalahan dicatat pada lembar catatan | * + Memahami pesan kesalahan dari hasil proses kompilasi dan proses *link/build*   + Mengetahui cara membetulkan program berdasarkan pesan kesalahan yang muncul pada proses kompilasi maupun pada proses *link/build*   + Mendokumentasikan *”history”* kesalahan dan perbaikannya | * + Melakukan analisis terhadap informasi kesalahan yang terjadi pada saat proses kompilasi dan proses *link/build*   + Melakukan analisa terhadap kesalahan dan membetulkan kesalahan berdasarkan hasil proses kompilasi maupun proses *link/build*   + Mencatat *history* kesalahan yang terjadi selama pengembangan program / perangkat lunak | * + Dengan cermat menganalisis hasil proses kompilasi, dan pesan kesalahan yang muncul (jika ada) dan mendokumentasikannya dengan lengkap | 2.5 | 5 |

* 1. **Unit Kompetensi : Membuat program basis data berbasis MYSQL/PostGreSQL**

**Kode Unit : TIK.PR08.007.01**

**Durasi Pembelajaran : 19 Jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGE-TAHUAN** | **KETERAM-PILAN** |
| * 1. Menjelaskan kebutuhan *software* | * 1. Persyaratan sistem operasi (Linux, Windows) diindefitikasi.   2. Kalibrasi *database* agar berjalan secara optimal dilakukan.   3. Pengujian *database* apakah *database* berjalan dengan optimal dilakukan. | * Konsep sistem operasi * Melakukan instalasi MySQL / PostgreSQL * Memahami konsep basis data | * Membangun *database* * Menguji kesiapan *database* | * Menganalisa kebutuhan sistem yang tepat * Menjalankan tahapan penginstalan dengan teratur | 2 | 3 |
| * 1. Mempersiap-kan *security* | 2.1 Persiapan privileges dan access level baik untuk *groups*, *user*s, *databases* dan *tables* dilakukan | * Pengetahuan mengenai hak akses dan *priviliges* * Pengetahuan mengenai *group*, *user*s, *database* dan tabel | * Membuat *database*, *user*s, group ataupun tabel sesuai kebutuhan * Memberi hak akses kepada masing-masing sesuai kebutuhan | * Menganalisa kepentingan tiap *user* akan penggunaan *database* * Memberi hak sesuai dengan autoritas masing-masing *user* | 2 | 2 |
| * 1. Menggunakan sintaks-sintaks khusus *MySQL / PostgreSQL* | 3.1 Sintaks-sintaks di MySQL/PostgreSQL secara efektif digunakan. | * Pengetahuan tentang sintaks-sintaks pada MySQL / PostgreSQL * Memahami kegunaan dari tiap sintaks * Mengetahui konsep DML | * Menggunakan sintaks yang tepat untuk melihat isi dari *database* * Melakukan modifikasi pada *database* dengan menggunakan sintaks yang tepat * Menggunakan sintaks secara tepat guna untuk mendapatkan informasi yang optimal | * Menggunakan sintaks secara tepat guna dan teliti | 2,5 | 2,5 |
| * 1. Melakukan pengaksesan *database* | 4.1 Konsep dari inisialisasi dan terminasi koneksi *database* dijelaskan.  4.2 *Persistent* dan non-*persistent* *connection* dimanfaatkan secara optimal. | * Pengetahuan tentang koneksi ke *database* * Pengetahuan tentang koneksi *persistent* dan non-*persistent* | * Menjalankan koneksi ke *database* dengan benar * Menganalisa apakah perlu koneksi *persistent* atau non-*persistent* * Menutup koneksi ke *database* | * Menganalisa kebutuhan untuk koneksi ke *database* * Melakukan koneksi sesuai dengan langkah-langkah yang diajarkan | 3 | 2 |

* 1. **Unit Kompetensi : Membuat Dokumen Kode Program**

**Kode Unit : TIK.PR02.019.01**

**Durasi Pembelajaran : 10 Jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGETAHU-AN** | **KETERAM-PILAN** |
| 1. Mengidentifikasi kode program | | * 1. Modul program berdasarkan fungsi, prosedur dan data di identifikasi.   2. Parameter data yang menjadi masukan dan keluaran pada fungsi atau prosedur di ketahui.   3. Data dan algoritma pada deskripsi fungsi dan prosedur di jelaskan. | * Mengetahui program berdasarkan fungsi, prosedur dan data. * Mengetahui parameter data yang menjadi masukan dan keluaran pada fungsi atau prosedur. * Mengenal data dan algoritma. | * Mampu mengidentifikasi modul program berdasarkan fungsi, prosedur dan data. * Mengidentifikasi parameter pada fungsi dan prosedur. * Mampu mengenal data dan algoritma pada fungsi dan prosedur. * Bertanya dan mendengarkan secara aktif dilakukan untuk menambah informasi. | * Mengikuti pelatihan sesuai bahan yang telah diberikan dengan seksama * Menjaga ketertiban selama proses pelatihan. | 2 | 1 |
| 1. Membuat dokumentasi modul program | | * 1. Dokumentasi modul di buat sesuai dengan identitas untuk memudahkan pelacakan.   2. Nama fungsi dan prosedur yang ada di dalam modul di jelaskan pada dokumentasi modul. | * Mengetahui cara mendokumentasikan modul. * Mengetahui nama fungsi dan prosedur dan dijelaskan pada dokumentasi modul. | * Mampu melakukan dokumentasi modul program. | * Mengikuti pelatihan sesuai bahan yang telah diberikan dengan seksama. * Menjaga ketertiban selama proses pelatihan. | 2 | 5 |

* 1. **Unit Kompetensi : Membuat Halaman Web Dinamis Dasar**

**Kode Unit : TIK.PR04.002.01**

**Durasi Pembelajaran : 12 Jam (@60 menit)**

| **ELEMEN KOMPETENSI** | **KRITERIA UNJUK KERJA** | **MATERI PELATIHAN** | | | **JAM PELATIHAN** | |
| --- | --- | --- | --- | --- | --- | --- |
| **PENGETAHUAN** | **KETERAMPILAN** | **SIKAP KERJA** | **PENGE-TAHUAN** | **KETERAM-PILAN** |
| 01 Mempersiapkan lingkungan teknis | 1. Lingkungan pengembangan dan *software server* tersedia dan dapat diakses. 2. Akses atau *server* jaringan disesuaikan dengan kebijaksanaan dan prosedur yang ditetapkan organisasi. 3. Direktori *virtual* dibuat dan disimpan dalam direktori yang tepat untuk akses *server*. | * Konsep *server* *side* *scripting* * Instalasi *web* *server* dan *program* *edit* *web* *page* * *Setting* dan penggunaan *web server program* *edit* *web* *page* | * Melakukan persiapan dan proses instalasi *web* *server* dan *program* *edit* *web* *page* * Mengaktivasi *service web* *server* | * Memperhatik-an tuntutan spesifikasi aplikasi *web* yang dibuat | 1 | 1,5 |
| 02 Membuat halaman dinamis | * 1. Halaman dibuat dan disimpan dengan ekstensi yang tepat kemudian disimpan ke dalam direktori/*folder* yang tepat.   2.2 Kode ditambahkan di antara *tag* pembatas.     * 1. *Break* dan *indent* yang tepat pada baris kode dibuat ke dalam baris untuk memudahkan pembacaan.   2. *Tag HTML* digunakan pada format halaman dan konten ditambahkan sesuai dengan kebutuhan.   3. Halaman disimpan dalam direktori yang tepat.   4. Akses *file* melalui *http://localhost*, alamat *IP* lokal, atau *URL* lengkap ditentukan.   5. Modifikasi dibuat sesuai kebutuhan. | * *Tags* *HTML* yang umum digunakan * Perbedaan, karakteristik dan keuntungan yang diperoleh membuat halaman *web* dinamis * Peletakan *file*-*file* *web* sehingga memiliki struktur yang baik dan terorganisir * Menampilkan *web* yang dibuat | * Membuat halaman *web* dengan *tag* *HTML* yang sesuai * Mengimplementasikan penggunaan halaman *web* dinamis dengan tepat * Membuat struktur *web* yang terorganisir * Mencoba tampilan *web* dengan *browser* | * Melakukan perencanaan dari struktur *web* yang akan dibuat * Membuat halaman *web* sesuai permintaan * Mencoba mengakses *web* yang dibuat | 2 | 2 |
| 03 Menambahkan fungsi-fungsi pada halaman dinamis | 1. Variabel-variabel dan *array* digunakan dan dideklarasikan menurut persyaratan. 2. Nilai variabel, *string* dan *konstansta* ditetapkan. 3. Fungsi-fungsi internal digunakan pada halaman *web*. 4. *Control structure* (seperti *statement If,* *While, for, swicth*) digunakan sesuai kebutuhan. | * + Konsep pemrograman dasar, seperti variabel, *array*, fungsi, dan *control structure*   + Penggunaan konsep pemrograman dasar tersebut sesuai sintaks bahasa pemrograman *web* yang digunakan   + Mengetahui fungsi-fungsi standar yang digunakan dan mampu membuat fungsi sendiri | * + Membuat variabel dan *array* sesuai keperluan   + Menggunakan *control structure* untuk validasi dan proses perulangan yang diperlukan   + Menggunakan fungsi *built-in* dan membuat fungsi sendiri | * + Mengidentifi-kasikan fungsi-fungsi yang diperlukan pada halaman dinamis * Menggunakan sintaks-sintaks dasar pemrograman tersebut sesuai keperluan * Mampu mengecek logika *program* yang dibuat dan men*debug*nya | 1 | 1,5 |
| 04 Menguji dan mengakhiri halaman dinamis | * 1. Halaman disimpan pada *server* di lokasi yang tepat.   2. Halaman dinamis ditampilkan pada *browser* dan diuji.   3. Fungsi format dinamis secara keseluruhan dibuat sesuai dengan kebutuhan bisnis dan pelanggan.   4. Halaman klien diakhiri sesuai persyaratan bisnis. | * Menginstall *web* pada *server* sesuai konfigurasi yang   diperlukan   * Menampilkan dan mengetes aplikasi *web*      * Implementasi *web* sesuai kebutuhan bisnis klien | * Implementasi aplikasi *web* yang dibuat pada   *environment server* *hosting*   * Melakukan testing pada aplikasi *web* apakah sudah sesuai dengan keinginan klien | * Memperhatik-an perbedaan lingkungan tes   lokal dan lingkungan *server* *hosting* sehingga implementasi berhasil   * Membuat aplikasi *web* yang berhasil memenuhi keinginan klien | 1 | 2 |