

DAFTAR UNIT KOMPETENSI CODER

Sektor : **Kominfo (TIK)**
Sub Sektor : **Teknologi Informasi**
Area Pekerjaan : **Pembangunan dan Implementasi Perangkat Lunak**
Bidang : **Pemrograman**

- A. Melaksanakan penulisan kode program pada kegiatan pembangunan instalasi jaringan lunak sesuai dengan kebutuhan, rencana dan jadwal yang sudah ada;
- B. Melaksanakan penulisan kode program pada kegiatan pengembangan instalasi jaringan lunak sesuai dengan kebutuhan, rencana dan jadwal yang sudah ada;
- C. Melaksanakan penulisan kode program pada kegiatan pemeliharaan instalasi jaringan lunak sesuai dengan kebutuhan, rencana dan jadwal yang sudah ada;
- D. Melaksanakan penulisan kode program pada kegiatan perbaikan instalasi jaringan lunak sesuai dengan kebutuhan, rencana dan jadwal yang sudah ada;
- E. Melaporkan hasil kegiatan secara berkala kepada pimpinan

SKKNI 2016-282

- 1. J.620100.003.01 Melakukan Identifikasi Library, Komponen atau Framework yang Diperlukan
- 2. J.620100.005.02 Mengimplementasikan User Interface
- 3. J.620100.010.01 Menerapkan Perintah Eksekusi Bahasa Pemrograman Berbasis Teks, Grafik, dan Multimedia
- 4. J.620100.014.01 Menerapkan Metode dan Praktik Penggunaan Kembali (Reusable) Subrutin-Subrutin
- 5. J.620100.016.01 Menulis Kode dengan Prinsip Sesuai Guidelines dan Best Practices
- 6. J.620100.017.02 Mengimplementasikan Pemrograman Terstruktur
- 7. J.620100.018.02 Mengimplementasikan Pemrograman Berorientasi Objek
- 8. J.620100.019.02 Menggunakan Library atau Komponen PreExisting
- 9. J.620100.021.02 Menerapkan akses basis data
- 10. J.620100.022.02 Mengimplementasikan Algoritma Pemrograman
- 11. J.620100.023.02 Membuat Dokumen Kode Program
- 12. J.620100.025.02 Melakukan Debugging
- 13. J.620100.026.01 Menggunakan Source Code Versioning
- 14. J.620100.033.02 Melaksanakan Pengujian Unit Program

A. SILABUS PELATIHAN BERBASIS KOMPETENSI

I. KELOMPOK UNIT KOMPETENSI

- 1.1 Unit Kompetensi : Melakukan Identifikasi Library, Komponen atau Framework yang Diperlukan
 Kode Unit : J.620100.003.01
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Menganalisis library, komponen, atau framework yang sesuai dengan konteks 2. Membuat proof of concept library, komponen atau Framework berdasarkan konteks kebutuhan 3. Merancang integrasi dan batasan penggunaan library, komponen atau framework	1. Pemahaman analisa library, komponen dan framework yang akan digunakan 2. Pemahaman fungsi dan kegunaan library, komponen dan framework	1. Mengidentifikasi library, komponen atau framework yang dibutuhkan 2. Memahami fungsi library, komponen dan framework sesuai kegunaannya	1. Analisis system requirements, library, komponen dan framework 2. Pengertian library, jenis dan fungsinya 3. Pengertian komponen, jenis dan fungsinya 4. Pengertian framework, jenis dan fungsinya
Asesmen			

*) coret yang tidak perlu

- 1.2 Unit Kompetensi : Mengimplementasikan User Interface

Kode Unit : J.620100.005.02
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Mengidentifikasi rancangan user interface 2. Melakukan implementasi rancangan user interface	1. Identifikasi rancangan user interface dari program yang sudah akan dibangun 2. Pemahaman terhadap urutan dari akses komponen user Interface dialog 3. Kesesuaian rancangan rancangan user interface sesuai dengan implementasinya	1. Mengidentifikasi rancangan user interface yang sudah ada 2. Memahami urutan akses komponen user interface dialog 3. Mengimplementasikan user interface sesuai dengan rancangan	1. Komponen pada user interface 2. Alur akses komponen pada user interface 3. Menerapkan rancangan user interface dari aplikasi yang akan dikembangkan
Asesmen			

*) *coret yang tidak perlu*

1.3 Unit Kompetensi : Menerapkan Perintah Eksekusi Bahasa Pemrograman Berbasis Teks, Grafik, dan Multimedia
 Kode Unit : J.620100.010.01
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Mengidentifikasi mekanisme running atau eksekusi source code 2. Mengeksekusi source code 3. Mengidentifikasi hasil eksekusi	1. Kesesuaian tools yang digunakan untuk melakukan eksekusi terhadap source code yang dibuat sesuai dengan spesifikasi yang dibutuhkan 2. Kesesuaian mekanisme eksekusi source code yang dibuat dengan tools yang digunakan. 3. Source code dapat dieksekusi dengan baik sesuai skenario yang direncanakan.	1. Memilih tools yang akan digunakan untuk melakukan proses eksekusi source code yang telah dibuat sesuai dengan spesifikasi dari tools tersebut. 2. Melakukan langkah kerja untuk mengeksekusi source code yang dibuat menggunakan tools yang telah dipilih. 3. Melakukan langkah kerja untuk dapat memastikan source code dapat dieksekusi dengan baik sesuai skenario yang direncanakan	1. Klasifikasi dan Jenis-jenis Bahasa Pemrograman. 2. Pengertian running, debugging, compiling, interpreting, dan pembuatan executable file 3. Perangkat aplikasi yang digunakan untuk melakukan proses eksekusi bahasa-bahasa pemrograman 4. Cara melakukan eksekusi source code dari bahasa-bahasa pemrograman
Asesmen			

*) coret yang tidak perlu

1.4 Unit Kompetensi : Menerapkan Metode dan Praktik Penggunaan Kembali (Reusable)
 Subrutin-Subrutin
 Kode Unit : J.620100.014.01
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Mengimplemen- tasikan fungsi/prosedu- r/modul yang bisa digunakan secara berulang 2. Menggunakan reusable fungsi/prosedu- r/modul	1. Mengetahui jenis passing parameter berdasarkan alamat atau nilai parameternya 2. Mampu mengimplementasik- an fungsi/prosedur reusable	1. Mengetahui jenis passing parameter berdasarkan alamat atau nilai parameternya 2. Mengimplemen- tasikan fungsi/prosedur/ modul yang reusable 3. Menentukan dan menggunakan reusable fungsi/prosedur/ modul yang telah ada	1. Pengertian dan jenis passing parameter 2. Pengenalan konsep penggunaan Kembali (reuse) pada pemrograman 3. Bagaimana menentukan reusable fungsi/prosedur/modul 4. Penggunaan reusable fungsi/prosedur/modul
Asesmen			

*) coret yang tidak perlu

1.5 Unit Kompetensi : Menulis Kode dengan Prinsip Sesuai Guidelines dan Best Practices
 Kode Unit : J.620100.016.01
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
-------------------	-------------------------	------------------	------------------

1. Menerapkan coding-guidelines dan best practices dalam penulisan program (kode sumber) 2. Menggunakan ukuran performansi dalam menuliskan kode sumber	1. Menulis source code sesuai prinsip penulisan kode 2. Menangani galat/error 3. Menerapkan efisiensi penggunaan memori	1. Menulis source code meliputi penamaan, komentar dan indentasi sesuai prinsip penulisan kode 2. Menangani galat/error sesuai Bahasa pemrograman terkait 3. Menerapkan efisiensi memori dan kemudahan interaksi	1. Coding Guidelines 2. Paradigma Pemrograman 3. Cara menulis source code sesuai guidelines dan best practices 4. Jenis-jenis dan penanganan galat/error 5. Efisiensi penggunaan memori dalam menuliskan kode
Asesmen			

*) coret yang tidak perlu

1.6 Unit Kompetensi : Mengimplementasikan Pemrograman Terstruktur
Kode Unit : J.620100.017.02
Perkiraan Waktu Pelatihan : 3 JP @50 menit
Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Menggunakan tipe data dan kontrol program 2. Membuat program sederhana	1. Penentuan variabel dan tipe data sesuai dengan jenis-jenisnya	1. Membuat program baca tulis sederhana 2. Menerapkan struktur kontrol percabangan dan	1. Penjelasan dan jenis-jenis tipe data pemrograman 2. Algoritma pemrograman

3. Membuat program menggunakan prosedur dan fungsi 4. Membuat program menggunakan array 5. Membuat program untuk akses file 6. Mengkompilasi program	2. Membuat program sederhana terstruktur 3. Menerapkan prosedur, fungsi dan array pada program 4. Mengakses file melalui program sederhana.	perulangan dalam pembuatan program sederhana 3. Menerapkan prosedur dan fungsi sesuai aturan penulisan program 4. Menerapkan array dalam pembuatan program sederhana 5. Membuat program untuk mengakses data yang ada di media penyimpanan 6. Mengatasi permasalahan pada program	3. Source code membuat program input dan output menggunakan keyboard 4. Struktur kontrol percabangan dan perulangan 5. Cara membuat prosedur dan/atau fungsi sesuai aturan penulisan program 6. Tipe data array, jenis dimensi array dan cara implementasinya 7. Cara mengurutkan array 8. Jenis media penyimpanan data 9. Implementasi akses file untuk membaca dan menulis data dari/ke media penyimpanan 10. Jenis kesalahan pada program 11. Cara mengatasi masalah pada program
Asesmen			

**) coret yang tidak perlu*

1.7 Unit Kompetensi : Mengimplementasikan Pemrograman Berorientasi Objek
Kode Unit : J.620100.018.02
Perkiraan Waktu Pelatihan : 3 JP @50 menit

Metode Pelatihan

: ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
<ol style="list-style-type: none"> 1. Membuat program berorientasi objek dengan memanfaatkan class 2. Menggunakan tipe data dan control program pada metode atau operasi dari suatu kelas 3. Membuat program dengan konsep berbasis objek 4. Membuat program object-oriented dengan interface dan paket 5. Mengkompilasi program 	<ol style="list-style-type: none"> 1. Membuat kode program dengan konsep Object Oriented Programming 2. Membuat kode program yang bebas dari kesalahan/error 	<ol style="list-style-type: none"> 1. Class yang dibuat sesuai dengan coding guideline dan best practice 2. Class member yang dibuat sesuai dengan kebutuhan kasus yang diberikan 3. Object digunakan pada kode program sesuai dengan coding guideline dan best practice 4. Pemanfaatan konsep Object Oriented Programming seperti inheritance, polymorphism, overloading, interface, dan package dilakukan sesuai dengan coding guideline dan best practice 5. Proses debugging dilakukan dengan benar dan tepat untuk memastikan kode program terbebas dari kesalahan/error 	<ol style="list-style-type: none"> 1. Pengenalan konsep class dan object 2. Cara membuat class pada bahasa pemrograman yang digunakan 3. Macam-macam class Member 4. Macam-macam access modifier/hak akses 5. Cara membuat class member pada bahasa pemrograman yang digunakan 6. Cara instansiasi object pada bahasa pemrograman yang digunakan 7. Pengenalan, Inheritance, Polymorphism, Overloading, Overriding, Interface, Package
Asesmen			

**) coret yang tidak perlu*

1.8 Unit Kompetensi : Menggunakan Library atau Komponen PreExisting
 Kode Unit : J.620100.019.02
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Melakukan pemilihan unit-unit reuse yang potensial 2. Melakukan integrasi library atau komponen pre-existing dengan source code yang ada 3. Melakukan pembaharuan library atau komponen pre-existing yang digunakan	1. Menggunakan komponen-komponen reuse untuk mendukung pengembangan aplikasi	1. Memilih library yang sesuai 2. Menginstal library menggunakan composer 3. Memperbaharui library	1. Penjelasan terkait library atau komponen pre-existing 2. Jenis-jenis library atau komponen pre-existing 3. Penjelasan lisensi library 4. Pemilihan library yang sesuai dengan source code 5. Penggunaan composer untuk instalasi library 6. Memperbaharui library yang sudah obsolete
Asesmen			

**) coret yang tidak perlu*

1.9 Unit Kompetensi : Menerapkan akses basis data

Kode Unit : J.620100.021.02
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Membuat berbagai operasi terhadap basis data 2. Membuat prosedur akses terhadap basis data 3. Membuat koneksi basis data 4. Menguji program basis data	1.	1. Menyimpan data dalam format basis data 2. Menghasilkan Informasi yang diinginkan menggunakan query tersebut 3. Mempergunakan Indeks untuk mempercepat akses 4. Menerapkan Library akses basis data 5. Menentukan keamanan koneksi dan hak setiap pengguna 6. Menyiapkan skenario pengujian basis data	
Asesmen			

*) *coret yang tidak perlu*

1.10 Unit Kompetensi : Mengimplementasikan Algoritma Pemrograman
 Kode Unit : J.620100.022.02
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Menjelaskan varian dan invarian 2. Membuat alur logika pemrograman 3. Menerapkan teknik dasar algoritma umum 4. Menggunakan prosedur dan fungsi 5. Mengidentifikasi kompleksitas algoritma	1. Mengimplemen tasikan rancangan alur logika pemrograman (flowchart) ke dalam kode program	1. Menulis kode program dengan penggunaan variabel dan konstanta sesuai coding guideline dan best practice 2. Menulis kode program dengan penggunaan tipe sesuai coding guideline dan best practice 3. Merancang flowchart sesuai dengan kasus yang diberikan 4. Menulis kode program sesuai dengan flowchart yang dibuat 5. Membuat prosedur dan fungsi sesuai dengan konsep	1. Konsep Dasar Pemrograman. 2. Fungsi dan Cara Kerja Algoritma Pemrograman 3. Pengertian Syntax dan Elemen Fungsinya 4. Pengertian, Fungsi dan Struktur Pseudocode 5. Pengenalan variabel dan konstanta pada bahasa pemrograman yang digunakan 6. Pengenalan tipe data pada bahasa pemrograman yang digunakan 7. Macam-macam struktur dasar algoritma

		<p>penggunaan kembali pada pemrograman</p> <p>6. Menggunakan prosedur dan fungsi sesuai dengan coding guideline dan best practice</p>	<p>8. Pengenalan flowchart</p> <p>9. Cara merancang flowchart</p> <p>10. Macam-macam control program pada bahasa pemrograman yang digunakan</p> <p>11. Cara mengimplementasikan flowchart ke dalam kode program</p>
Asesmen			

*) coret yang tidak perlu

1.11 Unit Kompetensi : Membuat Dokumen Kode Program
 Kode Unit : J.620100.023.02
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Melakukan identifikasi kode program 2. Membuat dokumentasi modul program 3. Membuat dokumentasi fungsi, prosedur, atau method program	1. Mengidentifikasi modul program 2. Mengidentifikasi parameter yang digunakan 3. Menjelaskan cara kerja algoritma 4. Memberikan Komentar	1. Mampu menjelaskan modul-modul dalam suatu program. 2. Mampu mengidentifikasi modul-modul program 3. Mampu menjelaskan parameter yang dipergunakan.	1. Modul-modul dalam suatu program 2. Parameter yang digunakan 3. Pengertian dan cara kerja algoritma 4. Penggunaan komentar pada program 5. Pembuatan dokumentasi modul sesuai dengan identitas untuk memudahkan pelacakan.

4. Men-generate dokumentasi	<p>setiap baris kode termasuk data, eksepsi, fungsi, prosedur dan class(bila ada)</p> <p>5. Membuat dokumentasi modul sesuai dengan identitas untuk memudahkan pelacakan.</p> <p>6. Menerapkan identifikasi dokumen</p> <p>7. Menjelaskan kegunaan modul</p> <p>8. Merevisi dokumen sesuai perubahan kode program</p> <p>9. Membuat dokumentasi fungsi, prosedur atau metod</p> <p>10. Menjelaskan kemungkinan eksepsi</p> <p>11. Mengidentifika si tool untuk generate dokumentasi</p> <p>12. Melakukan generate dokumentasi</p>	<p>4. Mampu mengidentifikasi Parameter yang dipergunakan</p> <p>5. Menjelaskan algoritma</p> <p>6. Mampu menunjukkan cara kerja algoritma</p> <p>7. Mampu menjelaskan kegunaan komentar pada program.</p> <p>8. Mampu memberikan komentar pada setiap baris kode.</p> <p>9. Mampu menjelaskan pembuatan dokumentasi modul sesuai dengan identitas untuk memudahkan pelacakan.</p> <p>10. Mampu membuat dokumentasi modul sesuai dengan identitas untuk memudahkan pelacakan</p> <p>11. Mampu menjelaskan cara mengidentifikasi dokumentasi.</p> <p>12. Mampu menerapkan identifikasi dokumentasi.</p>	<p>6. Cara mengidentifikasi dokumentasi</p> <p>7. Kegunaan modul</p> <p>8. Cara merevisi dokumen sesuai perubahan kode program</p> <p>9. cara membuat dokumentasi fungsi, prosedur atau metod</p> <p>10. Penjelasan kemungkinan eksepsi</p> <p>11. Revisi dokumen sesuai perubahan kode program</p> <p>12. Tools untuk generate dokumentasi</p> <p>13. Cara men-generate dokumentasi</p>
-----------------------------	---	---	--

		<p>13. Mampu menjelaskan kegunaan modul.</p> <p>14. Mampu mengidentifikasi kegunaan modul</p> <p>15. Mampu menjelaskan cara merevisi dokumen sesuai perubahan kode program.</p> <p>16. Mampu merevisi dokumen sesuai perubahan kode program.</p> <p>17. Mampu menjelaskan cara membuat dokumentasi fungsi, prosedur atau metod.</p> <p>18. Mampu membuat dokumentasi fungsi, prosedur atau metod</p> <p>19. Mampu menjelaskan Kemungkinan eksepsi.</p> <p>20. Mampu mengidentifikasi kemungkinan eksepsi.</p> <p>21. Mampu menjelaskan revisi dokumen sesuai perubahan kode program.</p> <p>22. Mampu merevisi dokumen sesuai</p>	
--	--	---	--

		perubahan kode program. 23. Mampu menjelaskan tools untuk generate dokumentasi. 24. Mampu mengidentifikasi tools untuk generate dokumentasi. 25. Mampu menjelaskan cara men-generate dokumentasi. 26. Mampu men-generate dokumentasi.	
Asesmen			

*) coret yang tidak perlu

1.12 Unit Kompetensi : Melakukan Debugging
 Kode Unit : J.620100.025.02
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Mempersiapkan kode program 2. Melakukan debugging	1.	1. Menyiapkan Kode program sesuai spesifikasi 2. Menyiapkan Debugging tools untuk melihat proses suatu modul	1. Cara menyiapkan kode program sesuai spesifikasi. 2. Cara menyiapkan debugging tools untuk melihat proses suatu modul. 3. Cara melakukan kompilasi kode program sesuai bahasa

3. Memperbaiki program		3. Menyiapkan Kode program dikompilasi sesuai bahasa pemrograman 4. Menganalisa Kriteria lulus build 5. Menganalisa Kriteria eksekusi aplikasi 6. Mencatat kode kesalahan 7. Merumuskan perbaikan terhadap kesalahan kompilasi maupun build 8. Mencatat perbaikan kesalahan	pemrograman yang digunakan. 4. Cara menganalisa kriteria lulus build. 5. Cara menganalisa kriteria eksekusi aplikasi. 6. Cara menganalisa kode kesalahan 7. Cara merumuskan perbaikan terhadap kesalahan kompilasi maupun build. 8. Cara melakukan perbaikan
Asesmen			

**) coret yang tidak perlu*

1.13 Unit Kompetensi : Menggunakan Source Code Versioning
Kode Unit : J.620100.026.01
Perkiraan Waktu Pelatihan : 3 JP @50 menit
Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Menerapkan konsep/metode pencatatan versi dari setiap program sumber	1.	1. Pengertian konsep penerapan versi kode program dapat dijelaskan. 2. Proses branching, merging, commit, check-in, check-out	1. Pemahaman konsep versi kode program 2. Pengertian branching, merging, commit, check-in, check-out dan cloning. 3. Konsep repository 4. Kegunaan Tools penyimpanan versi

2. Menggunakan suatu tools untuk menyimpan versi		<p>dan cloning dapat dijelaskan.</p> <p>3. Konsep repository dapat dijelaskan.</p> <p>4. Guna dari alat/tools dapat ditunjukkan.</p> <p>5. Alat/tools dapat diusulkan.</p> <p>6. Karakteristik dari tools/alat dapat dijelaskan atau ditunjukkan.</p> <p>7. Proses branching, merging, commit, check-in, check-out dan cloning dilakukan.</p>	<p>5. Tools penyimpanan versi yang diusulkan</p> <p>6. Karakteristik Tool penyimpanan versi</p> <p>7. Instalasi VCS</p> <p>8. Proses branching, merging, commit, check-in, check-out dan cloning.</p>
Asesmen			

**) coret yang tidak perlu*

1.14 Unit Kompetensi : Melaksanakan Pengujian Unit Program
 Kode Unit : J.620100.033.02
 Perkiraan Waktu Pelatihan : 3 JP @50 menit
 Metode Pelatihan : ~~Luring~~/Daring/~~Blended~~(*)

ELEMEN KOMPETENSI	CAPAIAN UNIT KOMPETENSI	KRITERIA CAPAIAN	POKOK PEMBAHASAN
1. Menentukan kebutuhan uji coba dalam pengembangan	1.	1. Mengidentifikasi Prosedur uji coba aplikasi sesuai dengan software development life cycle	1.

2. Mempersiapkan dokumentasi uji coba 3. Mempersiapkan data uji 4. Melaksanakan prosedur uji coba 5. Mengevaluasi hasil uji coba		2. Menentukan tools uji coba 3. Mengidentifikasi Standard dan kondisi uji coba 4. Menentukan kebutuhan-Kebutuhan untuk uji coba 5. Melaksanakan Uji coba dengan variasi kondisi 6. Membuat Skenario uji coba 7. Mengidentifikasi Data uji unit tes 8. Membangkitkan Data uji unit tes 9. Mendesain Skenario uji coba 10. Mendesain Prosedur uji coba dalam algoritma 11. Melaksanakan Uji coba 12. Mencatat Hasil uji coba 13. Menganalisis Hasil uji coba 14. Melaporkan Prosedur uji coba 15. Menyelesaikan Kesalahan/error	
Asesmen			

**) coret yang tidak perlu*