



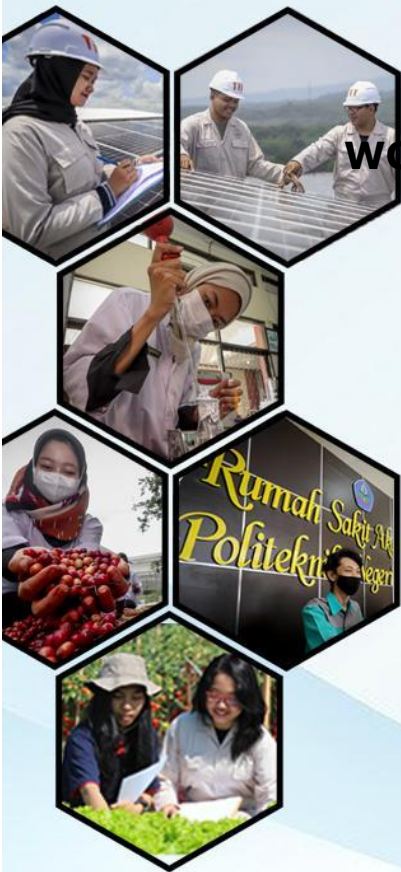
# **BUKU KERJA PRAKTIK MAHASISWA (BKPM)**

**WORKSHOP KUALITAS PERANGKAT LUNAK  
TIF130705  
SEMESTER 3**

**OLEH :**

- 1. Nanik Anita Mukhlisoh, S.ST., M.T.**
- 2. Mukhamad Angga Gumilang, S. Pd., M. Eng.**
- 3. Ahmad Fahriyannur Rosyady, S. Kom., M.T.**

**PROGRAM STUDI TEKNIK INFORMATIKA KAMPUS 3 NGANJUK  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI JEMBER  
TAHUN 2022**



LEMBAR PENGESAHAN

**WORKSHOP KUALITAS PERANGKAT LUNAK**

Mengetahui,

Koord. Program Studi,

Koord./Tim Mata Kuliah,

Penulis,



Ulfa Emi R., S.Kom., M.Kom  
NIP. 199706282022032018



Nanik Anita M., S.ST., M.T  
NIP. 198606092008122004



Nanik Anita M., S.ST., M.T  
NIP. 198606092008122004

Menyetujui,

Ketua Jurusan Teknologi Informasi

Hendra Yufit Riskiawan, S.Kom., M.Cs.  
NIP. 198302032006041003

**KATA PENGANTAR**

Puji Syukur kehadiran Tuhan Yang Maha Esa karena atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan Buku Kerja Praktikum Mahasiswa (BKPM) Workshop Kualitas Perangkat Lunak. Materi Workshop terdiri dari Quality Assurance Concepts, Menggunakan Tool Pengujian Perangkat Lunak, Komponen manajemen kualitas, perangkat lunak, Melakukan Proses Pengujian dalam Siklus Hidup Pengembangan Perangkat Lunak, Integration/System Test Planning, Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Website), Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Mobile), Melakukan Pengujian Kualitas Perangkat Lunak Secara Otomatis (Web), Melakukan Pengujian Kualitas Perangkat Lunak Secara Otomatis (Mobile), Melakukan Pengujian Aspek Kebergunaan Kepada Pengguna, Melakukan Pengujian Aspek Penerimaan Kepada Pengguna, Melakukan Pengujian Keamanan Perangkat Lunak, Mengevaluasi Pelaksanaan Penjaminan Kualitas ,Perangkat Lunak, serta Membuat Laporan dan Dokumentasi Pengujian. BKPM ini disusun berdasarkan metode *Student Center Learning* yaitu menempatkan mahasiswa sebagai pusat kegiatan belajar. BKPM ini terdiri dari Capaian Pembelajaran Mata Kuliah (CPMK), Indikator Penilaian, Dasar Teori, Alat dan Bahan, Prosedur Kerja, Hasil dan Pembahasan, dan Rubrik Penilaian.

Penulis menyadari masih banyak kekurangan dalam penyusunan BKPM ini. Oleh karena itu, penulis sangat mengharapkan kritik dan saran demi perbaikan dan kesempurnaan modul ini. Penulis mengucapkan terima kasih kepada berbagai pihak yang telah membantu proses penyelesaian BKPM ini. Semoga BKPM ini dapat bermanfaat bagi kita semua.

Jember, 12 Agustus 2022

Penulis

## DAFTAR ISI

<i>LEMBAR PENGESAHAN.....</i>	<i>ii</i>
<i>KATA PENGANTAR.....</i>	<i>iii</i>
<i>Acara 1 Quality Assurance Concept.....</i>	<i>1</i>
a.    Capaian Pembelajaran Mata Kuliah (CPMK) .....	1
b.    Penilaian Bertahap BNSP (Skill Passport).....	1
c.    Indikator Penilaian.....	1
d.    Dasar Teori .....	1
e.    Alat dan Bahan.....	3
f.    Prosedur Kerja.....	3
g.    Hasil dan Pembahasan.....	4
h.    Rubrik Penilaian.....	4
<i>Acara 2 Quality Assurance Concept .....</i>	<i>5</i>
a.    Capaian Pembelajaran Mata Kuliah (CPMK) .....	5
b.    Penilaian Bertahap BNSP (Skill Passport).....	5
c.    Indikator Penilaian.....	5
d.    Dasar Teori .....	5
e.    Alat dan Bahan.....	7
f.    Prosedur Kerja.....	8
g.    Hasil dan Pembahasan.....	8
h.    Rubrik Penilaian.....	8
<i>Acara 3 Komponen manajemen kualitas perangkat lunak.....</i>	<i>9</i>
a.    Capaian Pembelajaran Mata Kuliah (CPMK) .....	9
b.    Penilaian Bertahap BNSP (Skill Passport).....	9
c.    Indikator Penilaian.....	9
d.    Dasar Teori .....	9
e.    Alat dan Bahan.....	13
f.    Prosedur Kerja.....	13
g.    Hasil dan Pembahasan.....	13
h.    Rubrik Penilaian.....	13
<i>Acara 4 Komponen manajemen kualitas perangkat lunak.....</i>	<i>15</i>
a.    Capaian Pembelajaran Mata Kuliah (CPMK) .....	15
b.    Penilaian Bertahap BNSP (Skill Passport).....	15
c.    Indikator Penilaian.....	15
d.    Dasar Teori .....	15
e.    Alat dan Bahan.....	17
f.    Prosedur Kerja.....	17
g.    Hasil dan Pembahasan.....	18

h. Rubrik Penilaian.....	18
<i>Acara 5</i> Komponen manajemen kualitas perangkat lunak.....	19
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	19
b. Penilaian Bertahap BNSP (Skill Passport).....	19
c. Indikator Penilaian.....	19
d. Dasar Teori .....	19
e. Alat dan Bahan.....	21
f. Prosedur Kerja.....	22
g. Hasil dan Pembahasan.....	22
h. Rubrik Penilaian.....	22
<i>Acara 6</i> Komponen manajemen kualitas perangkat lunak.....	23
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	23
b. Penilaian Bertahap BNSP (Skill Passport).....	23
c. Indikator Penilaian.....	23
d. Dasar Teori .....	23
e. Alat dan Bahan.....	25
f. Prosedur Kerja.....	25
6. Hasil dan Pembahasan.....	25
7. Rubrik Penilaian.....	26
<i>Acara 7</i> Proses Pengujian dalam SDLC.....	27
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	27
b. Penilaian Bertahap BNSP (Skill Passport).....	27
c. Indikator Penilaian.....	27
d. Dasar Teori .....	27
e. Alat dan Bahan.....	30
f. Prosedur Kerja.....	30
g. Hasil dan Pembahasan.....	30
h. Rubrik Penilaian.....	30
<i>Acara 8</i> Proses Pengujian dalam SDLC.....	32
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	32
b. Penilaian Bertahap BNSP (Skill Passport).....	32
c. Indikator Penilaian.....	32
d. Dasar Teori .....	32
e. Alat dan Bahan.....	35
f. Prosedur Kerja.....	35
g. Hasil dan Pembahasan.....	35
h. Rubrik Penilaian.....	36
<i>Acara 9</i> Perencanaan Proses Pengujian .....	37
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	37

b.	Penilaian Bertahap BNSP (Skill Passport).....	37
c.	Indikator Penilaian.....	37
d.	Dasar Teori .....	37
e.	Alat dan Bahan.....	38
f.	Prosedur Kerja.....	38
g.	Hasil dan Pembahasan.....	39
h.	Rubrik Penilaian.....	40
<i>Acara 10 Perencanaan Proses Pengujian.....</i>		<i>41</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	41
b.	Penilaian Bertahap BNSP (Skill Passport).....	41
c.	Indikator Penilaian.....	41
d.	Dasar Teori .....	41
e.	Alat dan Bahan.....	42
f.	Prosedur Kerja.....	42
g.	Hasil dan Pembahasan.....	43
h.	Rubrik Penilaian.....	43
<i>Acara 11 Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Web).....</i>		<i>44</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	44
b.	Penilaian Bertahap BNSP (Skill Passport).....	44
c.	Indikator Penilaian.....	44
d.	Dasar Teori .....	44
e.	Alat dan Bahan.....	46
f.	Prosedur Kerja.....	46
g.	Hasil dan Pembahasan.....	46
h.	Rubrik Penilaian.....	46
<i>Acara 12 Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Web).....</i>		<i>48</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	48
b.	Penilaian Bertahap BNSP (Skill Passport).....	48
c.	Indikator Penilaian.....	48
d.	Dasar Teori .....	48
e.	Alat dan Bahan.....	49
f.	Prosedur Kerja.....	50
g.	Hasil dan Pembahasan.....	50
h.	Rubrik Penilaian.....	50
<i>Acara 13 Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Mobile) .....</i>		<i>51</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	51
b.	Penilaian Bertahap BNSP (Skill Passport).....	51
c.	Indikator Penilaian.....	51
d.	Dasar Teori .....	51



e.	Alat dan Bahan.....	53
f.	Prosedur Kerja.....	53
g.	Hasil dan Pembahasan.....	53
h.	Rubrik Penilaian.....	54
<i>Acara 14 Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Mobile).....</i>		<i>55</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	55
b.	Penilaian Bertahap BNSP (Skill Passport).....	55
c.	Indikator Penilaian.....	55
d.	Dasar Teori Dapat dilihat pada Minggu ke-6/2, Acara 12. ....	55
e.	Alat dan Bahan.....	55
f.	Prosedur Kerja.....	56
g.	Hasil dan Pembahasan.....	56
h.	Rubrik Penilaian.....	56
<i>Acara 15 dan 16 Melakukan Pengujian Kualitas Perangkat Lunak Secara Otomatis (Web).....</i>		<i>57</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	57
b.	Penilaian Bertahap BNSP (Skill Passport).....	57
c.	Indikator Penilaian.....	57
d.	Dasar Teori .....	57
e.	Alat dan Bahan.....	59
f.	Prosedur Kerja.....	59
a.	Hasil dan Pembahasan.....	62
b.	Rubrik Penilaian.....	62
<i>Acara 17 dan 18 Melakukan Pengujian Kualitas Perangkat Lunak Secara Otomatis (Mobile).....</i>		<i>64</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	64
b.	Penilaian Bertahap BNSP (Skill Passport).....	64
c.	Indikator Penilaian.....	64
d.	Dasar Teori .....	64
e.	Alat dan Bahan.....	64
f.	Prosedur Kerja.....	64
g.	Hasil dan Pembahasan.....	71
h.	Rubrik Penilaian.....	71
<i>Acara 19 Pengujian Aspek Kebergunaan Kepada Pengguna.....</i>		<i>72</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	72
b.	Penilaian Bertahap BNSP (Skill Passport).....	72
c.	Indikator Penilaian.....	72
d.	Dasar Teori .....	72
e.	Alat dan Bahan.....	75
f.	Prosedur Kerja.....	75
g.	Hasil dan Pembahasan.....	75

h. Rubrik Penilaian.....	76
<i>Acara 20 Pengujian Aspek Kebergunaan Kepada Pengguna.....</i>	<i>77</i>
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	77
b. Penilaian Bertahap BNSP (Skill Passport).....	77
c. Indikator Penilaian.....	77
d. Dasar Teori .....	77
e. Alat dan Bahan.....	79
f. Prosedur Kerja.....	80
g. Hasil dan Pembahasan.....	80
h. Rubrik Penilaian.....	80
<i>Acara 21 Pengujian Aspek Penerimaan Kepada Pengguna .....</i>	<i>81</i>
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	81
b. Penilaian Bertahap BNSP (Skill Passport).....	81
c. Indikator Penilaian.....	81
d. Dasar Teori .....	81
e. Alat dan Bahan.....	83
f. Prosedur Kerja.....	83
g. Hasil dan Pembahasan.....	84
h. Rubrik Penilaian.....	84
<i>Acara 22 Pengujian Aspek Penerimaan Kepada Pengguna .....</i>	<i>85</i>
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	85
b. Penilaian Bertahap BNSP (Skill Passport).....	85
c. Indikator Penilaian.....	85
d. Dasar Teori .....	85
e. Alat dan Bahan.....	85
f. Prosedur Kerja.....	85
g. Hasil dan Pembahasan.....	87
h. Rubrik Penilaian.....	89
<i>Acara 23 Keamanan Perangkat Lunak.....</i>	<i>90</i>
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	90
b. Penilaian Bertahap BNSP (Skill Passport).....	90
c. Indikator Penilaian.....	90
d. Dasar Teori .....	90
e. Alat dan Bahan.....	93
f. Prosedur Kerja.....	93
g. Hasil dan Pembahasan.....	94
h. Rubrik Penilaian.....	94
<i>Acara 24 Keamanan Perangkat Lunak.....</i>	<i>95</i>
a. Capaian Pembelajaran Mata Kuliah (CPMK) .....	95



b.	Penilaian Bertahap BNSP (Skill Passport).....	95
c.	Indikator Penilaian.....	95
d.	Dasar Teori .....	95
e.	Alat dan Bahan.....	100
f.	Prosedur Kerja.....	100
g.	Hasil dan Pembahasan.....	100
h.	Rubrik Penilaian.....	100
<i>Acara 25 dan 26 Mengevaluasi Pelaksanaan Penjaminan Kualitas Perangkat Lunak.....</i>		<i>101</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	101
b.	Penilaian Bertahap BNSP (Skill Passport).....	101
c.	Indikator Penilaian.....	101
d.	Dasar Teori .....	101
e.	Alat dan Bahan.....	102
f.	Prosedur Kerja.....	102
g.	Hasil dan Pembahasan.....	103
h.	Rubrik Penilaian.....	103
<i>Acara 27 dan 28 Membuat Laporan dan Dokumentasi Pengujian.....</i>		<i>104</i>
a.	Capaian Pembelajaran Mata Kuliah (CPMK) .....	104
b.	Penilaian Bertahap BNSP (Skill Passport).....	104
c.	Indikator Penilaian.....	104
d.	Dasar Teori .....	104
e.	Alat dan Bahan.....	106
f.	Prosedur Kerja.....	106
g.	Hasil dan Pembahasan.....	108
h.	Rubrik Penilaian.....	108

## Acara 1

Pokok Bahasan	: Quality Assurance Concepts
Acara Praktikum/Praktik	: Minggu 1 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 Menit

### a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengidentifikasi Dasar dan Prinsip Pengujian Perangkat Lunak
2. Mahasiswa mampu menentukan Dasar dan Prinsip Pengujian Perangkat Lunak pengembangan perangkat lunak
3. Mahasiswa mampu menganalisa acuan-acuan Dasar dan Prinsip Pengujian Perangkat Lunak

### b. Penilaian Bertahap BNSP (Skill Passport)

Nama SkemaSertifikasi:-

### c. Indikator Penilaian

Ketepatan menentukan acuan-acuan dasar dan prinsip dasar pengujian perangkat lunak

### d. Dasar Teori

Pengujian perangkat lunak merupakan suatu investasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji. Tujuannya untuk memberikan pandangan secara objektif dan independen yang bermanfaat dalam operasional bisnis untuk memahami tingkat resiko pada implementasinya. Pengujian perangkat lunak merupakan tahapan penting dalam pembangunan perangkat lunak.

Pengujian dilakukan dengan cara evaluasi konfigurasi (pembentukan susunan) yang terdiri dari :

- Spesifikasi kebutuhan
- Deskripsi perancangan
- Program yang dihasilkan

Hasil dari evaluasi konfigurasi kemudian dibandingkan dengan hasil uji yang diharapkan. Dan berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat atau baru melakukan pengujian.

Strategi software testing dapat dilakukan secara manual atau otomatis. Menurut strategi pengujian manual, pendekatan yang lebih tradisional, penguji menyiapkan suite tes yang menurut mereka akan lebih baik menjalankan program. Alat pengujian perangkat lunak otomatis membantu dalam menghasilkan kasus uji dari spesifikasi program atau dari data teks aslinya. Pengujian manual adalah teknik pengujian di mana insinyur penguji menyiapkan test case secara manual dan mengeksekusinya untuk mengidentifikasi cacat pada perangkat lunak. Mengotomatiskan pengujian perangkat lunak menggunakan bahasa scripting seperti Python, JavaScript atau Tool Command Language karena kasus uji dapat dengan mudah dieksekusi oleh mesin dengan sedikit intervensi manusia dan perhatian.

### **Sasaran-sasaran Pengujian Perangkat Lunak**

Pengujian perangkat lunak memiliki aturan yang berfungsi sebagai sasaran. Yaitu :

1. Pengujian : yaitu proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. Test Case : yaitu sekumpulan dari input test atau kondisi yang akan diuji. Kondisi yang baik adalah yang memiliki probabilitas tinggi untuk menentukan kesalahan yang belum pernah ditentukan sebelumnya.
3. Pengujian yang sukses : yaitu pengujian yang mengungkap semua kesalahan yang belum pernah ditentukan sebelumnya.

Prinsip-prinsip Pengujian Perangkat Lunak

- Semua pengujian harus dapat ditelusuri sampai kepada persyaratan pelanggan.
- Pengujian harus direncanakan jauh hari sebelum rencana itu dimulai.
- Prinsip Pareto mengimplikasikan 80% dari semua kesalahan yang ditemukan akan dapat ditelusuri dari semua modul program.
- Dimulai dari hal yang kecil dan berkembang ke arah pengujian yang lebih besar.
- Harus dilakukan oleh pihak ketiga yang independent untuk perangkat lunak yang paling efektif.

### **Proses Testing**

1. System Testing : Pengujian terhadap integrasi sub sistem, yaitu ketekhubungan antar sub sistem.
2. Acceptance Testing : Pengujian terakhir sebelum sistem dipakai oleh user, melibatkan pengujian dengan data dari pengguna sistem. Dikenal sebagai

Alpha test (beta test untuk perangkat lunak komersial yang dilakukan oleh potensial customer).

3. Unit testing : Pengujian yang masing-masing unit komponen program untuk memastikan bahwa program sudah dijalankan secara benar.
4. Module Testing : Pengujian terhadap koleksi unit-unit komponen yang saling berhubungan.
5. Sub System Testing : Pengujian terhadap koleksi modul-modul yang membentuk sub sistem.
6. System Testing : Pengujian terhadap keterhubungan sub sistem.

### **The Testing Process**

1. Component Testing : Pengujian komponen-komponen program, biasanya dilakukan oleh component developer (kecuali untuk sistem kritis).
2. Integration Testing : Pengujian kelompok komponen yang terintegrasi untuk membentuk sub sistem ataupun sistem yang dilakukan oleh tim penguji yang independent. Pengujian dilakukan berdasarkan spesifikasi sistem.

### **Testing Approach**

White Box Testing adalah proses testing yang fokus pada struktur code (source code) pada aplikasi yang akan dites, apakah fungsi-fungsi internal telah berjalan dengan baik dan benar sesuai dengan spesifikasi atau belum. Test ini biasanya dilakukan saat proses testing unit level. Black Box Testing testing ini biasa dikenal dengan Behavioral/Input-Output testing, karena sebuah tester tidak perlu memahami logika yang terjadi didalam struktur kodenya (source code).

### **e. Alat dan Bahan**

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen Modul Pengujian Perangkat Lunak

### **f. Prosedur Kerja**

1. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
2. Silahkan Review Jurnal / Referensi lain yang berisi Prosedur Pengujian– sebagai dasar pengujian perangkat lunak dibagi dengan tiap anggota kelompok.
3. Tuangkan kedalam bentuk PPT dan Di diskusikan didepan kelas.

4. Sertakan daftar rujukan yang digunakan.

**g. Hasil dan Pembahasan**

Dokumen Dasar Pengujian Perangkat Lunak – sebagai dasar pengujian perangkat lunak

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai dengan contoh implementasi dan atau penerapan basis data pada lingkungan sekitar	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

## **Acara 2**

Materi Pembelajaran	: Quality Assurance Concepts
Acara Praktikum/Pertemuan	: Minggu 1 / 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu mengidentifikasi Dasar dan Prinsip Pengujian Perangkat Lunak
2. Mahasiswa mampu menentukan Dasar dan Prinsip Pengujian Perangkat Lunak pengembangan perangkat lunak
3. Mahasiswa mampu menganalisa acuan-acuan Dasar dan Prinsip Pengujian Perangkat Lunak

### **b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

### **c. Indikator Penilaian**

Ketepatan dalam memahami konsep pengujian perangkat Lunak

### **d. Dasar Teori**

Sebuah perangkat lunak yang berkualitas membutuhkan proses pengujian sebelum digunakan. Pengujian perangkat lunak diperlukan untuk menemukan cacat/kesalahan yang ada dalam program, baik itu kesalahan penulisan kode program maupun kesalahan tampilan program. Agar perangkat lunak tersebut terhindar ataupun bebas dari kesalahan/cacat dan dapat bekerja dengan baik sesuai yang diharapkan, maka diperlukan adanya pengujian perangkat lunak/testing. Pengujian perangkat lunak harus dilakukan secara optimal, teliti dan harus sesuai dengan tujuan pengembangan perangkat lunak. Oleh karena itu dalam melakukan pengujian perangkat lunak perlu kiranya memegang teguh prinsip pengujian dasar sebagai berikut :

#### **a. Pengujian menunjukkan adanya BUG / Kesalahan**

Dalam melakukan pengujian perangkat lunak proses yang dilakukan sebenarnya adalah mencari Bug / kesalahan pada perangkat lunak yang dikembangkan. Seorang tester harus membuat perangkat lunak menjadi gagal. Hal itu diperlukan agar proses pengujian perangkat lunak bisa menemukan kegagalan program sebanyak mungkin, sehingga bisa segera diperbaiki dan perangkat lunak yang



dihasilkan berkualitas. Meskipun dalam beberapa proses pengujian sangat mustahil untuk memastikan bahwa sebuah perangkat lunak 100% bebas dari cacat/bug. Namun, dengan adanya pengujian perangkat lunak sebelum perangkat lunak tersebut dirilis ke publik dapat mengurangi bug/cacat dari perangkat lunak meskipun tidak 100%.

### **b. Mustahil Melakukan Pengujian Secara Menyeluruh**

Melakukan pengujian perangkat lunak secara menyeluruh itu sangat tidak mungkin, karena membutuhkan waktu yang panjang sehingga akan berdampak pada waktu rilis program dan biaya dan tenaga yang diperlukan dalam proses pengembangan perangkat lunak tersebut. Oleh karena itu, dalam pengujian perangkat lunak yang harus diutamakan adalah mempertimbangkan risiko dan prioritas perangkat lunak yang dikembangkan. Langkah yang diperlukan adalah melakukan pengujian pada fungsional utama dari perangkat lunak tersebut. Khususnya pada fungsional yang membutuhkan masukan user dan keluaran informasi ke user. Pengujian dilakukan untuk melihat respon aplikasi jika menerima inputan benar maupun inputan salah, hasilnya harus sesuai dengan yang diperlukan

### **c. Pengujian Dilakukan Lebih Awal**

Melakukan pengujian di awal proses pengembangan perangkat lunak sangat disarankan, karena akan menekan biaya perbaikan. Selain itu juga akan meningkatkan kinerja perangkat lunak itu sendiri. Sehingga proses pengujian perangkat lunak dari awal sejak tahapan analisis akan membuat pengembangan dan pengujian perangkat lunak akan semakin efektif dan efisien.

### **d. Mengelompokkan Bug**

Dalam melakukan pengelompokan bug/kerusakan perangkat lunak dapat menggunakan prinsip Pareto dengan aturan 80-20, dimana 80% masalah yang ditemukan bersumber dari 20% modul yang ada pada perangkat lunak, kemudian 20% masalah yang tersisa merupakan masalah dari 80% modul perangkat lunak. Sehingga dalam melakukan pengujian perangkat lunak yang paling utama adalah menekan 20% modul, karena dalam 20% modul yang akan dihadapi adalah 80% masalah dari perangkat lunak. Artinya dalam pengelompokan cacat sebuah perangkat lunak pada proses pengujian lebih diutamakan melakukan pengujian pada modul atau fungsional program yang kecil, karena berdasarkan prinsip pareto diatas dalam modul yang kecil akan ada masalah yang besar/banyak.

### **e. Paradoks Pestisida**

Analogi yang tepat untuk menjelaskan tentang Paradok Pestisida adalah sebagaimana penamaannya. Jika anda selalu menggunakan pestisida yang sama dalam mengusir hama di kebun/sawah, maka hama tersebut lama kelamaan akan menjadi kebal karena sudah bisa menyesuaikan. Dalam buku yang berjudul *Software Testing Techniques* (1990) Boris Beizer memperkenalkan istilah *Pesticide Paradox* dimana proses pengujian yang dilakukan secara berulang pada akhirnya tidak akan menemukan cacat dari program. Karena mungkin pengembang (Developer) perangkat lunak sudah mengetahui/belajar dari proses kesalahan sebelumnya. Sehingga seorang tester harus selalu memperbaharui teknik pengujian agar bisa menemukan bug yang lebih banyak ketika melakukan pengujian perangkat lunak. Hal itu diperlukan agar terhindar dari paradok pestisida.

### **f. Pengujian Bergantung Pada konteks**

Proses pengujian perangkat lunak harus sesuai dengan konteks perangkat lunaknya. Dalam melakukan pengujian terhadap perangkat lunak perbankan tentu tidak bisa menggunakan cara pengujian untuk perangkat lunak penjualan. Sehingga proses pengujian perangkat lunak harus benar-benar sesuai dengan konteks dari perangkat lunak itu sendiri. Setiap perangkat lunak memiliki cara pengujiannya masing-masing tidak bisa dipukul rata sama semua.

### **g. Tidak Ada Bug Merupakan Kesalahan**

Pengujian perangkat lunak diperlukan untuk menghasilkan perangkat lunak yang berkualitas. Salah satu indikator perangkat lunak berkualitas adalah dibuat sesuai dengan kebutuhan pengguna. Oleh karena itu, jika dalam pengujian tidak ditemukan adanya bug/cacat maka itu perlu dicurigai apakah perangkat lunak tersebut sudah sesuai dengan kebutuhan bisnis pengguna. Atau jangan-jangan tidak adalah bug karena memang tidak sesuai dengan kebutuhan. Makanya pengujian yang menyatakan tidak ada bug dari perangkat lunak perlu ditinjau ulang untuk memastikan proses pengujian yang dilakukan sudah benar dan perangkat lunak yang dikembangkan memang sudah sesuai dengan kebutuhan bisnis pengguna agar perangkat lunak yang dirilis bisa digunakan sebagaimana yang diharapkan.

### **e. Alat dan Bahan**

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen Modul Pengujian Perangkat Lunak

**f. Prosedur Kerja**

1. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
2. Carilah jurnal yang berhubungan dengan prinsip pengujian perangkat lunak
3. Buatlah Laporan yang berisi tentang prinsip pengujian perangkat lunak dibagi dengan tiap anggota kelompok.
4. Diskusikan di depan Kelas
5. Sertakan daftar rujukan yang digunakan.

**g. Hasil dan Pembahasan**

1. Laporan Prinsip Pengujian Perangkat Lunak.
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.polije.ac.id/elearning>

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai dengan contoh implementasi dan atau penerapan basis data pada lingkungan sekitar	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

### **Acara 3**

Materi Pembelajaran	: Komponen manajemen kualitas perangkat lunak
Acara Praktikum/Pertemuan	: Minggu 2 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

#### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Memanajemen Pengujian Perangkat Lunak
2. Mahasiswa mampu Menentukan Tolls Pengujian Perangkat Lunak
3. Mahasiswa mampu Menganalisa acuan-acuan Materi Pokok Bahasan Tools Pengujian Perangkat Lunak

#### **b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

#### **c. Indikator Penilaian**

Ketepatan dalam memanajemen pengujian perangkat lunak serta menentukan tools pengujian perangkat lunak

#### **d. Dasar Teori**

Testing tools atau alat pengujian adalah perangkat yang penting untuk menguji aplikasi Anda sebelum dirilis kepada pengguna serta untuk mengetahui kecepatan dan kesalahan-kesalahan pada aplikasi yang baru dibuat Jenis testing tools ada bermacam-macam. Berikut ini adalah rekomendasi aplikasi terbaik untuk automation testing :

##### **1. Katalon Studio**

Katalon Studio adalah salah satu software testing tool yang dikatakan sebagai solusi pengujian berkelanjutan yang komprehensif yang membahas otomatisasi pengujian permintaan yang tajam di CI / CD dan DevOps. Katalon Studio mendukung pengujian aplikasi web, API, seluler, dan desktop di seluruh sistem operasi Windows, macOS, dan Linux. Ini adalah serangkaian fitur terpadu yang dimiliki oleh Katalon Studio mulai dari merekam langkah-langkah pengujian, menjalankan kasus pengujian hingga menyediakan infrastruktur, laporan analitik, dan integrasi CI/CD. Katalon Studio mendukung pengujian berkelanjutan melalui integrasi Jira Agile yang mulus dan integrasi CI / CD asli dengan alat CI paling populer (misalnya Jenkins, Bamboo, Azure, dan CircleCI). Katalon Studio dikenal mudah digunakan, dengan pembuatan tes tanpa kode

untuk pemula dan dilengkapi ekstensi tingkat lanjut untuk para ahli. Plugin di Katalon Store dapat membantu Anda memperluas kemampuan otomatisasi dan dengan mudah berintegrasi dengan perangkat lunak CI / CD lainnya. Katalon Studio juga menawarkan serangkaian lengkap laporan berwawasan, pemantauan waktu nyata, dan mekanisme umpan balik instan. Anda sepenuhnya dapat mengontrol kualitas produk serta kinerja tim Anda. Putaran umpan balik instan Katalon akan membantu Anda mendeteksi masalah yang muncul segera setelah muncul. Katalon Studio pun bersifat open-source, jadi Anda bisa menggunakannya secara gratis dan tersedia versi berbayar khusus untuk perusahaan.

## 2. Selenium

Selenium adalah salah satu testing tool yang sangat populer dalam otomatisasi pengujian. Testing tool ini dapat dikatakan sebagai testing tool tertua dan telah dikembangkan sejak tahun 2014. Selenium merupakan testing tool yang ideal untuk digunakan dalam pengujian yang sering. Selenium juga memiliki Selenium WebDriver, sebuah framework untuk melakukan otomatisasi web dan dapat digunakan untuk menjalankan uji otomatisasi regresi berbasis browser yang andal. Banyak pengembang web lebih memilih Selenium karena alat ini kompatibel dengan berbagai bahasa pemrograman dan dapat diintegrasikan dengan kerangka pengujian. Selenium mendukung berbagai OS populer (Windows, macOS, Linux) dan browser (Chrome, Firefox, Safari) untuk pengujian lintas lingkungan.

Selenium pun bersifat open-source dan dapat dengan mudah diintegrasikan dengan alat dan framework lain. Namun, Selenium juga memiliki kekurangan yaitu ketika mengintegrasikan Selenium dengan alat lain di pipeline CI / CD, Anda harus memiliki pengetahuan dan keterampilan teknis tertentu.

## 3. Eggplant

Eggplant sebelumnya dikenal dengan Testplant adalah testing tool yang mendukung berbagai teknologi dan platform otomatisasi, dari web, seluler, hingga aplikasi Point of Sales (POS). EggPlant adalah alat otomatisasi pengujian GUI black box. Untuk melakukan pengujian perangkat lunak, Eggplant menawarkan berbagai macam alat otomatisasi pengujian yang dapat Anda gunakan untuk melakukan berbagai jenis pengujian. Ada EggPlant Functional untuk pengujian fungsional dan EggPlant Performance untuk pengujian beban dan kinerja. Eggplant juga dapat dengan mudah diintegrasikan dengan alat manajemen

pengujian yang populer. Dengan satu skrip pengujian, EggPlant dapat membuat beberapa skenario perangkat.

Dengan kurva pembelajaran yang singkat, Eggplant adalah testing tool berkelanjutan yang paling cocok untuk pengujian manual. Tools ini menawarkan metode unik otomatisasi pengujian dengan solusi berbasis gambar. Eggplant berinteraksi dengan Applications Under Test (AUTs) dengan mereplikasi sudut pandang pengguna melalui screenshot dan gambar.

Eggplant menyediakan test lab, layanan manajemen perangkat cloud yang memungkinkan akses 24/7 bagi tim pengembangan untuk digunakan dalam pengujian berkelanjutan dan penerapan berkelanjutan.

Selain itu, Eggplant mendukung integrasi dengan ekosistem CI / CD seperti Jenkins, Bamboo, CA Agile Requirements Designers. Anda juga dapat mengintegrasikan Eggplant dengan Undo (alat rekam dan pemutaran) untuk membantu pengembang mempersingkat waktu mendeteksi dan merespons kegagalan.

#### 4. Appium

Appium adalah software testing tool seluler otomatis. Appium pun bersifat open-source dan cocok digunakan untuk native web app, aplikasi hibrid, dan seluler untuk iOS maupun Android. Appium adalah pilihan tepat untuk kerangka kerja otomatisasi pengujian karena dapat digunakan untuk semua jenis aplikasi/web yang berbeda ini.

Testing tool ini dianggap sebagai alat otomatisasi pengujian seluler terbaik di pasar saat ini. Ini dapat disesuaikan dan dapat digunakan sesuai dengan kebutuhan.

Appium saat ini banyak dimanfaatkan dan disesuaikan secara fleksibel untuk kebutuhan pengujian berkelanjutan. Selain itu, sejumlah besar penyedia perangkat cloud telah mengizinkan pengguna menjalankan pengujian Appium pada infrastrukturnya sendiri. Menggunakan Appium memungkinkan Anda untuk menulis, mengunggah, mengeksekusi, dan melihat hasil tes langsung di cloud.

Kekurangan Appium terletak pada langkah awal integrasi dengan ekosistem CI / CD. Mirip dengan Selenium, pengguna Appium membutuhkan pengetahuan lanjutan tentang kerangka kerja pengujian untuk menyiapkan dan menerapkan. Hal ini membuat Appium paling cocok untuk tim yang telah memiliki pemahaman mendalam tentang kerangka kerja ini dan alat CI di industri.



## 5. Test Complete

TestComplete adalah platform otomatisasi pengujian fungsional yang dikembangkan oleh SmartBear. TestComplete memberi penguji perangkat lunak kemampuan untuk membuat pengujian otomatis untuk aplikasi web, Microsoft Windows, iOS dan Android. Alat ini mendukung berbagai bahasa skrip termasuk Python, Javascript, VBScript. TestComplete memungkinkan Anda melakukan pengujian berdasarkan kata kunci atau berdasarkan data. Pembuat TestComplete baru-baru ini memperkenalkan fitur AI untuk pengenalan dan pemeliharaan objek uji yang dinamis. TestComplete dapat secara otomatis mendeteksi dan memperbarui pengujian jika ada perubahan dengan UI AUT.

TestComplete mendukung integrasi terbuka dengan ekosistem CI / CD melalui plugin. Anda dapat menggunakan plugin ini untuk berintegrasi dengan alat CI / CD populer seperti Jenkins, GIT, Zephyr (alat manajemen pengujian oleh Smartbear) atau Anda dapat mengembangkan plugin khusus untuk diintegrasikan dengan sistem yang ada.

SmartBear mengklaim TestComplete sebagai yang paling mudah digunakan dalam hal pengujian UI otomatis. Kelebihan dari TestComplete adalah Anda dapat menjalankan uji regresi paralel dengan pembuatan otomatisasi dan membuat uji regresi yang stabil. TestComplete pun dapat secara otomatis menjadwalkan dan menjalankan tes regresi tanpa campur tangan manusia sehingga lebih menghemat waktu pengujian dan biaya pelatihan secara signifikan.

## 6. Wireshark

Awalnya dikenal sebagai Ethereal, Wireshark menampilkan data dari ratusan protokol berbeda di semua jenis jaringan utama. Paket data dapat dilihat secara real-time atau dianalisis secara offline. Wireshark mendukung lusinan format file tangkap / lacak (capture / trace), termasuk CAP dan ERF. Alat dekripsi terintegrasi menampilkan paket terenkripsi untuk beberapa protokol umum, termasuk WEP dan WPA / WPA2.

Wireshark adalah penganalisa lalu lintas jaringan terkemuka di dunia, dan alat penting untuk profesional keamanan atau administrator sistem. Software gratis ini memungkinkan Anda menganalisis lalu lintas jaringan secara real time, dan seringkali merupakan alat terbaik untuk mengatasi masalah pada jaringan Anda.

Masalah umum yang dapat dipecahkan oleh Wireshark termasuk paket yang hilang, masalah latensi, dan aktivitas jahat di jaringan Anda. Ini memungkinkan Anda menempatkan lalu lintas jaringan Anda di bawah

mikroskop, dan menyediakan alat untuk menyaring dan menelusuri lalu lintas itu, memperbesar penyebab utama masalahnya. Administrator menggunakan software ini untuk mengidentifikasi peralatan jaringan yang rusak yang menjatuhkan paket, masalah latensi yang disebabkan oleh mesin yang merutekan lalu lintas di belahan dunia, dan pengelupasan data atau bahkan upaya peretasan terhadap Anda.

Wireshark adalah alat yang ampuh yang membutuhkan pengetahuan dasar-dasar jaringan. Untuk sebagian besar perusahaan modern, itu berarti memahami TCP / IP stack, cara membaca dan menafsirkan header paket, dan bagaimana routing, port forwarding, dan DHCP berfungsi.

**e. Alat dan Bahan**

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

Dokumen Modul Pengujian Perangkat Lunak

**f. Prosedur Kerja**

1. Tugas mandiri
2. Carilah jurnal yang berhubungan dengan Tolls Pengujian perangkat Lunak
3. Buatlah Laporan yang berisi tentang perbandingan kelebihan dan kekurangan tools Pengujian Perangkat Lunak
4. Diskusikan di depan Kelas
5. Sertakan daftar rujukan yang digunakan.

**g. Hasil dan Pembahasan**

1. Laporan Kelebihan dan Kekurangan Tools Pengujian Pengujian Perangkat Lunak
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.polije.ac.id/elearning>

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai dengan contoh implementasi dan atau penerapan basis data pada lingkungan sekitar	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		

## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 4**

Materi Pembelajaran	: Komponen manajemen kualitas perangkat lunak
Acara Praktikum/Pertemuan	: Minggu 2 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Memanajemen Pengujian Perangkat Lunak
2. Mahasiswa mampu Menentukan Tolls Pengujian Perangkat Lunak
3. Mahasiswa mampu Menganalisa acuan-acuan Materi Pokok Bahasan Tools Pengujian Perangkat Lunak

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

**c. Indikator Penilaian**

Ketepatan dalam memanajemen pengujian perangkat lunak serta menentukan tools pengujian perangkat lunak

**d. Dasar Teori****Statistic Testing Tools**

Statistic testing atau verifikasi merupakan proses saat program belum dijalankan atau kode program belum dieksekusi. Tujuannya untuk melakukan checking pada kode program, dokumen dan file. Biasanya berlangsung pada tahap pengembangan atau development karena itu disebut juga verification testing. 4 teknik uji statis:

- a. Inspection: tujuan utamanya untuk menemukan defects. Dilakukan oleh moderarator dengan list pengecekan yang sudah disiapkan.
- b. Walkthrough: teknik testing dengan melakukan meeting sehingga partisipan dapat memberikan saran atau masukan.
- c. Technical Reviews: teknik ini untuk memastikan pembuatan source code telah memenuhi standar yang telah ditentukan.
- d. Informal Reviews: dokumen direview secara informal dan tidak berdasarkan prosedur. Tujuannya untuk meningkatkan kualitas dokumen.

### White-box Testing

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya White Box Testing ini menguji dengan cara melihat Pure Code dari suatu aplikasi/software yang diuji tanpa memperdulikan Tampilan atau UI dari aplikasi tersebut.

1. Teknik White-box Testing, sebagai berikut :

- Basis Path Testing.

Basis path testing merupakan metode yang memungkinkan perancang testcase untuk membuat pengukuran kompleksitas logikal dari rancangan prosedural dan menggunakan pengukuran ini sebagai panduan untuk mendefinisikan himpunan basis dari jalur eksekusi. Test case yang dibuat untuk menguji himpunan basis dijamin akan mengeksekusi setiap statement di dalam program sekurangnya sekali pada saat pengujian

- Flow Graph

Flow graph merupakan notasi sederhana untuk merepresentasi control flow.

- Cyclomatic Complexity

Cyclomatic complexity digunakan untuk mengetahui jumlah jalur yang perlu dicari. Cyclomatic complexity adalah metric software yang menyediakan ukuran kuantitatif dari kompleksitas logikal program. Nilai yang dihitung bagi cyclomatic complexity menentukan jumlah jalur-jalur yang independen dalam kumpulan basis suatu program dan memberikan jumlah tes minimal yang harus dilakukan untuk memastikan bahwa semua pernyataan telah dieksekusi sekurangnya satu kali.

Cyclomatic complexity mempunyai fondasi dalam teori graph dan dapat dihitung dengan satu dari tiga cara :

- Jumlah region sama dengan cyclomatic complexity.
- Cyclomatic complexity,  $V(G)$ , untuk sebuah flow graph,  $G$ , didefnisikan sebagai:  $V(G) = E - N + 2$   $E$  adalah jumlah edge pada flow graph, dan  $N$  adalah jumlah node pada flow graph.
- Cyclomatic complexity,  $V(G)$ , untuk flow graph,  $G$ , juga didefinisikan sebagai:  $V(G) = P + 1$   $P$  adalah jumlah predicate nodes yang terdapat pada flow graph  $G$ .
- Graph Matrix

- Prosedur untuk membuat flow graph dan menentukan himpunan basis path dapat diterima berdasarkan mekanisme. Untuk mengembangkan software yang membantu pengujian basis path, sebuah struktur data yang disebut graph matrix, dapat sangat bermanfaat. Graph matrix adalah matriks kotak yang ukurannya (jumlah baris dan kolom) sama untuk jumlah node pada flow graph. Setiap baris dan kolom berhubungan dengan node yang teridentifikasi, dan data matriks berhubungan dengan koneksi (edge)

### **Kelemahan White Box-Testing:**

1. Sangat mahal untuk dilakukan karena membutuhkan tester yang terampil untuk melakukan pengujian.
2. Pada perangkat lunak yang jenisnya besar, metode white box testing ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya.
3. Tidak mempedulikan Tampilan UI aplikasinya.

### **Penggunaan metode pengujian white box dilakukan untuk :**

1. Memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali
2. Menggunakan semua keputusan logis untuk semua kondisi true atau false
3. Mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.
4. Menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

### **e. Alat dan Bahan**

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

Dokumen Modul Pengujian Perangkat Lunak

### **f. Prosedur Kerja**

1. Tugas mandiri
2. Carilah jurnal yang berhubungan dengan White Box Testing
3. Buatlah Laporan yang berisi tentang White Box Testing
4. Diskusikan di depan Kelas
5. Sertakan daftar rujukan yang digunakan.



**g. Hasil dan Pembahasan**

1. Laporan Kelebihan dan Kekurangan Tools Pengujian Pengujian Perangkat Lunak
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.polije.ac.id/elearning>

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai dengan contoh implementasi dan atau penerapan basis data pada lingkungan sekitar	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

## **Acara 5**

Materi Pembelajaran	: Komponen manajemen kualitas perangkat lunak
Acara Praktikum/Pertemuan	: Minggu 3 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Mengidentifikasi Pengelolaan Pengujian Perangkat Lunak
2. Mahasiswa mampu Menentukan Pengelolaan Pengujian Perangkat Lunak
3. Mahasiswa mampu Menganalisa acuan-acuan Materi Pokok Bahasan Pengelolaan Pengujian Perangkat Lunak

### **b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

### **c. Indikator Penilaian**

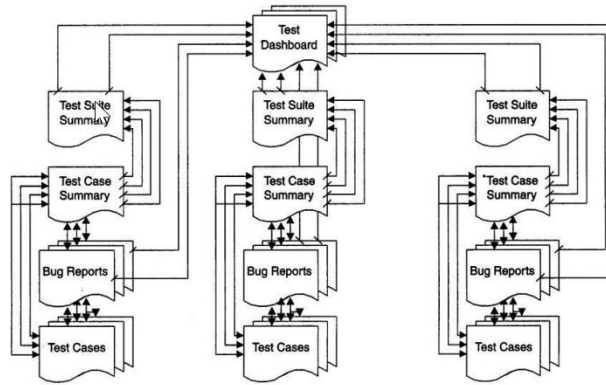
Ketepatan dalam menentukan komponen Kualitas perangkat lunak

### **d. Dasar Teori**

Apabila kita sudah menangani proyek pengujian perangkat lunak satu atau dua kali maka kita akan mengalami masa-masa kegentingan yang sangat memeras tenaga dan membingungkan. Masalah yang dihadapi: kenyataan yang dihadapi tidak sesuai yang diharapkan, perencanaan yang selalu berubah, dan ketergantungan yang tidak jelas.

Beberapa Saran dalam Pengelolaan proses pengujian Perangkat Lunak.

- Tetap maju sementara mendapatkan seluruh kenyataan: keinginan terhadap kepastian, kemajuan yang tetap harus terlaksana.
- Ketergantungan, Jadwal, dan peringatan: pentingnya untuk menindaklanjuti.
- Masalah Pengiriman: Proses Revisi dan pengeluaran (release).
- Masalah instalasi: Konfigurasi lingkungan pengujian.
- Pemeriksaan (auditing) dan memperbaharui (update) hasil pengujian.
- Mendefinisikan proses pelaksanaan pengujian untuk Mengaplikasikan Exploratory Testing.
- Ketika pengujian gagal: meminimumkan kesalahan pada interpretasi hasil pengujian.



Gambar 1 Hubungan Antara Test Tracking Dan Laporan

Ada 2 metode untuk melakukan pengelolaan terhadap pengujian Perangkat lunak ketika melakukan pengelolaan pengujian system yaitu :  
Basis Data Logistik (Logistics Database)

Basis Data Logistik (Logistics Database) dapat menelusuri lokasi, konfigurasi, dan kebutuhan pengujian akan perangkat keras, perangkat lunak, infrastruktur, dan staf. Basis data ini dapat melakukan penelusuran terhadap :

Lokasi, instalasi, dan perpindahan perangkat keras.

Konfigurasi perangkat lunak pada saat ini, pada masa lalu, dan konfigurasi yang akan direncanakan.

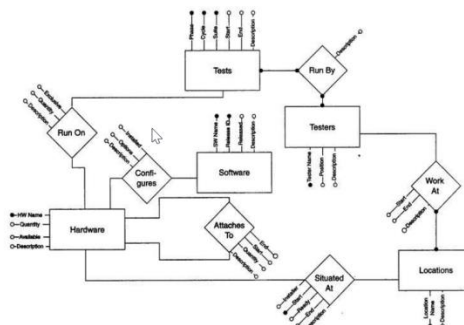
Interkoneksi perangkat keras dan jaringan computer.

Lokasi pengujian.

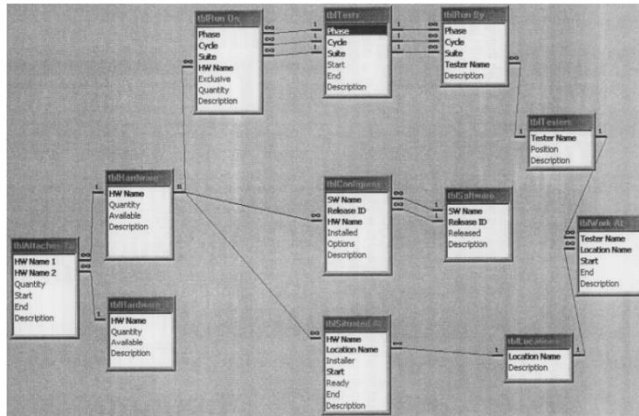
Infrastruktur pengujian

Penugasan engineer pengujian dan lokasinya.

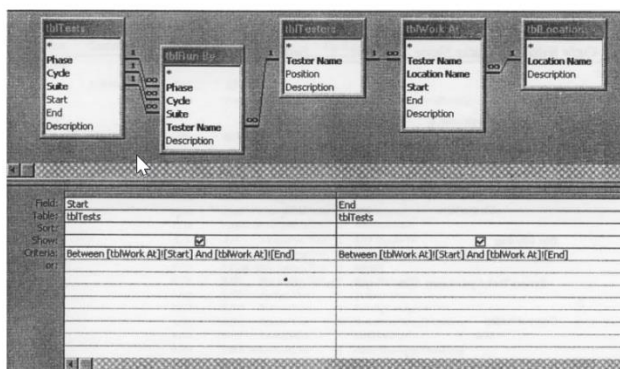
Penyebaran sumber daya manusia.



Gambar 2 ER Diagram



Gambar 3 penerapan basis data logistic



Gambar 4 Query untuk Pengujian

## Basis Data Pengelola Perubahan Sederhana (Simple Change Management Database)

Basis Data Pengelola Perubahan Sederhana (Simple Change Management Database) membantu dalam menanggapi bagian perbaikan yang tak terelakan yang muncul pada setiap proyek pengujian. Bagaimanapun sempurnanya suatu perencanaan, langkah penanganan yang hati-hati, keefektifan kerjasama dengan tim penguji, rekan kerja, dan manajer tetap saja akan menghadapi tanggapan yang tidak diharapkan. Perubahan tidak dapat dihindari dan tidak seluruh perubahan dapat direncanakan sebelumnya. Perubahan seringkali merupakan manifestasi dari proses pembelajaran. Pengelolaan perubahan dapat dilakukan secara formal dengan CCB (change control board) atau secara informal dengan simple change management database. Menggunakan Data Pengelolaan Perubahan, Dengan mengumpulkan data mengenai perubahan, pengaruhnya dan perencanaan pemulihan (recovery plan) akan dapat memenuhi beberapa komunikasi yang penting dan tanggung jawab perencanaan. Dengan data ini maka komunikasi dengan manajemen proyek dapat dilakukan secara terstruktur.

ID	Field Name	Data Type	Description
1	Project Name	Text	Unique identifier for the change.
2	Date Noted	Date/Time	Name of project on which the change occurred.
3	Impact Date	Date/Time	Date on which the change was first noted.
4	Change Description	Memo	Date on which the change will first affect testing.
5	Change Type	Text	Short summary of what changed.
6	Cross-References	Memo	Type of change that is affecting testing.
7	Impact Description	Memo	List of the specific test documents that are affected by the change, with page and paragraph numbers.
8	Schedule Impact	Number	Short list of what this change will mean for testing.
9	Resource Impact	Number	Number of calendar-days delay or advance as a result of this change.
10	Other Costs/Savings	Currency	Number of person-days lost or saved as a result of this change.
11	Test Impact	Memo	Other costs/savings associated with this change.
12	Recovery Plan	Memo	Short list of the test case IDs affected by this change.
13	Status Log	Memo	Short description of how we intend to accommodate this change, including any contingency plans identified in the test or project plan.
			Log of progress toward recovering from this problem.

Gambar 5 Tabel sederhana untuk mencatat pengelolaan perubahan perangkat Lunak

## e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

Dokumen Modul Pengujian Perangkat Lunak

**f. Prosedur Kerja**

1. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
2. Carilah jurnal / Referensi yang berhubungan dengan Pengelolaan Pengujian Perangkat Lunak Berdasarkan Logistic Database dan Simple Management database perangkat lunak
3. Buatlah Laporan yang berisi tentang Logistic Database dan Simple Management database perangkat lunak.
4. Diskusikan di depan Kelas
5. Sertakan daftar rujukan yang digunakan.

**g. Hasil dan Pembahasan**

1. Laporan tentang Logistic Database dan Simple Management database perangkat lunak
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.polije.ac.id/elearning>

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Entitas, atribut, relasi, kardinalitas, dan kunci	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

## **Acara 6**

Materi Pembelajaran	: Komponen manajemen kualitas perangkat lunak
Acara Praktikum/Pertemuan	: Minggu 3 / 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Mengidentifikasi Pengelolaan Pengujian Perangkat Lunak
2. Mahasiswa mampu Menentukan Pengelolaan Pengujian Perangkat Lunak
3. Mahasiswa mampu Menganalisa acuan-acuan Materi Pokok Bahasan Pengelolaan Pengujian Perangkat Lunak

### **b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

### **c. Indikator Penilaian**

Ketepatan dalam menentukan komponen Kualitas perangkat lunak

### **d. Dasar Teori**

#### **1. Pengelolaan Tim Pengujian**

Dalam sebuah pengujian perangkat lunak, diperlukan pengelolaan terhadap Tim pengujian agar pengujian yang di inginkan dapat berjalan dengan baik dan semua anggota tim dapat bersinergi dalam melakukan pengujian perangkat lunak. Beberapa hal yang perlu diperhatikan adalah sebagai berikut  
Most Basic Principle for Managing Staff :

1. Hiring (Memberikan pekerjaan dalam mengembangkan diri)
2. Motivating
3. Retaining (mempertahankan/ memelihara)

What kind of people make good Test Engineers ?:

1. Professional Pessimism
2. Balanced Curiosity
3. Focus: No Space Cadets
4. Avoid the Aspiring Hero
5. Shun the Sloth
6. Reject Casper Milquetoast

Defining the Test Team



1. Size
2. Skills : Technology, Application Domain, Testing
3. Education and training
4. Positions, experience and goals

### 2. Jumlah Personel

Ada dua cara untuk menentukan jumlah anggota suatu test team, yaitu :

- Berdasarkan tugas/task yang harus dikerjakan.
- Berdasarkan rasio antara tester dan developer. Rasio yang ideal antara tester dan developer 1:1, umumnya berkisar 1:2 s.d. 1:5

### 3. Keahlian

Dalam beberapa kasus testing diperlukan tenaga yang memiliki suatu keahlian khusus. Harus dapat membaca dan belajar secara cepat. Dapat menulis laporan dengan baik. Dalam beberapa kasus diperlukan tenaga yang menguasai statistik dan matematika.

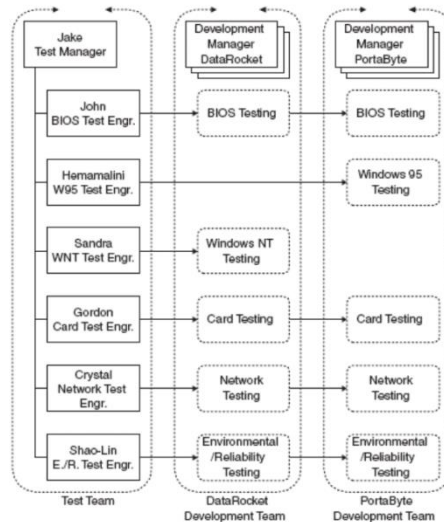
### 4. Jabatan

1. Test engineer : programmer, mechanical engineer, dan electrical engineer.
2. Test Technician.
3. Test Manager
4. Dalam beberapa kasus mungkin diperlukan juga release management engineer dan configuration management engineer

### 5. Model Organisasi

**Skill-based Model:** sangat direkomendasikan pada sistem2 yang rumit dan berteknologi canggih.

**Project-bases Organization :** pembagian tugas berdasarkan proyek yang dikerjakan, bukan berdasarkan spesialisasinya



Gambar 6 Contoh Skill Based Organization

### e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

Dokumen Modul Pengujian Perangkat Lunak

### f. Prosedur Kerja

1. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
2. Carilah jurnal / Referensi yang berhubungan dengan Pengelolaan TIM Pengujian Perangkat Lunak
3. Buatlah Laporan yang berisi tentang Pengelolaan Tim Pengujian perangkat lunak.
4. Diskusikan di depan Kelas
5. Sertakan daftar rujukan yang digunakan.

### 6. Hasil dan Pembahasan

1. Laporan tentang Logistic Database dan Simple Management database perangkat lunak
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.polije.ac.id/elearning>

**7. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Entitas, atribut, relasi, kardinalitas, dan kunci	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

## **Acara 7**

Materi Pembelajaran	: Proses Pengujian dalam SDLC
Acara Praktikum/Pertemuan	: Minggu 4 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Mengidentifikasi Pengelolaan Pengujian Perangkat Lunak
2. Mahasiswa mampu Menentukan Pengelolaan Pengujian Perangkat Lunak
3. Mahasiswa mampu Menganalisa acuan-acuan Materi Pokok Bahasan Pengelolaan Pengujian Perangkat Lunak

### **b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

### **c. Indikator Penilaian**

Ketepatan membuat proses Pengujian Perangkat Lunak menggunakan metode SDLC

### **d. Dasar Teori**

Integration Testing (Pengujian Terintegrasi) – sebagai dasar pengujian perangkat lunak

- Pengujian integrasi adalah teknik yang sistematis untuk mengkonstruksi susunan program sambil melakukan pengujian untuk memeriksa kesalahan yang nantinya digabungkan dengan interface.
- Sasarannya adalah untuk mengambil modul yang dikenai pengujian unit dan membangun struktur program yang telah ditentukan oleh desain.
- Kesulitannya adalah lokalisasi error yang sulit ditemukan pada saat proses.
- Terdapat interaksi yang rumit, antara komponen sistem dan ketika ditemukan output yang menyimpang, mungkin sulit untuk menemukan sumber error tersebut.

#### **1. Integrasi Non-Inkremental**

Program diuji sebagai kesatuan. Serangkaian kesalahan akan terjadi. Koreksi sulit dilakukan karena isolasi penyebab diperumit oleh luasnya program keseluruhan. Sekali kesalahan tersebut dibetulkan, maka akan muncul kesalahan yang baru dan proses itu terus berlanjut dalam loop yang kelihatannya tidak akan pernah berhenti.

## **2. Integrasi Inkremental**

Program dibangun dan diuji ke dalam segmen-segmen kecil, sehingga kesalahan lebih mudah diisolasi dan dibetulkan. Interface lebih mungkin untuk diuji secara lengkap, dan pendekatan pengujian secara sistematis dapat diaplikasikan.

### **2.1. Top-Down Integration**

Adalah pendekatan inkremental terhadap struktur program. Modul diintegrasikan dengan menggerakkan kebawah dengan hirarki kontrol, dimulai dengan modul kontrol utama (program utama). Memverifikasi kontrol utama dan keputusan pada saat awal proses pengujian. Pada struktur program yang dibuat dengan baik, keputusan akan dikerjakan pada tingkat atas hirarki. Stub mengganti modul tingkat rendah pada awal pengujian top-down, sehingga tidak ada data yang penting yang dapat mengalir ke atas pada struktur program.

Proses integrasi dilakukan dalam 5 (lima) langkah:

- Modul kontrol utama digunakan sebagai test driver, dan stub ditambahkan pada semua modul yang secara langsung subordinat terhadap modul kontrol utama.
- Stub subordinat diganti pada suatu saat dengan modul aktual.
- Pengujian dilakukan pada saat masing-masing modul diintegrasikan.
- Pada kelengkapan masing-masing rangkaian pengujian, stub yang lain diganti dengan modul real.
- Pengujian regresi dapat dilakukan untuk memastikan bahwa kesalahan baru belum dimunculkan.

### **2.2. Bottom-Up Integration**

Dapat dinyatakan dengan penyusunan yang dimulai dan diuji dengan atomic modul (modul tingkat paling bawah pada struktur program). Modul diintegrasikan dari bawah keatas sehingga pemrosesan yang diperlukan untuk modul subordinat yang selalu diberikan akan selalu tersedia dan kebutuhan akan stub dapat dieliminasi.

Strategi bottom-up integration dapat diterapkan dengan urutan langkah-langkah sebagai berikut:

- Modul tingkat paling bawah digabungkan ke dalam cluster (sering disebut build) yang melakukan subfungsi perangkat lunak spesifik.
- Driver (program kontrol untuk pengujian) ditulis untuk mengkoordinasi input dan output test case.
- Cluster diuji.

- Driver diganti dan cluster digabungkan dengan menggerakkannya ke atas di dalam struktur program.

### **3. Regression Testing (Pengujian Regresi)**

Adalah aktivitas yang membantu memastikan bahwa perubahan (karena pengujian atau alasan lain) tidak menimbulkan tingkah laku yang tidak diharapkan atau kesalahan tambahan. Regression Testing merupakan eksekusi ulang dari beberapa subset yang telah dilakukan untuk memastikan bahwa perubahan tidak menimbulkan efek samping yang tidak diharapkan. Pengujian yang menggunakan pengujian regression ini akan menghasilkan kesalahan, dan setiap kesalahan harus dikoreksi. Setiap kali perangkat lunak dikoreksi, maka banyak aspek konfigurasi perangkat lunak (program, dokumentasi atau data yang mendukung) akan diubah.

Pengujian regresi (subset dari pengujian yang akan dieksekusi) berisi tiga kelas test case yang berbeda yaitu:

- Sample representatif dari pengujian yang akan menggunakan semua fungsi perangkat lunak.
- Pengujian tambahan yang akan berfokus pada fungsi-fungsi perangkat lunak yang mungkin dipengaruhi oleh perubahan tersebut.
- Pengujian yang berfokus pada komponen perangkat lunak yang diubah.

Pemilihan strategi integrasi, tergantung pada karakteristik perangkat lunak dan kadang-kadang juga pada jadwal proyek. Secara umum pendekatan yang digabungkan (sandwich testing), yang menggunakan strategi top-down untuk tingkat yang lebih tinggi dari struktur program, dipasangkan dengan strategi bottom-up untuk tingkat subordinat.

Pada saat pengujian integrasi dilakukan, penguji harus mengidentifikasi modul kritis. Modul kritis memiliki karakteristik sebagai berikut:

- Menekankan beberapa persyaratan perangkat lunak.
- Memiliki tingkat kontrol yang tinggi.
- Kompleks (Cyclomatic Complexity dapat digunakan sebagai indikator).
- Memiliki persyaratan kinerja yang terbatas.

**Scope of Testing** merangkum fungsi yang spesifik, kinerja dan karakteristik desain internal yang akan diuji. Pengujian dibatasi; kriteria perlengkapan dari masing-masing fase pengujian digambarkan; dan batasan jadwal didokumentasikan.

**Test Plan** menggambarkan seluruh strategi interaksi. Pengujian dibagi ke dalam phases dan builds yang menekankan fungsional spesifik dan karakteristik

tingkah laku dari perangkat lunak tersebut. Misalnya pengujian integrasi untuk sebuah sistem komputer yang berorientasi pada grafik dapat dibagi ke dalam fase-fase pengujian sebagai berikut:

- Interaksi pemakai (seleksi perintah, representasi tampilan, pemrosesan, dan representasi kesalahan).
- Manipulasi dan analisis data (pembuatan simbol, penentuan dimensi, rotasi, komputasi sifat fisis).
- Pemrosesan dan pemunculan tampilan (tampilan dua dimensi, tampilan tiga dimensi, grafis dan bagan).
- Manajemen database (akses, update, integritas, kinerja).
- Kriteria pengujian yang sesuai diaplikasikan untuk semua fase pengujian:
- Integritas interface. Antar muka internal dan eksternal diuji pada saat masing-masing modul (kluster) ditambahkan ke dalam struktur.
- Validitas fungsional. Pengujian yang didesain untuk mengungkap kesalahan fungsional yang dilakukan.
- Isi Informasi. Pengujian dilakukan untuk mengungkap kesalahan yang berhubungan dengan struktur data global atau lokal yang dilakukan.
- Kinerja. Pengujian didesain untuk memeriksa batasan kinerja yang dibangun selama desain perangkat lunak dilakukan.

### e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

Dokumen Modul Pengujian Perangkat Lunak

### f. Prosedur Kerja

1. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
2. Buatlah makalah yang berisi Prosedur Integration testing – sebagai dasar pengujian perangkat lunak dibagi dengan tiap anggota kelompok.
3. Rangkum prosedur Integration testing yang dilakukan.
4. Sertakan daftar rujukan yang digunakan..

### g. Hasil dan Pembahasan

1. Makalah Prosedur Pengujian Unit – sebagai dasar pengujian perangkat lunak
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.polije.ac.id/elearning>

### h. Rubrik Penilaian



No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Entitas, atribut, relasi, kardinalitas, dan kunci	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

## **Acara 8**

Materi Pembelajaran	: Proses Pengujian dalam SDLC
Acara Praktikum/Pertemuan	: Minggu 4 / 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganju
Alokasi Waktu	: 120 menit

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Mengidentifikasi Pengelolaan Pengujian Perangkat Lunak
2. Mahasiswa mampu Menentukan Pengelolaan Pengujian Perangkat Lunak
3. Mahasiswa mampu Menganalisa acuan-acuan Materi Pokok Bahasan Pengelolaan Pengujian Perangkat Lunak

### **b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

### **c. Indikator Penilaian**

Ketepatan membuat proses Pengujian Perangkat Lunak menggunakan metode SDLC

### **d. Dasar Teori**

Validation Testing (Pengujian Validasi) – sebagai dasar pengujian perangkat lunak dilakukan setelah integration testing dilakukan serta kesalahan-kesalahan yang ditemukan telah diperbaiki. Validasi berhasil jika fungsi-fungsi yang ada pada perangkat lunak telah sesuai dengan yang diharapkan oleh pemakai. Merupakan kumpulan pengujian black-box yang memperlihatkan atau menunjukkan sesuai dengan yang diperlukan. Garis besar rencana pengujian dikerjakan dan prosedur pengujian didefinisikan dengan test case yang spesifik untuk menunjukkan sesuai dengan yang diperlukan. Rencana dan prosedur dirancang untuk menjamin seluruh keperluan fungsional telah terpenuhi, seluruh performansi dapat dicapai, dokumentasi dilakukan dengan benar.

Setelah pengujian dikerjakan, ada satu kemungkinan dari dua kondisi yang ada, yaitu:

- ~ Karakteristik performansi fungsi sesuai dengan spesifikasi dan dapat diterima.
- ~ Penyimpangan dari spesifikasi ditemukan dan daftar penyimpangan dibuat.

### **1. Kajian Konfigurasi**

Merupakan satu elemen penting dari proses validasi. Tujuannya untuk memastikan apakah semua elemen konfigurasi perangkat lunak telah dikembangkan dengan tepat, dikatalog, dan memiliki detail yang perlu untuk mendukung fase pemeliharaan dari siklus hidup perangkat lunak. Sering disebut *audit*.

Alpha dan Beta Testing Sangat tidak mungkin bagi pengembang perangkat lunak untuk meramalkan bagaimana pelanggan akan benar-benar menggunakan sebuah program. Instruksi yang digunakan dapat disalah-interprestasikan, kombinasi data yang aneh dapat dipakai secara reguler, dan output yang kelihatannya sudah jelas bagi penguji tidak dapat dimengerti oleh pemakai di lapangan. Bila perangkat lunak biasa dibangun bagi satu pelanggan, maka acceptance test dapat dilakukan untuk memungkinkan pelanggan memvalidasi semua persyaratan. Acceptance Test dilakukan karena memungkinkan pelanggan untuk menemukan kesalahan yang lebih terinci dan membiasakan pelanggan lebih memahami perangkat lunak yang telah dibuat. Jika perangkat lunak dibuat dan dikembangkan untuk dipakai oleh banyak pelanggan, maka tidak praktis untuk melakukan pengujian satu per satu terhadap perangkat lunak tersebut. Maka digunakan Alpha dan Beta Testing.

*Alpha Testing* adalah tahap pengembangan dimana perangkat lunak atau perangkat keras yang telah dibuat dikirim ke kelompok pemakai atau pembeli yang potensial kemudian mereka akan menggunakan produk ini untuk melaporkan jika produk ini gagal. Pengujian alpha dilakukan pada sebuah lingkungan yang terkontrol.

Pengujian *beta* dilakukan oleh pelanggan yang merupakan pemakai akhir perangkat lunak. *Pengujian Beta* merupakan sebuah aplikasi “live” dari perangkat lunak dari suatu lingkungan yang tidak dapat dikontrol oleh pengembang. Pelanggan merekam semua masalah yang ditemui selama pengujian beta dan melaporkannya kepada pengembang. Pengembang melakukan modifikasi kemudian mempersiapkan pelepasan produk ke seluruh pelanggan.

## 2. Kajian Konfigurasi

Perangkat lunak merupakan salah satu dari sistem yang berbasis komputer yang sangat besar. Perangkat lunak diintegrasikan dengan elemen sistem lainnya (hardware, informasi) dan serangkaian integrasi sistem dan validasi test dilakukan. Jika pengujian gagal atau diluar scope dari

pengembangan sistem dan tidak hanya dikerjakan oleh programmer, maka langkah yang diambil selama perancangan dan pengujian dapat diperbaiki.

System testing adalah sederetan pengujian yang berbeda-beda dengan tujuan utama mengerjakan keseluruhan elemen dalam sistem yang telah dikembangkan.

Peran Analisis Sistem antara lain:

1. Menangani kesalahan-kesalahan yang muncul dari elemen-elemen perangkat lunak.
2. Mengerjakan serangkaian pengujian.
3. Mencatat hasil pengujian.
4. Berpartisipasi dalam perencanaan dan merancang pengujian sistem untuk menjamin kualitas perangkat lunak.

### **3. Stress Testing (Pengujian Stres)**

Didesain untuk menghadapi situasi yang tidak normal, pada saat program mengalami pengujian. Dilakukan oleh sistem untuk kondisi-kondisi volume data yang tidak normal (melebihi atau kurang dari batasan) frekuensi dan lain-lain. Intinya penguji berusaha untuk mencari kelemahan dengan cara melakukan sedikit modifikasi untuk mencari fatal error pada program.

### **4. Recovery Testing (Pengujian Perbaikan)**

Adalah pengujian sistem yang memaksa perangkat lunak untuk mengalami kegagalan dalam berbagai cara dan melakukan verifikasi sesuai dengan performansi yang tepat. Banyak sistem yang berbasis komputer harus melindungi dari kesalahan dan melanjutkan prosesnya dalam waktu yang telah ditentukan. Sistem harus toleran terhadap kesalahan-kesalahan. Kesalahan pemrosesan tidak boleh keseluruhan fungsi berhenti.

### **5. Security Testing (Pengujian Keamanan)**

Adalah pengujian yang akan melakukan verifikasi dari mekanisme perlindungan yang akan dibuat oleh sistem, melindungi dari hal-hal yang mungkin terjadi. Penguji memerankan individu yang akan menembus sistem. Pengujian untuk mencoba menembus tingkat keamanan sebuah perangkat.

*Strategi Sandwich Compromise*, menguji perangkat lunak dengan melakukan pengujian mulai dari entry point tertentu kemudian bergerak keatas atau kebawah.

*Volume Testing*, menguji perangkat lunak dengan data yang berlebihan.

Configuration Testing, *menguji berbagai variasi perangkat lunak di berbagai lingkungan perangkat lunak.*

Compability Testing, *menguji kesesuaian sebuah perangkat lunak dengan sistem yang sedang dimanfaatkan.*

Timing Testing, *melakukan pengujian terhadap perangkat lunak untuk evaluasi terhadap waktu tanggap dan waktu proses yang dibutuhkan untuk menyelesaikan sebuah tugas.*

Enviromental Testing, *menguji toleransi perangkat lunak terhadap suhu, kelembaban, gerak dan perpindahan.*

Human Factor Testing, *menguji antar muka perangkat lunak bersama-sama dengan pemakai.*

### **6. Interface Testing (Pengujian Interface)**

Dilakukan ketika modul atau subsistem diintegrasikan untuk membuat sistem yang lebih besar. Setiap modul atau subsistem memiliki interface yang terdefinisi yang dipanggil oleh komponen program lain.

Tujuannya adalah untuk mendeteksi kesalahan yang mungkin telah masuk ke dalam sistem karena error interface atau asumsi invalid mengenai interface. Penting untuk pengembangan berorientasi objek.

#### **e. Alat dan Bahan**

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

Dokumen Modul Pengujian Perangkat Lunak

#### **f. Prosedur Kerja**

1. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
2. Buatlah makalah yang berisi Prosedur Validation testing – sebagai dasar pengujian perangkat lunak dibagi dengan tiap anggota kelompok.
3. Rangkum prosedur Validation testing yang dilakukan.
4. Sertakan daftar rujukan yang digunakan.

#### **g. Hasil dan Pembahasan**

1. Makalah Prosedur Pengujian Unit – sebagai dasar pengujian perangkat lunak
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.polije.ac.id/elearning>

**h. Rubrik Penilaian**

<b>No</b>	<b>INDIKATOR KINERJA</b>	<b>Bobot (%)</b>	<b>Penilaian</b>	<b>Nilai Akhir</b>
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Mapping ERD ke Table	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

## **Acara 9**

Materi Pembelajaran	: Perencanaan Proses Pengujian
Acara Praktikum/Pertemuan	: Minggu 5 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa dapat Melakukan Perencanaan Proses Pengujian
2. Mahasiswa dapat menyusun skenario testing
3. Mahasiswa dapat menyusun test case

### **b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

### **c. Indikator Penilaian**

Ketepatan mengimplementasikan skenario dalam studi kasus uji coba

### **d. Dasar Teori**

Hal — hal yang perlu diperhatikan dalam membuat sebuah skenario test antara lain :

1. Requirement Description. Pada aspek ini seorang QA harus memahami behaviour sebuah aplikasi yang diuji, bagaimana cara aplikasi tersebut berjalan serta mengetahui goals yang sesuai dengan requirement aplikasi yang sedang dikembangkan agar tetap selaras.
2. Inputs and outputs or actions (positive-negative scenarios) and expected results. Seorang QA harus mengetahui hasil dari eksekusi langkah — langkah pengujian yang ditulis dalam skenario test
3. Updates. Jika seorang QA menemukan skenario baru atau bahkan jika terdapat penambahan/perubahan task (karena diterapkannya scrumb) maka ia harus terus memperbarui skenario tersebut. Sehingga skenario dapat mencakup seluruh aspek aplikasi
4. Konsisten. Pada aspek ini merupakan konsisten dalam hal penamaan. Agar Skenario test tersebut lebih rapi, mudah dipahami dan tidak membingungkan. contoh : Setelah menggunakan istilah 'login', namun setelah itu menggunakan istilah 'Sign in.
5. Simple dan Transparan. Pada saat membuat skenario, semua langkah pelaksanaan harus jelas dan tidak ada yang terlewat



6. Mudah Dimengerti. Saat menyusun skenario alangkah baiknya menggunakan bahasa yang sederhana dan mudah dipahami. Sehingga orang lain pun dapat memahami skenario yang dibuat
7. End-user mind. Seorang QA tester harus bisa bermindset bahwa ia adalah seorang end-user. maka dari itu hal yang lain yang perlu diperhatikan adalah kemudahan penggunaan aplikasi sehingga konsumen mendapat kepuasan terhadap produk yang dikembangkan tersebut

## e. Alat dan Bahan

1. Kertas folio
2. Kertas A4
3. Spidol
4. Microsoft Excell
5. Jaringan Internet
6. Penggaris

## f. Prosedur Kerja

### Implementasi Pada Studi Kasus

Pembuatan Skenario test biasanya dilakukan saat aplikasi sedang dikerjakan oleh developer. sehingga saat modul atau fitur dari suatu aplikasi sudah didevelop, skenario test telah siap dan seorang QA dapat langsung melakukan pengujian berdasarkan skenario test yang telah dibuat sebelumnya. Berikut ini adalah Skenario Test dari beberapa fitur di task Maintenance Ticket Aplikasi Andriod OSVAS.

Modul	Features	Test Steps	Expected Result	Status	Note
Maintenance Ticket	View List Active Ticket		User diarahkan pada halaman Ticket	PASS	
			User diarahkan pada halaman Maintenance ticket di tab Active	PASS	
		Pada halaman My Account, User klik menu Ticket	Pada halaman maintenance ticket terdapat Button Active dan Close	PASS	
			Terdapat Button Sorting	PASS	
			Terdapat add ticket maintenance ticket	PASS	
			Button berwarna gelap (aktif) dan menampilkan list data ticket dengan status active	PASS	
			List ticket akan menampilkan data sebagai berikut : 1. Ticket Number 2. Lanlord Name 4. Property Area	PASS	
		User melihat page maintenance ticket Active	Terdapat aksi Add Note	PASS	
	Add Note	User berada di halaman tab active maintenance ticket	User berada di halaman tab active maintenance ticket	PASS	
		User klik aksi Add Note	Akan menampilkan text field Note dan button Add	PASS	
		User mengisi inputan note	Note terisi sesuai inputan	PASS	
			Terdapat alert succes	PASS	
		User klik button Add	Note akan tersimpan	PASS	

Dalam Sebuah Skenario Test komponen — komponen yang disertakan antara lain :

1. **Modul.** Pada section modul merupakan nama kasus uji/task aplikasi tersebut pada contoh di atas nama modulnya adalah Maintenance Ticket
2. **Feature.** Pada kolom features berisi fitur/kasus — kasus yang lebih detail. pada gambar di atas adalah beberapa contoh fitur dalam task maintenance ticket yaitu View List Active Ticket dan fitur add note

3. **Test Steps.** Pada Test Step berisi langkah — langkah user dalam mengoperasikan sebuah fitur pada suatu task.

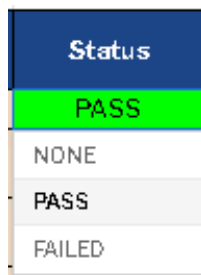
contoh : Test Steps pada fitur Add Note

User berada di halaman tab active maintenance ticket	User berada di halaman tab active maintenance ticket
User klik aksi Add Note	Akan menampilkan field Note berupa text dan button Add
User menginputkan note	Note terisi sesuai inputan
User klik button Add	Terdapat alert sukses Note akan tersimpan

4. **Expected Result.** Pada expected result berisi hasil yang diharapkan atau yang sesuai dengan requirement aplikasi dari setiap test steps yang ditulis. berikut ini merupakan contoh expected result dari fitur Add note

User berada di halaman tab active maintenance ticket	User berada di halaman tab active maintenance ticket
User klik aksi Add Note	Akan menampilkan field Note berupa text dan button Add
User menginputkan note	Note terisi sesuai inputan
User klik button Add	Terdapat alert sukses Note akan tersimpan

5. **Status.** Pada Kolom Status, disajikan berupa dropdown sehingga jika skenario telah dieksekusi maka QA dapat mengubah status tersebut apakah aplikasi sudah sesuai dengan ekspektasi yang diharapkan



6. **Note.** Pada kolom ini dapat diisi dengan catatan khusus jika hasil testing gagal.

### g. Hasil dan Pembahasan

1. Pada saat dilakukan testing, QA tester dapat menggunakan Skenario Test seperti pada contoh diatas sebagai acuan. QA tester juga dapat mengupdate skenario test tersebut saat menemukan case baru baik positif case maupun negative case . Pada saat actual result fitur yang diuji tidak sesuai dengan expected result yang ada pada skenario, maka QA dapat langsung melapor ke developer dengan menyertakan expected result dan step — step pengujian berdasarkan pada skenario test sehingga developer juga mengetahui dengan jelas seperti apa flow yang benar pada sistem yang sedang dikembangkan.
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.polije.ac.id/elearning>
3. Skenario Test dibuat sebagai acuan pada saat QA melakukan testing. kegiatan

testing berjalan lebih efektif serta menghindari adanya step yang terlewat saat proses testing dilakukan sehingga dapat langsung diketahui apakah fitur pada aplikasi berjalan normal atau tidak

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Mapping ERD ke Table	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 10**

Materi Pembelajaran : Perencanaan Proses Pengujian  
Acara Praktikum/Pertemuan : Minggu 5 / 2  
Tempat : Laboratorium PSDKU Kampus 3 Kab. Nganjuk  
Alokasi Waktu : 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa dapat Melakukan Perencanaan Proses Pengujian
2. Mahasiswa dapat menyusun skenario testing
3. Mahasiswa dapat menyusun test case

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

**c. Indikator Penilaian**

Ketepatan mengimplementasikan skenario dalam studi kasus uji coba

**d. Dasar Teori**

Beberapa hal yang perlu diperhatikan dalam membuat test case yaitu:

- Test Case dibuat Sederhana dan Transparan

Dalam pembuatan test case, penguji harus membuat test case yang sederhana dengan penjelasan jelas dan ringkas supaya orang lain dapat mengeksekusi. Gunakan bahasa yang mudah dipahami, sehingga membantu pemahaman pengujian menjadi mudah dan pelaksanaan tes atau pengujian menjadi lebih cepat.

- Test Case dibuat dengan End User in Mind

Tujuan utama dari pembangunan sebuah perangkat lunak adalah untuk membuat kasus uji yang memenuhi persyaratan client yang mudah digunakan dan dioperasikan. Dengan demikian, penguji harus mampu membuat kasus uji dengan mengingat perspektif dari pengguna akhir

- Hindari Pengulangan Kasus Uji

Dalam membuat test case, usahakan untuk tidak membuat pengujian yang berulang. Jika sebuah test case diperlukan untuk mengeksekusi beberapa test case lain, penguji dapat memanggil test case tersebut ke dalam kolom pre-condition.

- Jangan Berasumsi  
Dalam membuat test case, jangan mengasumsikan mengenai fitur dan fungsi dari perangkat yang sedang diuji. Pengujian harus sesuai dengan ketentuan dan dokumen spesifikasi.
- Pastikan bahwa Pengujian Sudah Mencakup Semuanya  
Dalam membuat pengujian, pastikan bahwa penguji sudah memeriksa semua fitur sistem secara keseluruhan serta memenuhi persyaratan perangkat lunak seperti yang disebutkan dalam dokumen spesifikasi. Penguji dapat menggunakan Traceability Matrix untuk memastikan bahwa tidak ada fungsi yang belum teruji
- Test Case Harus Dapat Diidentifikasi  
Nama dari test case id harus dapat diidentifikasi sehingga memudahkan penguji ketika akan melacak error atau mengidentifikasi persyaratan perangkat lunak pada tahap berikutnya
- Repeatable and Self-standing  
Dalam membuat test case, test case harus menghasilkan hasil yang sama setiap kali, tidak peduli siapa pengujinya.
- Peer Review  
Test case yang dibuat harus bisa ditinjau oleh orang lain. Hal ini memungkinkan orang lain menemukan sebuah cacat yang terlewatkan

### **e. Alat dan Bahan**

1. Kertas folio
2. Kertas A4
3. Spidol
4. Ms.Excell
5. Jaringan Internet
6. Penggaris

### **f. Prosedur Kerja**

1. Berkelompok, buatlah skenario dan test case yang sesuai dengan aplikasi mobile dan website.
2. Berikut contoh format test case

## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

Test Case ID	Test Case Description	Pre-Condition	Test Step	Expected Result	Actual Result	Status
Login Gmail	Valid	User login gmail account using valid data User access : https://www.google.com/intl/in/gmail/about/#	1. Click button "Login" 2. Input email: penikurniawati7@gmail.com 3. Click button "Next" 4. Input password: ***** 5. Click button "Next"	Login success and go to gmail inbox page	As Expected	Pass
	Invalid	User login gmail account using invalid data User access : https://www.google.com/intl/in/gmail/about/#	1. Click button "Login" 2. Input email: penikurniawati7@gmail.com 3. Click button "Next"	Show error message	As Expected	Pass

### 3. Menerapkan Teknik Pengujian

1. Dalam melakukan sebuah pengujian, penguji tidak memeriksa setiap kondisi yang ada dalam perangkat lunak. Teknik pengujian membantu penguji dalam memilih beberapa kasus uji dengan kemungkinan maksimum menemukan cacat.
2. Boundary Value Analysis (BVA): Teknik ini mendefinisikan pengujian batas untuk rentang nilai tertentu.
3. Equivalen Partition (EP): Teknik ini membagi rentang menjadi bagian atau kelompok yang sama yang cenderung memiliki perilaku sama.
4. State Transition Technique: Teknik ini digunakan ketika perilaku perangkat lunak berubah dari satu negara ke negara lain setelah tindakan tertentu.
5. Error Guessing Technique: Teknik ini digunakan untuk mengantisipasi kesalahan yang mungkin terjadi saat proses pengujian. Teknik ini bukan metode formal dan mengambil keuntungan dari penguji dengan aplikasi.

### g. Hasil dan Pembahasan

1. Dokumentasi terkait tugas
2. File dikumpulkan dalam berbentuk .doc dan dikumpulkan pada <http://jti.poliije.ac.id/elearning>

### h. Rubrik Penilaian

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Perencanaan Proses Pengujian	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 11**

Materi Pembelajaran	: Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Web)
Acara Praktikum/Pertemuan	: Minggu 6 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Menyiapkan data uji yang sesuai dengan kebutuhan penggunaan perangkat lunak yang diuji.
2. Mahasiswa mampu Menguasai penggunaan minimal satu standar proses penjaminan kualitas perangkat lunak.
3. Mahasiswa mampu Membaca dan memahami dokumen spesifikasi desain perangkat lunak, spesifikasi produk perangkat lunak, dan skenario uji perangkat lunak

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

Ketepatan mengidentifikasi spesifikasi kebutuhan perangkat lunak yaitu pada aplikasi web, spesifikasi perancangan perangkat lunak (pada aplikasi web), dan spesifikasi produk aplikasi web yang diuji dan ketepatan pemilihan dan penggunaan data uji.

**d. Dasar Teori**

White-box Testing

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya White Box Testing ini menguji dengan cara melihat Pure Code dari suatu aplikasi/software yang diuji tanpa memperdulikan Tampilan atau UI dari aplikasi tersebut.

Teknik White-box Testing, sebagai berikut :

6. Basis Path Testing



Basis path testing merupakan metode yang memungkinkan perancang testcase untuk membuat pengukuran kompleksitas logikal dari rancangan prosedural dan menggunakan pengukuran ini sebagai panduan untuk mendefinisikan himpunan basis dari jalur eksekusi. Test case yang dibuat untuk menguji himpunan basis dijamin akan mengeksekusi setiap statement di dalam program sekurangnya sekali pada saat pengujian

### 7. Flow Graph

Flow graph merupakan notasi sederhana untuk merepresentasi control flow.

### 8. Cyclomatic Complexity

Cyclomatic complexity digunakan untuk mengetahui jumlah jalur yang perlu dicari. Cyclomatic complexity adalah metric software yang menyediakan ukuran kuantitatif dari kompleksitas logikal program. Nilai yang dihitung bagi cyclomatic complexity menentukan jumlah jalur-jalur yang independen dalam kumpulan basis suatu program dan memberikan jumlah tes minimal yang harus dilakukan untuk memastikan bahwa semua pernyataan telah dieksekusi sekurangnya satu kali. Cyclomatic complexity mempunyai fondasi dalam teori graph dan dapat dihitung dengan satu dari tiga cara :

- ✓ Jumlah region sama dengan cyclomatic complexity.
- ✓ Cyclomatic complexity,  $V(G)$ , untuk sebuah flow graph,  $G$ , didefinisikan sebagai:  $V(G) = E - N + 2$   $E$  adalah jumlah edge pada flow graph, dan  $N$  adalah jumlah node pada flow graph.
- ✓ Cyclomatic complexity,  $V(G)$ , untuk flow graph,  $G$ , juga didefinisikan sebagai:  $V(G) = P + 1$   $P$  adalah jumlah predicate nodes yang terdapat pada flow graph  $G$ .

### 9. Graph Matrix

Prosedur untuk membuat flow graph dan menentukan himpunan basis path dapat diterima berdasarkan mekanisme. Untuk mengembangkan software yang membantu pengujian basis path, sebuah struktur data yang disebut graph matrix, dapat sangat bermanfaat. Graph matrix adalah matriks kotak yang ukurannya (jumlah baris dan kolom) sama untuk jumlah node pada flow graph. Setiap baris dan kolom berhubungan dengan node yang teridentifikasi, dan data matriks berhubungan dengan koneksi (edge) antara :

Kelemahan White Box-Testing:

10. Sangat mahal untuk dilakukan karena membutuhkan tester yang terampil untuk melakukan pengujian.

11. Pada perangkat lunak yang jenisnya besar, metode white box testing ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya.
12. Tidak mempedulikan Tampilan UI aplikasinya.

Penggunaan metode pengujian white box dilakukan untuk :

- ✓ Memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali
- ✓ Menggunakan semua keputusan logis untuk semua kondisi true atau false
- ✓ Mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.
- ✓ Menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

### e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Formulir scenario uji dan hasil uji
3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen Modul standart proses review penjamin kulaitas proses dan produk
2. Matix penilaian

### f. Prosedur Kerja

1. Mendefinisikan semua alur logika yang ada di aplikasi web pada fitur register dan login website
2. Menstrukturkan scenario pengujian pada pada fitur register dan login website dengan menggunakan salah satu Teknik White Box testing pada aplikasi web
3. Mengevaluasi semua hasil pengujian dengan menggunakan metrics pengujian
4. Melakukan pengujian secara menyeluruh
5. Membuat Dokumentasi hasil pengujian

### g. Hasil dan Pembahasan

Dokumentasi hasil pengujian pada fitur register dan login website

### h. Rubrik Penilaian

## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen hasil pengujian	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 12**

Materi Pembelajaran	:Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Web)
Acara Praktikum/Pertemuan	: Minggu 6 / 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Menyiapkan data uji yang sesuai dengan kebutuhan penggunaan perangkat lunak yang diuji.
2. Mahasiswa mampu Menguasai penggunaan minimal satu standar proses penjaminan kualitas perangkat lunak.
3. Mahasiswa mampu Membaca dan memahami dokumen spesifikasi desain perangkat lunak, spesifikasi produk perangkat lunak, dan skenario uji perangkat lunak

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

Ketepatan mahasiswa dalam mengidentifikasi spesifikasi kebutuhan perangkat lunak yaitu pada aplikasi web, spesifikasi perancangan perangkat lunak (pada aplikasi web), dan spesifikasi produk aplikasi web yang diuji dan ketepatan pemilihan dan penggunaan data uji.

**d. Dasar Teori****Black-box Testing**

Pada Black Box Testing dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh customer. Black-box Testing ini lebih menguji ke Tampilan Luar(Interface) dari suatu aplikasi agar mudah digunakan oleh Customer. Pengujian ini tidak melihat dan menguji source code program. Black-box Testing bekerja dengan mengabaikan struktur control sehingga perhatiannya hanya terfokus pada informasi domain.

Keuntungan dari Black-box Testing :

1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu

2. Pengujian yang dilakukan berdasarkan sudut pandang user agar dapat mengungkapkan inkonsistensi atau ambiguitas dalam spesifikasi.
3. Programmer dan tester memiliki ketergantungan satu sama lain

Kekurangan Black-box Testing :

1. Uji kasus sulit disain tanpa spesifikasi yang jelas
2. Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer
3. Beberapa bagian back end tidak diuji sama sekali.

### **Teknik Black-box Testing**

1. Equivalence Partitioning: Cara kerja teknik ini adalah dengan melakukan partition atau pembagian menjadi beberapa partisi dari input data.
2. Boundary Value Analysis: Teknik ini lebih fokus kepada boundary, dimana adakah error dari luar atau sisi dalam software, minimum, maupun maximum nilai dari error yang didapat.
3. Fuzzing: Fuzz merupakan teknik untuk mencari bug / gangguan dari software dengan menggunakan injeksi data yang terbilang cacat ataupun sesi semi-otomatis.
4. Cause-Effect Graph: Ini adalah teknik testing dimana menggunakan graphic sebagai acuannya. Dimana dalam grafik ini menggambarkan relasi diantara efek dan penyebab dari error tersebut.
5. Orthogonal Array Testing: Dapat digunakan jika input domain yang relatif terbilang kecil ukurannya, tetapi cukup berat untuk digunakan dalam skala besar.
6. All Pair Testing: Dalam teknik ini, semua pasangan dari test case di desain sedemikian rupa agar dapat di eksekusi semua kemungkinan kombinasi diskrit dari seluruh pasangan berdasar input parameternya. Tujuannya testing ini adalah memiliki pasangan test case yang mencakup semua pasangan tersebut.
7. State Transition: Testing ini berguna untuk melakukan pengetesan terhadap kondisi dari mesin dan navigasi dari UI dalam bentuk grafik

### **e. Alat dan Bahan**

Alat :

1. Komputer atau sejenisnya
2. Formulir scenario uji dan hasil uji
3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen Modul standart proses review penjamin kulaitas proses dan produk
2. Matix penilaian

**f. Prosedur Kerja**

1. Mendefinisikan semua alur logika yang ada di aplikasi web pada fitur register dan login website
2. Menstrukturkan scenario pengujian pada pada fitur register dan login website dengan menggunakan salah satu Teknik Black Box testing pada aplikasi web
3. Mengevaluasi semua hasil pengujian dengan menggunakan metrics pengujian
4. Melakukan pengujian secara menyeluruh
5. Membuat Dokumentasi hasil pengujian

**g. Hasil dan Pembahasan**

Dokumentasi tugas mahasiswa berupa pembuatan dokumentasi hasil pengujian pada fitur register dan login website

Skenario Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Mapping ERD ke Table	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		



**Acara 13**

Materi Pembelajaran	: Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Mobile)
Acara Praktikum/Pertemuan	: Minggu 7 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Menyiapkan data uji yang sesuai dengan kebutuhan penggunaan perangkat lunak yang diuji
2. Mahasiswa mampu Menguasai penggunaan minimal satu standar proses penjaminan kualitas perangkat lunak
3. Mahasiswa mampu Membaca dan memahami dokumen spesifikasi desain perangkat lunak, spesifikasi produk perangkat lunak, dan skenario uji perangkat lunak

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

**c. Indikator Penilaian**

Ketepatan mahasiswa dalam mengidentifikasi spesifikasi kebutuhan perangkat lunak yaitu pada aplikasi mobile, spesifikasi perancangan perangkat lunak (pada aplikasi mobile), dan spesifikasi produk aplikasi mobile yang diuji dan ketepatan pemilihan dan penggunaan data uji

**d. Dasar Teori****White-Box Testing**

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya White Box Testing ini menguji dengan cara melihat Pure Code dari suatu aplikasi/software yang diuji tanpa memperdulikan Tampilan atau UI dari aplikasi tersebut.

### Teknik White-box Testing, sebagai berikut :

#### 1. Basis Path Testing.

Basis path testing merupakan metode yang memungkinkan perancang testcase untuk membuat pengukuran kompleksitas logikal dari rancangan prosedural dan menggunakan pengukuran ini sebagai panduan untuk mendefinisikan himpunan basis dari jalur eksekusi. Test case yang dibuat untuk menguji himpunan basis dijamin akan mengeksekusi setiap statement di dalam program sekurangnyanya sekali pada saat pengujian

#### 2. Flow Graph

Flow graph merupakan notasi sederhana untuk merepresentasi control flow.

#### 3. Cyclomatic Complexity

Cyclomatic complexity digunakan untuk mengetahui jumlah jalur yang perlu dicari. Cyclomatic complexity adalah metric software yang menyediakan ukuran kuantitatif dari kompleksitas logikal program. Nilai yang dihitung bagi cyclomatic complexity menentukan jumlah jalur-jalur yang independen dalam kumpulan basis suatu program dan memberikan jumlah tes minimal yang harus dilakukan untuk memastikan bahwa semua pernyataan telah dieksekusi sekurangnyanya satu kali.

Cyclomatic complexity mempunyai fondasi dalam teori graph dan dapat dihitung dengan satu dari tiga cara :

- Jumlah region sama dengan cyclomatic complexity.
- Cyclomatic complexity,  $V(G)$ , untuk sebuah flow graph,  $G$ , didefinisikan sebagai:  $V(G) = E - N + 2$   $E$  adalah jumlah edge pada flow graph, dan  $N$  adalah jumlah node pada flow graph.
- Cyclomatic complexity,  $V(G)$ , untuk flow graph,  $G$ , juga didefinisikan sebagai:  $V(G) = P + 1$   $P$  adalah jumlah predicate nodes yang terdapat pada flow graph  $G$ .

#### 4. Graph Matrix

Prosedur untuk membuat flow graph dan menentukan himpunan basis path dapat diterima berdasarkan mekanisme. Untuk mengembangkan software yang membantu pengujian basis path, sebuah struktur data yang disebut graph matrix, dapat sangat bermanfaat. Graph matrix adalah matriks kotak yang ukurannya (jumlah baris dan kolom) sama untuk jumlah node pada flow graph. Setiap baris dan kolom berhubungan dengan node yang teridentifikasi, dan data matriks berhubungan dengan koneksi (edge) antara

#### 5. Kelemahan White Box-Testing:

- a) Sangat mahal untuk dilakukan karena membutuhkan tester yang terampil untuk melakukan pengujian.

- b) Pada perangkat lunak yang jenisnya besar, metode white box testing ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya.
  - c) Tidak mempedulikan Tampilan UI aplikasinya.
6. Penggunaan metode pengujian white box dilakukan untuk :
- a) Memberikan jaminan bahwa semua jalur independen suatu modul digunakan minimal satu kali
  - b) Menggunakan semua keputusan logis untuk semua kondisi true atau false
  - c) Mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi.
  - d) Menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

**e. Alat dan Bahan**

Alat :

- 1. Komputer atau sejenisnya
- 2. Formulir scenario uji dan hasil uji
- 3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

- 1. Dokumen Modul standart proses review penjamin kulaitas proses dan produk
- 2. Matix penilaian

**f. Prosedur Kerja**

- 1. Mendefinisikan semua alur logika yang ada di aplikasi web pada fitur register dan login website
- 2. Menstrukturkan scenario pengujian pada pada fitur register dan login website dengan menggunakan salah satu Teknik Black Box testing pada aplikasi web
- 3. Mengevaluasi semua hasil pengujian dengan menggunakan metrics pengujian
- 4. Melakukan pengujian secara menyeluruh
- 5. Membuat Dokumentasi hasil pengujian

**g. Hasil dan Pembahasan**

- 1. Dokumentasi hasil pengujian pada fitur register dan login mobile
- 2. Dokumentasi hasil pengujian pada fitur register dan login mobile

## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

Skenario Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan

### h. Rubrik Penilaian

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Dokumentasi hasil pengujian	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 14**

Materi Pembelajaran	: Melakukan Pengujian Kualitas Perangkat Lunak Secara Manual (Mobile)
Acara Praktikum/Pertemuan	: Minggu 7 / 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu Menyiapkan data uji yang sesuai dengan kebutuhan penggunaan perangkat lunak yang diuji
2. Mahasiswa mampu Menguasai penggunaan minimal satu standar proses penjaminan kualitas perangkat lunak
3. Mahasiswa mampu Membaca dan memahami dokumen spesifikasi desain perangkat lunak, spesifikasi produk perangkat lunak, dan skenario uji perangkat lunak

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

Ketepatan mengidentifikasi spesifikasi kebutuhan perangkat lunak yaitu pada aplikasi mobile, spesifikasi perancangan perangkat lunak (pada aplikasi mobile), dan spesifikasi produk aplikasi mobile yang diuji dan ketepatan pemilihan dan penggunaan data uji.

**d. Dasar Teori**

Dapat dilihat pada Minggu ke-6/2, Acara 12.

**e. Alat dan Bahan**

Alat :

1. Komputer atau sejenisnya
2. Formulir scenario uji dan hasil uji
3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen Modul standart proses review penjamin kualitas proses dan produk

## 2. Matix penilaian

**f. Prosedur Kerja**

1. Mendefinisikan semua alur logika yang ada di aplikasi web pada fitur register dan login website
2. Menstrukturkan scenario pengujian pada pada fitur register dan login aplikasi mobile dengan menggunakan salah satu Teknik Black Box testing pada aplikasi mobile
3. Mengevaluasi semua hasil pengujian dengan menggunakan *metrics* pengujian
4. Melakukan pengujian secara menyeluruh
5. Membuat Dokumentasi hasil pengujian

**g. Hasil dan Pembahasan**

1. Lakukan pengujian unit pada fitur login/registrasi aplikasi web yang kelompok Anda kembangkan.
2. Buat dokumen hasil testing yang telah dilakukan
3. Dokumen hasil testing

Skenario Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi Dokumen hasil testing	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 15 dan 16**

Materi Pembelajaran	: Melakukan Pengujian Kualitas Perangkat Lunak Secara Otomatis (Web)
Acara Praktikum/Pertemuan	: Minggu 9 / 1 dan 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu melakukan pengujian perangkat lunak secara otomatis
2. Mahasiswa mampu menyusun dokumen hasil pengujian perangkat lunak

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

**c. Indikator Penilaian**

Keberhasilan dalam menggunakan tools dalam pelaksanaan pengujian perangkat lunak, ketepatan dalam menyusun dokumen hasil pengujian perangkat lunak

**d. Dasar Teori****1. Pengujian Perangkat Lunak Otomatis**

Pengujian otomatis adalah aplikasi perangkat lunak untuk mengotomatisasi proses manual yang digerakkan oleh manusia untuk meninjau dan memvalidasi produk perangkat lunak. Sebagian besar proyek perangkat lunak agile dan DevOps modern sekarang menyertakan pengujian otomatis sejak awal. Namun, untuk sepenuhnya menghargai nilai pengujian otomatis, ada baiknya untuk memahami seperti apa kehidupan sebelum diadopsi secara luas.

Kembali ketika pengujian manual adalah norma, itu adalah praktik umum bagi perusahaan perangkat lunak untuk mempekerjakan tim QA penuh waktu. Tim ini akan mengembangkan kumpulan 'rencana pengujian,' atau daftar periksa langkah demi langkah yang menegaskan fitur proyek perangkat lunak berperilaku seperti yang diharapkan. Tim QA kemudian akan secara manual menjalankan daftar periksa ini setiap kali pembaruan atau perubahan baru didorong ke proyek perangkat lunak, kemudian mengembalikan hasil rencana pengujian ke tim teknik untuk ditinjau dan pengembangan lebih lanjut untuk mengatasi masalah.



Proses ini lambat, mahal, dan rawan kesalahan. Pengujian otomatis membawa keuntungan besar untuk efisiensi tim dan ROI tim jaminan kualitas. Pengujian otomatis menempatkan tanggung jawab kepemilikan di tangan tim teknik. Rencana pengujian dikembangkan bersamaan dengan pengembangan fitur peta jalan reguler kemudian dijalankan secara otomatis oleh alat integrasi berkelanjutan perangkat lunak. Pengujian otomatis mempromosikan ukuran tim QA yang ramping dan memungkinkan tim QA untuk fokus pada fitur yang lebih sensitif.

Continuous delivery (CD) adalah tentang mengirimkan rilis kode baru secepat mungkin kepada pelanggan. Pengujian otomatis sangat penting untuk tujuan itu. Tidak ada cara untuk mengotomatiskan pengiriman ke pengguna jika ada langkah manual yang memakan waktu dalam proses pengiriman. CD adalah bagian dari jalur penyebaran yang lebih besar. CD adalah penerus dan juga bergantung pada continuous integration (CI). CI bertanggung jawab penuh untuk menjalankan pengujian otomatis terhadap setiap perubahan kode baru dan memverifikasi bahwa perubahan tersebut tidak merusak fitur yang sudah ada atau memperkenalkan bug baru. CD dipicu setelah langkah integrasi berkelanjutan melewati rencana pengujian otomatis.

Hubungan antara pengujian otomatis, CI, dan CD ini menghasilkan banyak manfaat bagi tim perangkat lunak berkecepatan tinggi. Pengujian otomatis memastikan kualitas di setiap tahap pengembangan dengan memastikan komitmen baru tidak menimbulkan bug apa pun, sehingga perangkat lunak tetap siap digunakan setiap saat.

Pengujian otomatis dapat meningkatkan efisiensi tim QA. Beberapa manfaat antara lain:

- Akurasi yang lebih tinggi
- Kemampuan pelaporan yang lebih baik
- Peningkatan cakupan
- Peningkatan efisiensi sumber daya
- Deteksi bug yang ditingkatkan
- Peningkatan penggunaan kembali

## **2. Unit Testing**

Unit testing adalah suatu pekerjaan yang harus dilakukan ketika mengembangkan suatu produk, terutama produk software. Karena pekerjaan ini sangat penting untuk memastikan kualitas setiap komponen yang ada di dalamnya. Umumnya, unit testing dikerjakan oleh seorang software tester atau software developer. Nah, pada kesempatan kali ini kita akan membahas secara lengkap tentang unit testing dan berbagai tools yang biasa digunakan. unit testing adalah suatu jenis software testing yang dilakukan untuk

menguji coba suatu bagian ataupun komponen yang ada pada software. Unit tersebut bisa berbentuk fungsi, kode, metode, prosedur, modul ataupun objek itu sendiri.

Unit testing dilakukan agar bisa memastikan setiap unit kode yang terdapat pada software bisa berjalan sebagaimana mestinya. Laman resmi SmartBear menjelaskan “smaller is better”, maksudnya, semakin kecil unit yang diuji, maka Anda akan melihat dan juga memastikan kinerja software tersebut semakin detail.

### **3. Tools Unit Testing**

Seperti yang sudah dijelaskan sebelumnya, unit testing adalah suatu pengujian software yang dilakukan secara manual ataupun otomatis dengan menggunakan alat bantu atau tools khusus. Tapi, kebanyakan developer saat ini lebih memilih metode otomatis. Berikut ini adalah beberapa tools yang bisa Anda pilih untuk melakukan unit testing.

- JUnit: tools unit testing untuk aplikasi yang menggunakan bahasa pemrograman Java
- NUnit: framework unit testing untuk aplikasi yang menggunakan bahasa pemrograman .NET
- JMockit: tools unit testing untuk aplikasi yang berbasis open source
- EMMA: tools untuk menganalisis dan melaporkan kode yang menggunakan bahasa pemrograman Java
- PHPUnit: tools unit testing untuk PHP programmer

#### **e. Alat dan Bahan**

Alat:

- Laptop

Bahan:

- PHP Unit

#### **f. Prosedur Kerja**

1. Lakukan instalasi aplikasi PHP unit. Dapat di download pada link <https://phpunit.de/>, pada tutorial ini menggunakan versi 8.
2. Pertama-tama kita membutuhkan framework PHPUnit. Halaman website PHPUnit dapat ditemukan pada alamat <https://phpunit.de/>. Saat ini PHPUnit sudah mencapai versi 9. Nah, untuk latihan kali ini kita akan gunakan versi 8 yang terdaftar secara default pada repository Composer. Silahkan buka folder latihan-php-unit, lalu jalankan perintah ini pada CLI :

```
composer require phpunit/phpunit
```

Tampilannya akan seperti di bawah ini:

```
oriza@LAPTOP-PFHSSPTC: /var/www/html/lab/latihan-php-unit$ cd ../
oriza@LAPTOP-PFHSSPTC: /var/www/html/lab$ clear
oriza@LAPTOP-PFHSSPTC: /var/www/html/lab$ cd latihan-php-unit/
oriza@LAPTOP-PFHSSPTC: /var/www/html/lab/latihan-php-unit$ composer require phpunit/phpunit
Using version ^8.5 for phpunit/phpunit
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 29 installs, 0 updates, 0 removals
- Installing sebastian/version (2.0.1): Loading from cache
- Installing sebastian/type (1.1.3): Loading from cache
- Installing sebastian/resource-operations (2.0.1): Loading from cache
- Installing sebastian/recursion-context (3.0.0): Loading from cache
- Installing sebastian/object-reflector (1.1.1): Loading from cache
- Installing sebastian/object-enumerator (3.0.3): Loading from cache
- Installing sebastian/global-state (3.0.0): Loading from cache
- Installing sebastian/exporter (3.1.2): Loading from cache
- Installing sebastian/environment (4.2.3): Loading from cache
- Installing sebastian/diff (3.0.2): Loading from cache
- Installing sebastian/comparator (3.0.2): Loading from cache
- Installing phpunit/php-timer (2.1.2): Loading from cache
- Installing phpunit/php-text-template (1.2.1): Loading from cache
- Installing phpunit/php-file-iterator (2.0.2): Loading from cache
- Installing theseer/tokenizer (1.1.3): Loading from cache
- Installing sebastian/code-unit-reverse-lookup (1.0.1): Loading from cache
- Installing phpunit/php-token-stream (3.1.1): Loading from cache
- Installing phpunit/php-code-coverage (7.0.10): Loading from cache
- Installing doctrine/instantiator (1.3.0): Loading from cache
- Installing phpdocblock/reflection-common (2.0.0): Loading from cache
- Installing symfony/polyfill-type (v1.13.1): Loading from cache
- Installing webmozart/assert (1.6.0): Loading from cache
- Installing phpdocblock/type-resolver (1.0.1): Loading from cache
- Installing phpdocblock/reflection-docblock (5.0.0): Loading from cache
```

3. Jika Anda buka folder latihan-php-unit sekarang akan terisi file composer.json, composer.lock beserta folder vendor. Sampai sini kita sudah berhasil meinstall PHP Unit.
4. Membuat class contoh. Dalam kasus sekarang kita ceritanya mempunyai sebuah library yang ingin kita test. Library ini kita beri nama Wordcount.php. Yang berisi sebuah method untuk menghitung jumlah kata dalam kalimat. Silahkan buat sebuah file bernama Wordcount.php berisi kode berikut :

```
<?php
// File : WordCount.php
class Wordcount
{
    public function countWords($sentence)
    {
        return count(explode(" ", $sentence));
    }
}
?>
```

5. Membuat test. Untuk membuat test kita membutuhkan komponen class dari PHPUnit. Komponen tersebut sudah terinstall dalam folder vendor saat menjalankan Composer. Sekarang buatlah file bernama SimpleTest.php pada folder latihan-php-unit berisi kode berikut ini :

```
// Path to run ./vendor/bin/phpunit --bootstrap vendor/autoload.php FileName.php
// Butuh Framework PHPUnit
use PHPUnit\Framework\TestCase;

// Class yang mau di TEST.
require_once "Wordcount.php";

// Class untuk run Testing.
class SimpleTest extends TestCase
{
    public function testCountWords()
    {
        // Kita pakai class yang mau kita test.
        $Wc = new WordCount();

        // Kita masukan parameter 4 kata, yang harusnya dapat output 4.
        $TestSentence = "My name is Joko"; // 4 Kata ..
        $WordCount = $Wc->countWords($TestSentence);

        // Kita assert equal, ekspektasi nya harus 4, jika benar berarti Wordcount berfungsi dengan baik.
        $this->assertEquals(4, $WordCount);
    }
}
```

Pada kode tersebut kita use komponen PHPUnit yang sudah ada dalam folder vendor. Lalu kita juga require class yang sudah kita bikin sebelumnya yaitu Wordcount.php. Dalam kode kita juga membuat class bernama SimpleTest yang meng-extend TestCase milik PHPUnit. Dalam penggunaannya, kita minimal harus membuat sebuah method dalam TestCase, disini kita bernama testCountWords.

Sebenarnya penamaannya bebas terserah teman-teman, nama pada contoh diatas adalah nama yang paling relevan dengan kasus kita. Pada method testCountWords kita membuat instance dari class yang ingin kita test, lalu menjalankan assertEquals.

6. Jalankan Test. Silahkan buka folder proyek latihan-php-unit lalu jalankan perintah ini :

```
./vendor/bin/phpunit --bootstrap vendor/autoload.php SimpleTest.php
```

Disini kita memanggil binary program phpunit dan memerintah untuk mengetest file SimpleTest.php yang sudah kita buat sebelumnya. Silahkan klik enter/run maka hasilnya akan seperti ini :

## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

```
oriza@LAPTOP-PFHSSPTC: /var/www/html/lab/latihan-php-unit$ ./vendor/bin/phpunit --bootstrap
vendor/autoload.php SimpleTest.php
PHPUnit 8.5.2 by Sebastian Bergmann and contributors.

.
1 / 1 (100%)

Time: 104 ms, Memory: 4.00 MB

OK (1 test, 1 assertion)
oriza@LAPTOP-PFHSSPTC: /var/www/html/lab/latihan-php-unit$
```

Perintah `assertEqual` akan membandingkan parameter pertama (output yang diinginkan) dan parameter kedua (method yang di run), jika match, berarti test berhasil. Kita hanya membandingkan hasil yang diinginkan dengan hasil real output class yang ingin di test. Jika sesuai ekspektasi maka si class tersebut berjalan dengan baik. Jika ternyata saat di test hasil tidak sama, berarti class kita ada bug / error. String "My name is Joko" harus selalu mengeluarkan hasil 4.

### d. Tugas Praktikum

1. Lakukan pengujian unit pada fitur login/registrasi aplikasi mobile yang kelompok Anda kembangkan.
2. Buat dokumen hasil testing yang telah dilakukan

### a. Hasil dan Pembahasan

1. Dokumen Hasil Testing.

### b. Rubrik Penilaian

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen Hasil Testing	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		



## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		



### Acara 17 dan 18

Materi Pembelajaran	: Melakukan Pengujian Kualitas Perangkat Lunak Secara Otomatis (Mobile)
Acara Praktikum/Praktik	: Minggu 10 / 1 dan 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk
Alokasi Waktu	: 2 x 120 menit

#### a. **Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu melakukan pengujian perangkat lunak secara otomatis
2. Mahasiswa mampu menyusun dokumen hasil pengujian perangkat lunak

#### b. **Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

#### c. **Indikator Penilaian**

Keberhasilan dalam menggunakan tools dalam pelaksanaan pengujian perangkat lunak, ketepatan dalam menyusun dokumen hasil pengujian perangkat lunak

#### d. **Dasar Teori**

Dapat dilihat pada Minggu ke-8 / 1 dan 2 , Acara 15-16.

#### e. **Alat dan Bahan**

Alat:

- Laptop

Bahan:

- Android studio

#### f. **Prosedur Kerja**

##### **I. Membuat pengujian lokal**

Logika aplikasi dapat dievaluasi menggunakan pengujian unit lokal saat Anda perlu menjalankan pengujian lebih cepat dan tidak memerlukan fidelitas serta keyakinan yang terkait dengan menjalankan pengujian di perangkat sebenarnya. Dengan pendekatan ini, Anda biasanya dapat memenuhi hubungan dependensi menggunakan Robolectric atau framework tiruan, seperti Mockito. Biasanya, jenis dependensi yang terkait dengan pengujian menentukan alat mana yang Anda gunakan:



4. Jika memiliki dependensi pada framework Android, khususnya yang membuat interaksi kompleks dengan framework, sebaiknya Anda menyertakan dependensi framework menggunakan Robolectric.
5. Jika pengujian memiliki dependensi minimal pada framework Android, atau jika pengujian hanya bergantung pada objek Anda sendiri, Anda dapat menyertakan dependensi tiruan menggunakan framework tiruan seperti Mockito.

### II. Menyiapkan lingkungan pengujian

1. Dalam project Android Studio, Anda harus menyimpan file sumber untuk pengujian unit lokal di `module-name/src/test/java/`. Direktori ini sudah ada saat Anda membuat project baru.
2. Anda juga perlu mengonfigurasi dependensi pengujian untuk project Anda agar dapat menggunakan API standar yang diberikan oleh framework JUnit 4. Jika pengujian Anda perlu berinteraksi dengan dependensi Android, sertakan Robolectric atau library Mockito untuk menyederhanakan pengujian unit lokal Anda.
3. Di file `build.gradle` tingkat teratas aplikasi, tentukan library berikut sebagai dependensi:

```
dependencies {  
    // Required -- JUnit 4 framework  
    testImplementation 'junit:junit:4.12'  
    // Optional -- Robolectric environment  
    testImplementation 'androidx.test:core:1.0.0'  
    // Optional -- Mockito framework  
    testImplementation 'org.mockito:mockito-core:1.10.19'  
}
```

### III. Membuat class pengujian unit lokal

Class pengujian unit lokal Anda harus ditulis sebagai class pengujian JUnit 4. JUnit adalah framework pengujian unit yang paling populer dan banyak digunakan untuk Java. JUnit 4 memungkinkan Anda untuk menulis pengujian dengan cara yang lebih rapi dan fleksibel daripada versi pendahulunya karena JUnit 4 tidak mengharuskan Anda untuk melakukan hal-hal berikut:

1. Perluas class `junit.framework.TestCase`.
2. Berikan awalan pada nama metode pengujian Anda dengan kata kunci 'test'.
3. Gunakan class dari paket `junit.framework` atau `junit.extensions`.
4. Untuk membuat class pengujian JUnit 4 dasar, buat class yang berisi satu atau beberapa metode pengujian. Sebuah metode pengujian dimulai

dengan anotasi `@Test` dan berisi kode untuk menerapkan serta memverifikasi fungsionalitas tunggal dalam komponen yang ingin Anda uji.

5. Contoh berikut menunjukkan cara mengimplementasikan class pengujian unit lokal.

Metode pengujian `email Validator_CorrectEmailSimple_ReturnsTrue` memverifikasi bahwa metode `isValidEmail()` di aplikasi yang diuji menampilkan hasil yang benar.

Kotlin:

```
import com.google.common.truth.Truth.assertThat
import org.junit.Test

class EmailValidatorTest {
    @Test
    fun emailValidator_CorrectEmailSimple_ReturnsTrue() {
        assertThat(EmailValidator.isValidEmail("name@email.com")).isTrue()
    }
}
```

Java:

```
import com.google.common.truth.Truth.assertThat;
import org.junit.Test;

public class EmailValidatorTest {
    @Test
    public void emailValidator_CorrectEmailSimple_ReturnsTrue() {
        assertThat(EmailValidator.isValidEmail("name@email.com")).isTrue();
    }
}
```

Untuk membuat pengujian dapat dibaca yang mengevaluasi apakah komponen-komponen dalam aplikasi Anda menunjukkan hasil yang diharapkan, sebaiknya gunakan library Truth dan class dari Android Assertion, seperti yang ditunjukkan di contoh sebelumnya. Untuk mempelajari lebih lanjut jenis validasi logika yang didukung oleh Truth dan Android Assertion, lihat bagian yang menjelaskan cara membuat pernyataan yang lebih mudah dibaca.

Jika Anda lebih senang membandingkan antara hasil yang diharapkan dengan hasil sebenarnya menggunakan metode `junit.Assert` atau `Hamcrest matchers` (seperti metode `is()` dan `equalTo()`), Anda dapat menggunakan library tersebut sebagai gantinya.

#### IV. Menyertakan dependensi framework

Jika pengujian Anda berinteraksi dengan beberapa dependensi framework Android, atau berinteraksi dengan dependensi tersebut secara kompleks, gunakan artefak Robolectric yang diberikan oleh Pengujian AndroidX.

Robolectric menjalankan kode framework Android asli dan kode framework native palsu di JVM lokal atau di perangkat sebenarnya.

Contoh berikut menunjukkan cara membuat pengujian unit yang menggunakan Robolectric:

app/build.gradle

```
android {
    // ...
    testOptions {
        unitTests.includeAndroidResources = true
    }
}
```

MyLocalUnitTestClass

Kotlin:

```
import android.content.Context
import androidx.test.core.app.ApplicationProvider
import com.google.common.truth.Truth.assertThat
import org.junit.Test

private const val FAKE_STRING = "HELLO_WORLD"

class UnitTestSample {
    val context = ApplicationProvider.getApplicationContext<Context>()

    @Test fun readStringFromContext_LocalizedString() {
        // Given a Context object retrieved from Robolectric...
        val myObjectUnderTest = ClassUnderTest(context)

        // ...when the string is returned from the object under test...
        val result: String = myObjectUnderTest.getHelloWorldString()

        // ...then the result should be the expected one.
        assertThat(result).isEqualTo(FAKE_STRING)
    }
}
```

Java

```
import android.content.Context;
import androidx.test.core.app.ApplicationProvider;
import org.junit.Test;

import static com.google.common.truth.Truth.assertThat;

public class UnitTestSampleJava {
    private static final String FAKE_STRING = "HELLO_WORLD";
    private Context context = ApplicationProvider.getApplicationContext();

    @Test
    public void readStringFromContext_LocalizedString() {
        // Given a Context object retrieved from Robolectric...
        ClassUnderTest myObjectUnderTest = new ClassUnderTest(context);

        // ...when the string is returned from the object under test...
        String result = myObjectUnderTest.getHelloWorldString();

        // ...then the result should be the expected one.
        assertThat(result).isEqualTo(FAKE_STRING);
    }
}
```

### V. Menyertakan class android builder

Jika membuat pengujian unit lokal yang dijalankan di lingkungan Robolectric atau di perangkat sebenarnya, Anda dapat menggunakan builder yang disediakan Pengujian AndroidX untuk beberapa class framework umum. Builder tersebut memungkinkan Anda untuk membuat instance dari class berikut tanpa perlu menggunakan tiruan atau refleksi:

- ~ Application Info
- ~ PackageInfoMotionEvent
- ~ MotionEvent.PointerCoords
- ~ MotionEvent.PointerProperties

### VI. Menggunakan class utilitas parcelables

Selain itu, library juga memberikan class utilitas untuk objek Parcelable. Dengan memberikan objek Creator, class ini melakukan unmarshal kepada objek Parcelable yang diberikan, lalu melakukan marshal kepada objek Parcelable duplikat.

### VII. Menyertakan dependensi tiruan

Secara default, Plugin Android untuk Gradle menjalankan pengujian unit lokal Anda terhadap versi modifikasi library android.jar, yang tidak berisi kode aktual. Sebagai gantinya, metode yang memanggil ke class Android dari pengujian unit Anda akan memunculkan pengecualian. Perilaku ini memastikan Anda hanya menguji kode Anda dan tidak bergantung pada perilaku tertentu dari platform Android (yang belum dibuat atau ditiru secara eksplisit).

### VIII. Meniru dependensi android

Jika Anda memiliki dependensi Android minimal serta perlu menguji interaksi spesifik antara komponen dan dependensinya dalam aplikasi, gunakan framework tiruan untuk menghentikan dependensi eksternal dalam kode Anda. Dengan begitu, Anda dapat dengan mudah menguji apakah komponen berinteraksi dengan dependensi menggunakan cara yang diharapkan. Dengan mengganti dependensi Android dengan objek tiruan, Anda dapat memisahkan pengujian unit dari sistem Android lainnya sambil memverifikasi bahwa metode yang benar dalam dependensi tersebut dipanggil. Framework tiruan Mockito untuk Java (versi 1.9.5 dan yang lebih tinggi) menawarkan kompatibilitas dengan pengujian unit

Android. Dengan Mockito, Anda dapat mengonfigurasi objek tiruan untuk menampilkan nilai tertentu saat dipanggil.

Untuk menambahkan objek tiruan ke pengujian unit lokal Anda menggunakan framework ini, ikuti model pemrograman berikut:

1. Sertakan dependensi library Mockito dalam file build.gradle, seperti yang dijelaskan dalam Menyiapkan lingkungan pengujian Anda.
2. Di awal definisi class pengujian unit, tambahkan anotasi `@RunWith(MockitoJUnitRunner.class)`. Anotasi ini memberi tahu runner pengujian Mockito untuk memvalidasi bahwa penggunaan framework sudah benar dan menyederhanakan inisialisasi objek tiruan Anda.
3. Untuk membuat objek tiruan untuk dependensi Android, tambahkan anotasi `@Mock` sebelum deklarasi kolom.
4. Untuk menghentikan perilaku dependensi, Anda dapat menentukan kondisi dan mengembalikan nilai saat kondisi terpenuhi dengan menggunakan metode `when()` dan `thenReturn()`.

Contoh berikut menunjukkan cara membuat pengujian unit yang menggunakan objek Context tiruan.

### Kotlin

```
import android.content.Context
import com.google.common.truth.Truth.assertThat
import org.junit.Test
import org.junit.runner.RunWith
import org.mockito.Mock
import org.mockito.Mockito.`when`
import org.mockito.junit.MockitoJUnitRunner

private const val FAKE_STRING = "HELLO WORLD"

@RunWith(MockitoJUnitRunner::class)
class UnitTestSample {

    @Mock
    private lateinit var mockContext: Context

    @Test
    fun readStringFromContext_LocalizedString() {
        // Given a mocked Context injected into the object under test...
        `when`(mockContext.getString(R.string.hello_word))
            .thenReturn(FAKE_STRING)
        val myObjectUnderTest = ClassUnderTest(mockContext)

        // ...when the string is returned from the object under test...
        val result: String = myObjectUnderTest.getHelloWorldString()

        // ...then the result should be the expected one.
        assertThat(result, `is`(FAKE_STRING))
    }
}
```

## Java

```
import android.content.Context;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.mockito.Mock;
import org.mockito.junit.MockitoJUnitRunner;

import static com.google.common.truth.Truth.assertThat;
import static org.mockito.Mockito.when;

@RunWith(MockitoJUnitRunner.class)
public class UnitTestSample {

    private static final String FAKE_STRING = "HELLO WORLD";

    @Mock
    Context mockContext;

    @Test
    public void readStringFromContext_LocalizedString() {
        // Given a mocked Context injected into the object under test...
        when(mockContext.getString(R.string.hello_world))
            .thenReturn(FAKE_STRING);
        ClassUnderTest myObjectUnderTest = new ClassUnderTest(mockContext);

        // ...when the string is returned from the object under test...
        String result = myObjectUnderTest.getHelloWorldString();

        // ...then the result should be the expected one.
        assertThat(result, is(FAKE_STRING));
    }
}
```

## IX. Menjalankan pengujian unit lokal

Untuk menjalankan pengujian unit lokal, gunakan langkah-langkah berikut:

1. Pastikan project Anda disinkronkan dengan Gradle dengan mengklik Sync Project di toolbar.
2. Jalankan pengujian Anda dengan salah satu cara berikut:
3. Untuk menjalankan pengujian tunggal, buka jendela Project, lalu klik kanan pengujian dan klik Run .
4. Untuk menguji semua metode di suatu class, klik kanan class atau metode di file pengujian dan klik Run .
5. Untuk menjalankan semua pengujian di suatu direktori, klik kanan pada direktori dan pilih Run tests .

Plugin Android untuk Gradle mengompilasi kode pengujian unit lokal yang terletak di direktori default (src/test/java/), membuat aplikasi pengujian, dan menjalankannya secara lokal menggunakan class runner pengujian default. Android Studio kemudian menampilkan hasilnya di jendela Run.



**g. Hasil dan Pembahasan**

Dokumentasi latihan mahasiswa berupa Dokumen Hasil Testing.

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen Hasil Testing	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		



**Acara 19**

Materi Pembelajaran	: Pengujian Aspek Kebergunaan Kepada Pengguna
Acara Praktikum/Praktik	: Minggu 11 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk Alokasi
Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. melakukan Pengujian Aspek Kebergunaan Kepada Pengguna
2. menyusun dokumen hasil Pengujian Aspek Kebergunaan Kepada Pengguna
3. Menganalisa acuan-acuan Pengujian Aspek Kebergunaan Kepada Pengguna

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

1. Mahasiswa mampu melakukan Pengujian Aspek Kebergunaan Kepada Pengguna
2. Mahasiswa mampu menyusun dokumen hasil Pengujian Aspek Kebergunaan Kepada Pengguna
3. Mahasiswa mampu Menganalisa acuan-acuan Pengujian Aspek Kebergunaan Kepada Pengguna

**d. Dasar Teori**

Membangun sebuah website, apalagi untuk keperluan bisnis merupakan ide yang cemerlang. Tapi, bagaimana caranya agar tahu apakah website Anda sudah cukup baik dan ideal untuk digunakan? Usability testing adalah jawabannya. Dengan melakukan usability testing, Anda bisa mengukur kualitas website Anda untuk menciptakan user experience yang baik dan mengadopsi sistem navigasi yang mudah. Metode usability testing memerlukan aturan tertentu untuk memastikan hasilnya benar-benar valid.

Usability testing adalah sebuah metode untuk mengevaluasi user experience terhadap software ataupun website yang dibuat. Umumnya, metode ini dilakukan oleh para UX developer dengan melibatkan beberapa user (pengguna) tertentu untuk diteliti bagaimana proses mereka selama berinteraksi dengan website.

Bisa dibilang kalau metode ini sebaiknya tidak dilewatkan, mengingat user experience merupakan faktor terpenting dalam perilsan sebuah website ataupun software.

Terdapat 5 macam manfaat dari usability testing, yaitu:

1. Mengatasi masalah internal — Usability testing menawarkan pemeriksaan yang tidak bias, akurat, dan langsung berkaitan dengan pengalaman pengguna website. Bahkan, dengan melakukan metode ini, segala masalah internal yang terdapat di dalam tim Anda bisa cepat terselesaikan. Sebab Anda dan tim pun bisa tahu kapan waktu yang pas untuk merilis website.
2. Proses mudah dan cepat — Menggarap proses usability testing juga sangat mudah dan cepat. Yang perlu Anda siapkan hanyalah ruangan kosong, beserta alat perekam untuk mengabadikan proses pengujian antara Anda dengan pengguna.
3. Mengetahui setiap aktivitas pengunjung — Sudah cukup jelas, kalau usability testing akan membantu Anda untuk tahu apa saja yang dilakukan pengguna selama berada di dalam website Anda dan mengapa mereka melakukan hal tersebut.
4. Mengecek kekurangan website — Usability testing memungkinkan Anda untuk mengetahui apa saja kekurangan atau kesalahan yang terdapat di dalam website Anda. Sehingga, biaya yang dikeluarkan selama merancang website pun menjadi tidak sia-sia.
5. Strategi mengembangkan bisnis — Metode ini juga bermanfaat pada pertumbuhan bisnis. Karena user experience merupakan salah satu pendukung untuk membuat produk Anda laku di pasaran.

✓ Langkah Metode Usability Testing untuk Website :

1. Tentukan bagian website yang akan diuji
2. Siapkan prototype
3. Buat parameter keberhasilan

Berikut adalah tiga parameter yang bisa Anda gunakan:

- Efektifitas — kemudahan pengguna dalam mencapai tujuan yang Anda inginkan
- Efisiensi — waktu yang diperlukan pengguna untuk mencapai tujuan tersebut
- Kepuasan — tingkat kesenangan yang didapatkan pengguna selama proses untuk mencapai tujuan tersebut

4. Mempersiapkan skenario pengujian
5. Menentukan calon pengguna yang akan diuji
6. Memulai tahap pengujian

Contoh completion rate

### Completion Rate

	Step 1	Step 2	Step 3
User 1	●		
User 2			
User 3			
User 4			
User 5			

- Cepat
- Sedikit lambat
- Stop

## 7. Analisis hasil data pengujian

Intinya, dari setiap tahap proses pengujian, Anda perlu fokus kepada segala inti masalah yang muncul untuk selanjutnya dikembangkan dalam bentuk versi terbaru. Perlu diingat, bahwa Usability Testing sangat dianjurkan bagi Anda yang hendak melakukan pembaharuan terhadap setiap versi. Tujuannya adalah untuk menemukan apa saja kendala yang terdapat di bagian awal sebelum website siap dirilis.

✓ Beberapa Contoh Usability Testing adalah sebagai berikut :

### 1. In-House Usability Testing

Usability testing ini dilakukan secara in-house. Artinya, subjek dan developer harus berada di lokasi yang sama. Namun, meski dilakukan bersama-sama, sebaiknya developer memilih subjek dari kalangan luar perusahaan. Hal ini dilakukan agar hasil tes objektif atau tidak bias terhadap perusahaan.

Contoh teknik yang digunakan dalam in-house usability testing adalah:

A/B testing — tes untuk membandingkan dua desain

Eye tracking — tes untuk melihat arah pandangan subjek ketika mengakses sebuah halaman website

Card sorting — tes untuk menentukan navigasi dan struktur website

### 2. Moderated Remote Usability Testing

Pelaksanaan moderated remote usability testing seperti tes in-house. Bedanya, developer dan subjek tidak berada di tempat yang sama. Namun, tes tetap dilaksanakan dalam waktu bersamaan. Dalam pelaksanaan moderated remote usability testing, developer bisa menggunakan teknik-teknik yang dilakukan secara in-house. Ini termasuk card

sorting dan A/B testing. Namun, terhubung dilakukan secara remote, developer harus menyiapkan tool-nya terlebih dahulu.

Jika menghendaki tes dengan model moderated remote, Anda bisa menggunakan tool online seperti Lookback atau UserLytics.

### 3. Unmoderated Remote Usability Testing

Dalam unmoderated remote usability testing, developer dan subjek tidak berada di tempat yang sama. Masing-masing subjek pun tidak harus menjalani tes di waktu yang bersamaan.

Karena sifatnya yang tidak terikat tempat dan waktu, tes ini bisa jadi pilihan jika target pengguna Anda tersebar di berbagai belahan dunia.

Untuk melaksanakan unmoderated remote usability testing, Anda tentu perlu tool yang memadai juga. Contohnya Gazepoint untuk eye tracking dan Optimal Workshop untuk card sorting.

#### **e. Alat dan Bahan**

Alat:

- Komputer atau sejenisnya
- Aplikasi Microsoft word, Excel, Power Point atau sejenisnya
- Tools Pengujian Perangkat lunak

Bahan:

- Dokumen Usability Testing

#### **f. Prosedur Kerja**

1. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
2. Lakukan pengujian unit pada UI / Antarmuka aplikasi kelompok Anda
3. Lakukan sesuai step yang telah didefinisikan
4. Buat dokumen hasil testing yang telah dilakukan.
5. Presentasikan di depan kelas

#### **g. Hasil dan Pembahasan**

Dokumentasi latihan mahasiswa berupa Dokumen Hasil Testing.

**h. Rubrik Penilaian**

<b>No</b>	<b>INDIKATOR KINERJA</b>	<b>Bobot (%)</b>	<b>Penilaian</b>	<b>Nilai Akhir</b>
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen Hasil Testing	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

## Acara 20

Materi Pembelajaran	: Pengujian Aspek Kebergunaan Kepada Pengguna
Acara Praktikum/Praktik	: Minggu 11 / 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk Alokasi
Alokasi Waktu	: 120 menit

### a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu melakukan Pengujian Aspek Kebergunaan Kepada Pengguna
2. Mahasiswa mampu menyusun dokumen hasil Pengujian Aspek Kebergunaan Kepada Pengguna
3. Mahasiswa mampu Menganalisa acuan-acuan Pengujian Aspek Kebergunaan Kepada Pengguna

### b. Penilaian Bertahap BNSP (Skill Passport)

Nama Skema Sertifikasi :-

### c. Indikator Penilaian

Mahasiswa memahami dan Menganalisa acuan-acuan Pengujian Aspek Kebergunaan Kepada Pengguna .

### d. Dasar Teori

**Merancang Alat Ukur *Usability***  
*Case Study Aplikasi Belajar Sarapan*



Gambar 1.27 Contoh Tampilan Antar Muka

Alat ukur usability disesuaikan dengan komponen usability yaitu efektivitas, efisiensi dan kepuasan pengguna. Efektivitas dan efisiensi diukur menggunakan lembar observasi, sedangkan kepuasan diukur menggunakan post task kuesioner. Selain itu, diperlukan scenario untuk memandu responden dalam menggunakan aplikasi. Skenario adalah kumpulan tugas yang harus dikerjakan responden pada saat menggunakan aplikasi. Skenario disusun untuk semua fungsi aplikasi F01 sampai dengan F04. Berikut adalah contoh skenario untuk fungsi F05 dapat dilihat pada **Error! Reference source not found.** dibawah ini :

<b>Tugas yang harus dikerjakan</b>	<b>Ya/Tidak</b>
Pada menu utama pilih menu Profil Giziku	
Baca dan perhatikan baik-baik Status Gizimu	
Pilih tombol Kembali pada keypad untuk kembali ke menu utama	

Lembar observasi dibuat untuk menilai efektivitas dan efisiensi semua fungsi (F01 sampai dengan F04). Berikut adalah contoh lembar observasi efektivitas dan efisiensi untuk fungsi F03 .

<b>Elemen observasi</b>	<b>Ya/Tidak</b>
Siswa membuka menu Rekomendasi sarapan dengan mudah	
Siswa membaca tulisan yang digunakan dengan mudah	
Siswa berpindah ke menu utama dengan mudah	

Berikut adalah contoh lembar observasi efektivitas dan efisiensi untuk fungsi rekomendasi.

<b>Elemen evaluasi</b>	<b>Ya/Tidak</b>
Siswa memahami informasi mengenai Rekomendasi Sarapan (Observator menanyakan siswa mengenai informasi Rekomendasi Sarapan)	
Frekuensi siswa bertanya kepada observator rendah	
Frekuensi panduan dan bantuan dari observator rendah	
Error dan kesalahan yang dilakukan siswa hanya sedikit	

Kuissoner menggunakan Bahasa yang mudah dimengerti oleh responden. Missal

- Sangat setuju
- Setuju
- Baik
- Cukup
- Kurang

Atau.:





1



2



3



4



5

**Analisis Usability dapat didefinisikan sebagai berikut :**

Tingkat efektifitas dan efisiensi diukur dengan menggunakan user's succes rate yaitu persentase tugas yang diselesaikan dengan benar oleh pengguna, dan dapat dihitung sebagai berikut :

$$\text{Efektivitas, Efisiensi (\%)} = \frac{(\sum_{i=1}^n X_i)}{n} \times 100\%$$

dengan  $X_i$  adalah nilai keberhasilan responden ke-i,  $X_i = \{0,1\}$

Kepuasan adalah persentase perbandingan antara nilai kepuasan responden ke-i ( $x_i$ ) dengan perkalian antara bobot maksimum skala likert dengan jumlah responden ( $n$ )

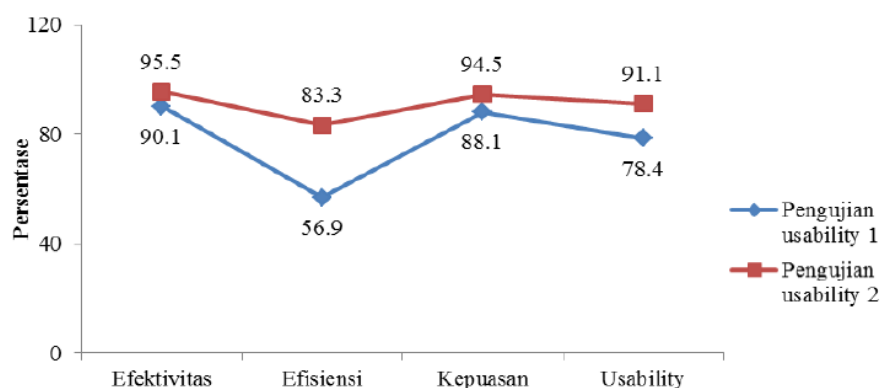
$$\text{Kepuasan (\%)} = \frac{\sum_{i=1}^n X_i}{5 \times n} \times 100\%$$

dengan  $X_i$  adalah nilai keberhasilan responden ke-i,  $X_i = \{0,1,2,3,4,5\}$

usability aplikasi adalah rataaan dari efektivitas, efisiensi dan kepuasan. seperti tertulis pada persamaan berikut :

$$\text{Usability (\%)} = \frac{(\text{Efektivitas} + \text{Efisiensi} + \text{Kepuasan})}{3} \times 100\%$$

Sehingga didapatkan grafik sebagai berikut :



Contoh implementasi dapat dilihat pada jurnal dibawah ini :

<https://media.neliti.com/media/publications/245243-pengujian-usability-untuk-meningkatkan-a-5b871c0b.pdf>

## e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Formulir scenario uji dan hasil uji
3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen Usability Testing

**f. Prosedur Kerja**

1. Buka kembali hasil pembuatan database dan tabel serta atribut/field terkait
2. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
3. Lakukan pengujian unit pada UI / Antarmuka aplikasi kelompok Anda
4. Lakukan Perbandingan Usability
5. Sertakan hasil responden
6. Buat dokumen hasil testing yang telah dilakukan.
7. Presentasikan di depan kelas

**g. Hasil dan Pembahasan**

Dokumentasi tugas mahasiswa berupa pembuatan Dokumen Testing.

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen Hasil Testing	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 21**

Materi Pembelajaran	: Pengujian Aspek Kebergunaan Kepada Pengguna
Acara Praktikum/Praktik	: Minggu 12 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk Alokasi
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

Mahasiswa dapat melakukan pengujian aspek penerimaan kepada pengguna.

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi : -

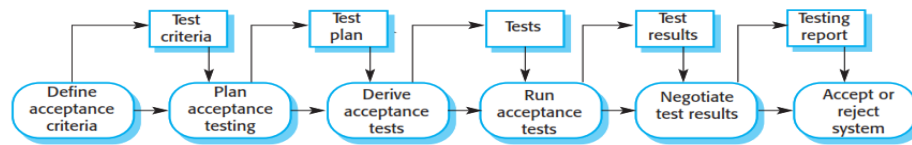
**c. Indikator Penilaian**

Ketepatan teknik dan laporan hasil pengujian aspek kebergunaan.

**d. Dasar Teori**

User testing atau customer testing merupakan tahapan dalam proses pengujian dimana pengguna atau pelanggan memberikan masukan dan saran atas pengujian sistem. Pengguna bereksperimen dengan produk perangkat lunak baru untuk melihat apakah mereka menyukainya dan untuk memeriksa apakah produk tersebut memenuhi kebutuhan atau persyaratan atau kebutuhan fungsi yang mereka mereka inginkan. User testing sangat penting, bahkan ketika pengujian sistem dan rilis yang komprehensif telah dilakukan. Pengaruh dari lingkungan kerja pengguna dapat berdampak besar pada keandalan, kinerja, kegunaan, dan ketahanan sistem.

Secara praktis tidak mungkin bagi pengembang perangkat lunak untuk mereplikasi lingkungan kerja sistem, karena pengujian di lingkungan pengembang pasti hanya buatan. Misalnya, sistem yang dimaksudkan untuk digunakan di rumah sakit digunakan dalam lingkungan klinis rumah sakit di mana hal-hal lain terjadi, seperti keadaan darurat pasien dan percakapan dengan kerabat pasien. Ini semua memengaruhi penggunaan sistem, tetapi pengembang tidak dapat memasukkannya ke dalam lingkungan pengujian yang pengembang lakukan. Intinya User Testing ini pengujian sebenarnya untuk melihat apakah sistem tersebut sesuai dengan kebutuhan fungsional pengguna.



Gambar 1. User Acceptance Testing

Acceptance testing adalah bagian inheren dari pengembangan sistem kustom. Customer menguji sistem, menggunakan data mereka sendiri, dan memutuskan apakah itu harus diterima dari pengembang sistem. Penerimaan menyiratkan bahwa pembayaran akhir harus dilakukan untuk perangkat lunak.

### 1. Define acceptance criteria

Tahap ini idealnya dilakukan di awal proses sebelum kontrak sistem ditandatangani. Kriteria penerimaan harus menjadi bagian dari kontrak sistem dan disetujui oleh Customer dan Pengembang perangkat lunak. Tapi dalam praktiknya, sulit untuk menentukan kriteria di awal proses. Persyaratan terperinci mungkin tidak tersedia, dan persyaratan tersebut hampir pasti akan berubah selama proses pengembangan.

### 2. Plan acceptance testing

Tahap ini melibatkan penentuan sumber daya, waktu, dan anggaran untuk pengujian penerimaan dan penetapan jadwal pengujian. Rencana acceptance testing juga harus membahas cakupan persyaratan yang diperlukan dan urutan fitur sistem yang diuji. Hal ini harus menentukan risiko pada proses pengujian seperti sistem crash dan kinerja sistem yang tidak memadai, dan membahas bagaimana risiko ini dapat dikurangi.

### 3. Derive acceptance tests

Setelah kriteria penerimaan ditetapkan, tes harus dirancang untuk memeriksa apakah suatu sistem dapat diterima atau tidak. Acceptance tests harus bertujuan untuk menguji karakteristik fungsional dan non-fungsional (misalnya, kinerja) dari sistem. Idealnya pengembang sistem harus menyediakan perlindungan lengkap dari persyaratan sistem. Tapi dalam praktiknya, sulit untuk menetapkan kriteria penerimaan yang sepenuhnya obyektif. Seringkali ada ruang untuk argumen tentang apakah suatu tes menunjukkan bahwa suatu kriteria telah terpenuhi atau tidak.

### 4. Run acceptance tests

Acceptance tests yang disepakati dijalankan pada sistem. Idealnya, langkah ini harus dilakukan di lingkungan aktual di mana sistem akan digunakan, tetapi ini mungkin mengganggu dan tidak praktis. Oleh karena itu, lingkungan pengujian pengguna mungkin harus disiapkan untuk menjalankan pengujian ini. Sulit untuk mengotomatiskan proses ini karena bagian dari acceptance tests mungkin

melibatkan pengujian interaksi antara pengguna akhir dan sistem. Beberapa pelatihan pengguna akhir mungkin diperlukan.

5. Negotiate test results

Sangat kecil kemungkinannya bahwa semua acceptance tests yang ditentukan akan lulus dan tidak akan ada masalah dengan sistem. Jika ini masalahnya, maka acceptance tests selesai dan sistem dapat diserahkan. Lebih umum, beberapa masalah akan ditemukan. Dalam kasus seperti itu, pengembang dan customer harus bernegosiasi untuk memutuskan apakah sistem tersebut cukup baik untuk digunakan. Mereka juga harus menyetujui bagaimana pengembang akan memperbaiki masalah yang teridentifikasi.

6. Reject/accept the system

Tahap ini melibatkan pertemuan antara pengembang dan pelanggan untuk memutuskan apakah sistem harus diterima atau tidak. Jika sistem tidak cukup baik untuk digunakan, maka pengembangan lebih lanjut diperlukan untuk memperbaiki masalah yang teridentifikasi. Setelah selesai, tahap pengujian penerimaan diulangi.

**e. Alat dan Bahan**

Alat :

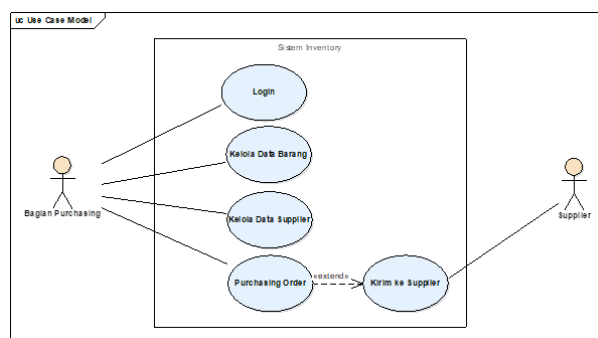
1. Komputer atau sejenisnya
2. Formulir scenario uji dan hasil uji
3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen User Acceptance Testing

**f. Prosedur Kerja**

User Acceptance Test merupakan pengujian yang tidak melihat source code tapi hanya melihat persyaratan yang sudah ditentukan. Prinsipnya sama seperti Black Box Testing, tapi model pengujiannya menguji fungsional yang sudah dipersyaratkan. Pengujian UAT ini mengambil pengujiannya adalah dari Use Case Diagram. Jika use case diagram tersebut memiliki Generalisasi yang diujikan adalah pada use case child nya.



Gambar 2. Use Case Diagram Sistem Inventory Sederhana

## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

Maka dokumen pengujiannya seperti dibawah ini.

Dokumen User Acceptance Testing					
Nama Proyek	:	Rancang Bangun Sistem Informasi Inventory Menggunakan Metode FIFO Berbasis Web	Logo		
Studi Kasus	:	PT. Angin Sepoy-Sepoy			
Manajer Proyek	:	Davina Y. Putri			

Proses Pengujian						
No	Use Case		Hasil Uji [Berhasil   Gagal]	Nama Penguji *	Tanggal Pengujian	Catatan Penguji
1	Usecase Uji	: Login	Berhasil	Bambang	15 Mei 2020	
	Deskripsi	: Melakukan verifikasi terhadap pengguna yang terdaftar dalam sistem				
	Kasus Pengujian		Berhasil	Yulianingsih	20 Mei 2020	
	Username	: purchasing				
	Password	: 123				
Chapcha	: input sesuai dengan hasil generate-code chapcha					
	Hasil yang diharapkan:					
	- Jika login berhasil akan masuk kedalam halaman purchasing					
	- Jika login tidak berhasil tidak akan masuk kedalam halaman purchasing, dan menampilkan pesan kesalahan melalui display.					
	- Chapcha akan terverifikasi berhasil sesuai dengan kode chapcha yang ditampilkan					

### g. Hasil dan Pembahasan

1. Buatlah sebuah dokumen UAT untuk aplikasi website dan mobile yang sudah dikerjakan!

### h. Rubrik Penilaian

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen UAT	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 22**

Materi Pembelajaran : Pengujian Aspek Kebergunaan Kepada Pengguna  
Acara Praktikum/Praktik : Minggu 12 / 2  
Tempat : Laboratorium PSDKU Kampus 3 Kab. Nganjuk Alokasi  
Alokasi Waktu : 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

Mahasiswa dapat melakukan pengujian aspek penerimaan kepada pengguna.

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

Ketepatan teknik dan laporan hasil pengujian aspek kebergunaan.

**d. Dasar Teori**

UAT (User Acceptance Test) adalah suatu proses pengujian yang dilakukan oleh pengguna dengan hasil output sebuah dokumen hasil uji yang dapat dijadikan bukti bahwa software sudah diterima dan sudah memenuhi kebutuhan yang diminta. UAT tidak jauh beda dengan kusioner pada tahap awal pembuatan aplikasi.

**e. Alat dan Bahan**

Alat :

1. Komputer atau sejenisnya
2. Formulir scenario uji dan hasil uji (GForm)
3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen User Acceptance Testing

**f. Prosedur Kerja**

Untuk mengetahui tanggapan responden (user) terhadap aplikasi pembelajaran yang akan diimplementasikan, maka dilakukan pengujian dengan memberikan 4 pertanyaan kepada 36 responden (siswa) dimana jawaban dari pertanyaan tersebut terdiri dari tingkatan yang dapat dipilih sebagai berikut:



## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

**Tabel Pilihan Jawaban UAT**

A	Sangat : Mudah/Bagus/Sesuai/Jelas
B	Mudah/Bagus/Sesuai/Jelas
C	Netral
D	Cukup : Sulit/Bagus/Sesuai/Jelas
E	Sangat : Sulit/Jelek/Tidak Sesuai/Tidak Jelas

**Tabel Bobot Nilai Jawaban**

Jawaban	Bobot
A. Sangat : Mudah/Bagus/Sesuai/Jelas	5
B. Mudah/Bagus/Sesuai/Jelas	4
C. Netral	3
D. Cukup : Sulit/Bagus/Sesuai/Jelas	2
E. Sangat : Sulit/Jelek/Tidak Sesuai/Tidak Jelas	1

Template kuisisioner UAT adalah sebagai berikut :

**Tabel Pertanyaan Kuesioner**

No	Pertanyaan	A	B	C	D	E
1	Apakah tampilan media pembelajaran <i>web</i> ini menarik ?					
2	Apakah menu-menu media pembelajaran <i>web</i> ini mudah dipahami ?					
3	Apakah materi media pembelajaran <i>web</i> ini mudah dipahami ?					
4	Apakah adanya contoh-contoh membantu memahami materi biologi ?					
5	Apakah evaluasi pada media pembelajaran <i>web</i> ini sudah sesuai dengan silabus ?					
6	Apakah evaluasi membantu mengukur pemahaman materi ?					
7	Apakah media pembelajaran <i>web</i> ini dapat dijadikan media bantu belajar ?					
8	Apakah media pembelajaran <i>web</i> ini sudah cukup baik?					

Data yang didapat di atas diolah dengan cara mengalikan setiap poin jawaban dengan bobot yang sudah ditentukan sesuai dengan tabel bobot nilai jawaban. Dari hasil perhitungan dengan mengalikan setiap jawaban bobot yang sudah ditentukan maka didapat hasil sebagai berikut Tabel Data Kuesioner Siswa Setelah Diolah

Contoh hasil data yang didapat :

Tabel Data Jawaban Kuesioner Siswa

No	Pertanyaan	Jawaban					Presentase				
		A	B	C	D	E	A	B	C	D	E
1	Apakah tampilan media pembelajaran <i>web</i> ini menarik ?	12	11	8	3	2	33%	31%	22%	8%	6%
2	Apakah menu-menu media pembelajaran <i>web</i> ini mudah dipahami ?	5	15	12	3	1	14%	42%	33%	8%	3%
3	Apakah materi media pembelajaran <i>web</i> ini mudah dipahami ?	7	13	10	6	0	19%	36%	28%	17%	0%
4	Apakah adanya contoh-contoh membantu memahami materi biologi ?	8	16	8	4	0	22%	45%	22%	11%	0%
5	Apakah evaluasi pada media pembelajaran <i>web</i> ini sudah sesuai dengan silabus ?	15	12	6	3	0	42%	33%	17%	8%	0
6	Apakah evaluasi membantu mengukur pemahaman materi ?	13	14	5	4	0	36%	39%	14%	11%	0
7	Apakah media pembelajaran <i>web</i> ini dapat dijadikan media bantu belajar ?	14	12	8	2	0	39%	33%	22%	6%	0%
8	Apakah media pembelajaran <i>web</i> ini sudah cukup baik?	17	12	5	2	0	47%	33%	14%	6%	0%

## g. Hasil dan Pembahasan

Contoh hasil pengolahan data:

No	Pertanyaan	Nilai					Jml
		Ax5	Bx 4	Cx 3	Dx 2	Ex 1	
1	Apakah tampilan media pembelajaran <i>web</i> ini menarik ?	60	44	24	6	2	136
2	Apakah menu-menu media pembelajaran <i>web</i> ini mudah dipahami ?	25	60	36	6	1	128
3	Apakah materi media pembelajaran <i>web</i> ini mudah dipahami ?	35	52	30	12	0	129
4	Apakah adanya contoh-contoh membantu memahami materi biologi ?	40	64	24	8	0	137
5	Apakah evaluasi pada media pembelajaran <i>web</i> ini sudah sesuai dengan silabus ?	75	48	18	6	0	147
6	Apakah evaluasi membantu mengukur pemahaman materi ?	65	56	15	8	0	144
7	Apakah media pembelajaran <i>web</i> ini dapat dijadikan media bantu belajar ?	75	48	24	4	0	151
8	Apakah media pembelajaran <i>web</i> ini sudah cukup baik?	85	48	15	4	0	152

a. Analisa pertanyaan pertama

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 36 responden untuk pertanyaan pertama adalah 136. Nilai rata-ratanya adalah  $136/36 = 3.7$ . Prosentase nilainya adalah  $3.7/5 \times 100 = 74\%$ .

b. Analisa pertanyaan kedua

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 36 responden untuk pertanyaan kedua adalah 128. Nilai rata-ratanya adalah  $128/36 = 3.5$ . Prosentase nilainya adalah  $3.5/5 \times 100 = 70\%$ .

c. Analisa pertanyaan ketiga

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 36 responden untuk pertanyaan ketiga adalah 129. Nilai rata-ratanya adalah  $129/36 = 3.6$ . Prosentase nilainya adalah  $3.6/5 \times 100 = 72\%$ .

d. Analisa pertanyaan keempat

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 36 responden untuk pertanyaan keempat adalah 137. Nilai rata-ratanya adalah  $137/36 = 3.8$ . Prosentase nilainya adalah  $3.8/5 \times 100 = 76\%$ .

e. Analisa pertanyaan kelima

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 36 responden untuk pertanyaan keempat adalah 147. Nilai rata-ratanya adalah  $147/36 = 4.1$ . Prosentase nilainya adalah  $4.1/5 \times 100 = 82\%$ .

f. Analisa pertanyaan keenam.

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 36 responden untuk pertanyaan keempat adalah 144. Nilai rata-ratanya adalah  $144/36 = 4$ . Prosentase nilainya adalah  $4/5 \times 100 = 80\%$ .

g. Analisa pertanyaan ketujuh.

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 36 responden untuk pertanyaan keempat adalah 151. Nilai rata-ratanya adalah  $151/36 = 4.2$ . Prosentase nilainya adalah  $4.2/5 \times 100 = 84\%$ .

h. Analisa pertanyaan kedelapan

Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 36 responden untuk pertanyaan keempat adalah 137. Nilai rata-ratanya adalah  $152/36 = 4.2$ . Prosentase nilainya adalah  $4.2/5 \times 100 = 84\%$ .

Dari data diatas dapat disimpulkan bahwa media pembelajaran biologi kelas VIII tersebut mempunyai tampilan yang menarik, menu-menu pada media web mudah dipahami, isi atau materi mudah dimengerti dan dipahami, adanya contoh-contoh juga membantu dalam memahami materi biologi, evaluasi yang tersedia sudah sesuai dengan silabus serta evaluasi membantu mengukur pemahaman terhadap materi, media pembelajaran web ini juga dapat dijadikan media bantu belajar dan media pembelajaran web ini sudah cukup baik.

#### **h. Rubrik Penilaian**

<b>No</b>	<b>INDIKATOR KINERJA</b>	<b>Bobot (%)</b>	<b>Penilaian</b>	<b>Nilai Akhir</b>
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen UAT	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 23**

Materi Pembelajaran	: Pengujian Aspek Kebergunaan Kepada Pengguna
Acara Praktikum/Praktik	: Minggu 13 / 1
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk Alokasi
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu menentukan aspek keamanan perangkat lunak yang akan diuji
2. Mahasiswa mampu melaksanakan pengujian keamanan perangkat lunak
3. Mahasiswa mampu menilai kebijakan mengenai keamanan yang digunakan

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

Keberhasilan dalam menggunakan tools dalam pelaksanaan pengujian keamanan perangkat lunak, ketepatan dalam menyusun dokumen hasil pengujian perangkat lunak.

**d. Dasar Teori****1. Pengujian Keamanan Perangkat Lunak**

Security testing adalah jenis pengujian perangkat lunak yang dilakukan untuk mengidentifikasi kerentanan serta memastikan bahwa data dan sumber daya sistem di dalamnya sudah terlindungi dengan baik dari para penyusup. Pengujian ini dilakukan dengan tujuan untuk menemukan semua celah dan kelemahan sistem yang dapat mengakibatkan hilangnya informasi atau reputasi perusahaan.

Pada umumnya, security testing akan dilakukan setiap kali Anda melakukan perubahan pada sistem yang Anda kembangkan. Meskipun demikian, perusahaan-perusahaan perlu menyadari bahwa security testing perlu dilakukan secara berkala karena peretas akan selalu mencari cara untuk menyusup ke dalam sistem. Dengan rutin melakukan security testing, Anda dapat menjamin bahwa sistem yang digunakan selalu memiliki tingkat keamanan yang baik.

Dalam security testing khususnya yang dilakukan pada situs website dan aplikasi, akan ada empat area utama yang akan diperhatikan. Keempat area tersebut adalah:

- Network security: Pengujian ini dilakukan untuk mencari kerentanan dalam infrastruktur jaringan.

- System software security: Pengujian untuk menilai bagaimana tingkat kelemahan berbagai perangkat lunak tempat aplikasi bekerja seperti operating system, database system, dan lain-lain.
  - Client-side application security: pengujian ini dapat memastikan bahwa sisi klien seperti browser tidak dapat dimanipulasi.
  - Server-side application security: pengujian untuk memastikan bahwa sisi server memiliki keamanan yang kuat dan dapat memblokir beragam gangguan.
2. Jenis-jenis pengujian keamanan perangkat lunak

a) Vulnerability scanning

Vulnerability Scanning (vuln scan) adalah pengujian keamanan yang dilakukan melalui software otomatis untuk memindai aplikasi web. Software ini akan mencari kerentanan keamanan yang ada di dalam sistem seperti pembuatan Cross site scripting, SQL Injection, Command Injection, Path Traversal, serta konfigurasi server yang tidak aman. Tool ini sering disebut sebagai bagian dari Dynamic Application Security Testing (DAST).

Secara garis besar, Vulnerability Scanning memiliki beberapa jenis pemindaian, yaitu:

- External vulnerability scans:

External vulnerability scans merupakan penilaian kerentanan yang dilakukan dengan menargetkan ekosistem IT yang tidak dibatasi untuk penggunaan internal. Pemindaian ini akan fokus pada beberapa area seperti applications, ports, websites, services, networks, dan sistem yang dapat diakses dari luar oleh customer atau user.

- Internal vulnerability scans:

Internal vulnerability scans adalah pemindaian yang memiliki target utama jaringan internal perusahaan. Pemindaian ini akan mengidentifikasi kerentanan di dalam jaringan untuk menghindari kerusakan. Pemindaian internal juga memungkinkan perusahaan atau organisasi untuk bisa melindungi dan memperkuat sistem keamanan aplikasi dari dalam.

b) Security scanning

Security scanning merupakan pemindaian yang digunakan untuk menemukan kerentanan atau modifikasi file yang tidak diinginkan dalam aplikasi berbasis web, situs web, jaringan, atau sistem file. Pemindaian ini dapat dilakukan secara otomatis ataupun manual. Pemindaian ini akan memberikan Anda insight yang mendalam serta menyediakan rekomendasi solusi untuk memperbaiki masalah yang ditemukan.



Security scanning dapat dilakukan sebagai one-time check atau pemeriksaan satu kali. Meskipun demikian, sebagian besar perusahaan pengembang perangkat lunak lebih memilih untuk melakukan pemindaian keamanan secara teratur untuk memastikan sistem yang dikembangkan benar-benar aman.

c) Penetration testing

Penetration testing adalah proses pengujian dengan melakukan simulasi serangan cyber terhadap sistem yang akan diuji. Pengujian ini akan dilakukan secara manual oleh pentester profesional dan bersertifikat menggunakan beragam pentest tools dan teknik. Ketika pengujian penetration testing dilakukan, Anda akan menemukan beragam kerentanan yang dapat dieksploitasi oleh para penjahat cyber. Dengan demikian, maka proses penambalan atau perbaikan dapat segera dilakukan sebelum pihak peretas menemukannya.

d) Risk assessment

Melalui risk assessment, risiko keamanan yang dihadapi oleh aplikasi, software, dan jaringan akan diidentifikasi dan dianalisis. Risiko keamanan tersebut kemudian akan diklasifikasikan ke dalam beberapa kategori yaitu tinggi, sedang, dan rendah. Salah satu jenis security testing ini dapat membantu Anda memastikan bahwa kontrol keamanan cyber yang Anda lakukan sebelumnya sudah sesuai dengan risiko keamanan yang dihadapi organisasi/perusahaan Anda.

e) Security auditing

Security Auditing adalah metode terstruktur untuk mengevaluasi langkah-langkah keamanan di dalam perusahaan. Dengan melakukan audit secara rutin, Anda akan terbantu dalam mengidentifikasi titik lemah dan kerentanan dalam infrastruktur IT perusahaan, memverifikasi kontrol keamanan, memastikan kepatuhan terhadap peraturan keamanan, dan masih banyak lagi.

f) Ethical hacking

Ethical hacking merupakan pengujian keamanan yang dilakukan menggunakan semua teknik peretasan dan teknik serangan komputer terkait lainnya. Proses pengujian ini dilakukan oleh ethical hacker yang sudah memperoleh izin untuk menjelajahi infrastruktur IT perusahaan secara lebih luas.

g) Posture assessment

Posture Assessment mengacu pada metodologi yang dilakukan untuk meningkatkan kemampuan manajemen risiko di perusahaan. Sebagian besar orang menganggap bahwa posture Assessment merupakan gabungan dari beberapa jenis security testing yaitu Ethical Hacking, Security Scanning, dan Risk Assessment.



Penilaian ini dapat menjadi langkah berharga untuk mengetahui bagaimana kondisi keamanan perusahaan. Dengan melakukan Posture Assessment, sebuah Anda akan memiliki pandangan yang jelas tentang status keamanan perusahaan serta mampu mengidentifikasi ancaman keamanan yang mungkin terjadi.

### 3. Metode atau Pendekatan Pengujian Keamanan Perangkat Lunak

#### a) Tiger box

Metodologi pengujian ini pada umumnya dilakukan melalui laptop atau OS yang memiliki seperangkat alat untuk hacking. Dengan metodologi ini, penguji dapat mengevaluasi kerentanan serangan cyber.

#### b) Black box

Black box testing adalah metode pengujian perangkat lunak dimana struktur atau desain internal dari item yang perlu diuji tidak diketahui oleh penguji. Pengujian black box juga sering disebut sebagai functional testing atau closed-box testing.

#### c) Grey box

Grey Box testing dilakukan pada suatu perangkat lunak dengan informasi fungsionalitas internal yang terbatas. Biasanya, pihak penguji akan memiliki beberapa pengetahuan dasar tentang kode dan algoritma perangkat lunak. Meskipun demikian informasi tersebut tetap terbatas sehingga pengujian akan tetap dilakukan secara normal dari sudut pandang penguji.

### 4. Tools pengujian keamanan perangkat lunak

#### a) Intruder

#### b) Owasp

#### c) Acunetix

#### d) Wireshark

#### e) W3af

## e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Formulir scenario uji dan hasil uji (GForm)
3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Security testing process
2. Test plan
3. Test scenario

## f. Prosedur Kerja

1. Diskusikan dengan kelompok Anda untuk membuat dokumen test plan untuk security testing mulai dari requirements hingga fase support.
2. Sesuaikan prosesnya dengan pengembangan aplikasi web atau mobile yang kelompok Anda kembangkan (dapat memilih salah satu web atau mobile).
3. Buatlah test scenario untuk pengujian keamanan perangkat lunak pada fitur login/registrasi pada aplikasi Anda.
4. Lakukan pengujian keamanan aplikasi Anda dengan pendekatan Black Box dan dokumentasikan hasilnya.

**g. Hasil dan Pembahasan**

Dokumen test plan, test scenario dan hasil testing pengujian keamanan perangkat lunak yang dikembangkan.

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen security Testing	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 24**

Materi Pembelajaran	: Pengujian Aspek Kebergunaan Kepada Pengguna
Acara Praktikum/Praktik	: Minggu 13 / 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk Alokasi
Alokasi Waktu	: 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu menentukan aspek keamanan perangkat lunak yang akan diuji
2. Mahasiswa mampu melaksanakan pengujian keamanan perangkat lunak
3. Mahasiswa mampu menilai kebijakan mengenai keamanan yang digunakan

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

Keberhasilan dalam menggunakan tools dalam pelaksanaan pengujian keamanan perangkat lunak, ketepatan dalam menyusun dokumen hasil pengujian perangkat lunak.

**d. Dasar Teori**

Pada bagian ini akan dijelaskan konfigurasi lab yang digunakan dalam melakukan pengujian. Lab pengujian dapat dilakukan langsung dari perangkat mobile atau bisa juga menggunakan emulator. Berikut detail kedua konfigurasi lab tersebut:

**1. Langsung dari Device**

Berikut ini adalah daftar kebutuhan untuk konfigurasi lab pengujian langsung dari device:

- Mobile device (MD)
- Kabel data
- Access point (AP)
- Hub
- Kabel LAN
- Komputer (PC)
- Untuk akses langsung ke device dari komputer digunakan kabel data.

PC → kabel data → MD

Access point, Hub dan kabel LAN diperlukan untuk pengujian pada sisi keamanan jaringan.

Berikut ini adalah diagram skema topologi pengujian keamanan jaringan aplikasi.

Internet → Hub → AP → MD → Komputer

### 2. Menggunakan Emulator

Apabila tidak memiliki mobile device bukan berarti tidak bisa melakukan pengujian. Pengujian aplikasi mobile dapat memanfaatkan emulator yang ada. Platform mobile pada paper ini dibatasi pada Android. Pada paper ini tidak akan dijelaskan bagaimana menyiapkan: instalasi & konfigurasi SDK Android. Untuk petunjuk bagaimana menyiapkan SDK Android dapat dilihat pada situs pengembang Android.

Usai SDK Android siap, dapat dipilih versi Android mana yang emulatoarnya ingin disiapkan. Silakan pilih sesuai kebutuhan. Emulator pada Android disebut dengan Android Virtual Device (AVD). Versi terakhir emulator Android terbaru saat paper ini ditulis adalah Android 4.0.3 dengan API Level 15.



Gambar2 emulator android

### 3. Aplikasi Wireshark

Wireshark adalah aplikasi open source yang menangkap dan menampilkan data yang berjalan bolak-balik di jaringan. Wireshark ini biasanya digunakan untuk memecahkan masalah jaringan dan menguji perangkat lunak karena memberikan kemampuan untuk menelusuri dan membaca konten setiap paket.

Tutorial cara menggunakan wireshark dalam artikel ini berlaku untuk software Wireshark 3.0.3 untuk Windows dan Mac.

Wireshark adalah penganalisa lalu lintas jaringan terkemuka di dunia, dan alat penting untuk profesional keamanan atau administrator sistem. Software gratis ini memungkinkan Anda menganalisis lalu lintas jaringan secara real time, dan seringkali merupakan alat terbaik untuk mengatasi masalah pada jaringan Anda. Masalah umum yang dapat dipecahkan oleh Wireshark termasuk paket yang hilang, masalah latensi, dan aktivitas jahat di jaringan Anda. Ini memungkinkan Anda menempatkan lalu lintas jaringan Anda di bawah mikroskop, dan menyediakan alat untuk menyaring dan menelusuri lalu lintas itu, memperbesar penyebab utama masalahnya. Administrator menggunakan software ini untuk mengidentifikasi peralatan jaringan yang rusak yang menjatuhkan paket, masalah latensi yang disebabkan oleh mesin yang merutekan lalu lintas di belahan dunia, dan pengelupasan data atau bahkan upaya peretasan terhadap Anda

#### 4. Case Study Aplikasi Wordpress

Wordpress merupakan aplikasi F/OSS Content Management System (CMS) blog. Selain dikembangkan aplikasi blog engine Wordpress itu sendiri, dikembangkan juga aplikasi Wordpress untuk Android. Aplikasi Wordpress untuk Android ini merupakan aplikasi klien yang digunakan untuk mengakses blog Wordpress. Pada lab pengujian ini, diinstall aplikasi Wordpress pada server, dan aplikasi Wordpress untuk Android pada ponsel Android. Skema jaringan mengikuti Gambar 3 hanya saja tidak perlu tersambung ke Internet karena aplikasi Wordpress terinstall pada server sendiri (self hosted Wordpress blog). Alasan menggunakan self hosted Wordpress blog adalah agar analisis dapat dilakukan lebih mendalam dan detail sekaligus lebih sederhana karena tidak memerlukan akses Internet.

Berikut detail versi aplikasi yang digunakan:

- Wordpress: 3.3.1
- Wordpress untuk Android: 2.0.7

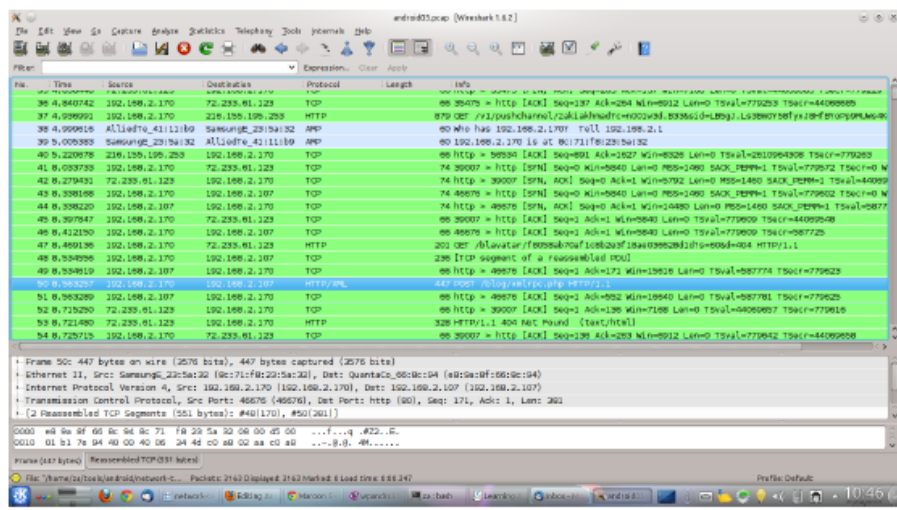
Pertama akan dilakukan analisis network traffic. Tujuan dari aktivitas ini adalah mencari informasi sensitif yang tidak dienkripsi. Aplikasi Wordpress untuk Android akan digunakan untuk mengakses self-hosted Wordpress blog dan melakukan aktivitas sebagai

berikut:

- Akses sebagai publik (tanpa otentikasi)
- Melakukan otentikasi, masuk sebagai authorized user.
- Menulis tulisan baru
- Akses menu konfigurasi Wordpress
- Mengubah password

## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

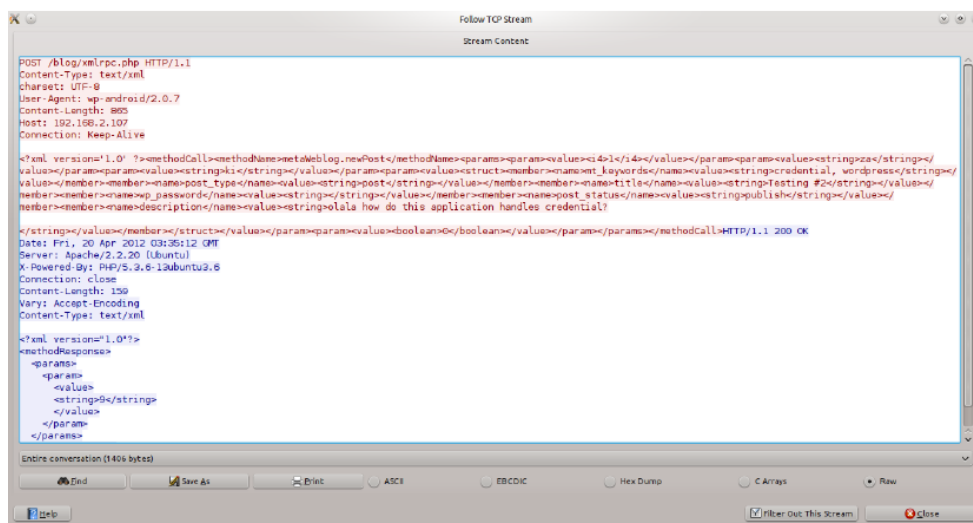
Semua aktivitas ini akan dianalisis dari hasil packet capture-nya. Berikut ini merupakan hasil analisis berkas pcapnya.



Gambar 3 Berkas Packet Capture Wordpress

Berdasarkan gambar 3 diatas , dapat terlihat bahwa IP mobile device yang digunakan adalah 192.168.2.170. Hal ini penting untuk diketahui dalam melakukan analisis lebih dalam dan menyusun formula filter packet pada Wireshark yang digunakan sebagai tool menganalisis packet capture. Lalu kotak merah pada Gambar 4 yang kedua menunjukkan request POST yang dilakukan oleh aplikasi Wordpress untuk Android. Request ini menarik untuk ditelusuri lebih lanjut. Untuk itu dilakukan follow TCP stream pada paket ini.

Lalu kotak merah pada Gambar yang kedua menunjukkan request POST yang dilakukan oleh aplikasi Wordpress untuk Android. Request ini menarik untuk ditelusuri lebih lanjut. Untuk itu dilakukan follow TCP stream pada paket ini.

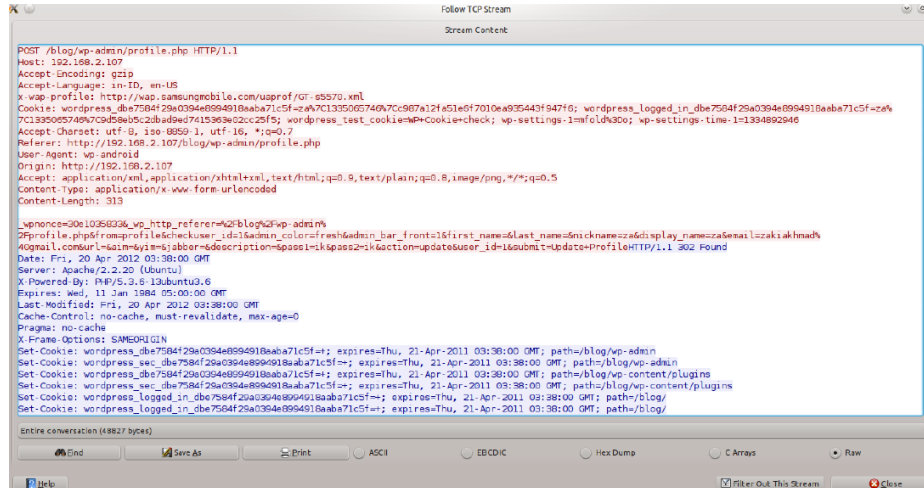


Gambar 4 String Username & Password



## BKPM –WORKSHOP KUALITAS PERANGKAT LUNAK

Dari hasil analisis paket terlihat saat user akan menulis tulisan baru, username dan password dikirimkan dalam bentuk string tanpa melalui kanal terenkripsi. Hasil penelusuran lebih lanjut menunjukkan bahwa username dan password selalu disertakan dalam setiap request aplikasi saat akan mengakses server.



Gambar 5 Ubah Password

Saat password user Wordpress diubah bukan melalui aplikasi Android, maka aplikasi Android tidak lagi bisa mengakses blog. Hal ini dilakukan untuk mengetahui bagaimana aplikasi menyimpan password.



Gambar 6 Login Usai Password Diganti

Ternyata aplikasi menyimpan password pada semacam berkas konfigurasi. Jadi, perlu dilakukan konfigurasi ulang pada aplikasi dengan mengubah nilai password untuk bisa kembali login. Dari keseluruhan pengujian, terlihat bahwa desain aplikasi wordpress untuk Android tidak aman. Username dan password dikirim dalam bentuk string dalam setiap request melalui kanal tak terenkripsi. Cara sederhana meningkatkan keamanan aplikasi Wordpress untuk Android dapat dilakukan dengan menggunakan kanal terenkripsi. Pengguna Android hanya bisa



mengakses self-hosted Wordpress blog menggunakan protokol HTTPS. Selanjutnya perlu dipikirkan desain aplikasi yang lebih baik yang tidak harus selalu mengirimkan username dan password dalam setiap requestnya. Persisnya pada bagian mekanisme otentikasi aplikasi.

### e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Wireshark
3. Aplikasi Microsoft word, Excel, Power Point atau sejenisnya

Bahan :

1. Dokumen Dinamic Mobile Testing

### f. Prosedur Kerja

1. Bentuklah kelompok dengan jumlah anggota 5-6 orang.
2. Lakukan pengujian unit pada fitur login/registrasi aplikasi mobile yang kelompok Anda kembangkan menggunakan aplikasi wireshark.
3. Buat dokumen hasil testing yang telah dilakukan.
4. Presentasikan di depan kelas

### g. Hasil dan Pembahasan

Dokumentasi tugas mahasiswa berupa Dokumen testing.

### h. Rubrik Penilaian

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai contoh implementasi dokumen security Testing	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 25 dan 26**

Materi Pembelajaran	: Pengujian Aspek Kebergunaan Kepada Pengguna
Acara Praktikum/Praktik	: Minggu 14 / 1 dan 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk Alokasi
Alokasi Waktu	: 2 x 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK).**

1. Mahasiswa mampu mendokumentasikan hasil pelaksanaan penjaminan kualitas perangkat lunak
2. Mahasiswa mampu meninjau kembali pelaksanaan penjaminan kualitas perangkat lunak
3. Mahasiswa mampu meninjau kembali pelaksanaan tindakan perbaikan kualitas pengembangan perangkat lunak

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

Ketepatan dalam mendokumentasikan seluruh pelaksanaan penjaminan kualitas perangkat lunak dan meninjau pelaksanaannya

**d. Dasar Teori**

Software Quality Assurance ( SQA ) memiliki peran yang sangat penting untuk setiap proses pengembangan perangkat lunak. SQA dapat membantu meningkatkan produktivitas developer, meningkatkan kualitas produk, meningkatkan kepercayaan klien terhadap layanan perusahaan, menghemat biaya dan waktu pengembangan, dan masih banyak lagi.

Berdasarkan standar internasional ISO 9126, software dapat dinilai kualitasnya berdasarkan beberapa karakteristik, yaitu:

**1. Functionality (fungsionalitas)**

Fungsionalitas merupakan faktor penting untuk setiap produk. QA harus memastikan bahwa produk yang dikembangkan menyediakan fungsi yang diterapkan dengan benar serta sesuai dengan kebutuhan user.

**2. Reliability (Keandalan)**

Reliability didefinisikan sebagai kemampuan suatu sistem untuk terus berfungsi ketika digunakan dalam periode waktu tertentu. Faktor keandalan juga dapat dilihat dari performance produk ketika menghadapi suatu kondisi. Sebagai

contoh, ketika koneksi mati selama 20 detik dan menyala kembali, produk harus bisa recover dan bekerja kembali secara otomatis.

### 3. Usability (Kebergunaan)

Aplikasi harus dapat digunakan dengan mudah oleh pengguna. Agar produk mudah untuk dioperasi dan dinavigasi, maka penting bagi pihak pengembang untuk mengetahui siapa yang akan menjadi pengguna dari produk tersebut.

### 4. Efficiency (Efisiensi)

Karakteristik ini berkaitan dengan sumber daya sistem yang digunakan ketika software dijalankan. Sumber daya seperti jumlah ruang disk, memori, kapasitas prosesor, dan lain-lain harus dapat digunakan secara efisien oleh software yang dikembangkan.

### 5. Maintainability (Maintabilitas)

Software harus memiliki karakteristik maintainability yaitu kemampuan untuk bisa dimodifikasi. Maintainability juga dapat didefinisikan sebagai kemampuan software untuk mengidentifikasi dan memperbaiki kesalahan.

### 6. Portability (Portabilitas)

Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain. Software harus dapat beradaptasi dengan baik ketika digunakan pada program komputer yang berbeda seperti DOS, windows, dan lain-lain

## e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Power Point atau sejenisnya

Bahan :

1. Dokumen pelaksanaan kegiatan penjamin kualitas perangkat lunak

## f. Prosedur Kerja

1. Buatlah sebuah laporan pelaksanaan kegiatan penjaminan kualitas perangkat lunak dengan kelompok Anda.
2. Dokumen laporan berisi:
  - Halaman judul
  - Kata pengantar
  - Daftar isi
  - Bab 1 pendahuluan
  - Bab 2 penyusunan test case
  - Bab 3 hasil software test & dokumen rekomendasi

- Bab 4 kesimpulan dan saran
- Daftar pustaka

**g. Hasil dan Pembahasan**

Laporan kegiatan penjaminan kualitas perangkat lunak

**h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai dengan hasil Laporan kegiatan penjaminan kualitas perangkat lunak	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
	<b>Total</b>	<b>100</b>		

**Acara 27 dan 28**

Materi Pembelajaran	: Laporan dan Dokumentasi Pengujian
Acara Praktikum/Praktik	: Minggu 15 / 1 dan 2
Tempat	: Laboratorium PSDKU Kampus 3 Kab. Nganjuk Alokasi
Alokasi Waktu	: 2 x 120 menit

**a. Capaian Pembelajaran Mata Kuliah (CPMK).**

Mahasiswa mampu menyusun laporan dan dokumentasi pengujian aplikasi

**b. Penilaian Bertahap BNSP (Skill Passport)**

Nama Skema Sertifikasi :-

**c. Indikator Penilaian**

Ketepatan dalam mendokumentasikan seluruh pelaksanaan penjaminan kualitas perangkat lunak.

**d. Dasar Teori**

Panduan pengguna adalah panduan dalam format kertas atau elektronik (PDF atau XPS) yang memberikan instruksi tentang prosedur atau penggunaan sesuatu. Meskipun “panduan pengguna” sering kali diasosiasikan dengan panduan aplikasi komputer, panduan pengguna juga diberikan bersama komputer dan perangkat elektronik lainnya seperti televisi, stereo, sistem telepon, pemutar MP3, serta peralatan rumah tangga dan perlengkapan kebun dan halaman. Panduan pengguna yang baik mendidik pengguna tentang fitur produk sambil mengajarkan mereka bagaimana menggunakan fitur tersebut secara efektif. Panduan pengguna disusun sedemikian rupa sehingga mudah dibaca dan dirujuk. Berikut beberapa langkah yang perlu Anda pertimbangkan saat mengembangkan konten yang efektif dan merancang tata letak panduan pengguna.

**1. Tentukan siapa yang akan menggunakan panduan.** Untuk menulis panduan pengguna yang baik, Anda perlu mengembangkan profil pengguna, baik secara formal dengan membuat profil tertulis atau secara informal dengan meluangkan waktu untuk membuat asumsi yang masuk akal tentang karakteristik pengguna. Profil seperti itu berguna saat Anda menjadi bagian dari tim penulisan panduan pengguna dan saat mengerjakan produk yang dimaksud sejak masih berupa konsep hingga ke bentuk akhir. Pertimbangkan faktor berikut saat membuat profil pengguna:

- Lokasi pengguna akan menggunakan panduan tersebut, misalnya di rumah, di kantor, di tempat kerja yang terpencil, atau di mobil. Faktor ini tidak hanya menentukan konten, tetapi juga gaya penulisan yang akan diadopsi oleh panduan.
- Bagaimana pengguna akan menggunakan panduan. Jika mereka jarang menggunakannya atau hanya memerlukannya untuk mencari informasi, sebaiknya panduan dibuat dalam bentuk dokumen referensi. Di sisi lain, jika pengguna akan menggunakannya secara rutin di awal, bagian referensi harus dilengkapi dengan bagian "Cara Memulai" dan petunjuk tentang fungsi paling umum yang akan dilakukan produk tersebut.
- Berapa banyak pengalaman yang dimiliki pengguna terhadap produk atau barang sejenisnya. Jika produk terbilang baru atau sangat berbeda dari produk sejenis, Anda harus menyertakan penjelasan tentang bagaimana produk itu berbeda dibanding yang lain dan instruksi tentang cara menggunakannya. Jika produk berhubungan dengan sesuatu yang sering kali membuat pengguna merasa kesulitan, misalnya aplikasi komputer, Anda harus memberikan informasi dan detail yang sesuai dengan cara yang dapat dipahami.

**2 Buatlah panduan yang sesuai dengan kebutuhan pengguna dengan cara yang bisa mereka pahami.** Jika pengguna bukan orang-orang yang memiliki pengetahuan teknis, mungkin sebaiknya Anda menghindari bahasa yang sangat teknis dan memberikan penjelasan dengan gamblang dan sederhana. Teks juga harus diatur dengan cara yang menyerupai cara berpikir pengguna. Membuat daftar fitur produk yang dikelompokkan berdasarkan fungsi sering kali lebih masuk akal daripada berdasarkan fitur yang paling sering digunakan. Terkadang menggunakan istilah teknis tidak dapat dihindari, misalnya untuk aplikasi pembuat grafik yang mencakup grafik Fibonacci bersama dengan grafik lingkaran (*pie chart*) dan grafik batang yang lebih umum. Dalam kasus seperti ini, akan lebih baik jika Anda mendefinisikan istilah dan memberikan informasi, misalnya penjelasan tentang apa itu grafik Fibonacci dan bagaimana penggunaannya dalam analisis keuangan.

**3. Jelaskan masalah yang sedang coba dipecahkan pengguna, kemudian berikan solusi untuk masalah itu.** Menawarkan fitur sebagai solusi untuk masalah umum boleh-boleh saja saat mengiklankan produk, tetapi begitu pelanggan memiliki produk, dia perlu memahami cara menggunakannya.

Identifikasi masalah yang akan dihadapi pengguna, paparkan di dalam panduan pengguna, diikuti dengan instruksi untuk mengatasinya.

Jika masalahnya rumit, pecahlah menjadi bagian-bagian kecil. Tuliskan setiap bagian dengan instruksi bagaimana mengatasinya atau menghadapinya, kemudian diikuti dengan setiap bagian lain secara berurutan. Memecah informasi dengan cara seperti ini dikenal sebagai metode pengelompokan atau pemisahan.

Dokumentasi perangkat lunak yang baik, baik berupa dokumentasi spesifikasi bagi pemrogram dan penguji, dokumen teknis untuk pengguna internal, atau manual dan berkas pertolongan untuk pengguna akhir, akan membantu pengguna memahami fitur dan fungsi perangkat lunak. Dokumentasi yang baik adalah dokumentasi yang spesifik, jelas, dan relevan, dengan seluruh informasi yang diperlukan pengguna. Artikel ini akan memandu Anda untuk menulis dokumentasi perangkat lunak bagi pengguna teknis dan pengguna akhir.

### e. Alat dan Bahan

Alat :

1. Komputer atau sejenisnya
2. Aplikasi Microsoft word, Power Point atau sejenisnya

Bahan :

1. Dokumen pelaksanaan kegiatan penjamin kualitas perangkat lunak

### f. Prosedur Kerja

1. **Buatlah halaman sampul dan judul yang sesuai.** Anda membutuhkan sampul depan untuk panduan pengguna apa pun yang lebih dari sekadar kartu referensi dan halaman judul untuk panduan yang mencakup lebih dari sekadar selembarnya kertas terlipat (dengan panjang 4 halaman atau lebih).
  - Jika panduan dilindungi oleh hak cipta, pemberitahuan hak cipta harus dicantumkan pada sampul depan dan belakang, serta pada halaman judul.
  - Jika ada syarat dan ketentuan untuk menggunakan panduan dan produk yang terkait dengannya, cantumkan di bagian dalam sampul depan.
2. **Letakkan referensi ke dokumen terkait dalam kata pengantar.** Jika panduan pengguna lebih dari satu, buatlah referensi ke dokumen yang lain, dengan mencantumkan nomor versi yang benar, di sini. Kata pengantar juga merupakan tempat untuk meletakkan bagian "Cara Menggunakan Panduan Ini" jika ada.
3. **Sertakan daftar isi jika panduan lebih dari 10 halaman.**



4. **Letakkan instruksi/prosedur dan bahan referensi di batang tubuh panduan.** Dalam kebanyakan kasus, baik prosedur dan bahan referensi masing-masing harus memiliki bagian tersendiri. Namun, Anda dapat memberi tahu pengguna untuk merujuk pada konten spesifik dalam satu bagian dan membedakannya dengan bagian lainnya. Dengan cara ini, pengguna bisa dengan cepat menemukan informasi yang dia cari.
  - Prosedur harus ditulis dalam struktur yang konsisten di seluruh bagian instruksi pada panduan. Mulailah dengan gambaran umum fungsi produk, lalu jelaskan apa yang harus dilakukan pengguna dan hasil apa yang akan dia dapatkan atau lihat. Langkah-langkahnya harus diberi nomor dan dimulai dengan kata kerja tindakan, seperti cara penulisan langkah-langkah dalam setiap bagian artikel ini.
  - Bahan referensi dapat meliputi daftar pilihan, kiat-kiat pemecahan masalah, dan pertanyaan umum. Glosarium dan indeks dapat ditambahkan di akhir panduan, meskipun daftar istilah yang sering digunakan bisa diberikan di awal. Indeks bisa diabaikan jika panduan memiliki kurang dari 20 halaman.
5. **Gunakan gambar grafis sesuai kebutuhan untuk mendukung teks.** Gambar grafis, atau tangkapan layar, dapat memberikan ilustrasi poin-poin tertentu dalam panduan secara lebih baik dibanding teks, terutama dalam prosedur kompleks yang membuat pengguna membutuhkan konfirmasi visual untuk memastikan mereka mengikuti langkah-langkah dengan benar. Gambar grafis dapat dibuat menggunakan *computer-aided drafting* (CAD) atau aplikasi pengedit grafis, kamera digital dan aplikasi pengolah foto, atau jika Anda ingin menggunakan tangkapan layar, gunakan fitur tangkapan layar yang ada di komputer atau program grafis dengan kemampuan menangkap layar.
  - Begitu Anda membuat gambar grafis, simpan dalam format terkompresi sehingga Anda dapat menggunakannya dalam aplikasi pengolah kata atau *desktop publishing*. Anda juga perlu memperkecil ukuran gambar agar lebih sesuai dengan halaman, tetapi tetap memberikan detail yang lengkap untuk pengguna. (Jika perlu, Anda bisa memotong gambar asli menjadi beberapa bagian dan menunjukkan bagian yang relevan bersama teks yang mendukungnya).
  - Jika Anda menggunakan banyak gambar grafis dalam prosedur, pastikan gambar dibuat dengan ukuran yang konsisten, entah dalam dimensi panjang dan lebar yang sama atau dalam proporsi memperkecil ukuran gambar asli. Dengan begitu, gambar akan terlihat lebih menarik bagi pengguna. Hal yang sama juga berlaku saat membuat tangkapan layar

dari komputer; pastikan komputer menampilkan skema warna standar saat mengambil gambar jika panduan akan menampilkan tangkapan layar berwarna.

- Meskipun aplikasi pengedit gambar seperti Photoshop dan Paint Shop Pro memiliki kemampuan pengambilan gambar yang baik, program yang didedikasikan untuk mengambil tangkapan layar seperti SnagIt juga menawarkan kemampuan untuk memodifikasi, membuat katalog dan anotasi pada tangkapan layar dengan mudah.

#### **g. Hasil dan Pembahasan**

1. Buatlah dokumen panduan pengguna untuk aplikasi website dan mobile yang anda kembangkan !

##### **FormatLaporan**

- KertasA4
  - Format\*.pdf
  - StrukturLaporan: Cover, Pendahuluan, Hasil Praktik, KesimpulandanDaftarPustaka
  - BerikanIdentitasDiri:NIM, Nama,Golongan,TugasMingguKe-X
  - PenamaanFile:ACARA-X\_GOL\_NIM\_NAMA.pdf
- \*\*Contoh:\*\***ACARA- Kumpulkan pada <http://jti.polije.ac.id/elearning/>

#### **h. Rubrik Penilaian**

No	INDIKATOR KINERJA	Bobot (%)	Penilaian	Nilai Akhir
1	Memberikan penjelasan dan analisa secara terstruktur yang disertai dengan hasil dokumen panduan pengguna untuk aplikasi web dan mobile	30		
2	Memberikan bukti referensi dari jawaban	15		
3	Mempertanggung jawabkan hasil pekerjaan dengan presentasi	25		
4	Kerapian dalam menulis (bahasa dan struktur penulisan)	10		
5	Ketepatan waktu mengumpulkan	20		
<b>Total</b>		<b>100</b>		