

Testing and Implementation



Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion



Session 01

PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>

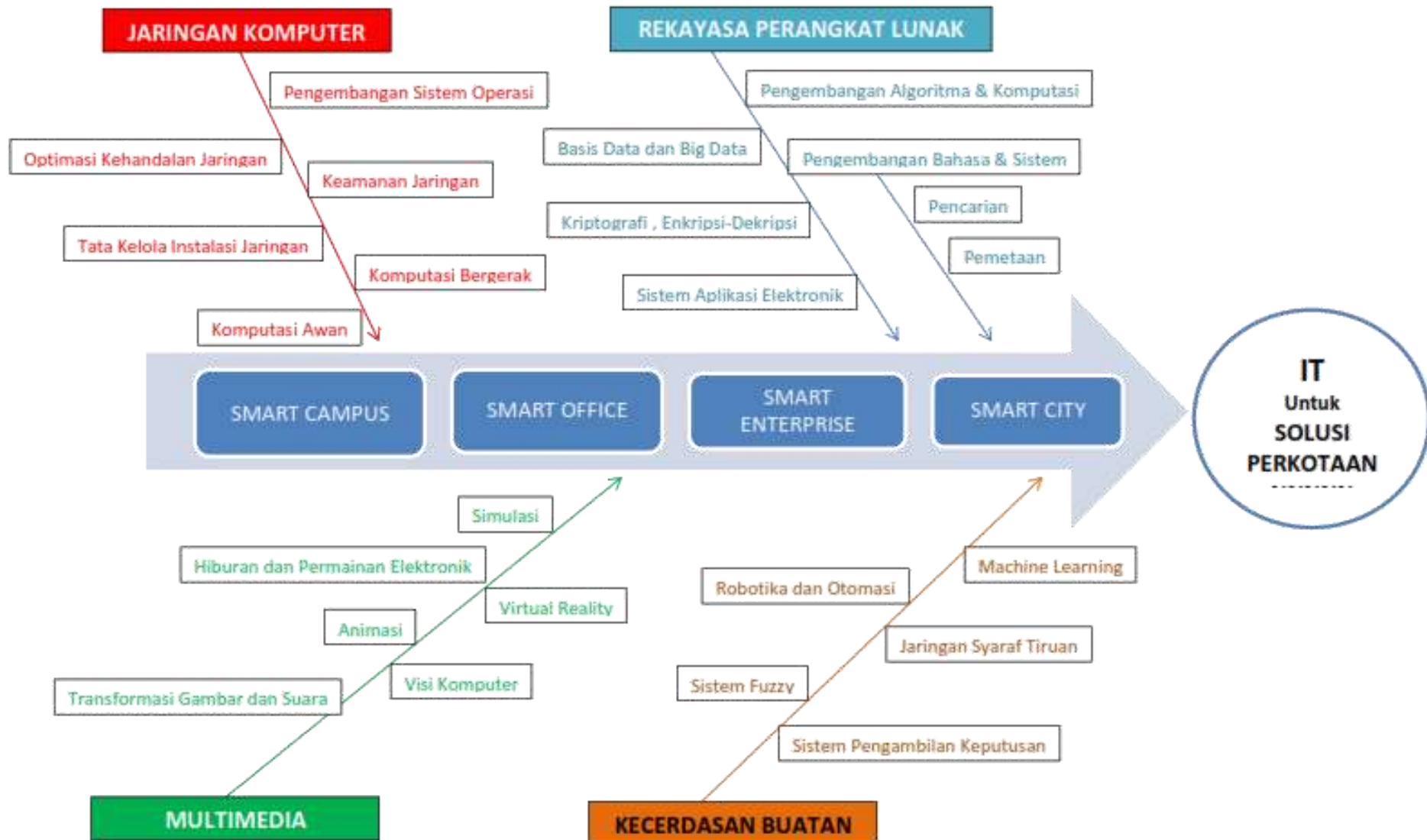
- **VISI.**

- √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.

- **MISI.**

1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhammadiyah.
2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkemajuan
3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.

Road Map TI UHAMKA





Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Keaktifan, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.

Tujuan/Objectives (Session.01)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Pengantar & Silabus.
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang RPS/SAP dan Silabus yang akan digunakan pada perkuliahan.



اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ
خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ اقْرَأْ وَرَبُّكَ الْأَكْرَمُ
الَّذِي عَلَّمَ بِالْقَلَمِ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ

- Qs. Al-Alaq, ayat 1 s/d 5
 - Bacalah dengan (menyebut) nama Tuhanmu yang menciptakan,
 - Dia telah menciptakan manusia dari segumpal darah.
 - Bacalah, dan Tuhanmulah Yang Mahamulia,
 - Yang mengajar (manusia) dengan pena.
 - Dia mengajarkan manusia apa yang tidak diketahuinya.

Testing and Implementation



Uhamka

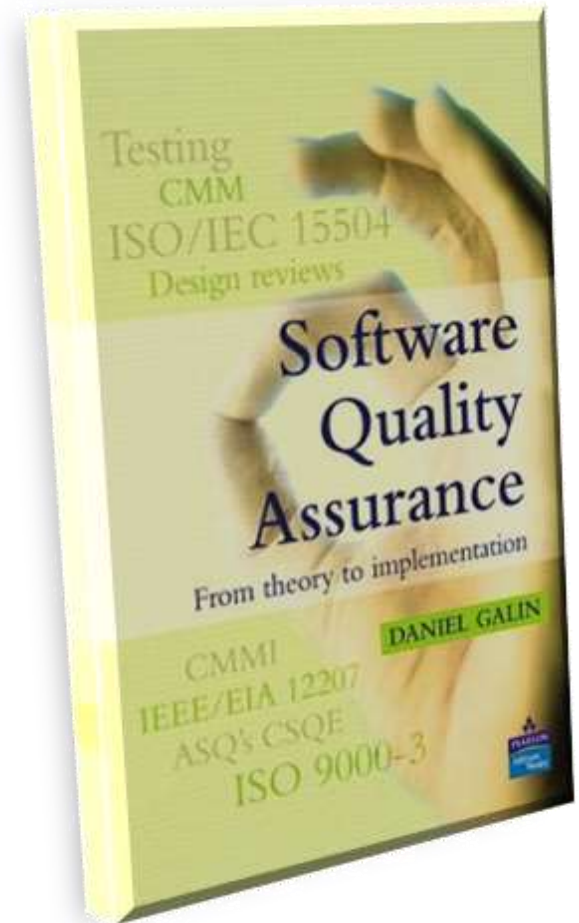
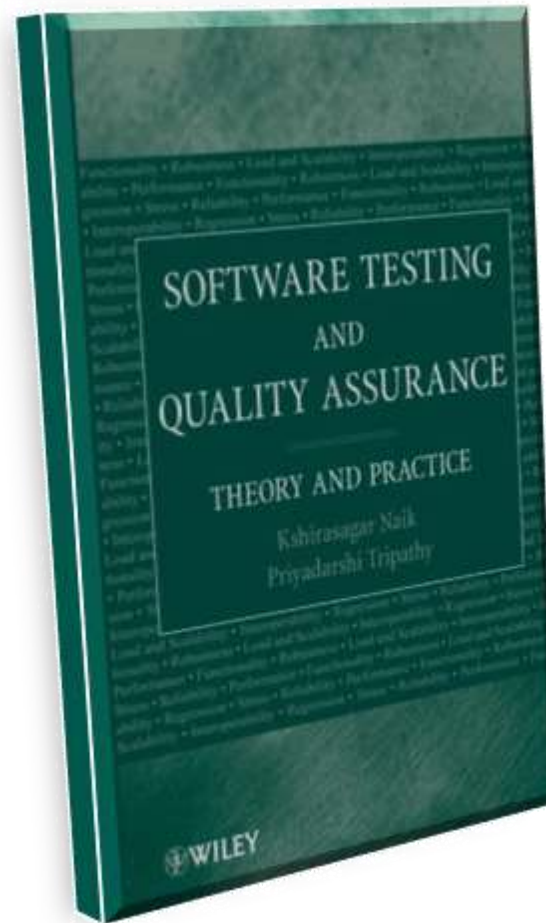
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion



Session 01

Pengantar & Silabus



RPS/SAP;

Rencana Pembelajaran Semester /
Satuan Acara Perkuliahan.



Session 01

- Kontrak Kuliah
- Perkenalan dan Silabus.
- Buku dan Referensi.

Session 03

- Konsep dasar testing.
- Testability

Session 05

- Unit testing.
- Memahami konsep unit testing dan cara menjalankannya.

Session 07

- Kategori test sistem.
- Memahami pengkategorian test system.

Session 02

- Pengantar Testing and Implementation

Session 04

- Tester, Defect
- Siklus Hidup dan Aktivitas Testing

Session 06

- Testing integrase sistem.
- Memahami testing pada saat integrase sistem.

Session 08

- U.T.S.

Session 09

- Perancangan test sistem.
- Memahami cara merancang test untuk sebuah sistem.

Session 11

- Eksekusi test sistem.
- Memahami cara eksekusi test sistem.

Session 13

- Kehandalan software.
- Memahami kehandalan software.

Session 15

- Project.

Session 10

- Perencanaan dan otomatisasi test sistem.
- Memahami proses perencanaan test sistem, dan otomatisasinya.

Session 12

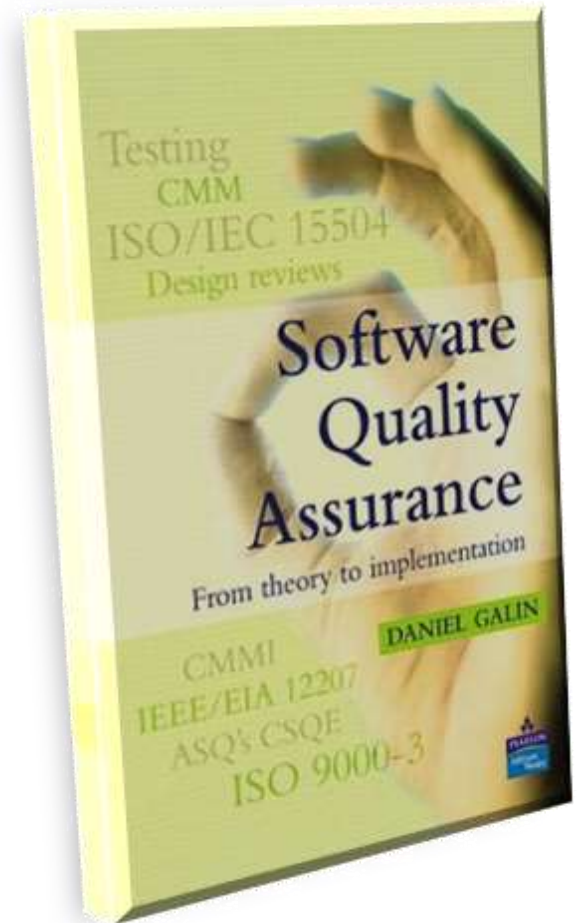
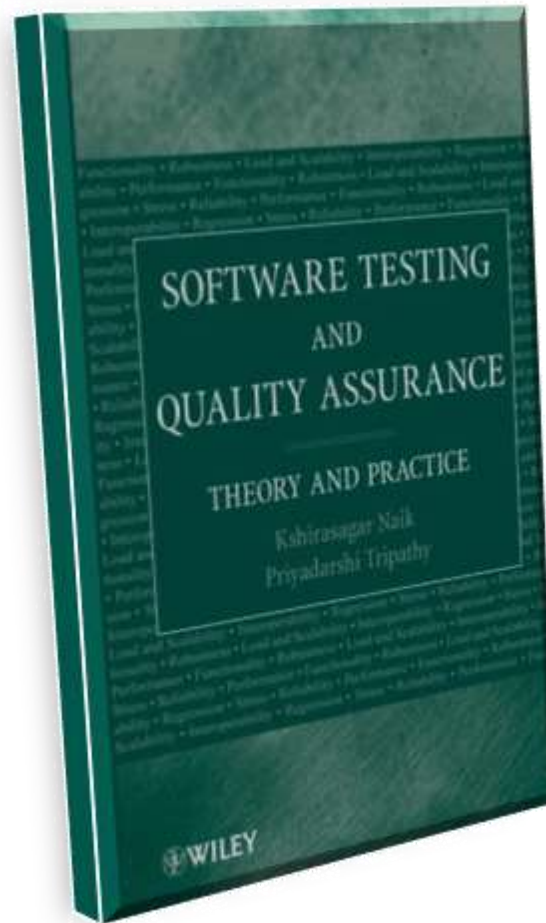
- Testing penerimaan.
- Memahami testing yang harus dilakukan pada saat produk siap untuk diserahkan ke klien.

Session 14

- Project.

Session 16

- U.A.S.



Pendahuluan;

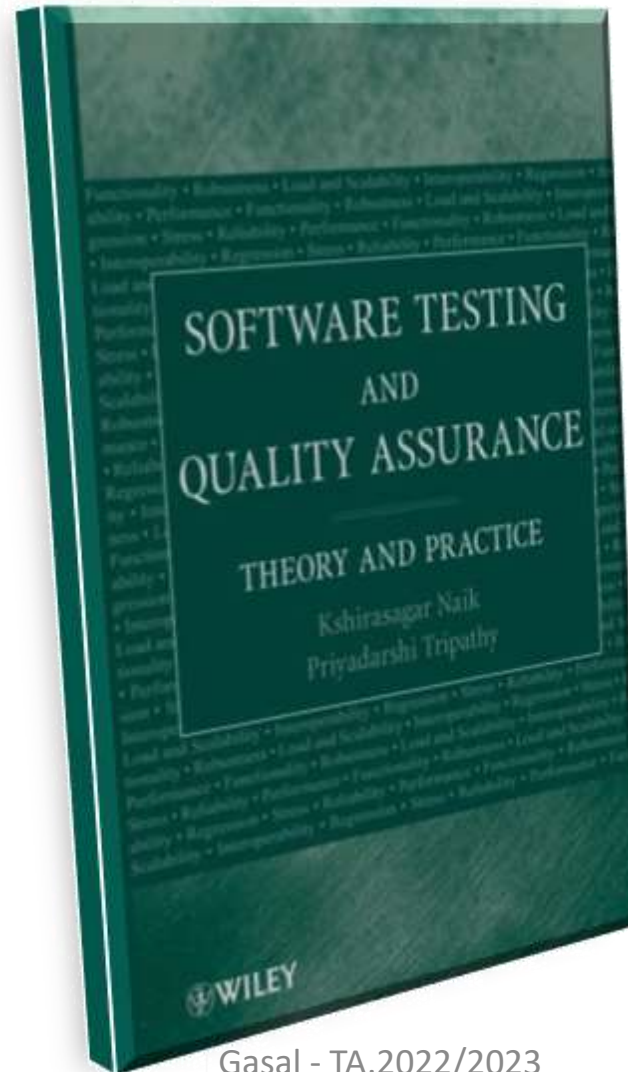
Referensi dan Buku-Buku



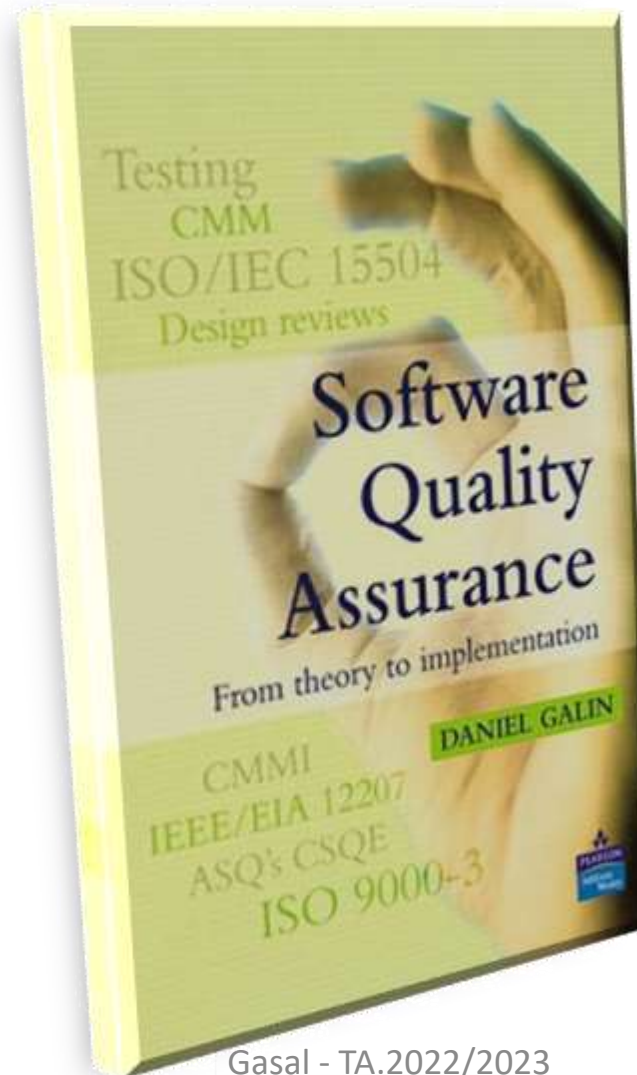
- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.



- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.



- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.



- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.

Dari Abu Hurairah رضي الله عنه, dia berkata



عَنِ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ:
مَا أَنْزَلَ اللَّهُ دَاءً
إِلَّا أَنْزَلَ لَهُ شِفَاءً.

Terima Kasih

"Dari Nabi ﷺ, beliau bersabda: "Tidaklah Allah menurunkan (-menakdirkan) suatu penyakit melainkan menurunkan pula obatnya." (HR. Bukhari).



Testing and Implementation



Session 02

PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>



- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhmadiyah.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkembang.
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.

Gasal - TA.2022/2023

2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.

Gasal - TA.2022/2023

3



Tujuan/Objectives (Session.02)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Pengantar Testing and Implementation
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang pengantar dari mata kuliah *testing and implementation*, pengertian *testing*, pengertian kualitas, hubungan antara kualitas dengan *testing*.

Gasal - TA.2022/2023

4



Muhasabah

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ
 خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ اقْرَأْ وَرَبُّكَ الْأَكْرَمُ
 الَّذِي عَلَّمَ بِالْقَلَمِ عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ

- Qs. Al-Alaq, ayat 1 s/d 5
 - Bacalah dengan (menyebut) nama Tuhanmu yang menciptakan,
 - Dia telah menciptakan manusia dari segumpal darah.
 - Bacalah, dan Tuhanmulah Yang Mahamulia,
 - Yang mengajar (manusia) dengan pena.
 - Dia mengajarkan manusia apa yang tidak diketahuinya.

Gasal - TA.2022/2023

5



Testing and Implementation



Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion



Session 02

Konsep Dasar *Testing*



Pengantar *Testing and Implementation*

Pendahuluan

Gasal - TA.2022/2023

7



Pendahuluan



- Masalah *testing* program muncul secara simultan bersamaan dengan pengalaman pertama dalam menulis program.

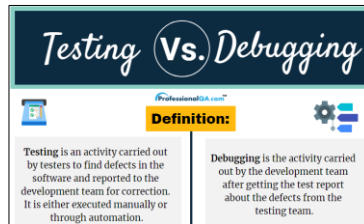
Gasal - TA.2022/2023

8



Pendahuluan (lanjutan)

- Di awal debutnya, *testing* merupakan aktifitas yang tidak hanya bertujuan untuk menemukan *error*, tapi juga bertujuan untuk mengoreksi/memperbaiki dan menghilangkan *bug* tersebut.
- Namun sudut pandang ini telah bergeser di tahun 1957, dimana *testing* program telah dibedakan secara jelas dengan *debugging*.



9



Pendahuluan (lanjutan)



- Sejak konferensi pertama tentang *testing software*, yang diadakan pada bulan Juni 1972 di *University of North Carolina*, mulai banyak konferensi dan *workshop* yang bertemakan tentang kualitas, reliabilitas dan rekayasa *software*, dimana secara bertahap telah memasukan disiplin *testing* sebagai elemen yang terorganisasi dalam teknologi *software*.

Gasal - TA.2022/2023

10



Pendahuluan (lanjutan)

Beberapa definisi tentang *testing*:

- Menurut Hetzel 1973:
 - *Testing* adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan.
- Menurut Myers 1979:
 - *Testing* adalah proses eksekusi program atau sistem secara intens untuk menemukan *error*.
- Menurut Hetzel 1983 (Revisi):
 - *Testing* adalah tiap aktivitas yang digunakan untuk dapat melakukan evaluasi suatu atribut atau kemampuan dari program atau sistem dan menentukan apakah telah memenuhi kebutuhan atau hasil yang diharapkan.
- Menurut Standar ANSI/IEEE 1059:
 - *Testing* adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*defects/errors/bugs*) dan mengevaluasi fitur-fitur dari entitas *software*.

Gasal - TA.2022/2023

11



Pendahuluan (lanjutan)

- Beberapa pandangan praktisi tentang *testing*, adalah sebagai berikut:
 - Melakukan cek pada program terhadap spesifikasi.
 - Menemukan *bug* pada program.
 - Menentukan penerimaan dari pengguna.
 - Memastikan suatu sistem siap digunakan.
 - Meningkatkan kepercayaan terhadap kinerja program.
 - Memperlihatkan bahwa program bekerja dengan benar.
 - Membuktikan bahwa *error* tidak terjadi.
 - Mengetahui akan keterbatasan sistem.
 - Mempelajari apa yang tak dapat dilakukan oleh sistem.
 - Melakukan evaluasi kemampuan sistem.
 - Verifikasi dokumen.
 - Memastikan bahwa pekerjaan telah diselesaikan.

Gasal - TA.2022/2023

12



Pengantar *Testing and Implementation*

Kualitas

Gasal - TA.2022/2023

13



Kualitas

- Definisi lain yang ditemukan dalam beberapa literatur, mendefinisikan *testing* sebagai pengukuran kualitas *software*.
- Sama halnya dengan *testing*, pengertian kualitas bagi tiap praktisi dapat berbeda-beda, karena kualitas memang merupakan suatu hal yang subjektif dan abstrak.
- Berikut ini beberapa definisi sederhana tentang kualitas:
 - Menurut CROSBY: Kualitas adalah pemenuhan terhadap kebutuhan.
 - Menurut ISO-8402: Kualitas adalah keseluruhan dari fitur yang menjadikan produk dapat memuaskan atau dipakai sesuai kebutuhan dengan harga yang terjangkau.
 - Menurut W.E. Perry: Kualitas adalah pemenuhan terhadap standar.
 - Menurut R. Glass: Kualitas adalah tingkat kesempurnaan.
 - Menurut J. Juran: Kualitas adalah tepat guna.

Gasal - TA.2022/2023

14



Hubungan *testing* dengan kualitas.

- Definisi *software* berkualitas adalah *software* yang bebas *error* dan *bug* secara obyektif, tepat waktu dan dana, sesuai dengan kebutuhan atau keinginan dan dapat dirawat (*maintainable*).
- *Testing* membuat kualitas dapat dilihat secara obyektif, karena *testing* merupakan pengukuran dari kualitas *software*.



- Dengan kata lain testing berarti pengendalian kualitas (*Quality Control* - QC), dan QC mengukur kualitas produk, sedangkan jaminan kualitas (*Quality Assurance* – QA) mengukur kualitas proses yang digunakan untuk membuat produk berkualitas.

Gasal - TA.2022/2023

15



Kualitas (lanjutan)

- Salah satu hal yang mendasar bila berbicara tentang kualitas dan bagaimana cara mengukurnya, adalah faktor-faktor apa yang menjadi tolak ukur bagi kualitas tersebut.
- Faktor-faktor kualitas *software* secara umum dapat dibedakan menjadi tiga faktor, yaitu :
 - 1) Fungsionalitas
 - 2) Rekayasa
 - 3) Adaptabilitas.
- Dimana ketiga faktor utama ini dapat juga disebut sebagai dimensi dari ruang lingkup kualitas *software*.



Gasal - TA.2022/2023

16



Kualitas (lanjutan)

- Berikut contoh yang mengilustrasikan beberapa faktor-faktor komponen yang sering digunakan:

Fungsionalitas (Kualitas Luar)

- Kebenaran (*Correctness*)
- Reliabilitas (*Reliability*)
- Kegunaan (*Usability*)
- Integritas (*Integrity*)

Rekayasa (Kualitas Dalam)

- Efisiensi (*Efficiency*)
- Testabilitas (*Testability*)
- Dokumentasi (*Documentation*)
- Struktur (*Structure*)

Adaptabilitas (Kualitas ke Depan)

- Fleksibilitas (*Flexibility*)
- Reusabilitas (*Reuseability*)
- Maintainabilitas (*Maintainability*)

Gasal - TA.2022/2023

17



Kualitas (lanjutan)

- Karena itu *testing* yang bagus harus dapat mengukur semua faktor-faktor yang berhubungan, yang tentunya tiap faktor komponen akan mempunyai tingkat kepentingan berbeda-beda antar satu aplikasi dengan aplikasi yang lain.
- Jadi agar *testing* dapat sepenuhnya efektif, maka harus dijalankan untuk melakukan pengukuran tiap faktor yang berhubungan dan juga menjadikan kualitas menjadi nyata dan terlihat.

Gasal - TA.2022/2023

18



Pengantar *Testing and Implementation*

Pendahuluan

Gasal - TA.2022/2023

19



Kualitas *Software & Organisasi Software*

- Secara natural pengembangan *software* bukanlah suatu hal yang mudah, bahkan mempunyai kecenderungan untuk mengalami kegagalan.
- Oleh karena itu berorientasi pada kualitas adalah salah satu usaha dalam menurunkan tingkat resiko terjadinya kegagalan proyek.
- Perlu diketahui dari data statistik di tahun 1995, perusahaan dan agen pemerintahan Amerika Serikat telah menghabiskan dana 81 milyar US\$ untuk proyek *software* yang dibatalkan, dengan rincian sebagai berikut:
 - 31 % Proyek dibatalkan sebelum selesai.
 - 53 % Proyek mengalami pembengkakan biaya sebesar 189% dari nilai estimasi.
 - 16 % Proyek selesai tepat waktu dan anggaran.

Gasal - TA.2022/2023

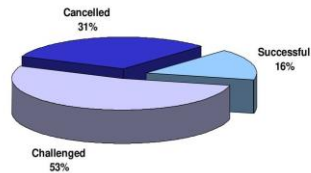
20



Kualitas *Software* & Organisasi *Software* (lanjutan)



Software Project Statistics 1995



Source: The Standish Group, 1995

Gasal - TA.2022/2023

21



Kualitas *Software* & Organisasi *Software* (lanjutan)

- Dari data statistik di atas terlihat bahwa, sebenarnya masalah utama dari kualitas *software* adalah biaya dan jadwal dengan akar penyebab dari masalah.
- Yaitu terutama kemampuan rekayasa *software* dari pihak pengembang (*developer*) yang tak mencukupi, dan kemampuan pelanggan (*client*) yang sangat kurang (bahkan tak mampu) untuk memberikan spesifikasi kebutuhan dari sistem.
- Dengan berorientasi pada kualitas, maka organisasi *software* akan dapat melakukan proses analisa, evaluasi, dan pengembangan yang berkesinambungan untuk mencapai suatu proses pengembangan *software* yang semakin lama semakin efektif, efisien, terukur, terkendali dan dapat diulang secara konsisten dalam menghasilkan suatu produk (*software*) yang berkualitas, tepat waktu, dan efisien dalam pendanaan.

Gasal - TA.2022/2023

22



Kualitas *Software* & Organisasi *Software* (lanjutan)



- Dimana hal ini akan memberikan suatu jaminan bagi pelanggan/klien untuk mendapatkan produk seperti yang diharapkan, sehingga akan menambah kepercayaan mereka akan kemampuan pengembang.
- Hal ini sangat dibutuhkan bagi organisasi *software* karena hubungan klien dan pengembangan adalah untuk jangka panjang dan berkesinambungan (*marital status*).

Gasal - TA.2022/2023

23



Referensi

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.

Gasal - TA.2022/2023

24

Dari Abu Hurairah رضي الله عنه, dia berkata

Terima Kasih



عَنِ النَّبِيِّ ﷺ قَالَ:
مَا أَنْزَلَ اللَّهُ دَاءً
إِلَّا أَنْزَلَ لَهُ شِفَاءً.

"Dari Nabi ﷺ, beliau bersabda: "Tidaklah Allah menurunkan (-menakdirkan) suatu penyakit melainkan menurunkan pula obatnya." (HR. Bukhari).



Testing and Implementation



Session 03



PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>

- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhmadiyah.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkemajuan
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.

Gasal - TA.2022/2023

2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.

Gasal - TA.2022/2023

3



Tujuan/Objectives (Session.03)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Konsep Dasar Testing
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang dasar-dasar *testing* seperti obyektivitas, prinsip-prinsip dasar *testing*, dan testabilitas.

Gasal - TA.2022/2023

4



Muhasabah

يَرْفَعُ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ

- Qs. Al-Mujadilah, ayat 11
 - Allah akan mengangkat (derajat) orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu beberapa derajat. Dan Allah Maha teliti apa yang kamu kerjakan.

Gasal - TA.2022/2023

5



Konsep Dasar *Testing*

Pendahuluan

Gasal - TA.2022/2023

7



Testing and Implementation



Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion

Session 03
Konsep Dasar *Testing*

Obyektivitas *Testing*

- Secara umum obyektifitas dari testing adalah untuk melakukan verifikasi, validasi dan deteksi *error* untuk menemukan masalah dan tujuan dari penemuan ini adalah untuk membenahinya.
- Namun terdapat pula beberapa pendapat dari praktisi yang dapat pula dipandang sebagai bagian dari obyektifitas *testing*, antara lain:
 - Meningkatkan kepercayaan bahwa sistem dapat digunakan dengan tingkat resiko yang dapat diterima.
 - Menyediakan informasi yang dapat mencegah terulangnya *error* yang pernah terjadi.
 - Menyediakan informasi yang membantu untuk deteksi *error* secara dini.
 - Mencari *error* dan kelemahan atau keterbatasan sistem.
 - Mencari sejauh apa kemampuan dari sistem.
 - Menyediakan informasi untuk kualitas dari produk *software*.

Gasal - TA.2022/2023

8



Pendahuluan



“Testing merupakan tugas yang tak dapat dihindari di tiap bagian dari tanggung jawab usaha pengembangan suatu sistem *software*”

William Howden

Gasal - TA.2022/2023

9



Misi dari Tim *Testing*

- Misi dari tim *testing* tidak hanya sekedar untuk melakukan *testing*, tapi juga untuk membantu meminimalkan resiko kegagalan proyek.
- Tester adalah individu yang memberikan hasil pengukuran dari kualitas produk.
- Tester mencari manifestasi masalah dari produk, masalah yang potensial, dan kehadiran dari masalah.
- Mereka mengeksplorasi, mengevaluasi, melacak, dan melaporkan kualitas produk, sehingga tim lainnya dari proyek dapat membuat keputusan terhadap pengembangan produk.
- Penting diingat bahwa tester tidak melakukan pembenahan atau pembedahan kode, tidak memermalukan atau melakukan komplain pada suatu individu atau tim, tetapi tester hanya menginformasikan.

Gasal - TA.2022/2023

10



Konsep Dasar *Testing*

Prinsip-prinsip *Testing*

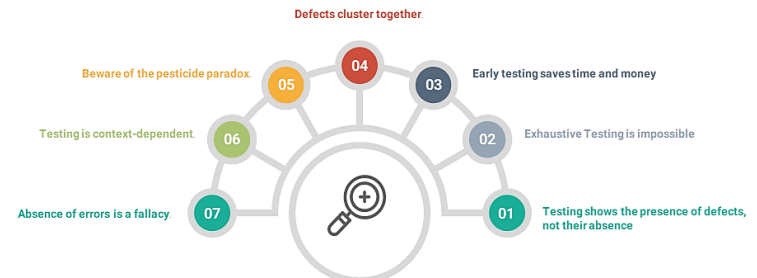
Gasal - TA.2022/2023

11



Prinsip-prinsip *Testing*

- Didalam melakukan sebuah *testing*, terdapat 7 prinsip utama yang wajib diperhatikan, yaitu ditunjukkan pada gambar dibawah ini:



Gasal - TA.2022/2023

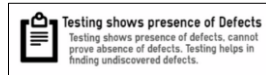
12



Prinsip-prinsip *Testing* (lanjutan)

1. *Testing shows presence of defects*

- Tujuan utama dari proses *testing* adalah menemukan cacat/*bug* pada suatu *software*.
- Pengujian dapat menunjukkan cacat yang dapat timbul, tetapi tidak dapat membuktikan bahwa tidak ada cacat.
- Bahkan setelah pengujian aplikasi atau produk secara menyeluruh kita tidak bisa mengatakan bahwa produk tersebut adalah 100% bebas cacat.
- Pengujian selalu mengurangi jumlah cacat yang belum ditemukan yang tersisa dalam perangkat lunak, tetapi bahkan jika tidak ada cacat yang ditemukan, maka itu bukan merupakan bukti kebenaran bahwa *software* tersebut bebas dari *bug*.



Gasal - TA.2022/2023

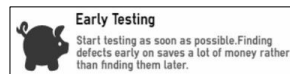
13



Prinsip-prinsip *Testing* (lanjutan)

3. *Early testing saves time and money*

- Dalam sebuah siklus *Software Development Life Cycle* (SDLC) kegiatan pengujian harus dimulai sedini mungkin dan harus difokuskan pada tujuan yang telah ditetapkan.
- Pengujian yang dilakukan sedini mungkin, ternyata dapat menghemat waktu dan biaya.



Gasal - TA.2022/2023

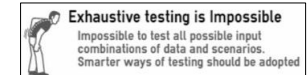
15



Prinsip-prinsip *Testing* (lanjutan)

2. *Exhausting testing is impossible*

- Menguji segala sesuatu termasuk semua kombinasi input dan prasyarat tidak mungkin.
- Jadi, bukannya melakukan pengujian lengkap, maka kita dapat menggunakan risiko dan prioritas untuk memfokuskan terhadap upaya pengujian.
- Jadi, mengakses dan mengelola risiko adalah salah satu kegiatan yang paling penting dan alasan untuk pengujian dalam setiap proyek.



Gasal - TA.2022/2023

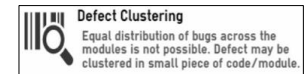
14



Prinsip-prinsip *Testing* (lanjutan)

4. *Defects cluster together*

- Sejumlah kecil modul berisi sebagian besar cacat ditemukan selama pengujian pra-rilis atau menunjukkan kegagalan yang paling operasional.
- Hal ini dapat memudahkan tester dalam mengidentifikasi.



Gasal - TA.2022/2023

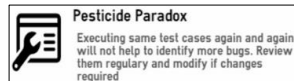
16



Prinsip-prinsip *Testing* (lanjutan)

5. *Beware of the Pesticide paradox*

- Jika jenis yang sama dari tes diulang lagi dan lagi, akhirnya set yang sama dari kasus uji tidak akan lagi dapat menemukan *bug* baru.
- Hal ini biasa disebut juga sebagai “*Pesticide Paradox*”.
- Untuk mengatasi hal ini maka sangat penting untuk meninjau kasus uji secara teratur, dan menggunakan tes baru yang berbeda harus ditulis/dicatat sebagai latihan bagian yang berbeda dari perangkat lunak atau sistem untuk berpotensi menemukan lebih banyak cacat.



Gasal - TA.2022/2023

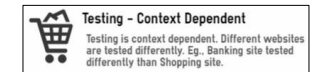
17



Prinsip-prinsip *Testing* (lanjutan)

6. *Testing is context-dependent*

- Pengujian pada dasarnya tergantung konteks.
- Berbagai jenis situs yang diuji berbeda.
- Misalnya, *testing software* yang berkaitan erat dengan keamanan tentu berbeda dari sebuah situs *e-commerce*.



Gasal - TA.2022/2023

18



Prinsip-prinsip *Testing* (lanjutan)

7. *Absence of errors fallacy*

- Jika sistem yang dibangun tidak dapat digunakan dan tidak memenuhi kebutuhan dan harapan dari pengguna, maka kemudian menemukan dan memperbaiki cacat/bugs, tidak akan berguna sama sekali.



Gasal - TA.2022/2023

19



Prinsip-prinsip *Testing* (lanjutan)

- Selain 7 prinsip utama dari *testing*, terdapat juga prinsip-prinsip lain yang perlu diperhatikan, yaitu:
 - Pengujian harus dilakukan oleh pihak independen.
 - Pengujian tidak boleh dilakukan oleh orang atau tim yang menjadi bagian didalam mengembangkan perangkat lunak tersebut, karena mereka cenderung membela kebenaran program.
 - Menetapkan personil terbaik untuk tugas itu.
 - Karena pengujian membutuhkan kreativitas tinggi dan tanggung jawab, maka hanya personil terbaik yang harus ditugaskan untuk merancang, melaksanakan, dan menganalisa uji kasus, data uji dan hasil tes.
 - Uji untuk kondisi input tidak valid dan tak terduga serta kondisi valid.
 - Program ini harus menghasilkan pesan yang salah saat tes dalam kondisi tidak valid ditemui, dan harus menghasilkan hasil yang benar ketika tes tersebut valid.
 - Jauhkan perangkat lunak statis selama tes.
 - Program ini tidak boleh diubah, selama dilakukan uji kasus atau selama dalam pengujian.
 - Memberikan hasil tes diharapkan jika memungkinkan.
 - Sebuah bagian penting dari dokumentasi uji spesifikasi hasil yang diharapkan, bahkan jika memberikan hasil seperti tidak praktis.

Gasal - TA.2022/2023

20



Konsep Dasar Testing

Testability

Gasal - TA.2022/2023

21



Testability

- Idealnya perekayasa perangkat lunak (*software engineering*) mendesain program komputer, sistem ataupun produk, dengan menempatkan testabilitas dalam benaknya (*mindset*).
- Hal ini akan memungkinkan untuk membantu *testing* dalam mendesain *test case* yang efektif dan lebih mudah.
- Secara sederhana, menurut James Bach, testabilitas *software* adalah seberapa mudah (suatu program komputer) dapat dites.
- Atribut-atribut di bawah ini disarankan oleh Bach untuk dapat digunakan oleh perekayasa *software* didalam mengembangkan suatu konfigurasi *software* (seperti program, data, dan dokumen) yang akan dapat membantu *testing*.

Gasal - TA.2022/2023

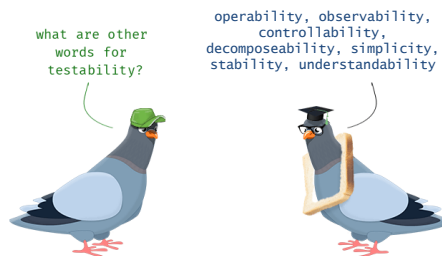
22



Testability (lanjutan)

- Berikut daftar sekumpulan karakteristik yang dapat mengarahkan pada *software* yang dapat dites.

- Operability*
- Observability*
- Controllability*
- Decomposeability*
- Simplicity*
- Stability*
- Understandability*



Gasal - TA.2022/2023

23



Testability (lanjutan)

a. *Operability*.

“Semakin baik *software* berkerja, maka akan membuat *software* dites dengan lebih efisien.”

- Sistem mempunyai *bug* baru (*bug* menambahkan biaya tak langsung pada proses *testing*, dengan adanya analisa dan pelaporan).
- Tidak ada *bug* yang menghentikan eksekusi tes
- Produk berubah dalam tahap fungsional (memungkinkan pengembangan dan *testing* yang simultan).

Gasal - TA.2022/2023

24



b. Observability

“Apa yang Anda lihat, adalah apa yang Anda tes.”

- Hasil dari setiap keluaran harus menunjukkan hasil dari masukan.
- Kondisi sistem dan variabel dapat dilihat atau diquery selama eksekusi berlangsung.
- Kondisi dan variabel sistem lama juga dapat dilihat atau di-query.
- Semua faktor yang mempengaruhi keluaran dapat dilihat.
- Keluaran yang salah dapat dengan mudah diidentifikasi
- Kesalahan internal dapat secara otomatis dideteksi oleh mekanisme tes yang menyeluruh.
- Kesalahan internal secara otomatis dilaporkan.
- *Source code* dapat diakses.

Gasal - TA.2022/2023

25

Testability (lanjutan)



c. Controllability

“Dengan semakin baik kita dapat mengendalikan *software*, semakin banyak testing dapat diotomatisasi dan dioptimalisasi.”

- Semua kemungkinan keluaran dihasilkan dari berbagai kombinasi masukan.
- Semua kode dieksekusi dari beberapa kombinasi masukan.
- Kondisi *hardware* dan *software* dan variabel dapat dikontrol secara langsung oleh *test engineer*.
- Format masukan dan keluaran harus konsisten dan terstruktur.
- *Testing* dapat dengan mudah dispesifikasikan, otomasi, dan dibuat ulang.

Gasal - TA.2022/2023

26

Testability (lanjutan)



d. Decomposeability

“Dengan pengendalian batasan testing, kita dapat lebih cepat dalam mengisolasi masalah dan melakukan *testing* ulang yang lebih baik.”

- Sistem *software* dibangun dari modul-modul yang independen.
- Modul *software* dapat di tes secara independen (sendiri-sendiri).

Gasal - TA.2022/2023

27

Testability (lanjutan)



e. Simplicity

“Semakin sedikit yang dites, semakin cepat kita melakukannya.”

- Kesederhanaan fungsi (fitur yang ada di buat seminimal mungkin untuk memenuhi kebutuhan yang ada).
- Kesederhanaan struktur (arsitektur dibuat sesederhana mungkin untuk menghindari kesalahan).
- Kesederhanaan kode (standar dari kode dibuat agar dengan mudah diinspeksi dan dirawat).

Gasal - TA.2022/2023

28

Testability (lanjutan)



Testability (lanjutan)

f. Stability

“Semakin sedikit perubahan, semakin sedikit masalah/gangguan *testing*.”

- Perubahan dari *software* terjadi kadang-kadang.
- Perubahan dari *software* tidak terkendali.
- Perubahan dari *software* tidak dapat divalidasi pada tes yang ada.
- *Software* dapat melakukan perbaikan untuk kembali berjalan dengan baik (*recovery*) dari kegagalan proses.

Gasal - TA.2022/2023

29



Testability (lanjutan)

g. Understandability

“Semakin banyak informasi yang kita miliki, kita akan dapat melakukan tes lebih baik.”

- Disain mudah dimengerti dan dipahami dengan baik.
- Keterkaitan antara internal, eksternal dan share komponen dipahami dengan baik.
- Perubahan disain dikomunikasikan.
- Dokumentasi teknis dapat dengan mudah diakses.
- Dokumentasi teknis diorganisasi dengan baik .
- Dokumentasi teknis berisi spesifikasi dan detail.
- Dokumentasi teknis yang akurat.

Gasal - TA.2022/2023

30



Referensi

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.

Gasal - TA.2022/2023

31





Testing and Implementation



Session 04

PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>



- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhammadiyah.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkembang.
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.

Gasal - TA.2022/2023

2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.

Gasal - TA.2022/2023

3



Tujuan/Objectives (Session.04)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Tester, dan Aktivitas Testing
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang tester, personalitas tester, *defect*, siklus hidup, dan aktivitas *testing*,

Gasal - TA.2022/2023

4



Muhasabah

لَقَدْ كَانَ لَكُمْ فِي رَسُولِ اللَّهِ أُسْوَةٌ حَسَنَةٌ
 لِمَن كَانَ يَرْجُوا اللَّهَ وَالْيَوْمَ الْآخِرَ وَذَكَرَ اللَّهَ كَثِيرًا

- Qs. Al-Ahzab, ayat 21
 - Sungguh, telah ada pada (diri) Rasulullah itu suri teladan yang baik bagimu (yaitu) bagi orang yang mengharap (rahmat) Allah dan (kedatangan) hari Kiamat dan yang banyak mengingat Allah.

Gasal - TA.2022/2023

5



Tester, Defect, dan Aktivitas Testing

skill Tester

Gasal - TA.2022/2023

7



Testing and Implementation



Uhamka
 UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA
integrity, trust, compassion

Session 04

Tester, Defect, dan Aktivitas Testing



Skill Tester

- Bila berbicara tentang karir di dunia *testing*, walaupun pada saat ini masih kurang menjadi perhatian utama di setiap organisasi berbasis teknologi informasi.
- Namun seiring dengan perkembangan dari tingkat kedewasaan proses pengembangan *software*, maka dapat disimpulkan bahwa karir untuk menjadi seorang tester mempunyai prospek yang cukup menjanjikan.



Gasal - TA.2022/2023

8



Skill Tester (lanjutan)

Kemampuan tester yang menjadi permintaan pada umumnya, adalah sebagai berikut:

- Kemampuan secara umum
 - Mempunyai kemampuan analisa yang kuat dan terfokus
 - Mempunyai kemampuan komunikasi yang baik
 - Mempunyai latar belakang QA (*Quality Assurance*)
- Pemahaman terhadap metodologi
 - Pengembangan rencana tes
 - Pembuatan dan perawatan lingkungan tes
 - Standar tes
 - Dokumentasi tes (seperti *test cases* dan *procedure test*)

Gasal - TA.2022/2023

9



Skill Tester (lanjutan)

- Pengetahuan akan pendekatan *testing*
 - *Integration testing*
 - *Acceptance Testing*
 - *Stress / Volume Testing*
 - *Regression testing*
 - *Functional testing*
 - *End-To-End Testing*
 - *GUI Testing*
- Pengetahuan tentang sistem (berhubungan dengan pasar dari organisasi bersangkutan)
 - Perbankan/Keuangan
 - Produk Komersial
 - Telecom
 - *Internet*
 - Y2K

Gasal - TA.2022/2023

10



Skill Tester (lanjutan)

- Pengetahuan dan pengalaman akan penggunaan alat bantu *testing*
 - Alat bantu *capture* atau *playback* (seperti WinRunner)
 - Alat bantu *Load testing* (seperti LoadRunner, RoboTest)
- Kemampuan terhadap lingkungan *testing*
 - *Mainframe* (seperti MVS, JCL).
 - *Client – Server* (seperti WinNT, UNIX)
- Kemampuan terhadap aplikasi
 - Dokumentasi (seperti office, excel, word, Lorus Notes)
 - *Database* (seperti oracle, access, 3GL, 4GL, SQL, RDBMS)
 - Pemrograman (seperti C++, VB, OO)

Gasal - TA.2022/2023

11



Skill Tester (lanjutan)

- R.Collard.1997, menyatakan bahwa kemampuan tester dibedakan menjadi tiga grup besar, yaitu :
 - Kemampuan fungsional dari subyek yang menjadi acuan
 - Basis teknologi
 - Teknik – teknik *testing* dan QA (*Quality Assurance*)



Gasal - TA.2022/2023

12



Tester, Defect, dan Aktivitas Testing

Personalitas Tester

Gasal - TA.2022/2023

13



Personalitas Tester (lanjutan)

Untuk itu perlu kiranya untuk mengetahui atribut-atribut personaliti yang diharapkan bagi seorang tester, yaitu:

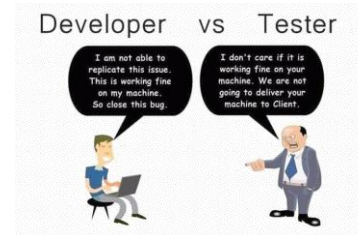
- **Atribut positif yang patut dikembangkan**
 - Terencana, sistematis, dan berhati-hati (tidak sembrono) → pendekatan logis terhadap testing.
 - Bermental juara → seperti penerapan standar kualitas yang tinggi dalam bekerja dalam suatu proyek.
 - Berpendirian teguh → tidak mudah menyerah
 - Praktikal → menyadari terhadap apa yang dapat dicapai terhadap batasan waktu dan anggaran tertentu.
 - Analitikal → memiliki intuisi dalam mengambil suatu pendekatan untuk menggali *error*.
 - Bermoral baik → berjuang untuk kualitas dan sukses, mengerti dan menyadari akan biaya-biaya yang terjadi terhadap suatu kualitas yang rendah

Gasal - TA.2022/2023

15



Personalitas Tester



- Seorang tester harus mempunyai personalitas yang lebih.
- Keberadaan testing akan dapat menjadi bumerang yang dapat menghancurkan keutuhan kerja sama tim pada organisasi tersebut bila tidak mendapat perhatian dan pengelolaan yang baik.

Gasal - TA.2022/2023

14



Personalitas Tester (lanjutan)

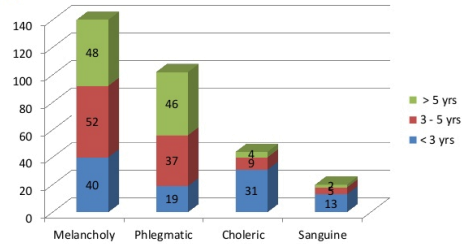
- **Atribut negatif yang patut dihindari**
 - Sedikit empati terhadap pengembang (*developers*) → mudah terpengaruh secara emosional yang kekanak-kanakan dalam hubungannya dengan pengembang (*developers*).
 - Kurang berdiplomasi → menciptakan konflik dengan pengembang (*developers*) dengan menunjukkan wajah yang tak bersahabat. Seorang tester akan tersenyum bila bertatap muka dengan pengembang (*developers*) saat menemukan defect, dan memberikan laporan defect yang disertai perhitungan statistik terjadinya defect dan bug.
 - Skeptis → meminta informasi dari pengembang (*developers*) dengan penuh kecurigaan (tidak percaya).
 - Keras kepala → tidak dapat fleksibel dalam mendiskusikan suatu proposal.

Gasal - TA.2022/2023

16



Software Tester's Personality A Statistics Sample



Statistics from personality tests of 306 testers from 4 IT companies in HCMC, done by people development programs of Talent Mind Education in 2012-2014

Gasal - TA.2022/2023

17

Personalitas Tester (lanjutan)



Tester, Defect, dan Aktivitas Testing

Defect

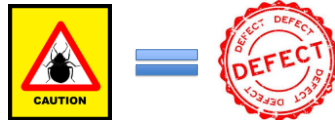
Gasal - TA.2022/2023

18



Defect

- Cacat perangkat lunak (*software defect*), didefinisikan sebagai defect pada perangkat lunak yang mungkin terjadi *defect* pada kode program, *defect* pada dokumentasi, pada desain, dan hal-hal lain yang menyebabkan kegagalan perangkat lunak.
- Software *defect* adalah suatu kondisi dalam produk perangkat lunak yang tidak memenuhi persyaratan perangkat lunak (sebagaimana tercantum dalam spesifikasi persyaratan) atau harapan pengguna akhir (yang mungkin tidak ditentukan tetapi masuk akal).



Gasal - TA.2022/2023

19



Defect (lanjutan)

Menurut Kaner, Falk, dan Nguyen, terdapat 13 kategori utama *defect* dari *software*, yaitu sebagai berikut:

- 1) *User interface errors*
 - Sistem memberikan suatu tampilan yang berbeda dari spesifikasi.
- 2) *Error handling*
 - Pengenalan dan perlakuan terhadap *error* bila terjadi.
- 3) *Boundary – related errors*
 - Perlakuan terhadap nilai batasan dari jangkauan mereka yang mungkin tidak benar.
- 4) *Calculation errors*
 - Perhitungan arimatika dan logika yang mungkin tidak benar.

Gasal - TA.2022/2023

20



Defect (lanjutan)

- 5) *Initial and later states*
 - Fungsi gagal pada saat pertama digunakan atau sesudah itu.
- 6) *Control flow errors*
 - Pilihan terhadap apa yang akan dilakukan berikutnya tidak sesuai untuk status saat ini.
- 7) *Errors in handling or interpreting data*
 - Melewatkan dan mengkonversi data antar sistem (dan mungkin komponen yang terpisah dari sistem) dapat menimbulkan *error*.
- 8) *Race conditions*
 - Apabila dua *event* diproses akan maka salah satu akan diterima berdasarkan prioritas sampai pekerjaan selesai dengan baik, baru pekerjaan berikutnya. Bagaimanapun juga kadang-kadang event lain akan diproses terlebih dahulu dan dapat menghasilkan sesuatu yang tidak diharapkan atau tidak benar.

Gasal - TA.2022/2023

21



Defect (lanjutan)

- 9) *Load conditions*
 - Saat sistem dipaksa pada batas maksimum, masalah akan mulai muncul, seperti *arrays*, *overflow*, *diskfull*.
- 10) *Hardware*
 - Antar muka dengan suatu device mungkin tidak dapat beroperasi dengan benar pada suatu kondisi tertentu seperti *device unavailable*.
- 11) *Source and Version Control*
 - Program yang telah kadaluwarsa mungkin akan dapat digunakan lagi bila ada revisi untuk memperbaikinya.
- 12) *Documentation*
 - Pengguna tak dapat melihat operasi yang telah dideskripsikan dalam dokumen panduan.
- 13) *Testing errors*
 - Tester membuat kesalahan selama *testing* dan berpikir bahwa sistem berkelakuan tak benar.

Gasal - TA.2022/2023

22



Biaya-biaya testing

- Biaya-biaya yang berkaitan dengan *testing* dapat dilihat sebagaimana terdapat pada tabel, di bawah ini.

Biaya Pencegahan Defects	Biaya Penilaian dan Evaluasi Defects
Pelatihan staf	Review disain
Analisa kebutuhan	Inspeksi kode
Pembuatan protipe awal	<i>Glass-box testing</i>
Disain <i>fault-tolerant</i>	<i>Black-box testing</i>
<i>Defensive programming</i>	Pelatihan tester
Analisa kegunaan	<i>Beta testing</i>
Spesifikasi yang jelas	Otomatisasi tes
Dokumentasi internal yang akurat	<i>Usability testing</i>
Evaluasi terhadap reliabilitas dari alat bantu pengembangan (sebelum membelinya) atau komponen lain dari produk yang potensial.	<i>Pre-release out-box testing</i> oleh staf <i>customer service</i>

- Kebanyakan atribut biaya *testing* menghabiskan sekitar 25 % dari pengembangan. Beberapa proyek bahkan dapat mencapai sekitar 80% dari dana pengembangan.

Gasal - TA.2022/2023

23



Biaya-biaya defects

Terutama bagi pengembang software, biaya-biaya defects dapat berupa hal-hal sebagai berikut:

- Kesiapan dukungan teknisi.
- Persiapan buku panduan FAQ.
- Investigasi complain pelanggan.
- Ganti rugi dan mengambil kembali produk.
- Coding atau testing dari pembenahan bugs.
- Pengiriman dari produk yang telah diperbaiki.
- Penambahan biaya terhadap dukungan berbagai versi dari produk yang telah di release.
- Tugas Public Relation untuk menjelaskan review dari defects.
- Hilangnya pangsa jual.
- Hilangnya kepercayaan pelanggan.
- Pemberian potongan harga pada penjual agar mereka tetap menjual produk.
- Garansi.
- Kewajiban.
- Investigasi pemerintah.
- Pinalti.
- Dan biaya lain yang berkaitan dengan hukum.

Gasal - TA.2022/2023

24



Defect (lanjutan)

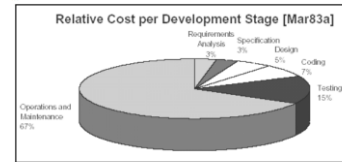
Biaya biaya Internal	Biaya-biaya Eksternal
Pembenahan <i>bugs</i>	Terbuangnya waktu
<i>Regression Testing</i>	Hilangnya data
Terbuangnya waktu <i>in-house user</i>	Kerugian bisnis
Terbuangnya waktu tester	Tercorengnya nama baik
Terbuangnya waktu penulis	Keluarnya karyawan akibat frustrasi
Terbuangnya waktu pemasaran	Hilangnya potesial presentasi
Terbuangnya promosi	Kegagalan pelanggan karena <i>software</i>
Biaya langsung dari keterlambatan pengiriman	Terjadinya <i>Failure</i> dari tugas-tugas yang hanya dapat dilakukan sekali
Biaya atas hilangnya kesempatan akibat keterlambatan pengiriman	Biaya penggantian produk
	Biaya rekonfigurasi sistem
	Biaya pembenahan <i>software</i>
	Biaya dukungan teknis
	Kecelakaan atau kematian

Gasal - TA.2022/2023

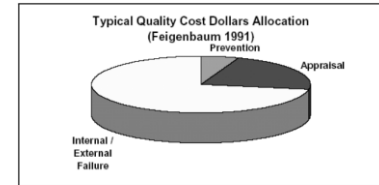
25



Defect (lanjutan)



Gambar 2.3 Biaya relatif pada tiap tahap pengembangan



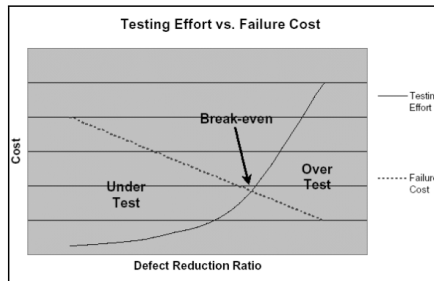
Gambar 2.4 Alokasi biaya (dalam dolar) kualitas secara umum.

Gasal - TA.2022/2023

26



Defect (lanjutan)



Gambar 2.5 Grafik hubungan usaha testing terhadap biaya *failure*.

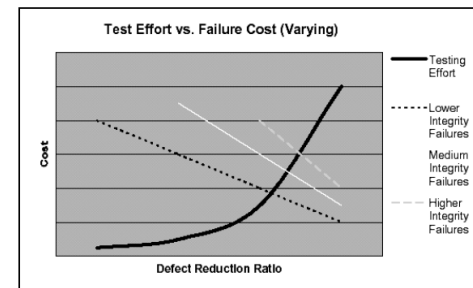
- Grafik diatas dapat dikorelasikan terhadap alokasi biaya, berdasar pada pengalaman dan estimasi, atau pengukuran internal dan analisa data.

Gasal - TA.2022/2023

27



Defect (lanjutan)



Gambar 2.6 Grafik hubungan usaha testing terhadap variasi biaya *failure*.

- Grafik diatas memperlihatkan bahwa semakin tinggi tingkat kritis suatu proyek, biaya *defect* juga meningkat. Hal ini mengindikasikan banyak sumber daya dapat dialokasikan untuk mencapai proporsi penghilangan defect yang lebih tinggi.

Gasal - TA.2022/2023

28



Tester, Defect, dan Aktivitas Testing

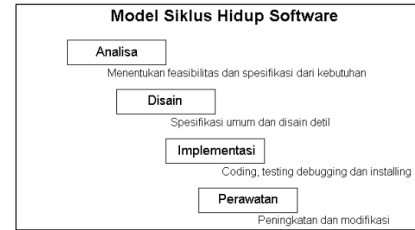
Siklus Hidup & Aktivitas Testing

Gasal - TA.2022/2023

29



Siklus Hidup & Aktivitas Testing



Gambar 2.7 Model siklus hidup software

- Kebanyakan organisasi menggunakan suatu standar untuk pengembangan *software* yang mendefinisikan suatu model siklus hidup (*life cycle model*), dan dibutuhkan tahap-tahap atau metodologi dalam pelaksanaannya.
- Semuanya memiliki tahap-tahap dasar yang sama, yaitu Analisa, Disain, Implementasi dan Perawatan,

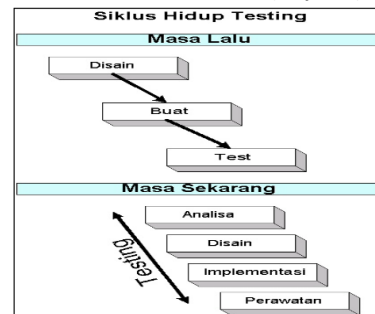
Gasal - TA.2022/2023

30



Siklus Hidup & Aktivitas Testing (lanjutan)

- Pada awalnya *testing* merupakan salah satu sub-fase dari fase pengembangan (*development*), setelah fase *coding*.
- Sistem didisain, dibangun dan kemudian dites dan di-*debug*.
- Sudut pandang *testing* yang tepat adalah dengan menyediakan suatu siklus hidup *testing* secara lengkap, yang merupakan suatu bagian dan menjadi satu kesatuan di dalam siklus hidup *software* secara keseluruhan.



Gambar 2.8 Siklus hidup testing

Gasal - TA.2022/2023

31



Siklus Hidup & Aktivitas Testing (lanjutan)

Apabila kita menggali lagi lebih dalam dari siklus hidup *testing*, tentang aktifitas apa saja yang terjadi di dalamnya, secara umum dan sederhana terdiri dari:

- Perencanaan
 - Rencana pendekatan umum
 - Menentukan obyektivitas *testing*
 - Memperjelas rencana umum
- Akuisisi
 - Disain tes
 - Menerapkan tes
- Pengukuran
 - Eksekusi tes
 - Cek terminasi
 - Evaluasi hasil

Gasal - TA.2022/2023

32



Siklus Hidup & Aktivitas *Testing* (lanjutan)

Sedangkan macam atau tipe *testing* secara umum ada tiga macam, secara urut berdasarkan waktu penggunaannya, adalah sebagai berikut:

- **Unit testing**
 - Testing penulisan kode-kode program dalam satuan unit terkecil secara individual.
- **System Testing**
 - Proses *testing* pada sistem terintegrasi untuk melakukan verifikasi bahwa sistem telah sesuai spesifikasi.
- **Acceptance Testing**
 - *Testing* formal yang dilakukan untuk menentukan apakah sistem telah memenuhi kriteria penerimaan dan memberdayakan pelanggan, untuk menentukan apakah sistem dapat diterima atau tidak.

Gasal - TA.2022/2023

33



Referensi

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.

Gasal - TA.2022/2023

34





Testing and Implementation



Session 05

PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>



- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhammadiyah.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkembang.
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.

Gasal - TA.2022/2023

2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.

Gasal - TA.2022/2023

3



Tujuan/Objectives (Session.05)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Test Case
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang *test case* sebagai salah satu dasar dari *testing*, disain *test case*, *white box testing*, dan *black box testing*.

Gasal - TA.2022/2023

4



Muhasabah

هَلْ جَزَاءُ الْإِحْسَانِ إِلَّا الْإِحْسَانُ
فَبِأَيِّ آلَاءِ رَبِّكُمَا تُكَذِّبِينَ

- Qs. Ar-Rahman, ayat 60 - 61
 - Sungguh, Tidak ada balasan untuk kebaikan selain kebaikan (pula).
 - Maka nikmat Tuhanmu yang manakah yang kamu dustakan ?

Gasal - TA.2022/2023

5



Testing and Implementation



Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion



Session 05

Test Case, White Box Testing, dan Black Box Testing



Test Case

Pendahuluan

Gasal - TA.2022/2023

7



Pendahuluan

- Test case merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi, ataupun hasil yang telah ditentukan sebelumnya.



- Adapun kegunaan dari test case ini, adalah sebagai berikut:
 - Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi – *Black Box Testing*.
 - Untuk melakukan testing kesesuaian suatu komponen terhadap disain – *White Box Testing*.

Gasal - TA.2022/2023

8



Pendahuluan (lanjutan)

- Variasi-variasi metode disain *test case* untuk *software* telah berkembang.
- Metode-metode saat ini menyediakan pengembang dengan pendekatan semantik terhadap *testing*.
- Yang lebih penting lagi, metode-metode ini menyediakan mekanisme yang dapat membantu untuk memastikan kelengkapan dari *testing* dan menyediakan kemungkinan tertinggi untuk mendapatkan *error* pada *software*.

Gasal - TA.2022/2023

9



Pendahuluan (lanjutan)

"Hanya ada satu aturan untuk mendisain test cases: disain test cases harus melingkupi semua fitur, namun jangan membuat terlalu banyak test cases."

Tsuneo Yamaura

Gasal - TA.2022/2023

10



Pendahuluan (lanjutan)

Tiap produk hasil rekayasa dapat di tes dalam dua cara:

- Pendekatan *Black Box Testing*.
 - Dengan berdasarkan pada fungsi yang dispesifikasikan dari produk, tes dapat dilakukan dengan mendemonstrasikan tiap fungsi telah beroperasi secara penuh sesuai dengan yang diharapkan, dan sementara itu, pada saat yang bersamaan, dilakukan pencarian error pada tiap fungsi.
- Pendekatan *White Box Testing*.
 - Dengan mengetahui operasi internal dari produk, tes dapat dilakukan untuk memastikan semua komponen berjalan sebagaimana mestinya, operasi internal berlaku berdasarkan pada spesifikasi dan semua komponen internal telah cukup diperiksa.

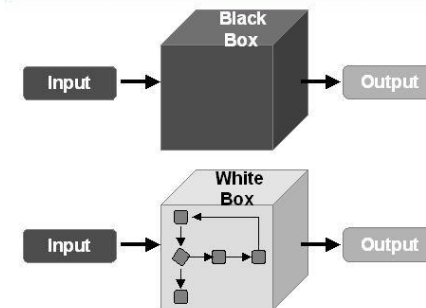
Gasal - TA.2022/2023

11



Pendahuluan (lanjutan)

Comparison among Black-Box & White-Box Tests



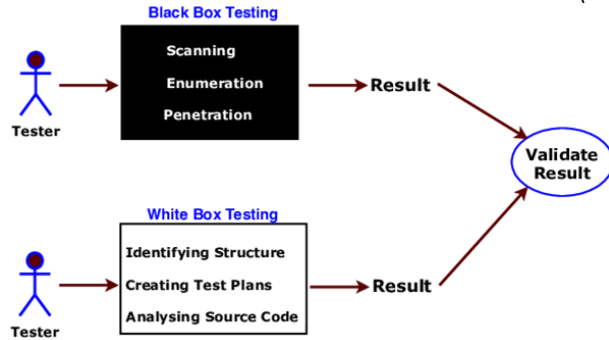
www.softwaretestinggenuis.com

Gasal - TA.2022/2023

12



Pendahuluan (lanjutan)



Pendahuluan (lanjutan)



- Hal yang perlu diingat, bahwa *testing* tidak dapat membuktikan kebenaran semua kemungkinan eksekusi dari suatu program.
- Namun dapat didekati dengan melakukan perencanaan dan disain *test case* yang baik, sehingga dapat memberikan jaminan efektifitas dari *software* sampai pada tingkat tertentu sesuai dengan yang diharapkan.



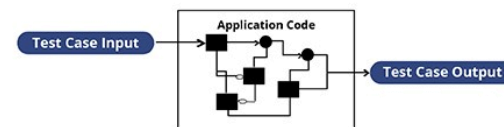
Test Case White Box Testing



White Box Testing

- *White Box Testing*, biasa disebut juga *glass box testing* atau *clear box testing*, adalah suatu metode disain *test case* yang menggunakan struktur kendali dari disain prosedural.

WHITE BOX TESTING APPROACH





White Box Testing (lanjutan)

- Metode desain *test case* dari *white box testing* ini dapat menjamin bahwa:
 - Semua jalur (*path*) yang independen/terpisah dapat dites setidaknya sekali tes.
 - Semua logika keputusan dapat dites dengan jalur yang salah dan atau jalur yang benar.
 - Semua *loop* dapat dites terhadap batasannya dan ikatan operasionalnya.
 - Semua struktur internal data dapat dites untuk memastikan validitasnya.
- Mengapa melakukan white box testing bilamana black box testing berfungsi untuk testing pemenuhan terhadap kebutuhan / spesifikasi?
 - Kesalahan logika dan asumsi yang tidak benar kebanyakan dilakukan ketika coding untuk "kasus tertentu". Dibutuhkan kepastian bahwa eksekusi jalur ini telah dites.
 - Asumsi bahwa adanya kemungkinan terhadap eksekusi jalur yang tidak benar. Dengan white box testing dapat ditemukan kesalahan ini
 - Kesalahan penulisan yang acak. Seperti berada pada jalur logika yang membingungkan pada jalur normal.

Gasal - TA.2022/2023

17



White Box Testing (lanjutan)

- Teknik-teknik pengujian menggunakan pendekatan white box testing, adalah sebagai berikut:
 - Cakupan Pernyataan (*statement*), Cabang (*branch*), dan Jalur (*path*).
 - Basis Path Testing*.
 - Cyclomatic Complexity*.
 - Graph Matrix*.
 - Control Structure Testing*.
 - Condition Testing*.
 - Data Flow Testing*.
 - Loop Testing*.
 - Lines of Code*.
 - Halstead's Metrics*.

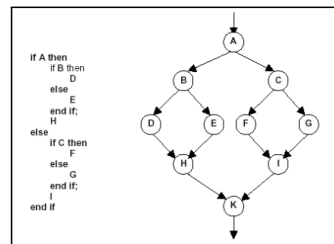
Gasal - TA.2022/2023

18



White Box Testing (lanjutan)

- Cakupan Pernyataan (*statement*), Cabang (*branch*), dan Jalur (*path*)
 - Cakupan pernyataan, cabang dan jalur adalah suatu teknik *white box testing* yang menggunakan alur logika dari program untuk membuat *test cases*.
 - Yang dimaksud dengan alur logika adalah cara dimana suatu bagian dari program tertentu dieksekusi saat menjalankan program.
 - Alur logika suatu program dapat direpresentasikan dengan *flow graph*.

Gambar 3.1 Contoh *flow graph* dari suatu kode program.

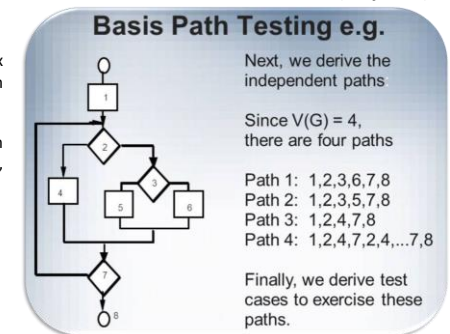
Gasal - TA.2022/2023

19



White Box Testing (lanjutan)

- Basis Path Testing*.
 - Merupakan teknik white box testing yang dikenalkan oleh Tom McCabe.
 - Basis path testing* hadir dalam 2 bentuk, yaitu: *Zero Path*, *One Path*.



Gasal - TA.2022/2023

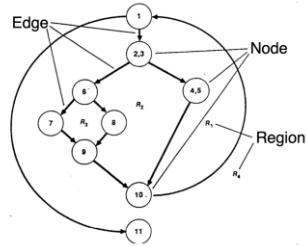
20



White Box Testing (lanjutan)

c) Cyclomatic Complexity.

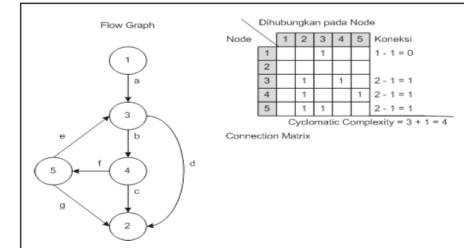
- Adalah pengukuran *software* yang memberikan pengukuran kuantitatif dari kompleksitas logika program.
- Pada konteks metode *basis path testing*, nilai yang dihitung bagi *cyclomatic complexity* menentukan jumlah jalur-jalur yang independen dalam kumpulan basis suatu program dan memberikan jumlah tes minimal yang harus dilakukan untuk memastikan bahwa semua pernyataan telah dieksekusi sekurangnya satu kali.



White Box Testing (lanjutan)

d) Graph Matrix.

- Adalah matrik berbentuk segi empat sama sisi, dimana jumlah baris dan kolom sama dengan jumlah *node*, dan identifikasi baris dan kolom sama dengan identifikasi *node*, serta isi data adalah keberadaan penghubung antar *node* (*edges*).



Gambar 3.11 Konversi flow graph ke graph matrix



White Box Testing (lanjutan)

e) Control Structure Testing.

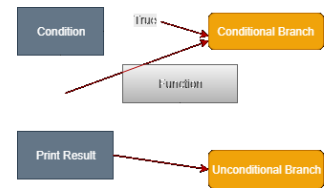
- Control structure testing meliputi teknik-teknik berikut:
 - Testing kondisi (*condition testing*).
 - Testing alur data (*data flow testing*).
 - Testing loop (*loop testing*).



White Box Testing (lanjutan)

f) Condition Testing.

- Suatu metode disain *test case* yang memeriksa kondisi logika yang terdapat pada modul program.
- Metode tes kondisi berfokus pada *testing* tiap kondisi dalam program.
- Tujuan tes kondisi disamping untuk mendeteksi *error* dari kondisi program juga untuk kesalahan lainnya dari program.

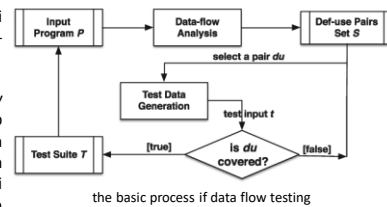




White Box Testing (lanjutan)

g) Data Flow Testing.

- Metode data flow testing memilih jalur program berdasarkan pada lokasi dari definisi dan penggunaan variabel-variabel pada program.
- Sebagai ilustrasi pendekatan *data flow testing*, diasumsikan bahwa tiap pernyataan dalam suatu program ditandai dengan suatu penomoran pernyataan yang unik sifatnya, sebagai identitas dari tiap pernyataan tersebut, dimana tiap fungsi tidak memodifikasi parameter atau variabel globalnya.



Gasal - TA.2022/2023

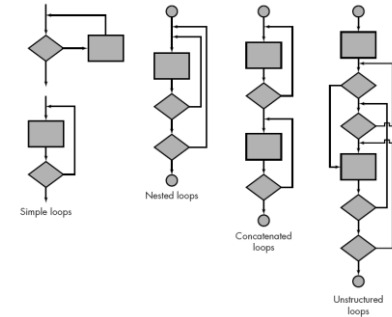
25



White Box Testing (lanjutan)

h) Loop Testing.

- Loop testing* merupakan suatu teknik *white box testing* yang berfokus pada validitas konstruksi *loop* secara eksklusif.
- Terdapat 4 (empat) tipe dari loop testing, yaitu:
 - Simple Loops
 - Nested Loops
 - Concatenated Loops
 - Unstructured Loops



Gasal - TA.2022/2023

26



White Box Testing (lanjutan)

i) Lines of Code.

- Yaitu merupakan pengukuran sederhana dengan cara menghitung jumlah baris kode dalam program, dan menggunakan perhitungan ini untuk mengukur tingkat kompleksitas.
- Berdasarkan penelitian dan pengalaman di lapangan, bahwa:
 - Program kecil mempunyai *error* rata-rata 1,3% sampai 1,8%.
 - Program besar mempunyai kenaikan *error* rata-rata dari 2,7% sampai 3,2%.

Gasal - TA.2022/2023

27



j) Halstead's Metric.

- Merupakan pengukuran yang berdasarkan pada penggunaan operator-operator (seperti *keyword*), dan operan-operan (seperti nama variabel, obyek *database*) yang ada didalam suatu program.

```
void sort ( int *a, int n ) {
    int i, j, t;

    if ( n < 2 ) return;
    for ( i=0 ; i < n-1; i++ ) {
        for ( j=i+1 ; j < n ; j++ ) {
            if ( a[i] > a[j] ) {
                t = a[i];
                a[i] = a[j];
                a[j] = t;
            }
        }
    }
}
```

- Ignore the function definition
- Count operators and operands

3	<	3	{	1	0
5	=	3	}	2	1
1	>	1	4	1	9
1	-	2	++	6	a
2	.	2	for	8	i
9	;	2	if	7	j
4	(1	int	3	n
4)	1	return	3	t
6	()				

	Total	Unique
Operators	N1 = 50	n1 = 17
Operands	N2 = 30	n2 = 7

n1 = jumlah operator yang unik (distinct) dalam program.
 n2 = jumlah operan yang unik (distinct) dalam program.
 Panjang program: $H = n1 \log_2 n1 + n2 \log_2 n2$
 N1 = perhitungan jumlah keseluruhan operator program.
 N2 = perhitungan jumlah keseluruhan operan program.
 Prediksi bug: $B = (N1 + N2) \log_2 (n1 + n2) / 3000$

$$V = 80 \log_2(24) \approx 392$$



White Box Testing (lanjutan)

Halstead's Metrics

■ Program Complexity

- **Volume:** $V = N \log_2 n$
 - Number of bits to provide a unique designator for each of the n items in the program vocabulary.

□ **Difficulty**

$$D = \frac{n_1}{2} \times \frac{N_2}{n_2}$$

□ **Program effort:** $E = D * V$

- This is a good measure of program comprehension (understandability)

Gasal - TA.2022/2023

29



Test Case

Black Box Testing

Gasal - TA.2022/2023

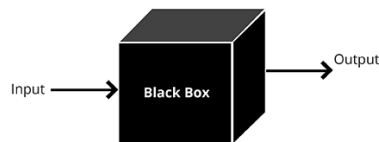
30



Black Box Testing

- Black box testing, dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites.
- Biasanya disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

BLACK BOX TESTING APPROACH



Gasal - TA.2022/2023

31



Black Box Testing (lanjutan)

- *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
- Dengan adanya *black box testing*, perekrutan *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program.
- *Black box testing* bukan teknik alternatif daripada *white box testing*, melainkan merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.

Gasal - TA.2022/2023

32



Black Box Testing (lanjutan)

- Kategori *error* yang akan diketahui melalui metode *black box testing*, adalah:
 - Fungsi yang hilang atau tak benar
 - *Error* dari antar-muka
 - *Error* dari struktur data atau akses eksternal database
 - *Error* dari kinerja atau tingkah laku
 - *Error* dari inisialisasi dan terminasi
- Tidak seperti metode *white box testing*, yang memang digunakan pada awal proses *testing*. *Black box testing* digunakan pada tahap akhir dan berfokus pada domain informasi.

Gasal - TA.2022/2023

33



Black Box Testing (lanjutan)

- *Tes case* yang didisain melalui metode *black box testing*, biasanya adalah untuk menjawab pertanyaan-pertanyaan sebagai berikut:
 - Bagaimana validasi fungsi yang akan dites ?
 - Bagaimana tingkah laku dan kinerja sistem dites ?
 - Kategori masukan apa saja yang bagus digunakan untuk *test cases* ?
 - Apakah sebagian sistem sensitif terhadap suatu nilai masukan tertentu ?
 - Bagaimana batasan suatu kategori masukan ditetapkan ?
 - Sistem mempunyai toleransi jenjang dan volume data apa saja ?
 - Apa saja akibat dari kombinasi data tertentu yang akan terjadi pada operasi sistem ?

Gasal - TA.2022/2023

34



Black Box Testing (lanjutan)

- Terdapat banyak jenis teknik disain tes yang dapat dipilih berdasarkan pada tipe *testing* yang akan digunakan, diantaranya adalah:
 - Equivalence Partitioning*
 - Boundary Value Analysis*
 - State Transitions Testing*
 - Cause-Effect Graphing*
 - Functional Analysis*
 - Use Cases*

Gasal - TA.2022/2023

35

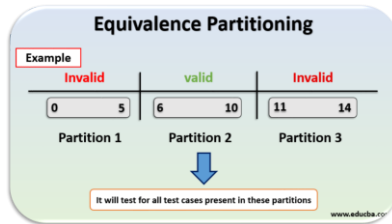


Black Box Testing (lanjutan)

- Equivalence Partitioning*.
 - Adalah metode *black box testing* yang membagi domain masukan dari suatu program kedalam kelas-kelas data, dimana *test cases* dapat diturunkan.
 - *Equivalence partitioning* berdasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang akan diperlakukan sama (*ekuivalen*) oleh komponen tersebut.
 - Sehingga dapat diasumsikan sebagai representasi dari semua nilai dalam partisi.

Gasal - TA.2022/2023

36



Black Box Testing (lanjutan)

AGE *Accepts value 18 to 56

EQUIVALENCE PARTITIONING		
Invalid	Valid	Invalid
<=17	18-56	>=57

MOBILE NUMBER *Must be 10 digits

EQUIVALENCE PARTITIONING		
Invalid	Valid	Invalid
987654321	9876543210	98765432109



Black Box Testing (lanjutan)

b) Boundary Value Analysis.

- *Boundary value analysis* adalah suatu teknik desain *test cases* yang berguna untuk melakukan pengujian terhadap nilai sekitar dari pusat domain masukan.
- Teknik *boundary value analysis* merupakan komplemen dari teknik *equivalence partitioning*.
- *Test cases* dilakukan untuk menguji nilai-nilai di kedua sisi dari batasan.



Black Box Testing (lanjutan)

AGE *Accepts value 18 to 56

BOUNDARY VALUE ANALYSIS		
Invalid (min -1)	Valid (min, +min, -max, max)	Invalid (max +1)
17	18, 19, 55, 56	57

Name *Accepts characters length (6 - 12)

BOUNDARY VALUE ANALYSIS		
Invalid (min -1)	Valid (min, +min, -max, max)	Invalid (max +1)
5 characters	6, 7, 11, 12 characters	13 characters



Black Box Testing (lanjutan)

c) Cause-Effect Graphing.

- Merupakan teknik desain *test cases* yang menggambarkan logika dari kondisi terhadap aksi yang dilakukan.
- Terdapat empat langkah, yaitu:
 - 1) Tiap penyebab (kondisi masukan) dan akibat (aksi) yang ada pada suatu modul didaftarkan.
 - 2) Membuat gambar sebab-akibat (*cause-effect graph*).
 - 3) Konversikan gambar/*cause-effect graph* ke tabel keputusan.
 - 4) Aturan-aturan yang ada di tabel keputusan di konversikan ke *test cases*.



Black Box Testing (lanjutan)

1

2

Simbol logika standar yang digunakan adalah AND (*), OR (v) dan NOT (!).

Garis mengkilap menunjukkan masukan-masukan yang dipengaruhi oleh simbol dimana ada lebih dari dua masukan yang termasuk di dalamnya.

3

Aturan	1	2	3	4	5	6	7	8
C1: transaksi jurnal kredit baru	F	F	F	F	T	T	T	T
C2: transaksi jurnal penarikan baru, tapi dengan batas penarikan tertentu.	F	F	T	T	F	F	T	T
C3: jurnal yang mempunyai pos penarikan.	F	T	F	T	F	T	F	T
A1: pemrosesan debit.	F	F	T	T	T	T	-	-
A2: penundaan jurnal penarikan	F	T	F	F	F	F	-	-
A3: pengimanan surat.	T	T	T	T	T	T	-	-

4

Test case	Tipe perkiraan	Sebab (cause) Batas penarikan	Nilai perkiraan saat ini	Jumlah debit	Akibat (effect) Nilai perkiraan baru	Kode aksi
1	kredit	100	-70	50	-70	L
2	tunda	1500	420	2000	420	S & L
3	kredit	250	650	900	-150	D & L
4	tunda	750	-500	200	-700	D & L
5	kredit	1000	2100	1200	900	D
6	tunda	500	250	150	100	D & L

Gasal - TA.2022/2023

41



Black Box Testing (lanjutan)

d) State Transition Testing.

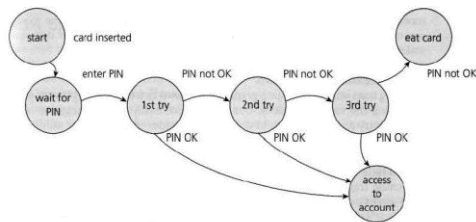
- State transition testing menggunakan model sistem yang terdiri dari:
 - Status yang terdapat di dalam program.
 - Transisi antar status-status tersebut.
 - Kejadian yang merupakan sebab dari transisi-transisi tersebut.
 - Aksi-aksi yang akan dihasilkan.
- Model umumnya direpresentasikan dalam bentuk *state transition diagram*.
- Test cases didisain untuk memeriksa validitas transisi antar status.
- Test cases tambahan juga akan didisain untuk *testing* terhadap transisi-transisi yang tidak termasuk dan tidak dispesifikasikan.

Gasal - TA.2022/2023

42



Black Box Testing State Transition Testing (lanjutan)



State Diagram for PIN Entry

Gasal - TA.2022/2023

43



Black Box Testing (lanjutan)

e) Functional Analysis.

- Teknik yang paling banyak dipakai untuk mengidentifikasi *test cases*.
- Dasar utama pemikirannya adalah melakukan analisa terhadap fungsi-fungsi yang terdapat pada suatu sistem, apakah fungsi-fungsi tersebut mempunyai kinerja sebagaimana yang diharapkan atau dispesifikasikan.
- Teknik ini membutuhkan jawaban atas pertanyaan sebagai berikut:
 - Fungsi utama apa saja yang harus ada pada sistem ?
 - Berdasarkan fungsi-fungsi yang ada, keluaran apa saja yang harus dihasilkan untuk membuktikan bahwa fungsi tersebut telah dipenuhi ?
 - Apa saja masukan dan inisialisasi yang dibutuhkan sistem untuk menghasilkan keluaran pada tiap fungsi yang bersangkutan ?

Gasal - TA.2022/2023

44



Black Box Testing (lanjutan)

- Karena itu, pendekatan pertama adalah mendapatkan informasi spesifikasi dari fungsi yang diharapkan dapat disediakan oleh sistem.
- Informasi ini umumnya terdapat pada dokumentasi spesifikasi fungsional sistem.
- Bagaimana bila tidak ada dokumentasi spesifikasi fungsional sistem ?, maka pengguna harus membuat spesifikasi fungsional.
- Proses pembuatan dapat dimulai dari struktur menu program atau buku panduan untuk pengguna (misal *Help File* atau *User Manual*).

Gasal - TA.2022/2023

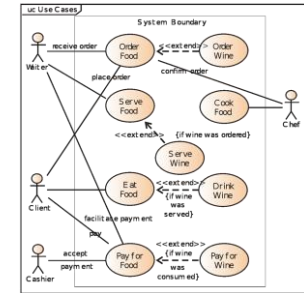
45



Black Box Testing (lanjutan)

f) Use Cases.

- Suatu *use case* adalah suatu sekuensial aksi yang dilakukan oleh sistem, yang akan secara bersama-sama memproduksi hasil yang dibutuhkan oleh pengguna sistem.
- *Uses cases* mendefinisikan alur proses sepanjang sistem yang berbasis pada kegunaan sebagaimana yang biasa dilakukan (secara manual).
- *Use cases* juga memasukkan interaksi atau fitur-fitur dan fungsi-fungsi yang berbeda dari sistem. Oleh karena alasan ini, maka tes yang dibuat dari *use cases* akan membantu dalam menemukan *errors* dari integrasi.



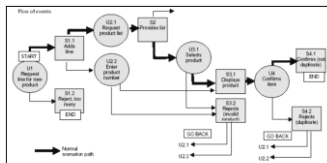
Gasal - TA.2022/2023

46



Black Box Testing (lanjutan)

- Setiap *use cases* memiliki:
 - *Preconditions*
 - *Postconditions*
 - *Flow of events*
- *Use cases* dan *test cases* akan bekerja dengan baik dalam dua cara, yaitu:
 - Jika *use cases* dari sistem komplit, akurat, dan jelas, maka pembuatan *test cases* dapat dilakukan secara langsung.
 - Jika *use cases* tidak dalam kondisi yang baik, maka pembuatan *test cases* akan membantu didalam melakukan *debug* terhadap *test cases*.



Gambar 3.36 Use case untuk pemilihan produk pada pesanan pembelian.

47



Referensi

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.
- Romeo, 2003. *Testing dan Implementasi Sistem*, Edisi Pertama, STIKOM.

Gasal - TA.2022/2023

48

HAKIKAT SEDEKAH

Dari Abu Hurairah radhiyallahu'anhu,
Rasulullah shallallahu 'alaihi wasallam bersabda,

مَا نَقَصَتْ صَدَقَةٌ مِنْ مَالٍ

**"Sedekah itu tidak akan
mengurangi harta."**

(Hadits Riwayat Muslim No. 4689)



Terima Kasih

Gasal - TA.2022/2023

49



Testing and Implementation



Session 06

PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>



- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhmadiyah.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkembang.
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.

Gasal - TA.2022/2023

2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **20%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **30%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.

Gasal - TA.2022/2023

3



Tujuan/Objectives (Session.06)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Teknik Testing Lain
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang teknik lain dari *disain test cases*.

Gasal - TA.2022/2023

4



Muhasabah

وَلَا تَهِنُوا وَلَا تَحْزَنُوا وَأَنْتُمْ الْأَعْلَوْنَ إِنْ كُنْتُمْ مُؤْمِنِينَ

- Qs. Ali-Imran, ayat 139
 - Dan janganlah kamu (merasa) lemah, dan jangan (pula) bersedih hati, sebab kamu paling tinggi (derajatnya), jika kamu orang beriman.

Gasal - TA.2022/2023

5



Teknik *Testing* Lainnya

Pendahuluan

Gasal - TA.2022/2023

7



Testing and Implementation



Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion

Session 06

Teknik *testing* lainnya



Teknik *Testing* Lainnya

- Selain teknik *testing* yang telah kita pelajari pada pertemuan sebelumnya, yaitu *white box testing* dan *black box testing* beserta metode-metodenya, maka ada banyak lagi teknik lain dari *disain test cases*.
- Teknik lain dari *disain test cases*, diantaranya adalah sebagai berikut:
 - a) *Comparison Testing*
 - b) *Test Factor Analysis*
 - c) *Risk Based Testing*
 - d) *Syntax Testing*
 - e) *Cross-Functional Testing*
 - f) *Operational Profiling*
 - g) *Table & Array Testing*



Gasal - TA.2022/2023

8



Teknik *Testing* Lainnya (lanjutan)

a) *Comparison Testing*.

- *Comparison testing* atau disebut *back-to-back testing*, biasa digunakan pada beberapa aplikasi yang mempunyai kebutuhan terhadap reliabilitas yang amat penting/kritis. Seperti misalnya sistem rem pada mobil, atau sistem navigasi pada pesawat terbang.
- Tes dilakukan pada tiap versi dengan data tes yang sama secara paralel dan *real time* untuk memastikan konsistensi (keluaran yang dihasilkan dari tiap versi identik).
- Bila keluaran berbeda atau terjadi *defect* pada satu atau lebih versi aplikasi, masing-masing diinvestigasi untuk menentukan dimana letak kesalahannya, dan versi aplikasi mana yang melakukan kesalahan.



Gasal - TA.2022/2023



Teknik *Testing* Lainnya (lanjutan)

b) *Test Factor Analysis*.

- *Test factor analysis* adalah suatu proses identifikasi faktor-faktor tes (variabel atau kondisi yang relevan terhadap sistem yang dites, dan dapat bervariasi selama proses *testing*), dan pilihan (*options*), kemudian dengan menggunakan kesamaan dan variasinya untuk menentukan kombinasi pilihan dari faktor yang akan dites.

Minimum testing: $(\sum Mi - N + 1)$, Mi = Jumlah opsi tiap faktor tes, N = Jumlah faktor-faktor tes

- Contoh:
Faktor 1 = konfigurasi komputer - sistem operasi Win 95/NT (2 opsi)
Faktor 2 = konfigurasi komputer - hard disk 1,5 GB / 2 GB (2 opsi)
maka:
Minimum testing = $4 - (2 + 1) = 1$

Gasal - TA.2022/2023

10



Teknik *Testing* Lainnya (lanjutan)

c) *Risk Based Testing*.

- *Risk based testing*, merupakan metode untuk menentukan prioritas dalam mendisain *test cases*.
- Efektifitas Test = Jumlah *defect* ditemukan / estimasi jumlah *defect*.

$$\text{Efektivitas Test} = \frac{\text{Jumlah defect ditemukan}}{\text{Estimasi jumlah defect}}$$

Gasal - TA.2022/2023

11



Teknik *Testing* Lainnya (lanjutan)

- Faktor resiko secara garis besar yang menentukan prioritas kebutuhan sistem / *test cases* adalah:

- Visibilitas dan akibat pada pelanggan.
- Resiko operasi bisnis.
- Sejarah terjadinya *defect* area yang baru / dimodifikasi.
- Kontinuitas bisnis.
- Tingkat kompleksitas pengembangan.
- Kondisi yang diharapkan (*positive testing*).
- Kondisi yang tak diharapkan (*negative testing*).
- Tingkat prioritas *testing* dari pengembang atau kontraktor.
- Tingkat kepercayaan pengembang.
- Lawan dari kepercayaan pengembang.
- Observasi tester terhadap kredibilitas pengembang.
- Keinginan dan perasaan dari tester dan pengguna.
- Cakupan.



Gasal - TA.2022/2023

12



Teknik *Testing* Lainnya (lanjutan)

d) *Syntax Testing*.

- *Syntax testing* menggunakan model sintaksis masukan sistem yang didisain secara formal, yang merupakan suatu cara penggunaan dan penggabungan kata-kata yang membentuk suatu frase.
- *Syntax testing* sangat berguna untuk sistem yang menggunakan metode baris-baris perintah untuk cara pengaksesannya.

```

test cases
syntax

module syntax/identifiers
language Tiger start symbol Id

test single lower case [[x]] parse succeeds
test single upper case [[X]] parse succeeds
test single digit      [[1]] parse fails

test single lc digit  [[x1]] parse succeeds
test single digit lc  [[1x]] parse fails
test single uc digit  [[X1]] parse succeeds
test single digit uc  [[1X]] parse fails
test double digit     [[11]] parse fails
  
```

Gasal - TA.2022/2023

13



Teknik *Testing* Lainnya (lanjutan)

e) *Cross-Functional Testing*.

- *Cross-functional testing* menggunakan matrik interaksi antar fitur dari sistem.
- Axis dari matrik X dan Y merupakan fitur dari sistem, dan sel mengindikasikan komponen yang di-*update* oleh satu fitur dan kemudian digunakan oleh lainnya.
- Tes didisain dari matrik untuk memeriksa interaksi antar fitur yang telah didefinisikan di dalam matrik tersebut.
- Interaksi dapat terjadi dalam dua tipe dependensi, yaitu: secara langsung dengan lewatnya pesan-pesan atau transaksi-transaksi diantara fitur-fitur, atau secara tidak langsung dengan adanya data umum yang dipakai bersama oleh fitur-fitur.
- Tiap dependensi dapat menyebabkan suatu perubahan status dan tingkah laku dari fitur yang terkait.

Gasal - TA.2022/2023

14



Teknik *Testing* Lainnya (lanjutan)

Berikut ini diberikan sekilas contoh *cross-functional testing*:

Fitur	F1	F2	F3
F1	-	C1	M2
F2	-	-	M3
F3	M1	-	-

Notasi-notasi dari sel-sel tabel di atas mempunyai arti bahwa fitur berinteraksi sebagai berikut:

- Fitur F1 meng-*update* hitungan C1 yang digunakan oleh fitur F2.
- Fitur F2 tidak meng-*update* hitungan C1.
- Fitur F3 mengirim pesan M1 ke F1.
- Fitur F1 mengirim pesan M2 ke F3.
- Fitur F2 mengirim pesan M3 ke F3.

Gasal - TA.2022/2023

15



Teknik *Testing* Lainnya (lanjutan)

f) *Operational Profiling*.

- *Operational profiling* (profil operasional) menjelaskan siapa pengguna, fitur-fitur apa yang digunakan, frekuensi penggunaan dan kondisi dimana fitur digunakan, lingkungan *hardware* dan *software* yang digunakan, serta prosedur pengoperasian dan mekanisme bagaimana fitur digunakan.
- Profil dari penggunaan, dapat diestimasi berdasarkan pada pola kerja, atau diekstrapolasi dari pengukuran aktual dari penggunaan yang ada.

Gasal - TA.2022/2023

16



Teknik *Testing* Lainnya (lanjutan)

g) *Table & Array Testing*.

- *Table* adalah suatu bentuk data yang biasanya berada di luar program, sedangkan *array* berada di dalam program, yang digunakan sebagai transfer data dari *table* (eksternal) untuk digunakan di dalam program.
- *Table* mempunyai dua bentuk utama, yaitu: sekuensial dan ber-indeks (*keyed / indexed*).

Gasal - TA.2022/2023

17



Teknik *Testing* Lainnya (lanjutan)

g) *Table & Array Testing*.

- *Table* adalah suatu bentuk data yang biasanya berada di luar program, sedangkan *array* berada di dalam program, yang digunakan sebagai transfer data dari *table* (eksternal) untuk digunakan di dalam program.
- *Table* mempunyai dua bentuk utama, yaitu: sekuensial dan ber-indeks (*keyed / indexed*).

Gasal - TA.2022/2023

18



Teknik *Testing* Lainnya (lanjutan)

- Tes penggunaan secara sekuensial dari *table* meliputi:
 - Menghapus data dari suatu *table* kosong.
 - Membaca data dari suatu *table* kosong.
 - Menambahkan data ke *table* yang penuh.
 - Menghapus satu data dari suatu *table* yang memiliki satu data.
 - Membaca data terakhir.
 - Membaca data berikutnya setelah data yang terakhir.
 - Menyimpan data baru setelah data terakhir muncul.
 - Menjalankan data secara sekuensial pada keseluruhan *table*.
 - Menyisipkan data di luar sekuensial data.
 - Menyisipkan data yang sama.

Gasal - TA.2022/2023

19



Teknik *Testing* Lainnya (lanjutan)

- Sedangkan tes penggunaan secara *keyed/indexed* dari *table* meliputi:
 - Menghapus logika data yang pertama.
 - Menghapus logika data yang ditengah.
 - Menghapus logika data yang terakhir.
 - Menambahkan logika baru di tengah data.
 - Menambahkan logika baru di awal data.
 - Menambahkan data yang sama.
 - Menambahkan suatu data dengan *key* yang salah.
 - Mengubah *key field* pada data yang telah ada.
 - Menghapus data yang tidak ada.
 - *Update* dan menulis kembali data yang telah ada.
 - Membaca data dari *file* yang tidak dibuka atau tidak ada.
 - Menuliskan data ke *file* yang berstatus hanya dapat dibaca.
 - Menuliskan data ke *file* yang versinya salah.

Gasal - TA.2022/2023

20



Teknik *Testing* Lainnya (lanjutan)

- Dari sekian banyak teknik disain *test cases* sebagaimana telah dijabarkan diatas, dan untuk lebih memudahkan didalam memahami penggunaannya, maka diberikan tabel sebagai ilustrasi penggunaan, daftar berdasarkan pada area penggunaannya dalam *testing* suatu sistem sebagai berikut:

Area aplikasi	Teknik Tes
Diskripsi fungsi dan fitur.	<i>Functional Analysis</i>
Spesifikasi logika keputusan (misal <i>If-Then-Else</i>).	<i>Black Box Path Analysis</i>
<i>Unit test code</i> .	<i>White Box Path Analysis</i>
Masukan (<i>GUI, queries, transaksi</i>).	<i>Boundary Value Analysis</i>
Data tersimpan.	<i>Table / Array Testing</i>
Sejumlah besar populasi item yang sama (<i>data fields of records</i>).	<i>Statistical Sampling</i>
Sejumlah besar variabel yang mempengaruhi <i>testing</i> .	<i>Test Factor Analysis</i>
Frekuensi penggunaan pola.	<i>Operational Profiling</i>
Pola resiko.	<i>Risk Assessment</i>
Perubahan sistem yang ada.	<i>Localized Test</i> <i>Volume Test</i> <i>Regression Test</i>

21



Referensi

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.
- Romeo, 2003. *Testing dan Implementasi Sistem*, Edisi Pertama, STIKOM.

Gasal - TA.2022/2023

22

Hidup sehat ala Rasulullah


Terima Kasih

Rasulullah *shallallahu 'alaihi wa sallam* bersabda,
"Tidak ada tempat yang lebih jelek daripada
memenuhi perut keturunan Adam.
Cukup keturunan Adam mengonsumsi yang dapat
menegakkan tulangnya. Kalau memang menjadi
suatu keharusan untuk diisi,
maka sepertiga untuk makannya,
sepertiga untuk minumannya,
dan sepertiga untuk nafasnya."


(HR. Ahmad, 4:132; Tirmidzi, no. 2380;
Ibnu Majah, no. 3349.
Syaikh Syaib Al-Arnauth mengatakan
bahwa perawi hadis ini *tsiqqah*, terpercaya).



Gasal - TA.2022/2023



Testing and Implementation




Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion

Session 08




PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>

- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhmadiyah.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkembang.
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.


Gasal - TA.2022/2023 2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.


Gasal - TA.2022/2023 3



Tujuan/Objectives (Session.08)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Strategi Testing
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang pendekatan-pendekatan yang dapat digunakan dalam menentukan strategi *testing software*, isu-isu strategi *testing* yaitu *unit testing, integration testing, validation testing, system testing*.


Gasal - TA.2022/2023 4


 **Muhasabah**

﴿وَاعْبُدُوا اللَّهَ وَلَا تُشْرِكُوا بِهِ شَيْئًا وَبِالْوَالِدَيْنِ إِحْسَانًا وَبِذِي الْقُرْبَىٰ وَالْيَتَامَىٰ وَالْمَسْكِينِ وَالْجَارِ ذِي الْقُرْبَىٰ وَالْجَارِ الْجُنُبِ وَالصَّاحِبِ بِالْجَنبِ وَابْنِ السَّبِيلِ وَمَا مَلَكَتْ أَيْمَانُكُمْ إِنَّ اللَّهَ لَا يُحِبُّ مَنْ كَانَ مُخْتَالًا فَخُورًا﴾

- Qs. An-Nisa, ayat 36
 - Dan sembahlah Allah dan janganlah kamu mempersekutukan-Nya dengan sesuatu apa pun.
 - Dan berbuat-baiklah kepada kedua orang tua, karib-kerabat, anak-anak yatim, orang-orang miskin, tetangga dekat dan tetangga jauh, teman sejawat, ibnu sabil dan hamba sahaya yang kamu miliki.
 - Sungguh, Allah tidak menyukai orang yang sombong dan membanggakan diri..

Gasal - TA.2022/2023 5

 **Testing and Implementation**




Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion


● ● ●
Session 08
Strategi *Testing*

 **Strategi *Testing***

Pendahuluan



Gasal - TA.2022/2023 7

 **Pendahuluan**

- *Testing* adalah suatu kumpulan aktifitas yang dapat direncanakan lebih lanjut dan dilakukan secara sistematis.
- Suatu strategi *testing software* harus cukup fleksibel untuk dapat mengakomodasi kustomisasi pendekatan *testing*.
- Pada saat yang bersamaan, harus juga cukup konsisten dan tegas agar dapat melakukan perencanaan yang masuk akal dan dapat melakukan manajemen perkembangan kinerja proyek.
- Setiap strategi *testing* harus menjadi satu kesatuan dengan perencanaan tes, *disain test cases*, eksekusi tes, dan pengumpulan serta evaluasi data hasil *testing*.

Gasal - TA.2022/2023 8

Pendahuluan
(lanjutan)



“Seperti kematian dan pajak, *Testing* sangat tidak menyenangkan dan tidak dapat dihindari”
Ed Yourdon

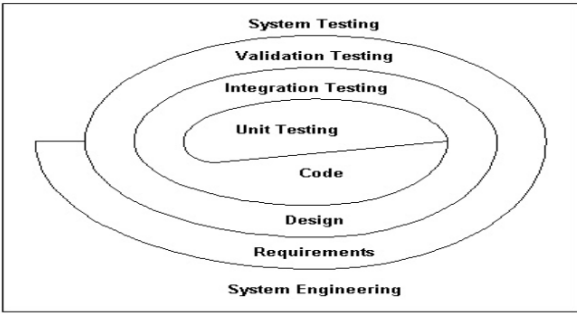
Gasal - TA.2022/2023 9

Pendahuluan
(lanjutan)

- Sejumlah strategi *testing software* diadakan untuk menyediakan kerangka *testing* bagi pengembang *software*, dengan karakteristik umum sebagai berikut:
 - *Testing* dimulai dari tingkat komponen terkecil sampai pada integrasi antar komponen pada keseluruhan sistem komputer tercapai.
 - Teknik *testing* berbeda-beda sesuai dengan waktu penggunaannya.
 - *Testing* dilakukan oleh suatu grup tes yang independen.
 - *Testing* dan *debugging* adalah aktivitas yang berlainan, tapi *debugging* harus diakomodasi disetiap strategi *testing*.

Gasal - TA.2022/2023 10

Pendahuluan
(lanjutan)



Gambar 4.1 Strategi testing.

Gasal - TA.2022/2023 11

Pendahuluan
(lanjutan)

- Betapapun bagusnya suatu strategi *testing*, akan dapat gagal jika tidak mempertimbangkan dengan baik isu-isu dari strategi *testing* dibawah ini :
 - Spesifikasi kebutuhan produk agar dapat dikuantitasi, maka harus diterapkan jauh sebelum *testing* dimulai.
 - Nyatakan obyektifitas *testing* secara eksplisit.
 - Memahami pengguna *software* dan mengembangkan profil untuk tiap kategori pengguna.
 - Mengembangkan rencana *testing* yang berdasar pada “*rapid cycle testing*”.
 - Membuat *software* yang tegar (*robust*), yang didisain untuk dapat melakukan tes terhadap dirinya sendiri.
 - Gunakan *Formal Technical Review* (FTR) yang efektif sebagai filter *testing* tertentu.
 - Lakukan FTR untuk menilai strategi tes dan *test cases* itu sendiri.
 - Kembangkan pendekatan pengembangan yang berkelanjutan untuk proses *testing*.

Gasal - TA.2022/2023 12


Pendahuluan
(lanjutan)

- Karakteristik Umum Strategi Pengujian *software*.
 - Pengujian dimulai pada level modul dan dilanjutkan terus hingga integrasi dari keseluruhan sistem.
 - Setiap saat pengujian mengimplementasikan teknik pengujian yang berbeda.
 - Pengujian dikelola oleh pengembang *Software* dan untuk yang berukuran besar dikelola oleh group penguji yang tidak terikat.
 - Pengujian dan *debugging* merupakan aktifitas yang berbeda, tetapi *debugging* selalu digunakan di setiap strategi pengujian.

Gasal - TA.2022/2023 13


Pendahuluan
(lanjutan)

– Siapa yang melakukan Pengujian PL ?



Pengembang

Memahami sistem, namun pada saat pengujian harus adil, Karena berpatokan pada penyebaran PL



Penguji Independent

Harus belajar mengenai sistem, namun saat ditemukan ketidakbenaran akan menghentikannya, karena berpatokan pada kualitas

Gasal - TA.2022/2023 14

Pendahuluan
(lanjutan)

- *Testing* dari *low-level* kepada *high level* (Tahapan *Testing*).
 - Sistem tidak diujikan sebagai suatu unit tunggal, kecuali untuk program yang kecil.
 - Sistem yang besar terdiri dari *sub-systems*, dimana masing-masing *sub-system* terdiri dari *modules* yang dibentuk oleh *procedures and functions*.
 - Proses pengujian dilakukan dalam beberapa langkah sehingga diproses secara *incrementally* dalam proses implementasi sistem.

Gasal - TA.2022/2023 15

Pendahuluan
(lanjutan)

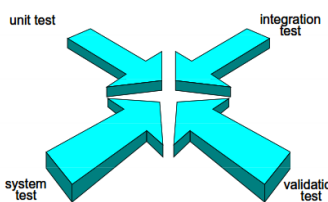
Testing Technique

Component testing	Unit Testing	Verification (Process Oriented)	White Box Testing Techniques (Tests that are derived from knowledge of the program's structure and implementation)
	Module Testing		
Integrated testing	Sub-System Testing	Validation (Product Oriented)	Black Box Testing Techniques (Tests are derived from the program specification)
	System Testing		
User testing	Acceptance Testing		

Gasal - TA.2022/2023 16

Pendahuluan
(lanjutan)

- Pendekatan strategis didalam pengujian perangkat lunak (*software*)
 - *Unit Testing* / Pengujian Unit
 - *Integration Testing* / Pengujian Integrasi
 - *Validation Testing* / Pengujian Validasi
 - *System Testing* / Pengujian Sistem



Gasal - TA.2022/2023 17

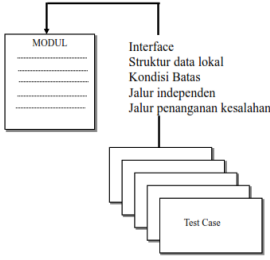
Strategi Testing
Unit Testing



Gasal - TA.2022/2023 18

Unit Testing

- *Unit Testing*, atau biasa disebut sebagai Pengujian Unit.
- Berfokus pada inti terkecil dari desain perangkat lunak yaitu modul.
- Biasanya berorientasi pada pendekatan metode *white box testing*.
- Interface modul diuji untuk memastikan bahwa informasi secara tepat mengalir masuk dan keluar dari modul yang diuji.

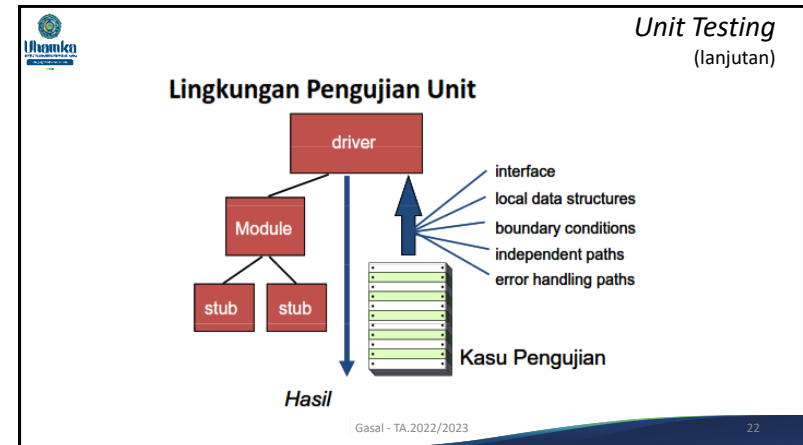
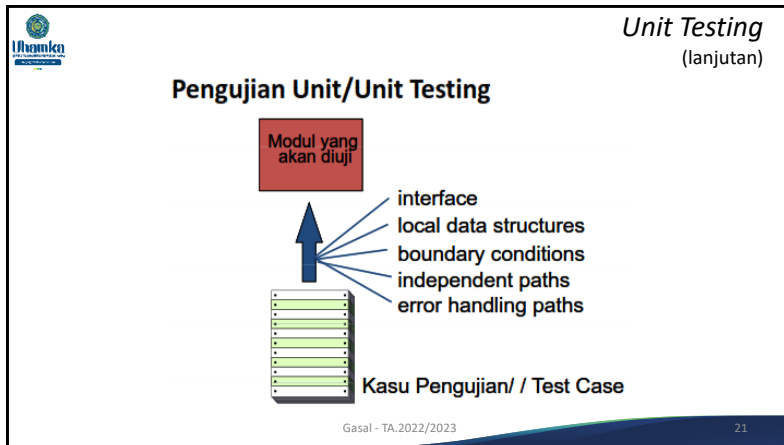


Gasal - TA.2022/2023 19

Unit Testing
(lanjutan)

- Test case harus didesain untuk mengungkap kesalahan dalam kategori.
- Pengetikan yang tidak teratur dan tidak konsisten
- Inisialisasi yang salah atau nilai-nilai default
- Nama variabel yang tidak benar
- Tipe data yang tidak konsisten
- Underflow, overflow dan pengecualian pengalaman

Gasal - TA.2022/2023 20




- Integration Testing**
- “Jika semua modul-modul *software* telah bekerja dengan baik secara individual, mengapa harus ada keraguan apakah modul-modul tersebut dapat bekerja sama sebagai satu kesatuan ?”
 - *Integration testing* adalah suatu teknik yang sistematis untuk pembangunan struktur program, dimana pada saat yang bersamaan melakukan *testing* untuk mendapatkan *errors* yang diasosiasikan dengan antar-muka.
 - Obyektifitasnya adalah untuk menindaklanjuti komponen-komponen yang telah melalui *unit testing* dan membangun suatu struktur program sesuai dengan disain yang telah dituliskan sebelumnya.
- Gasal - TA.2022/2023 24

Integration Testing
(lanjutan)

Strategi Pengujian Integrasi

Opsi : - Pendekatan "big bang"

- Strategi pengembangan inkremental



- Pengujian keseluruhan sistem atau sub-sistem yang terdiri dari komponen yang terintegrasi.
- *Integration testing* menggunakan pendekatan metode *black-box testing* dengan *test case* ditentukan dari spesifikasi.
- Kesulitannya adalah untuk menemukan/melokasikan.
- Penggunaan *Incremental integration testing* dapat mengurangi masalah tersebut.

Gasal - TA.2022/2023 25

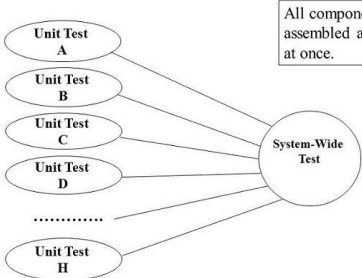
Integration Testing
(lanjutan)

- *Integration testing* ada 2 cara :
 - *Big-Bang Integration Testing*
 - Integrasi ini dilakukan dengan cara semua modul digabung seluruhnya. Setelah itu barulah dilakukan pengujian.
 - *Incremental Integration Testing*
 - Integrasi ini dilakukan untuk membangun dan menguji *interface* program dalam segmen-segmen kecil, sehingga kesalahan lebih mudah diisolasi dan dibetulkan. Interface lebih mungkin untuk diuji dengan lengkap.

Gasal - TA.2022/2023 26

Integration Testing
(lanjutan)

Big-Bang Integration Testing

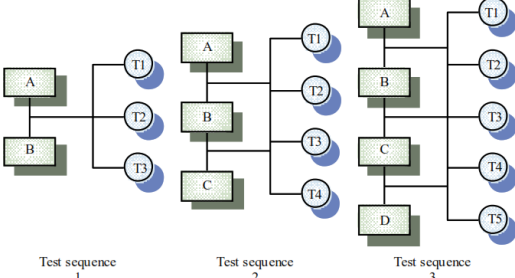


All components are assembled and tested at once.

Gasal - TA.2022/2023 27

Integration Testing
(lanjutan)

Incremental integration test



Gasal - TA.2022/2023 28

Integration Testing (lanjutan)

Pendekatan integration testing

- Pemilihan strategi integrasi (*top-down* atau *bottom-up*) tergantung pada karakteristik program dan juga jadwal proyek.
 - *Top-down testing*
 - *Bottom-up testing*
 - *Sandwich testing*

Gasal - TA.2022/2023 29

Integration Testing (lanjutan)

- **Top-down testing**
 - Berawal dari level-atas sistem dan terintegrasi dengan mengganti masing-masing komponen secara *top-down* dengan suatu stub (program pendek yg men-generate input ke sub-system yang diuji).

Integrasi TOP-DOWN

Gasal - TA.2022/2023 30

Integration Testing (lanjutan)

Top-down testing

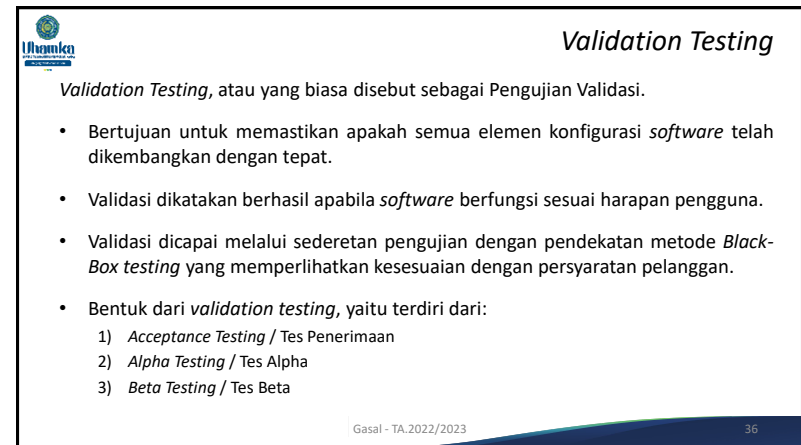
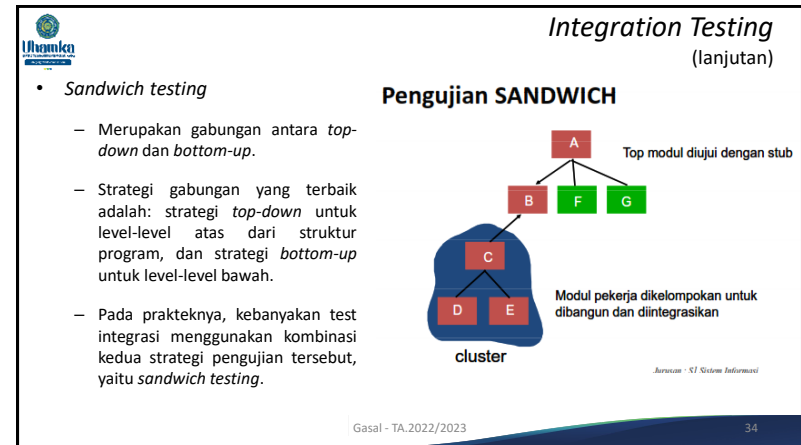
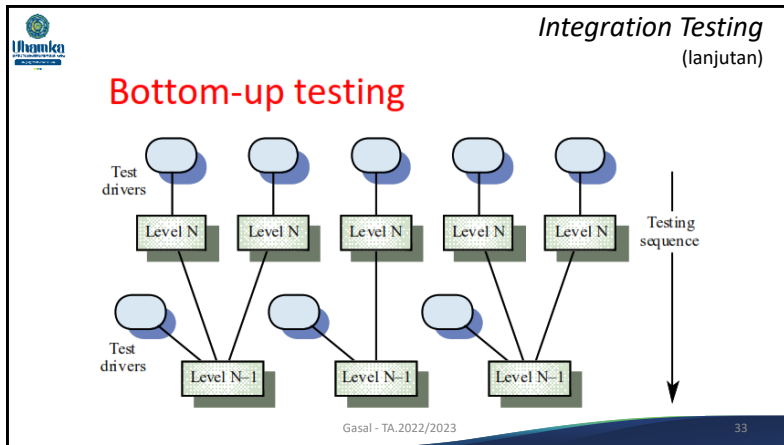
Gasal - TA.2022/2023 31


Integration Testing (lanjutan)

- **Bottom-up testing**
 - Integrasi components ini dimulai dari modul terkecil atau paling bawah, hingga sistem lengkap.

Integrasi BOTTOM-UP

Gasal - TA.2022/2023 32






Validation Testing (lanjutan)

1) Acceptance Testing

- Apabila *software* dibangun untuk satu pelanggan, maka sederetan pengujian penerimaan dapat dilakukan.
- Tujuannya adalah untuk memungkinkan pelanggan mem-validasi semua persyaratan (pengujian dilakukan oleh pelanggan).
- Pengujian terakhir sebelum sistem dipakai oleh user.
- Melibatkan pengujian dengan data dari pengguna sistem.

Gasal - TA.2022/2023 37




Validation Testing (lanjutan)

Acceptance Test | Penilaian Terhadap Faktor Usabilitas

A	Mudah digunakan	1	2	3	4	5
B	User Friendly	1	2	3	4	5
C	Mudah dimengerti	1	2	3	4	5
D	Tingkat Kepercayaan	1	2	3	4	5
E	Tingkat kesesuaian dengan yg dibutuhkan	1	2	3	4	5
F	Waktu Respons	1	2	3	4	5
G	Tingkat komfortabel	1	2	3	4	5

Gasal - TA.2022/2023 38




Validation Testing (lanjutan)

2) Alpha Testing

- Dilakukan pada sisi pengembang oleh user yang potensial.
- *Software* digunakan pada *setting* (pengaturan) yang natural (sebenarnya), sehingga apabila terjadi *error*, maka user dapat merekam masalah yang ada dan mungkin dapat terjadi.
- Dilakukan pada sebuah lingkungan yang terkontrol oleh pengembang.

Gasal - TA.2022/2023 39



Validation Testing (lanjutan)

3) Beta Testing

- Dilakukan oleh satu atau lebih user.
- Biasanya dilakukan oleh selain pengembang / pihak ketiga.
- Pengujian dilakukan diluar kontrol pengembang sistem.
- User merekam semua masalah yang mereka temukan dan melaporkan ke pengembang. Kemudian pengembang melakukan modifikasi dan akhirnya mempersiapkan pelepasan produk ke seluruh pelanggan.

Gasal - TA.2022/2023 40

Validation Testing
(lanjutan)

Alpha Testing VS Beta Testing

During Development

Alpha Testing → Beta Testing

at Customer End

Gasal - TA.2022/2023 41

Strategi Testing
System Testing

Gasal - TA.2022/2023 42

System Testing


- Bertujuan untuk memastikan bahwa semua elemen/komponen sistem saling berhubungan dengan tepat dan keseluruhan fungsi/kinerja sistem dapat tercapai.
- Bentuk tes sistem (system testing):
 - Tes / pengujian Perbaikan
 - Tes / pengujian Keamanan
 - Tes / pengujian Stress
 - Tes / pengujian Kinerja

Gasal - TA.2022/2023 43

System Testing
(lanjutan)

- Tes Perbaikan
 - Pengujian yang memaksa sistem untuk gagal (dirusakkan) dengan berbagai cara dan memeriksa apakah perbaikan dapat dilakukan dengan tepat.
- Tes Keamanan
 - Pengujian yang dilakukan untuk membuktikan apakah perlindungan (keamanan) yang ada pada sistem benar-benar dapat melindungi sistem dari gangguan.


Gasal - TA.2022/2023 44

 **System Testing**
(lanjutan)

3) Tes Stress

- Pengujian yang dilakukan untuk melihat bagaimana sistem dalam keadaan abnormal.
- Menguji sistem dengan nilai yang melebihi maksimum *load*. *Stressing* suatu sistem menyebabkan tidak mudah rusak.
- Contoh: melakukan *login* ke *server* ketika sejumlah besar *workstation* melakukan proses menjalankan perintah *sql database*.


Gasal - TA.2022/2023 45

 **System Testing**
(lanjutan)

4) Tes Kinerja

- Pengujian untuk menguji kinerja *run-time* (saat berjalan) dari *software* didalam konteks sistem yang terintegrasi.
- Tes kinerja sering digunakan bersamaan dengan tes *stress*, dimana kita bisa melihat bagaimana kinerja sistem dalam keadaan abnormal.

Gasal - TA.2022/2023 46

 **Referensi**

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.
- Romeo, 2003. *Testing dan Implementasi Sistem*, Edisi Pertama, STIKOM.


Gasal - TA.2022/2023 47

THE POWER OF QIRAAH
**SYAFA'AT
DI AKHIRAT**


عَنْ أَبِي أُمَامَةَ الْبَاهِلِيِّ رَضِيَ اللَّهُ عَنْهُ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ: اقْرَأُوا الْقُرْآنَ فَإِنَّهُ يَأْتِي يَوْمَ الْقِيَامَةِ شَفِيعًا لِأَصْحَابِهِ (حديث صحيح أخرجه مسلم)

Dari Abu Umamah Al Bahili, Rasulullah SAW bersabda: "Bacalah Al-Qur'an, maka sesungguhnya ia akan datang di hari kiamat memberi syafaat kepada pembacanya." (Hadis Shahih diriwayatkan oleh Imam Muslim)

Terima Kasih



Testing and Implementation




Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion

Session 09




PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>

- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhmadiyah.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkembang.
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.


Gasal - TA.2022/2023 2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.


Gasal - TA.2022/2023 3



Tujuan/Objectives (Session.09)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - GUI Testing
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang GUI *testing, needs of GUI testing, approaches, GUI testing tools, UI testcases with examples.*

Gasal - TA.2022/2023 4




Muhasabah

وَلَتَبْلُوَنَكُمْ بِشَيْءٍ مِّنَ الْخَوْفِ وَالْجُوعِ وَنَقْصٍ مِّنَ الْأَمْوَالِ وَالْأَنْفُسِ وَالثَّمَرَاتِ وَبَشِّرِ الصَّابِرِينَ
 الَّذِينَ إِذَا أَصَابَتْهُمُ مُصِيبَةٌ قَالُوا إِنَّا لِلَّهِ وَإِنَّا إِلَيْهِ رَاجِعُونَ
 أُولَئِكَ عَلَيْهِمْ صَلَوَاتٌ مِّن رَّبِّهِمْ وَرَحْمَةٌ وَأُولَئِكَ هُمُ الْمُهْتَدُونَ


- Qs. Al-Baqarah, ayat 155-157
 - Dan Kami pasti akan menguji kamu dengan sedikit ketakutan, kelaparan, kekurangan harta, jiwa, dan buah-buahan. Dan sampaikanlah kabar gembira kepada orang-orang yang sabar,
 - (yaitu) orang-orang yang apabila ditimpa musibah, mereka berkata "Inna lillahi wa inna ilaihi raji'un" (sesungguhnya kami milik Allah dan kepada-Nyalah kami kembali).
 - Mereka itulah yang memperoleh ampunan dan rahmat dari Tuhannya, dan mereka itulah orang-orang yang mendapat petunjuk.

Gasal - TA.2022/2023

5



Testing and Implementation



Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion

● ● ●

Session 09


Graphical User Interface Testing

Gasal - TA.2022/2023

6




GUI Testing



Pendahuluan

Gasal - TA.2022/2023

7



Pendahuluan

- Terdapat dua jenis antarmuka untuk aplikasi komputer.
- Antarmuka Baris Perintah (*Command Line Interface*), adalah tempat Anda mengetik teks dan komputer merespons perintah itu.
- GUI merupakan singkatan dari *Graphical User Interface*, di mana Anda berinteraksi dengan komputer menggunakan gambar daripada teks.

Gasal - TA.2022/2023

8


Pendahuluan (lanjutan)

- Berikut adalah salah satu elemen GUI yang dapat digunakan untuk interaksi antara pengguna dan aplikasi:
 - True False *Radio Button*
 - Check Check *Check Box*
 -
 -
- Pengujian GUI mencakup validasi dari elemen-elemen di atas.

Gasal - TA.2022/2023 9

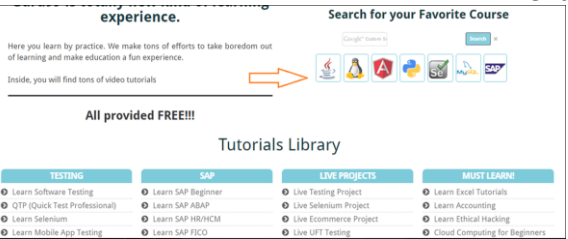
Pendahuluan (lanjutan)

- GUI *testing* adalah jenis pengujian perangkat lunak yang memeriksa Antarmuka Pengguna Grafis dari Perangkat Lunak.
- Tujuan GUI testing adalah, untuk memastikan fungsionalitas aplikasi perangkat lunak dapat bekerja sesuai spesifikasi yang telah ditentukan, antara lain dengan memeriksa layar dan kontrol seperti menu, tombol, ikon, dll.



Gasal - TA.2022/2023 10

Pendahuluan (lanjutan)



- GUI adalah apa yang dilihat pengguna. Katakanlah jika Anda mengunjungi sebuah *website*, maka apa yang akan Anda lihat pada halaman *homepage* itu adalah GUI dari situs tersebut.
- Seorang pengguna tidak akan melihat *source code* dari website tsb, yang dilihat hanyalah antarmuka yang terlihat oleh pengguna.

Gasal - TA.2022/2023 11

Pendahuluan (lanjutan)

- Dalam contoh di atas, jika kita harus melakukan pengujian GUI, pertama-tama kita periksa apakah gambar-gambar yang tersedia harus benar-benar terlihat di browser yang berbeda.
- Apakah *hyperlink*/tautannya tersedia, dan tombolnya akan berfungsi pada saat di-klik.
- Juga, jika pengguna mengubah ukuran layar, baik gambar maupun konten tidak boleh menyusut, atau terpotong, atau tumpang tindih.

Gasal - TA.2022/2023 12




GUI Testing
Need of GUI Testing

Gasal - TA.2022/2023 13

Need of GUI Testing

- Why do GUI testing ?
- Is it really needed ?
- Does testing of functionally and logic of Application is not more than enough ??
Then why to waste time on UI testing.



Gasal - TA.2022/2023 14

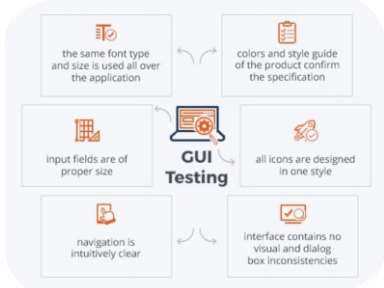
Need of GUI Testing (lanjutan)

- 1) GUI *testing* adalah proses pengujian GUI pada aplikasi untuk mengidentifikasi cacat yang terjadi di dalamnya selama tahap desain.
- 2) GUI *testing* dilakukan untuk memverifikasi fungsionalitas GUI sesuai spesifikasi dan tergantung pada teknologi yang digunakan.
- 3) GUI *testing* juga mengevaluasi kontrol seperti menu, tombol, ikon, kotak teks, daftar, kotak dialog, tata letak, warna, ukuran font, pemformatan teks, dll.
- 4) GUI *testing* dapat dilakukan secara manual atau otomatis dengan bantuan alat (*tools*), yang sering dilakukan oleh perusahaan pihak ketiga daripada pengembang atau pengguna.
- 5) GUI *testing* digunakan untuk mengeksekusi nilai properti untuk setiap objek GUI dan menjalankan cara-cara pengoperasian GUI seperti penekanan tombol atau klik *mouse*.

Gasal - TA.2022/2023 15

Need of GUI Testing (lanjutan)

- The following list suggests what should be exactly checked while performing GUI Testing;
 - Screen Validations
 - Size and position of GUI elements
 - Clear and well-aligned images
 - Navigations (links)
 - Font and alignment of text
 - Date and numeric fields
 - Usability conditions and data integrity
 - Error messages
 - Required fields
 - Abbreviations inconsistencies
 - Progress bars
 - Shortcuts




Gasal - TA.2022/2023 16



GUI Testing
Approaches

Gasal - TA.2022/2023 17



Approaches

- Pendekatan yang dilakukan pada saat GUI testing:
 - 1) *Manual Based Testing*
 - 2) *Record and Replay*
 - 3) *Model-Based Testing*

Gasal - TA.2022/2023 18

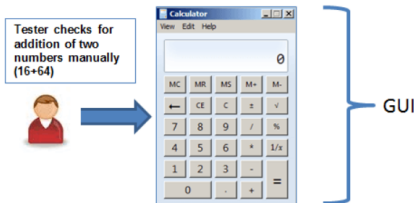


Approaches
(lanjutan)

1) *Manual Based Testing*

- Testers menerapkan pengetahuan yang mereka miliki, dan menguji tampilan grafis sesuai dengan persyaratan kebutuhan bisnis (*business requirements*).

Gasal - TA.2022/2023 19



Approaches
(lanjutan)

Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in the business requirements document.

Gasal - TA.2022/2023 20

Approaches
(lanjutan)

2) *Record and Replay*

- Pendekatan ini dapat dilakukan dengan menggunakan alat otomatisasi yang mendukung fitur/tindakan untuk Rekam (*Record*) dan Putar Ulang (*Replay*).
- Langkah-langkah pengujian yang dilakukan di-capture oleh tools tsb dan direkam, untuk selanjutnya dijalankan kembali (*Replay*) pada aplikasi yang diuji.

Gasal - TA.2022/2023 21

Approaches
(lanjutan)

GUI testing can be done using automation tools. This is done in 2 parts. During Record, test steps are captured by the automation tool. During playback, the recorded test steps are executed on the Application Under Test. Example of such tools – QTP.

Gasal - TA.2022/2023 22

Approaches
(lanjutan)

3) *Model Based Testing*

- Pengujian Berbasis Model dilakukan disesuaikan dengan perilaku sistem.
- Model-model ini dapat dikategorikan kedalam 3 jenis, seperti:
 - Event-based model*: Based on GUI events that are to occur at least once
 - State-based model*: Based on GUI states exercised at least once
 - Domain model*: Based on domain and functionality of the application

Gasal - TA.2022/2023 23

Approaches
(lanjutan)

GUI

Model Based Testing

Gasal - TA.2022/2023 24

Approaches
(lanjutan)

- Model adalah deskripsi grafis dari perilaku sistem.
- Model ini dapat membantu kita untuk memahami dan memprediksi perilaku sistem.
- Model membantu dalam menghasilkan kasus uji yang efisien menggunakan persyaratan sistem.
- Berikut ini perlu dipertimbangkan untuk pengujian GUI berbasis model ini:
 - *Build the model*
 - *Determine Inputs for the model*
 - *Calculate the expected output for the model*
 - *Run the tests*
 - *Compare the actual output with the expected output*
 - *A decision on further action on the model*

Gasal - TA.2022/2023 25

Approaches
(lanjutan)

- Beberapa teknik pemodelan yang dapat digunakan agar *test cases* dapat diturunkan:
 - *Bagan (charts)*
 - Menggambarkan keadaan sistem dan memeriksa keadaan setelah beberapa masukan.
 - *Tabel Keputusan (decision tables)*
 - Tabel yang digunakan untuk menentukan hasil untuk setiap input yang diterapkan.

Gasal - TA.2022/2023 26

Approaches
(lanjutan)

a) *Event-Based Model*

"Event-flow graph"

```

graph TD
    File --> New
    File --> Open
    File --> Save
    Edit --> Cut
    Edit --> Copy
    Edit --> Paste
    Edit --> Replace
    Edit --> Undo
    Edit --> Editbox0
    Edit --> Editbox1
    Edit --> Matchcase
    Editbox0 --> Findnext
    Editbox0 --> Replace
    Editbox1 --> Replace
    Editbox1 --> Replaceall
    Editbox1 --> Cancel
    Matchcase --> Cancel
    Toplevel((Top level))
  
```

TC: <S0, event1, event2, ...>
Oracle: <State1, State2, ...> & ICRASH

Gasal - TA.2022/2023 27

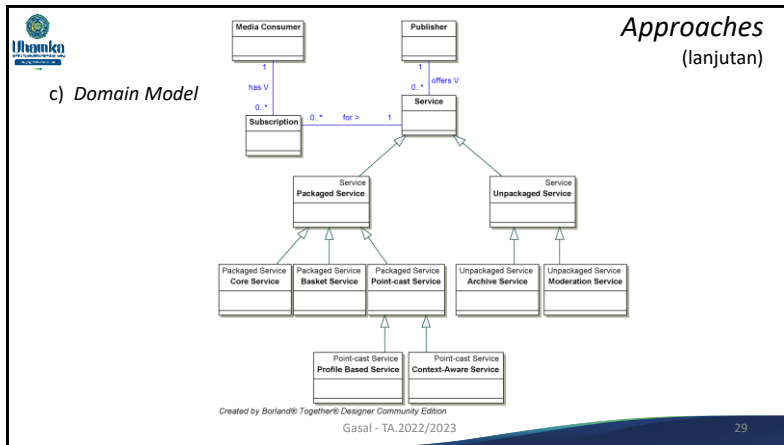
Approaches
(lanjutan)

b) *State-Based Model*

```

graph TD
    question -- CLICK_GOOD --> thanks
    question -- CLICK_BAD --> form
    form -- SUBMIT --> closed
    form -- CLOSE --> closed
    thanks -- CLOSE --> closed
    thanks -- Close --> Toplevel((Top level))
  
```

Gasal - TA.2022/2023 28



GUI Testing Tools

- Pengujian aplikasi khususnya GUI *testing*, dapat dilakukan secara manual maupun secara otomatis.
- Terdapat banyak *tools* yang dapat digunakan untuk melakukan GUI *testing*.
<https://www.softwaretestinghelp.com/best-gui-testing-tools/#Conclusion>
- Berikut beberapa di antaranya tercantum di bawah ini sesuai dengan popularitas dan penggunaannya.

GUI Testing Tools

Gasal - TA.2022/2023

31

GUI Testing Tools
(lanjutan)


eggPlant UI Automation Testing

eggPlant
built by
TestPlant


- eggPlant is a commercial GUI Automation and Software Testing tool designed by TestPlant
- It is good for GUI Testing and is used for the entire testing process
- It uses VNC to look at SUT and send mouse and keyboard commands
- Can test any device, any operating system, and any technology. Create tests quickly via eggDrive interface
- We can integrate eggplant into Jenkins, IBM Rational Quality Manager and Micro Focus Quality Centre for performing end to end QA process

Gasal - TA.2022/2023

32


 **GUI Testing Tools**
(lanjutan)

- Squish GUI testing tool




- Squish is a commercial tool for GUI automation testing
- It allows record and edit tests by using scripting languages like JavaScript, Perl, Python, and Ruby
- Facilitates verification of properties, screenshots, images, complex data, external files, and databases
- Possesses Eclipse-based integrated test development environment

Gasal - TA.2022/2023 33


 **GUI Testing Tools**
(lanjutan)

- AutoIT UI testing




- AutoIt is freeware scripting language with basic structures used to automate Windows GUI and general scripting
- It is a self-contained commercial tool which is a combination of keystrokes, mouse movements, and Windows Manipulation
- Tool avails COM support and script compilation with standalone executables
- The tool encompasses Built-in Editor, Basic-like Syntax, Rich function set etc

Gasal - TA.2022/2023 34


 **GUI Testing Tools**
(lanjutan)

- Ranorex Studio




- Reliable object identification, even for web elements with dynamic IDs.
- Shareable object repository and reusable code modules for efficient test creation and reduced maintenance.
- Customizable test report with video reporting of test execution.
- Run tests in parallel or distribute on a Selenium Grid with built-in Selenium Webdriver.
- Customizable test report.
- Integrates with tools like Jira, Jenkins, TestRail, Git, Travis CI, and more.

Gasal - TA.2022/2023 35

 **GUI Testing Tools**
(lanjutan)

- RIATest



- RIATest is a GUI test automation tool beneficial for Flex, HTML, JavaScript, jQuery or Windows 8 apps
- RIATest is added to the continuous integration system to find out issues quickly
- Helps to create a readable test script, uses the component inspector to recognize GUI elements
- It is a commercial tool that allows customized error handling through logging the error or throwing an exception

Gasal - TA.2022/2023 36



GUI Testing
User Interface (UI) TestCases with Examples

Gasal - TA.2022/2023 37

UI Testcases With Examples

- GUI Testing basically involves
 - 1) Testing the size, position, width, height of the elements.
 - 2) Testing of the error messages that are getting displayed.
 - 3) Testing the different sections of the screen.
 - 4) Testing of the font whether it is readable or not.
 - 5) Testing of the screen in different resolutions with the help of zooming in and zooming out like 640 x 480, 600 x 800, etc.
 - 6) Testing the alignment of the texts and other elements like icons, buttons, etc. are in proper place or not.

Gasal - TA.2022/2023 38

UI Testcases With Examples
(lanjutan)

- 7) Testing the colors of the fonts.
- 8) Testing the colors of the error messages, warning messages.
- 9) Testing whether the image has good clarity or not.
- 10) Testing the alignment of the images.
- 11) Testing of the spelling.
- 12) The user must not get frustrated while using the system interface.
- 13) Testing whether the interface is attractive or not.
- 14) Testing of the scrollbars according to the size of the page if any.
- 15) Testing of the disabled fields if any.

Gasal - TA.2022/2023 39

UI Testcases With Examples
(lanjutan)

- 16) Testing of the size of the images.
- 17) Testing of the headings whether it is properly aligned or not.
- 18) Testing of the color of the hyperlink.

Beberapa contoh penggunaan kasus uji (*test cases*) untuk GUI *testing*, dapat di-*download* pada **suplemen materia** yang disertakan pada slide ini.

Gasal - TA.2022/2023 40



GUI Testing

Conclusions

Gasal - TA.2022/2023 41

Conclusion

- GUI testing is necessary to improve the quality of an application. It is important but sometimes seems costly.
- GUI testing is better to perform using tools rather than manually. Tools help to detect loopholes in a better way.
- The success of a software product immensely depends on how the GUI interacts with the user and ease in using its various features. Hence, GUI testing is very important.
- Manual GUI testing can sometimes be repetitive and boring and hence error-prone. Automation is highly recommended for GUI testing.

Gasal - TA.2022/2023 42

Referensi

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.
- Romeo, 2003. *Testing dan Implementasi Sistem*, Edisi Pertama, STIKOM.


Gasal - TA.2022/2023 43

TOLAK BALA
dengan **SEDEKAH**


Terima Kasih

"Bersegeralah bersedekah,
karena bala bencana
tidak pernah bisa mendahului sedekah".
(HR. Baihaqi)





Testing and Implementation




Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion

Session 10




PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>

- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhmadiyah.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkembang
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.


Gasal - TA.2022/2023 2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%**, UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.


Gasal - TA.2022/2023 3



Tujuan/Objectives (Session.10)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Pengujian Arsitektur Client/Server
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang pengujian arsitektur *client/server*, *volume testing*, *stress testing*, *performance testing*, *data recovery testing*, *data backup* dan *restore testing*, *data security testing* .


Gasal - TA.2022/2023 4


 **Muhasabah**

وَاللّٰهُ اَخْرَجَكُمْ مِّنْ بُطُوْنِ اُمَّهَاتِكُمْ لَا تَعْلَمُوْنَ شَيْئًا
وَجَعَلَ لَكُمُ السَّمْعَ وَالْاَبْصَارَ وَالْاَفْئِدَةَ
لَعَلَّكُمْ تَشْكُرُوْنَ


- Qs. An-Nahl, ayat 78
 - Dan Allah mengeluarkan kamu dari perut ibumu dalam keadaan tidak mengetahui sesuatu pun, dan Dia memberimu pendengaran, penglihatan, dan hati nurani, agar kamu bersyukur.


Gasal - TA.2022/2023 5

 **Testing and Implementation**


Uhamka
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA
integrity, trust, compassion


● ● ●
Session 10
Pengujian Arsitektur *Client/Server*

 **Testing Arsitektur *Client/Server***



Pendahuluan

Gasal - TA.2022/2023 7

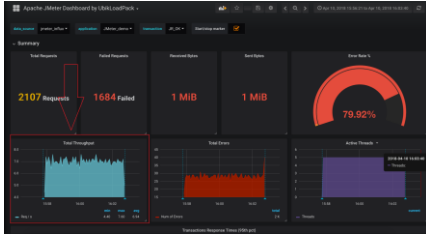
 **Pendahuluan**

- Testing arsitektur *client/server* merupakan pengujian yang berfungsi untuk menguji proses yang terjadi dari *client* ke *server*, atau sebaliknya.
- Pada pengujian arsitektur *client/server*, terdapat beberapa macam pengujian antara lain:
 - a) *Volume Testing*
 - b) *Stress Testing*
 - c) *Performance Testing*
 - d) *Data Recovery Testing*
 - e) *Data Backup dan Restore Testing*
 - f) *Data Security Testing*

Gasal - TA.2022/2023 8

Pendahuluan
(lanjutan)

- Untuk melakukan pengujian arsitektur *client/server*, ada berbagai aplikasi yang dapat membantu, dan salah satunya adalah aplikasi JMeter.
- JMeter adalah aplikasi *open source* berbasis Java yang dapat dipergunakan untuk melakukan *performance test*, *load/stress testing*, *web application testing*, *ftp application*, dan *database testing*.



Gasal - TA.2022/2023 9

Pendahuluan
(lanjutan)

- Untuk meng-unduh aplikasi JMeter, melalui tautan berikut:
https://jmeter.apache.org/download_jmeter.cgi
- User manual untuk penggunaan JMeter, dapat diakses pada tautan berikut:
<https://jmeter.apache.org/usermanual/index.html>

Gasal - TA.2022/2023 10



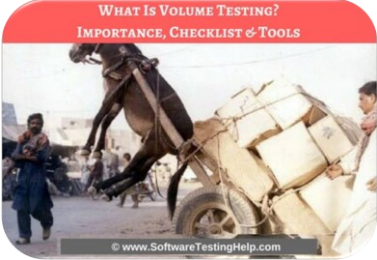
Pengujian Arsitektur *Client/Server*

a) *Volume Testing*

Gasal - TA.2022/2023 11

a) *Volume Testing*

- Pengujian ini dilakukan untuk menemukan kelemahan sistem selama melakukan pemrosesan data dalam jumlah yang besar dalam periode waktu yang singkat.



© www.SoftwareTestingHelp.com

Gasal - TA.2022/2023 12

a) Volume Testing
(lanjutan)

- Contoh dari pengujian ini adalah saat menguji proses antar *server* dan antar partisi *hardisk* pada satu *server*.

The diagram illustrates a network architecture for volume testing. On the left, multiple 'Users' (represented by laptop icons) connect to an 'Internet' cloud. This cloud connects to a 'FireWall' (represented by a shield icon). The traffic then passes through a 'Load Balancer' (represented by a server rack icon) to a group of 'Web Servers' (represented by server rack icons). These web servers connect to 'Application Servers' (represented by server rack icons), which in turn connect to 'Database Servers' (represented by server rack icons).

Gasal - TA.2022/2023 13

a) Volume Testing
(lanjutan)

The mind map 'Purpose of Volume Testing' branches into six points:

- 01 Evaluate the System response Time
- 02 Classifying the volume of the system
- 03 Design the scalability plans
- 04 Make sure there's no data loss
- 05 Receiving insight on the system performance under, low medium, and high data loads
- 06 Identify the error persuaded by force on the system components

- Tujuan melakukan pengujian ini adalah untuk meyakinkan bahwa sistem tetap melakukan pemrosesan data antar batasan fisik dan batasan logik.

Gasal - TA.2022/2023 14

The image shows three books: 'Concise Guide to Software Testing' by Gerard O'Regan, 'SOFTWARE TESTING AND QUALITY ASSURANCE: THEORY AND PRACTICE' by Wiley, and 'Testing CMM ISO/IEC 15504 Designing Software Quality Assurance' by Springer.

Pengujian Arsitektur Client/Server

b) Stress Testing

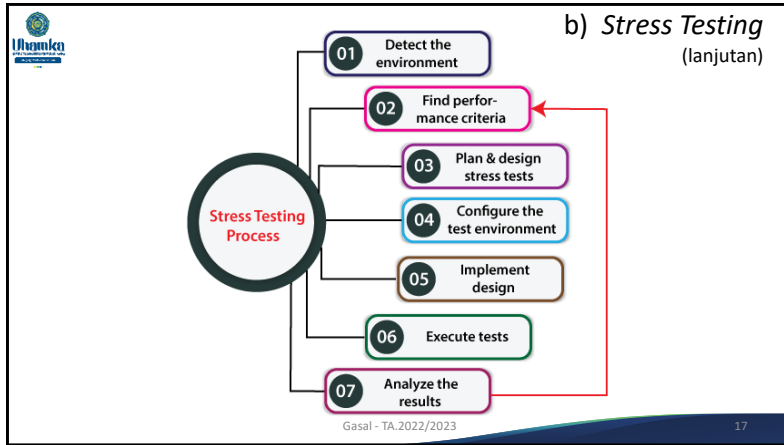
Gasal - TA.2022/2023 15

b) Stress Testing

- *Stress testing* dilakukan dengan tujuan untuk mengetahui kemampuan sistem dalam melakukan transaksi selama periode waktu puncak proses.
- Contohnya saat melakukan *login* ke *server* ketika sejumlah besar *workstation* melakukan proses menjalankan perintah (*query*) pada *sql database*.

The diagram shows a 'Stress Testing' scenario. On the left, several 'Workstations' (represented by laptop icons) connect to an 'Internet' cloud. The cloud connects to a server rack icon. A fire icon is placed between the internet cloud and the server rack, symbolizing high load or stress. On the right, a vertical stack of server rack icons represents the database or application servers.

Gasal - TA.2022/2023 16



What Is Performance Testing?

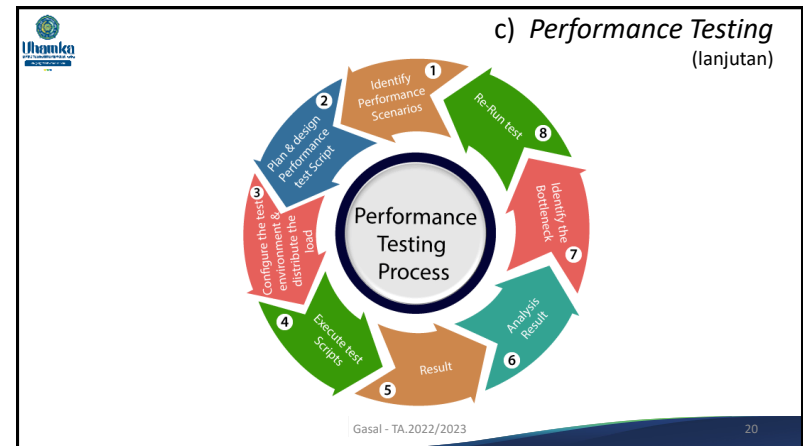
Performance Testing is a software testing process used for testing the speed, response time, stability, reliability, scalability, and resource usage of a software application under particular workload.

What does Performance Testing Check?

- Speed:** The speed at which the application responds.
- Scalability:** Determines maximum user load the software application can handle.
- Stability:** Determines if the application is stable under varying loads.

- Pengujian ini dilakukan secara paralel dengan *volume testing* dan *stress testing*.
- Tujuannya adalah untuk menguji kecepatan (*speed*), waktu respons (*response time*), stabilitas (*stability*), keandalan (*reliability*), skalabilitas (*scalability*), dan penggunaan sumber daya (*resource usage*) dari aplikasi perangkat lunak di bawah beban kerja tertentu

Gasal - TA.2022/2023 19





Pengujian Arsitektur *Client/Server*

d) *Data Recovery Testing*

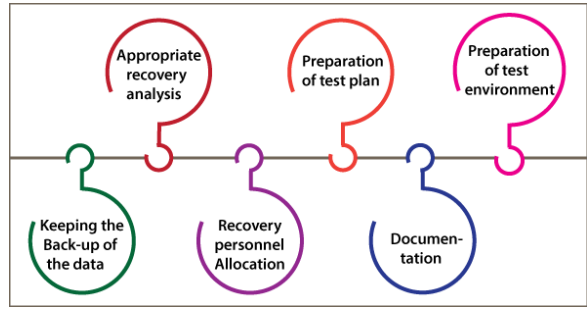
Gasal - TA.2022/2023 21

d) *Data Recovery Testing*

- *Data recovery testing* dilakukan untuk menginvestigasi dampak kehilangan data melalui proses *recovery* ketika terjadi kegagalan proses.
- Pengujian ini penting untuk dilakukan, karena data yang disimpan di *server* dapat dikonfigurasi dengan berbagai cara.
- Data dapat hilang akibat kegagalan sistem, *harddisk* rusak, penghapusan yang tidak sengaja, kecelakaan, virus, maupun oleh pencuri.

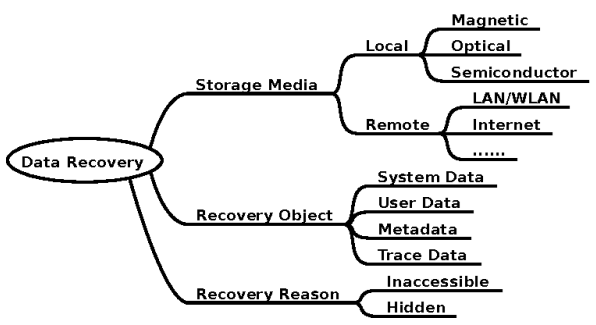
Gasal - TA.2022/2023 22

d) *Data Recovery Testing*
(lanjutan)



Gasal - TA.2022/2023 23

d) *Data Recovery Testing*
(lanjutan)



Gasal - TA.2022/2023 24




Pengujian Arsitektur *Client/Server*

e) *Data Backup dan Restore Testing*

Gasal - TA.2022/2023 25

e) *Data Backup dan Restore Testing*

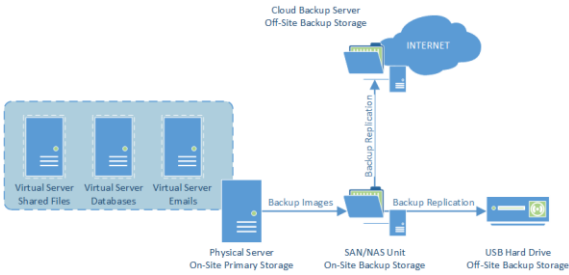
- Pengujian ini dilakukan untuk melihat prosedur *backup* dan *recovery*.
- Hal ini dilakukan dengan mensimulasikan beberapa kesalahan untuk menguji proses *backup* dan *restore*.
- Pengujian dapat dilakukan terhadap :
 - frekuensi *backup*,
 - waktu yang dibutuhkan,
 - mekanisme *backup* (manual/otomatis),
 - personal yang melaksanakan,
 - dan berapa lama *backup* akan disimpan.



Gasal - TA.2022/2023 26

e) *Data Backup dan Restore Testing*
(lanjutan)

- Switching* antara *live* dan *backup server* ketika terjadi kerusakan (*load log transaction* pada *back-up*, kemudian melakukan *recovery*)



Gasal - TA.2022/2023 27



Pengujian Arsitektur *Client/Server*

f) *Data Security Testing*


Gasal - TA.2022/2023 28



f) Data Security Testing

- Pengujian ini dilakukan ke beberapa *user* yang tidak memiliki *privilege access* ke *database*.
- Pengujian ini dapat dilakukan dengan cara *shutdown database engine*, melalui *operating system* yang dapat mematikan aplikasi *database*, atau dengan cara lainnya.

Gasal - TA.2022/2023 29



Referensi

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.
- Romeo, 2003. *Testing dan Implementasi Sistem*, Edisi Pertama, STIKOM.

Gasal - TA.2022/2023 30

Terima Kasih


Banyak ilmu, namun lupa belajar adab dan akhlak




Imam Malik *rahimahullah* pernah berkata pada seorang pemuda Quraisy, "Pelajarilah adab sebelum mempelajari suatu ilmu."

Kenapa sampai para ulama mendahulukan mempelajari adab?

Sebagaimana Yusuf bin Al Husain berkata, "Dengan mempelajari adab, maka engkau jadi mudah memahami ilmu."



Testing and Implementation




Uhamka

UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA

integrity, trust, compassion

Session 11




PROGRAM STUDI TEKNIK INFORMATIKA

Sumber: <https://ft.uhamka.ac.id/prodi/informatika/>

- VISI.
 - √ Menjadi program studi informatika yang menghasilkan karya unggul di bidang teknologi informasi melalui pembelajaran berlandaskan prophetic teaching untuk membentuk lulusan yang cerdas secara spiritual, intelektual, emosional, dan sosial.
- MISI.
 1. Melaksanakan pendidikan di bidang informatika yang terintegrasi dengan nilai-nilai Al Islam Kemuhmadiyahhan.
 2. Melaksanakan penelitian untuk menghasilkan karya teknologi informasi yang unggul untuk masyarakat berkembang
 3. Melaksanakan pengabdian kepada masyarakat untuk kesejahteraan umat.
 4. Menghasilkan mahasiswa yang berwawasan global dalam teknologi informasi.
 5. Mendorong jiwa kewirausahaan mahasiswa untuk berkiprah di tengah masyarakat dalam meningkatkan kegiatan ekonomi.
 6. Menerapkan standar mutu pendidikan informatika melalui layanan mutu yang berkesinambungan.
 7. Membangun kerja sama terpadu dengan mitra yang berkontribusi untuk kemajuan pendidikan di bidang informatika.


Gasal - TA.2022/2023 2



Sistem Penilaian

- SKS : 3
 - Teori 3 SKS – (14 Kali Pertemuan)
- Keaktifan (bobot **10%**)
 - Terdiri dari: Aktivitas Perkuliahan (*Kehadiran, Etika, dan Sikap*).
- Project (bobot **25%**)
 - Terdiri dari: Tugas Besar / Project (*Penguasaan Keterampilan*) dengan cara observasi, mengimplementasikan dan menghasilkan Proyek Akhir, serta mempresentasikannya.
- Tes (UTS = bobot **25%** , UAS = bobot **40%**)
 - Terdiri dari: UTS dan UAS (*Penguasaan Pengetahuan*) dengan cara tes tertulis maupun tes praktikum.


Gasal - TA.2022/2023 3



Tujuan/Objectives (Session.11)

- Objectives (Tujuan Instruksional Umum).
 - RPS/SAP.
 - Pengujian pada Dokumentasi & Fasilitas Help
- Indikator Kompetensi.
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang konsep, metode, strategi pengujian sistem, teknik pengukuran dan melakukan audit kegiatan dari sebuah sistem perangkat lunak dengan berbagai aspek
 - Mahasiswa mampu mengetahui, memahami, dan menjelaskan tentang pengujian pada dokumentasi (fase pertama, dan fase kedua), pengujian fasilitas *help* (FAQs, *admin online, search*)


Gasal - TA.2022/2023 4

 Muhasabah

وَلْتَكُنْ مِنْكُمْ أُمَّةٌ يَدْعُونَ إِلَى الْخَيْرِ وَيَأْمُرُونَ بِالْمَعْرُوفِ وَيَنْهَوْنَ عَنِ الْمُنْكَرِ
وَأُولَئِكَ هُمُ الْمُفْلِحُونَ

- Qs. Al Imron, ayat 104
 - Dan hendaklah di antara kamu ada segolongan orang yang menyeru kepada kebajikan, menyuruh (berbuat) yang makruf, dan mencegah dari yang mungkar.
 - Dan mereka itulah orang-orang yang beruntung.

Gasal - TA.2022/2023 5

 Testing and Implementation


Uhamka
 UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA
integrity, trust, compassion


● ● ●
 Session 11
 Pengujian pada Dokumentasi & Fasilitas Help

Pengujian Dokumentasi & Fasilitas Help

Pengujian Dokumentasi

Gasal - TA.2022/2023 7


 Pengujian Dokumentasi

- Istilah “pengujian perangkat lunak” memunculkan citra terhadap sejumlah besar *test case* yang disiapkan untuk menggunakan program komputer dan data yang dimanipulasi oleh program.
- Dengan melihat kembali definisi perangkat lunak yang disajikan di awal, penting untuk dicatat bahwa pengujian harus berkembang ke elemen ketiga dari konfigurasi perangkat lunak, yaitu “dokumentasi”.
- Pengujian dokumentasi, yaitu melakukan pengujian terhadap dokumentasi yang disediakan sebagai panduan pada saat *user* menggunakan program aplikasi.

Gasal - TA.2022/2023 8

Pengujian Dokumentasi
(lanjutan)

- Kesalahan dalam dokumentasi dapat menghancurkan penerimaan program seperti halnya kesalahan pada data atau kode sumber.
- Tidak ada yang lebih membuat frustrasi dibanding mengikuti tuntunan pengguna secara tepat, akan tetapi mendapatkan hasil atau tingkah laku yang tidak sesuai dengan yang diprediksi oleh dokumen.




- Karena itulah pengujian dokumentasi harus menjadi suatu bagian yang berarti dari setiap rencana pengujian perangkat lunak.

Gasal - TA.2022/2023 9

Pengujian Dokumentasi
(lanjutan)

- Pengujian dokumentasi dapat didekati dalam dua fase.
 - Fase pertama, yaitu kajian teknis formal yang menguji kejelasan editorial dokumen. Dalam fase ini perlu untuk melakukan *test plan* dan *test case*.
 - Fase kedua, yaitu *live test* menggunakan dokumentasi dalam kaitannya dengan penggunaan program aktual.



Gasal - TA.2022/2023 10



Pengujian Dokumentasi & Fasilitas Help

Pengujian Dokumentasi: Fase Pertama

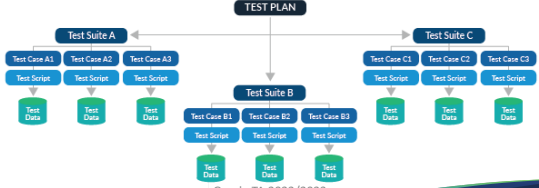
Gasal - TA.2022/2023 11

TEST PLAN vs. TEST CASE


TEST PLAN	TEST CASE
<ul style="list-style-type: none"> Comprehensive for Project Unit Integration Quality Assurance User Acceptance Regression Addresses Process Includes Test Case documentation 	<ul style="list-style-type: none"> Detailed Linked to Traceability

1st Phase

- Untuk melakukan pengujian pada dokumentasi, maka perlu untuk membuat dan melakukan *test plan* dan *test case*.



Gasal - TA.2022/2023 12




1st Phase
(lanjutan)

a) Test Plan

- *Test Plan* merupakan sebuah dokumen perencanaan yang berisikan definisi tujuan dan sasaran pengujian dalam sebuah *testing*.
- Tujuan dibuatnya sebuah *test plan* adalah untuk memudahkan *developer* untuk melakukan *testing*, agar *testing* yang dilakukan menjadi jelas sehingga hasilnya lebih berguna dan efisien.

Gasal - TA.2022/2023
13




1st Phase
(lanjutan)

CompuTech Website Test Plan for Firefox

HomePage

Test	User Requirement Addressed	Expected Result	Actual Result	Pass / Fail	Date	Corrective Action
Logo Displayed	The website uses the companies house style	CompuTech logo is displayed	As expected	Pass	01/12/09	None
Navigation Bar Appears	Website is easy to navigate	Navigation bar appears	As expected	Pass	01/12/09	None
Navigation Bar Drop Downs Work	Website is easy to navigate	When mouse goes over labels - menus drop down	Bar is static and drops do not work	Fail	01/12/09	Recode the JavaScript as it only works with IE
Navigation Bar Hyperlinks Work	Website is easy to navigate	When mouse goes over turns to red and can click on it to take you to correct page	As expected	Pass	01/12/09	Now I have fixed the previous error all hyperlinks work properly

Gasal - TA.2022/2023
14




1st Phase
(lanjutan)

TEST PLAN
(SISTEM PERPUSTAKAAN UNIVERSITAS XYZ)

Log In		Prepared By: Eghie	
Purpose / Goal	Untuk memastikan adanya pembatasan akses dan sistem log in dapat berjalan dengan baik		
Function Description	Sistem Peminjaman (Interface 1-2)		
Approach	<ol style="list-style-type: none"> Aktifkan kursor pada bagian Text Box KodePetugas Isikan Kode Petugas pada Text Box tersebut Klikkan kursor pada bagian Text Box PIN Isikan PIN pada Text Box tersebut Tekan tombol Masuk 		
Criteria / Data Test	Criteria	Standar	Status
	1. Kode Petugas diinputkan dengan 12345	1. Tampilan : 12345	Pass
	2. PIN diinputkan dengan 11111	2. Tampilan : *****	Pass
	3. Tekan tombol "Masuk"	3. Tombol peminjaman dan pengembalian aktif	Pass
	4. Kode Petugas diinputkan dengan AAAAA	4. Tampilan : (Harus dalam bentuk numeric)	Failed (Tidak bisa login, kembali ke form login ulang)
Notes	-		
	Tested By: Eghie		

Gasal - TA.2022/2023
15



1st Phase
(lanjutan)

Test Plan Template

Test Plan Template

Overview

Bounds

Scope

Definitions

Setting

Quality Risks

Proposed Schedule of Milestones

Test System Development

Test Execution

Key Participants

Test Case and Bug Tracking

Bug Isolation and Classification

Release Management

Test Cycles

Test Hours

Risks and Contingencies

Change History

Referenced Documents

Frequently Asked Questions

IEEE 829 Test Plan Template

Test Plan Identifier

Introduction

Test Items

Features To Be Tested

Features Not To Be Tested

Approach

Item Pass/Fail Criteria

Suspension Criteria And Resumption

Requirements

Test Deliverables

Testing Tasks

Environmental Needs

Responsibilities

Staffing And Training Needs

Schedule


Risks And Contingencies

Approvals

- Biasanya *test plan* dibuat berdasarkan *template* yang ada/disediakan.
- Berikut adalah *template* yang biasa digunakan adalah *template* yang dikeluarkan IEEE 829 (Institute of Electrical and Electronics Engineers)

Figure 2.1 A test plan template.
Figure 2.7 IEEE 829 Standard test plan template.

Gasal - TA.2022/2023
16




1st Phase (lanjutan)

Test Plan Identifier

- *Test Plan Identifier* adalah bagian untuk menjelaskan secara singkat mengenai objek yang akan di *test*.
- Dapat berupa penjelasan narasi atau berbentuk tabel dengan kategori-kategori tertentu.
- Informasi yang dijelaskan dapat berupa sekilas mengenai subjek *testing*, nama orang yang bertanggung jawab terhadap *testing*, penyusun *test plan*, tanggal dibuat *test plan* dan tanggal revisi, dll.

Gasal - TA.2022/2023 17




1st Phase (lanjutan)

- Contoh *Test Plan Identifier* pada modul soal di sistem informasi sekolah.

Subjek Testing	Cara menginput soal dan cara mengerjakan soal.
Penanggung Jawab	Joko
Penyusun	Budi
Tanggal Pembuatan	10 April 2018
Tanggal Revisi	16 April 2018

Gasal - TA.2022/2023 18




1st Phase (lanjutan)

Introduction

- Pada bagian *introduction* dibuat untuk menjelaskan secara narasi, mengenai *testing* yang akan dilakukan terhadap suatu objek *testing*.
- Bagian *Introduction* dapat dibuat lebih rinci dengan menambahkan sub-bab apabila perlu untuk dibuat.
- Contoh sub-bab yang dapat dibuat antara lain:
 - *Purpose* : Untuk menjelaskan tujuan testing secara spesifik.
 - *Background* : Latar belakang mengapa testing dilakukan.
 - *Scope* : Sejauh mana testing dilakukan.
 - *Definition and Acronyms* : Penjelasan mengenai singkatan dan istilah yang ada di dalam dokumen test plan.

Gasal - TA.2022/2023 19



1st Phase (lanjutan)

Test Items

- Bagian dari *test item* menjelaskan mengenai daftar komponen komponen dalam objek *testing* yang akan di *test* satu per-satu.
- Contohnya :

1. Functional Testing
2. Interface Testing
3. Database Testing
4. Compatibility Test
5. Performance Test
6. Security Test
7. Usability Test

Gasal - TA.2022/2023 20

1st Phase
(lanjutan)

The interfaces between the following subsystems will be tested:

- Manage Gallery
- Manage News and Events
- Manage Program
- Manage Semester
- Attendance
- Manage Course
- Manage Faculty
- Manage Batch
- Manage Student
- Achievements
- Club
- Profile
- Document
- Attendance
- Project and Theses
- Manage Group
- Manage Exam
- Manage Admission

The basic performance test:

- Add album
- Modify album
- Upload picture
- Delete picture
- Add achievements
- Add news/events
- Modify news/events
- Create program
- Modify program
- Create semester
- Modify semester
- Create course
- Modify course
- Add faculty member
- Modify faculty member
- Assign course
- Add Batch
- Modify Batch
- Add Student
- Modify Student
- Add club
- Modify club
- Create Profile
- Edit Profile
- Upload document
- Download document
- Delete document
- Attendance
- Add project/thesis
- Add group
- Create admission form

Gasal - TA.2022/2023 21

1st Phase
(lanjutan)

Features to be Tested

- Penjelasan dan daftar-daftar fitur yang akan di-test pada saat pelaksanaan *testing* dimulai.
- Penjelasan dan daftar fitur tsb dapat berupa tabel, seperti contoh berikut :

Features	Priority	Description
Outgoing Link (3)	2	Menguji link-link aplikasi atau website lain yang terhubung
Internal Link (5)	2	Menguji link menu yang ada pada website
Mail to Link	3	Menguji email yang terkirim
Test Form (3)	1	Menguji form yang terdapat pada website
Cookies	3	Cookies update dengan aktivitas website atau tidak
Bussiness Work Flow	1	Menguji urutan kerja

Gasal - TA.2022/2023 22

1st Phase
(lanjutan)

Features	Priority	Description
Add album	3	To show the photos of different events of IIT in an arranged order user can add an album
Modify album	3	Edit an existing album by adding or deleting photos or changing the description of album
Add achievements	3	Different achievements of IIT will be listed
Add news/events	1	This will work as notice board of IIT
Modify news/events	1	Change any components of the existing news or events
Upload picture	3	Upload photo in the album
Delete picture	3	Delete photo from the album
Create Program	1	Create program such as bachelor, masters
Modify Program	1	Edit an existing program
Create Semester	1	Create semesters for each program
Modify Semester	1	Edit an existing semester
Create Course	1	Create course under every semester

Gasal - TA.2022/2023 23


1st Phase
(lanjutan)

Features Not to be Tested

- Menjelaskan mengenai fitur-fitur apa saja yang ada di dalam objek *testing* namun, fitur tersebut tidak akan di-test pada saat pelaksanaan *testing* dan disertakan penjelasan singkat mengapa fitur tersebut tidak di test pada saat testing.
- Contoh :


Features	Description
White Box	Tidak dapat menguji codingan dari website orang lain.

Gasal - TA.2022/2023 24

 **1st Phase**
(lanjutan)

Features	Description
Registration	It needs not to be tested because registration will be done by admin manually.
Password Recovery	It can be done manually by admin. As there is a limited schedule we will skip this feature.
Change User Status	It will be done by admin manually, so this feature needs not to be tested.
Network Security	Testing network security is out of our scope.


Gasal - TA.2022/2023 25

 **1st Phase**
(lanjutan)

Approach / Test Strategy

- Bagian *Approach* adalah bagian yang digunakan untuk memberi deskripsi mengenai pendekatan yang dilakukan untuk melaksanakan *testing* dan disertakan dengan penjelasan mengenai *approach*/pendekatan yang digunakan.
- Contohnya :
 - *Unit Testing* menggunakan *Black Box Testing* dan atau *White box testing* beserta dengan penjelasan mengenai *approach*/ pendekatan yang digunakan.
 - *System testing* dengan *Performance testing* dan *Load testing*, dll.


Gasal - TA.2022/2023 26

 **1st Phase**
(lanjutan)

Item Pass / Fail Criteria

- Berisi tentang kriteria-kriteria yang harus dipenuhi sebelum berlanjut ke fase berikutnya (*item pass*), dan kriteria testing yang tidak dapat dipenuhi (*fail criteria*).
- Contoh :
 - 1) Jika suatu item di-*test* sebanyak 10 kali dan 9 kali diantaranya berhasil namun ada 1 dimana benar benar gagal maka item tersebut dinyatakan sebagai gagal/*fail*.
 - 2) Jika hasil dari suatu item sama dengan hasil yang diharapkan maka item tersebut dinyatakan berhasil/*pass*.
 - 3) *System Crash* akan dinyatakan sebagai *fail*, dsb.


Gasal - TA.2022/2023 27

 **1st Phase**
(lanjutan)

Suspension Criteria

- Berisi tentang spesifikasi kriteria-kriteria yang dapat digunakan untuk menghentikan sementara kegiatan *testing*, dan kegiatan *testing* tersebut dapat dilanjutkan di waktu lain.
- Contoh:
 - Beberapa *individual test cases* dapat di-*suspend*, di-*skip*, atau bahkan dikurangi, jika *test* yang diperlukan (*prerequisite test*) untuk *individual test cases* tersebut *fail* dan memerlukan perbaikan.
 - Apabila *individual test cases* tetap dilakukan meskipun *prerequisite test* yang diperlukan *fail*, maka hanya akan membuang sumber daya/*resources* saja.


Gasal - TA.2022/2023 28

 **1st Phase**
(lanjutan)

Test Deliverables

- *Test Deliverables*, adalah *list*/daftar yang berisi dokumen-dokumen apa saja yang akan dihasilkan setelah *testing* selesai dilakukan.
- Contohnya:
 - Master test plan (this document)
 - Individual test plans for each phase of the testing cycle
 - Test Design Specifications
 - Test log for each phase
 - Acceptance Test plan.
 - Unit test plan.
 - Screen Prototypes.
 - Test report.
 - Test scenario and expected result in an excel sheet.
 - System manual.

29


 **1st Phase**
(lanjutan)

Testing Task

- *Testing task*, menjelaskan kegiatan testing beserta dengan pihak yang akan melaksanakan kegiatan/*task* tersebut.
- Contoh:

Task	Assigned To
Create Acceptance Test Plan	Project Manager, Clients
Create Integration Test Plan	Developers, Project Manager
Define Unit Test rules and Procedures	Developers, Project Manager
Define Turnover procedures for each level	Developers
Verify prototypes of Screens	Clients, Project Manager
Verify prototypes of Reports	Clients, Project Manager

30

 **1st Phase**
(lanjutan)

Enviromental Needs

- Spesifikasi dan perincian segala sesuatu yang dibutuhkan dan digunakan selama proses *testing* berjalan, dapat berupa *hardware* yaitu spesifikasi komputer, atau hal lain selain *hardware*, misalnya izin akses seperti :
 - Membutuhkan akses ke dalam *database* untuk melaksanakan *testing*.
 - Membutuhkan akses ke dalam *official website*.


Client
Komputer netbook dengan spesifikasi:

- Processor Intel Atom N450 1.66GHz
- Memory 1 Gb
- Hardisk free 78 Gb
- Screen Monitor 10"
- Resolusi Monitor 1024 x 600
- Browser Mozilla Firefox

Server

- Komputer personal unit dengan spesifikasi:
- Memory 2 Gb
- Hardisk free 50 Gb
- Screen Monitor 16"
- Resolusi Monitor 1024 x 600
- Websrver apache
- Database MySql
- Browser Mozilla Firefox

31

 **1st Phase**
(lanjutan)


Responsibilities

- Berisi rincian pihak-pihak yang akan bertanggung jawab terhadap suatu kegiatan *task* di dalam serangkaian kegiatan *testing* yang akan dilaksanakan.

Responsibilities

Task	Penanggungjawab
Membuat Test Plan	Sulistiyono
Membuat Design Test	Muhammad Nasrulloh
Melakukan Testing	Sulistiyono, Muhammad Nasrulloh
Membuat Dokumen Final	Sulistiyono, Muhammad Nasrulloh

32

 **1st Phase**
(lanjutan)


Staffing and Training Needs

- Secara garis besar menjelaskan bagaimana melakukan pendekatan untuk menentukan peran para staff di dalam proyek, dan melakukan *training* apabila diperlukan untuk *testing*.

• Contoh: **I. Staffing and Training Needs**

Project Manager	bertanggung jawab mengatur implementasi total dalam website
Test Enginner	bertanggung jawab merancang desain dari test, membuat prosedur testing , membuat test data, dan menjalankan test.
Test Manager	bertanggung jawab melakukan pengecekan atau testing terhadap bug di dalam sebuah aplikasi atau program.

Gasal - TA.2022/2023 33


 **1st Phase**
(lanjutan)

Schedule

- Terdapat beberapa tujuan didalam membuat *schedule* pada *test plan*, antara lain:
 - Merincikan tolak ukur waktu pengerjaan *testing*.
 - Merincikan *event transmittal item*.
 - Estimasi waktu yang dibutuhkan untuk setiap *task*.
 - Menjadwalkan *Testing task* dan *Test Milestone*.
 - Merincikan periode pemakaian *testing resources*.

Fase Test	Penanggung Jawab	Waktu
Test Plan Creation	Test Manager	07/04/2018
Test Specification Creation	Test Leads	09/04/2018
Test Spec. Team Review	Project Team	10/04/2018
Unit Testing	Developer	Waktu pengembangan
Component Testing	Testing Team	12/04/2018
Integration Testing	Testing Team	15/04/2018
Use Case Validation	Testing Team	20/04/2018
User Interface Testing	Testing Team	23/04/2018
Load Testing	Testing Team	25/04/2018
Performance Testing	Testing Team	28/04/2018
Release To Production	Project Team	30/04/2018

Gasal - TA.2022/2023 34

 **1st Phase**
(lanjutan)

Risk and Contingencies

- Digunakan untuk memastikan agar hasil *testing* tetap berkualitas dengan memeriksa beberapa bagian yang tidak termasuk di dalam control pengerjaan *software*, namun bagian tersebut dapat berdampak langsung terhadap proses.

• Contoh :

- Sekuritas Layanan ketika masuk di dalam *software* masih belum diperhatikan, ketika *software* sedang dijalankan, belum ada proteksi keamanan oleh *software* ini sendiri sehingga akan sangat mudah apabila diinjeksi oleh pihak yang tidak bertanggung jawab.

Gasal - TA.2022/2023 35


 **1st Phase**
(lanjutan)

Approvals

- Approvals*, merupakan lembar persetujuan sebagai tanda bahwa seluruh tim/pimpinan telah menyetujui *Test Plan* yang telah dibuat.

Persetujuan				
Nama	Versi yang Disetujui	Jabatan	Tanggal	Tanda Tangan
Sulistiyono, Muhammad Nasrulloh	TP V.2	Pembuat	4/16/2012	
Sulistiyono, Muhammad Nasrulloh	TP V.3	Pembuat	4/17/2012	
Sulistiyono, Muhammad Nasrulloh	TP V.4	Pembuat	4/18/2012	
Sulistiyono, Muhammad Nasrulloh	TP V.5	Pembuat	4/19/2012	

Gasal - TA.2022/2023 36

 **1st Phase**
(lanjutan)

b) Test Case

- *Test case* adalah serangkaian tes yang digunakan untuk menentukan apakah satu hal tertentu bekerja dengan baik.
- Seringkali, itu berarti mencoba operasi yang sama berulang-ulang dengan sedikit dalam prosedur.
- Tes yang dilakukan biasanya meliputi:
 - White Box Testing, dan
 - Black box Testing.

Gasal - TA.2022/2023 37








Pengujian Dokumentasi & Fasilitas Help


Pengujian Dokumentasi: Fase Kedua

Gasal - TA.2022/2023 38

 **2nd Phase**

- Fase kedua, yaitu *live test* menggunakan dokumentasi dalam kaitannya dengan penggunaan program aktual.
- *Live test* untuk dokumentasi dapat didekati dengan menggunakan teknik yang analog dengan berbagai metode pengujian *black box*.
- Pengujian *graph-based* dapat digunakan untuk menggambarkan penggunaan program tersebut; partisi ekuivalensi dan analisis nilai batas dapat digunakan untuk menentukan berbagai kelas input dan interaksi yang sesuai.

Gasal - TA.2022/2023 39

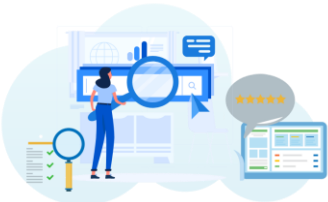
 **2nd Phase**
(lanjutan)

- Penelusuran dokumen dapat berupa:
 - Apakah penjelasan pada dokumen akurat ?
 - Apakah tabel dan isi dokumen sudah lengkap ?
 - Apakah desain dokumen mudah dipahami ?
 - Apakah semua pesan kesalahan ditampilkan bagi *user* dan digambarkan secara detail pada dokumen ?
 - Apakah contoh-contoh pada dokumen help akurat ?
 - Apakah *link hypertext* pada dokumen berfungsi ?

Gasal - TA.2022/2023 40

2nd Phase
(lanjutan)

- Satu-satunya cara yang dapat berjalan untuk menjawab pertanyaan-pertanyaan tersebut adalah dengan menggunakan bagian ketiga yang independen (misal : pengguna yang telah diseleksi), yang menguji dokumentasi di dalam konteks kegunaan program.
- Semua diskrepansi (ketidakcocokan / ketidaksesuaian) dicatat, dan area ambiguitas atau kelemahan dokumen ditentukan untuk penulisan ulang yang potensial.



Gasal - TA.2022/2023 41

Pengujian Dokumentasi & Fasilitas Help

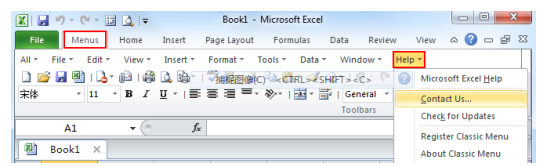

Fasilitas Help



Gasal - TA.2022/2023 42

Fasilitas Help

- Untuk menguji fasilitas *help* dapat menggunakan cara manual.

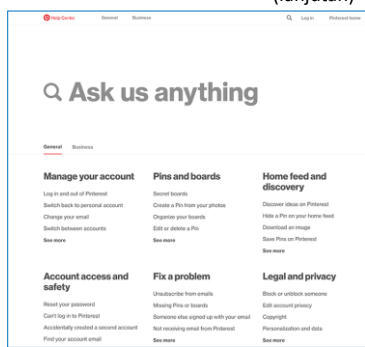
- Hal - hal yang dapat diuji dalam fasilitas *help*, antara lain :
 - FAQs (*Frequently Asked Questions*)
 - Fasilitas *admin online*
 - Fasilitas *search*

Gasal - TA.2022/2023 43

Fasilitas Help
(lanjutan)

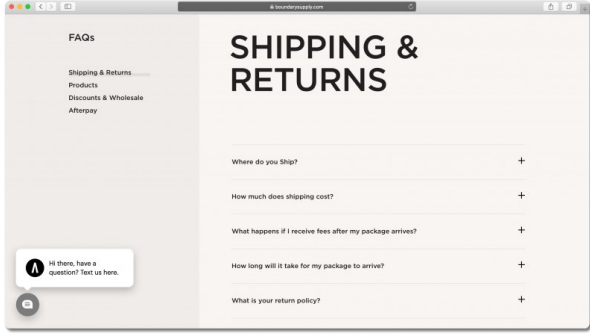
a) FAQs (*Frequently Asked Questions*)

- Pada bagian FAQs, apakah sudah diisi dengan jawaban yang sesuai dengan prosedur sebenarnya.



Gasal - TA.2022/2023 44

Fasilitas Help
(lanjutan)

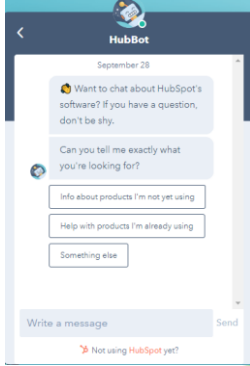


Gasal - TA.2022/2023 45

Fasilitas Help
(lanjutan)

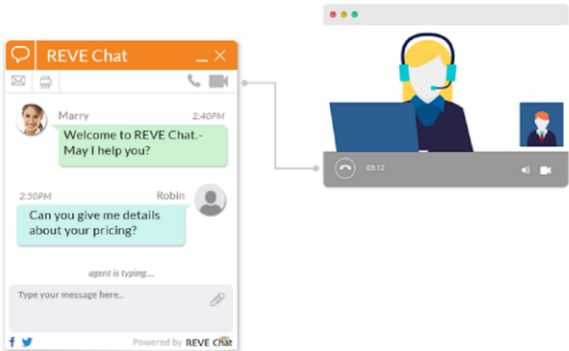
b) *Admin online*

- Fasilitas untuk bertanya dan memberi saran (*admin online*), apakah memberikan jawaban yang jelas sesuai dengan kebutuhan user.



Gasal - TA.2022/2023 46

Fasilitas Help
(lanjutan)



Gasal - TA.2022/2023 47

Fasilitas Help
(lanjutan)

c) *Search*

- Fasilitas *search* memberikan hasil sesuai dengan *query* yang diberikan, dan memberikan rekomendasi data sesuai dengan *query*.



Gasal - TA.2022/2023 48

Fasilitas Help
(lanjutan)

Gasal - TA.2022/2023 49

Referensi

- O'Regan, Gerard., 2019. *Concise Guide to Software Testing - Undergraduate Topics in Computer Science*, 1st edition, Springer.
- Naik, Kshirasagar., Tripathy, Priyadarshi., 2008. *Software Testing and Quality Assurance - Theory and Practice*, 1st edition, Wiley.
- Galin, Daniel., 2003. *Software Quality Assurance - From Theory to Implementation*, 1st edition, Pearson.
- Romeo, 2003. *Testing dan Implementasi Sistem*, Edisi Pertama, STIKOM.

Gasal - TA.2022/2023 50

JADILAH PELOPOR KEBAIKAN!

Rasulullah ﷺ bersabda,
 "Barangsiapa melakukan suatu amalan kebaikan lalu diamalkan oleh orang sesudahnya, maka akan dicatat baginya ganjaran semisal ganjaran orang yang mengikutinya dan sedikitpun tidak akan mengurangi ganjaran yang mereka peroleh. Sebaliknya, barangsiapa melakukan suatu amalan kejelekan lalu diamalkan oleh orang sesudahnya, maka akan dicatat baginya dosa semisal dosa orang yang mengikutinya, tanpa mengurangi dosanya sedikit pun."
 (HR. Muslim, no. 1017)

Terima Kasih