

Analisa perancang sistem informasi

Flowchart

1 jenis flow chart

- system flowchart bagan yang menunjukkan arus pekerjaan
- documen flowchart bagan yang menunjukkan laporan
- schematic flowchart bagan yang menggambarkan proesur diaam system
- program flowchart bagan yang menjelaskn secara rinci langkah langkah dari proses progam
- proses flowchart bagan berguna bagi analisis proses sistem untuk mengga,barkan proses dalam prosedur

Simbol-Simbol Flowchart / Bagan Alir

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

SDLC adalah system development life cycle merupakan langkah langkah dan metodologi ynaq harus dilakukan dalam mengembangkan software/ perangkat lunak.

Tahapan sdlc

1. project planning (menentukan jadwal kseluruhan proyek)
2. system analyst (menggali sedalam2nya kebutuhan user)
- 3.system design (pembuatn blue print sistem)
4. implementation (tahapa koding)
5. intregation dan testing (seluruh modul diintergrasikan secara lengkap dan di uji coba)
6. maintenana (menganalisa apakah aplikasi selama berjalana di production apakah timbul masalh atau tidak)

1. Planning: Tahap ini meliputi pengumpulan kebutuhan pengguna, analisis bisnis, estimasi biaya dan waktu, dan perencanaan proyek secara keseluruhan.
2. Analysis: Tahap ini meliputi pengumpulan dan analisis kebutuhan pengguna, perencanaan sistem, dan perancangan arsitektur perangkat lunak.
3. Design: Tahap ini meliputi perancangan detail dari sistem perangkat lunak, termasuk desain antarmuka pengguna, desain basis data, dan desain algoritma.
4. Implementation: Tahap ini meliputi pengembangan kode perangkat lunak, pengujian, dan integrasi dengan sistem yang sudah ada.
5. Testing: Tahap ini meliputi pengujian perangkat lunak secara menyeluruh, termasuk pengujian fungsional, pengujian performa, dan pengujian keamanan.
6. Deployment: Tahap ini meliputi instalasi perangkat lunak, pengaturan konfigurasi, dan peluncuran produk secara keseluruhan.
7. Maintenance: Tahap ini meliputi pemeliharaan perangkat lunak, termasuk pembaruan, perbaikan bug, dan dukungan pelanggan.

Metode sdlc

1. metode waterfall (air terjun)

Metode ini merupakn penegmbangan perangkat lunak tertua sebab sifatnya yang natural.

Urutan dalam metode waterfall = proses perncanaan, analisa , desain, pengkodean progam, uji coba progam dan implementasi dan pemeliharaan sistem

2. metode iterative (pengulangan)

Metode ini merupakn metode prototypinf model dan digunakn ketika requirement dari software aknterus berkembang dalam tahapan tahapan pengembangan aplikasi.

Type dari iterative model ini

- spiral model (dikembangkan dari sifat iterative prototyping dan sifat kinier waterfall)
- win win spiral model (model yang merupakn ektensi dari spiral model tim penegmbang dan pelanggan aka melakukam diskusi dan negoisasi terhadap requiremennya.
- component based developmen model (menitikberatkan pada penggunaan kembali dari kompenn2 ayng dibangun dalm sebuah aplikasi)

3 metode spiral boehm

Model ini mengadopsi dua model perangkat lunak yang ada yaitu model prototyping dengan pengulangnaya dan model waterfall dengn pengedalian dan sistematikanya.

4. metode fountain (airmancur) merupakan perbaikan logis dari model waterfll , langkah2 sama dengan waterfall

5. rapid application development (rad) adalah strtegi siklus hidup yang di tunjukan untuk menyedyiakan pengembangan yang jauh lebih cepat dan mendapatkan hasil dengn kualitas yg lebih baik.

6. metode BUID dan FIX. Adalah metode yang paling lemah dianatra metode sdlc yng lain, metode ini bertujuan memberikan kepercayaan terhadap pelanggan dengn cara meberrikan pelayann perbaikan dan perawatan terus menerus,

7. metode synchronize and stabilize model yang dgunakan mikrosoft model ini sama dengan model incremetal

8 metode extreme programing/ xp adalah model pengembangan perangkat lunak yng menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien adaptf, dan flexible.

9. metode agile adalah model pengembvngn jangka pendek yang memerlukan adaptasi cepet dan pengembangan terhadap perubahan dalam bentuk apapun.

10. metode v model pelaksana proses yang terjadi secara berurutan dalm bentuk v. dikenal juga sebgai model verifikasi dan validasi

11 model prototype merupakan suatu parakdigma baru dalam pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengenmbangan perangkat lunak tetepi juga merevolusi metode pengembangan perangkat lynak lama yaitu sistem skuensial yang biasa dikenal dengan sdlc atau waterfall model

12 model rational unified proses (rup)

Rup adalah metodologi pengembangan sistem berbasis objek
RUP adalah singkatan dari Rational Unified Process, yaitu sebuah metodologi pengembangan perangkat lunak yang dikembangkan oleh IBM. RUP merupakan sebuah proses iteratif dan incremental yang membantu tim pengembangan perangkat lunak dalam menentukan, merancang, membangun, dan mengirimkan produk perangkat lunak yang berkualitas. RUP memfokuskan pada perencanaan dan pengendalian proses pengembangan, memastikan bahwa produk perangkat lunak memenuhi kebutuhan klien dan memastikan bahwa proses pengembangan berjalan dengan efisien dan efektif.



SDLC (Software Development Life Cycle) adalah suatu metodologi untuk mengembangkan perangkat lunak. Berikut adalah beberapa macam metode SDLC:

1. Waterfall: Metode ini mengikuti aliran satu arah dari tahap analisis, perancangan, implementasi, pengujian, hingga pemeliharaan. Setiap tahap harus selesai sebelum masuk ke tahap berikutnya.
2. Agile: Metode ini berfokus pada kolaborasi antara tim pengembang dan pemangku kepentingan, dengan memecah proyek menjadi beberapa siklus pengembangan yang disebut "sprint". Setiap sprint menghasilkan produk yang dapat diuji dan diuji coba.
3. Prototype: Metode ini melibatkan pembuatan prototipe perangkat lunak sebelum membuat produk akhir. Prototipe ini digunakan untuk mengidentifikasi masalah dan mengevaluasi kelayakan dan fungsionalitas produk.
4. Spiral: Metode ini menggabungkan elemen dari model waterfall dan model prototyping. Proses pengembangan dibagi menjadi siklus, dengan setiap siklus yang meliputi tahap analisis, perancangan, implementasi, dan pengujian.
5. RAD (Rapid Application Development): Metode ini fokus pada pengembangan produk dalam waktu yang singkat, dengan memprioritaskan waktu dan biaya sebagai faktor utama. Ini dilakukan dengan menggabungkan prinsip-prinsip Agile dan prototyping untuk mencapai produk akhir dengan cepat.

Setiap metode SDLC memiliki kelebihan dan kelemahan, dan pemilihan metode yang tepat harus didasarkan pada kebutuhan proyek dan kondisi organisasi.

1. V-Model: Metode ini mengikuti aliran satu arah dari tahap analisis ke tahap verifikasi, dan kemudian dari tahap verifikasi ke tahap validasi. Metode ini menekankan pada pengujian produk perangkat lunak yang lebih menyeluruh dan lebih ketat.
2. Incremental: Metode ini mengembangkan produk secara bertahap dengan menambahkan fungsi baru pada setiap tahap. Setiap tambahan fungsi diuji dan diverifikasi sebelum tahap berikutnya dimulai.
3. DevOps: Metode ini mengintegrasikan pengembangan perangkat lunak dan operasi dalam suatu proses yang terus-menerus, dengan fokus pada otomatisasi, kolaborasi, dan pengiriman produk yang cepat dan stabil.
4. Lean: Metode ini berfokus pada mengurangi pemborosan dalam proses pengembangan perangkat lunak. Proses pengembangan disederhanakan untuk mengurangi waktu dan biaya, sementara fokus ditempatkan pada kualitas produk.
5. Iterative: Metode ini mengembangkan produk dalam beberapa siklus yang berulang-ulang. Setiap siklus mencakup tahap analisis, perancangan, implementasi, dan pengujian, dan kemudian siklus diulang untuk mengembangkan produk lebih lanjut.

Seperti yang disebutkan sebelumnya, pemilihan metode SDLC yang tepat harus didasarkan pada kebutuhan proyek dan kondisi organisasi. Setiap metode memiliki kelebihan dan kelemahan masing-masing, dan harus dipilih berdasarkan kebutuhan dan tujuan proyek yang diinginkan.

END USER COMPUTING

End user computing adalah (EUC) adalah salah satu metode pengembangan sistem berbasis komputer yang dilaksanakan oleh pemakai sendiri.

Kelebihan

- 1 pembuatan oleh user
2. sistem mengikuti kebutuhan user
3. cepat
4. Memungkinkan it mengerjakan tugas lain
5. flexibel

TEKNIK PENGUJIAN PERANGKAT LUNAK

Software testing merupakan sebuah metode untuk mengetahui apakah software yang dibuat, berfungsi dengan baik dan benar. Tanpa adanya software testing, kita tidak dapat mengetahui apakah sebuah software sudah memenuhi semua kriteria yang dibutuhkan untuk user. Software testing sendiri dilakukan berdasarkan SDLC (Software Development Life Cycle), dengan kata lain kita melakukan eksekusi program dan selama proses yang dilakukan, memungkinkan kita untuk menemukan kesalahan atau error.

Dalam software testing, ada siklus yang harus dilewati :

- a. Analisis Kebutuhan : Melakukan analisa dari fase SDLC yang dilalui oleh software
- b. Analisis Desain : Analisis ini merupakan bentuk analisis desain dimana melakukan analisa bagian desain dan parameter yang perlu dilakukan testing
- c. Test Desain : Menyempurnakan strategi testing yang akan dilakukan
- d. Run Test : Menjalankan testing dan mencari error, bug dan kesalahan yang ada
- e. Report Test : Memberikan report dari hasil testing kepada developer dan memberikan konklusi apakah software tersebut layak digunakan atau tidak.

WHITE-BOX TESTING

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

Teknik White-box Testing

- a. Basis Path Testing : Metode yang memungkinkan perancang testcase untuk membuat pengukuran kompleksitas logikal dari rancangan prosedural dan menggunakan pengukuran ini sebagai panduan untuk mendefinisikan himpunan basis dari jalur eksekusi. Test case yang dibuat untuk menguji himpunan basis dijamin akan mengeksekusi setiap statement di dalam program sekurangnya sekali pada saat pengujian
- b. Flow Graph : Notasi sederhana untuk merepresentasi control flow.
- c. Cyclomatic Complexity : Metric software yang menyediakan ukuran kuantitatif dari kompleksitas logikal program. Cyclomatic complexity digunakan untuk mengetahui jumlah jalur yang perlu dicari. 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.

(+) Kelebihan :

- Efisien dalam menemukan kesalahan dan masalah
- Diperlukan pengetahuan tentang internal perangkat lunak yang sedang diuji
- Bermanfaat untuk pengujian menyeluruh
- Memungkinkan menemukan kesalahan tersembunyi
- Membantu mengoptimalkan kode

(-) Kekurangan :

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

BLACK-BOX TESTING

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

Teknik Black-box Testing

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

(+) Kelebihan :

- Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu
- Pengujian yang dilakukan berdasarkan sudut pandang user agar dapat mengungkapkan inkonsistensi atau ambiguitas dalam spesifikasi.
- Programmer dan tester memiliki ketergantungan satu sama lain

(-) Kekurangan :

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

GREY-BOX TESTING

Grey Box Testing merupakan kombinasi dari Black Box dan White Box Testing, menguji perangkat lunak berdasarkan spesifikasi tetapi menggunakan cara kerja dari dalam sehingga membuat Grey Box dapat digunakan dengan baik dalam melakukan pengujian perangkat lunak.

Grey-box Testing mengacu pada suatu teknik pengujian sistem dengan pengetahuan yang terbatas dari suatu internal sistem. Pada pengujian grey box testing memiliki suatu akses ke desain dokumen dengan rinci melalui informasi di luar dari persyaratan dokumen. Grey box testing yang akan dihasilkan berdasarkan suatu informasi tersebut sebagai suatu state-based models atau architecture diagrams of the target system.

Teknik yang ada dalam pengujian Grey Box Testing, yaitu :

1. Matrix Testing berfungsi untuk menyatakan laporan atau status dari proyek
2. Regression testing berfungsi menyatakan status yang akan terjadi perubahan dalam kasus uji yang baru dibuat.
3. Pattern testing berfungsi untuk memverifikasi aplikasi yang baik untuk desain atau arsitektur dan pola.
4. Orthogonal array testing berfungsi sebagai bagian dari semua kemungkinan kombinasi.

(+) Kelebihan :

- Offer combined benefits merupakan pengambilan kelebihan dari White Box dan Black Box Testing, untuk melakukan percobaan terhadap pengujian
- Non Intrusive merupakan hal yang didasarkan pada spesifikasi fungsional dan tampilan arsitektur
- Unbiased Testing yang terlepas dari semua keuntungan yang ada di atas, dikarenakan Grey Box Testing mempertahankan batas terhadap pengujian antara tester dan developer

(-) Kekurangan :

- Partial Code Coverage dalam pengujian grey box merupakan sumber kode dan binary yang akan hilang karena akses yang terbatas pada struktur internal atau perangkat lunak yang akan menghasilkan akses terbatas terhadap kode path transversal.
- Defect Identification dalam perangkat lunak terdistribusi, sulit untuk mengidentifikasi bug.

SUSUNAN PENGUJIAN PERANGKAT LUNAK

ANALISA & PERANCANGAN
SISTEM INFORMASI | 33

Pada pengujian memiliki beberapa urutan yang perlu dilakukan. Berikut ini kategori dari pengujian perangkat lunak yang dapat disusun secara kronologis, yaitu:

Unit Testing merupakan pengujian yang dapat dilakukan pada setiap modul atau blok kode selama pengembangan dan bagian pengujian biasanya dilakukan oleh developer yang menulis kode tersebut.

Integration Testing merupakan pengujian yang dilakukan sebelum, selama dan setelah melakukan integrasi modul baru ke dalam sebuah paket perangkat lunak utama. Pengujian ini dapat melibatkan pengujian setiap modul kode ke beberapa individu. Satu perangkat lunak dapat berisikan beberapa modul yang sering dibuat oleh beberapa developer yang berbeda.

System Testing merupakan pengujian yang dapat dilakukan oleh agen pengujian yang profesional. Terdapat pada produk perangkat lunak yang telah selesai sebelum perangkat lunak tersebut dapat diperkenalkan ke umum.

Acceptance Testing merupakan pengujian beta dari beberapa produk yang dilakukan oleh pengujian akhir sebenarnya. Dalam Acceptance Testing sering kita kenal Alpha Testing dan Beta Testing.

- **Alpha testing** adalah pengujian yang dilakukan oleh user pada lingkungan pengembangan. Pengujian alpha berlangsung di situs pengembang oleh tim internal, sebelum rilis kepada pelanggan eksternal. Agar nantinya ketika pelanggan menggunakan sistem ini tidak kecewa karena masalah cacat atau kegagalan aplikasi. Pengujian ini dilakukan tanpa keterlibatan tim pengembangan. Selain itu, alpha testing sering digunakan untuk software sebagai bentuk penerimaan internal sebelum software menuju beta testing.
- **Beta testing** merupakan metode untuk memeriksa dan mengesahkan suatu software. Beta testing digunakan untuk menggambarkan proses testing external dimana software dapat diedarkan kepada orang lain seperti user yang berpotensi menggunakan software untuk kehidupan sehari-hari. Beta testing biasanya berpengaruh pada tahap akhir pengembangan software dan biasanya menjadi suatu pengesahan bahwa software sudah siap untuk digunakan oleh user. Tujuan dari beta testing dapat beraneka ragam, seperti kesempatan media pers untuk menuliskan masukan dari user untuk mengatasi bugs dan kesalahan yang ada.

JENIS PENGUJIAN PERANGKAT SISTEM

ANALISA & PERANCANGAN
SISTEM INFORMASI | 34

Pada pengujian sistem terdapat 50 jenis pengujian dan beberapa diantaranya sering digunakan oleh perusahaan pengembang software atau perangkat lunak besar. Berikut ini jenis pengujian yang sering digunakan, yaitu :

Usability Testing merupakan pengujian yang berfokus pada kemudahan pengguna dalam menggunakan aplikasi, lebih fleksibilitas dalam menangani kontrol dan kemampuan pada sistem untuk memenuhi tujuan dari perangkat lunak tersebut.

Load testing merupakan pengujian yang bertujuan untuk mengetahui cara kerja sebuah sistem dengan beban nyata.

Regression Testing merupakan pengujian yang dilakukan untuk dapat memastikan bahwa tidak ada perubahan yang dibuat selama proses pengembangan dapat menyebabkan bug baru muncul.

Recovery Testing merupakan pengujian yang dilakukan untuk menunjukkan suatu solusi dari sistem dapat diandalkan, dipercaya dan dapat berhasil untuk menutup kemungkinan terjadinya crash pada perangkat lunak.

Migration Testing merupakan pengujian yang dilakukan untuk memastikan bahwa sistem dapat dipindahkan dari beberapa infrastruktur sistem lama ke infrastruktur sistem yang baru tanpa terjadi masalah.

Functional Testing/Completeness Testing merupakan pengujian yang memerlukan pemikiran mengenai kemungkinan untuk terjadinya fungsi yang hilang. Pengujian ini dapat membuat daftar fungsional tambahan yang bisa dikembangkan oleh suatu produk selama proses pengujian.

Hardware/Software Testing merupakan pengujian yang terjadi ketika pengujian berfokus pada interaksi antara perangkat keras dan perangkat lunak selama proses pengujian sistem.

Jenis-jenis Stress Test

Jenis-jenis Stress Test pada Software

Stress test pada software bertujuan untuk menguji ketahanan software dalam beban kerja yang berat. Berikut ini adalah beberapa jenis stress test yang dapat dilakukan:

- **Load test** - Menguji kinerja software dalam beban pemakaian yang tinggi, seperti jumlah pengguna yang banyak secara simultan.
- **Volume test** - Menguji ketahanan software dalam jumlah data yang sangat besar, seperti database yang terlalu banyak data.
- **Endurance test** - Menguji ketahanan software dalam jangka waktu penggunaan yang lama secara berulang-ulang.
- **Spike test** - Menguji ketahanan software dalam kenaikan tiba-tiba jumlah beban kerja yang tinggi dan kemudian turun secara singkat.
- **Stress test gabungan** - Menguji software dalam berbagai kombinasi jenis stress test sekaligus.

JENIS-JENIS STRESS TEST

1. Pengujian stress terdistribusi

Dalam sistem server klien terdistribusi, pengujian dilakukan di semua klien dari server. Peran server stress adalah untuk mendistribusikan serangkaian tes stress untuk semua klien dan melacak statusnya. Setelah itu, server menambahkan nama klien dan mulai mengirim data untuk pengujian dengan mengirim sinyal atau detak jantung yang terhubung. Jika server tidak menerima sinyal apa pun dari mesin klien, maka perlu diselidiki lebih lanjut untuk melakukan debug.

2. Aplikasi stress testing

Pengujian ini berkonsentrasi untuk menemukan bug terkait penguncian dan pemblokiran data, masalah jaringan, dan kemacetan kinerja dalam aplikasi.

3. Pengujian stress transaksional

Merupakan pengujian pada satu atau lebih transaksi antara dua atau lebih aplikasi. Tes ini digunakan untuk fine-tuning dan mengoptimalkan sistem.

4. Pengujian stress sistemik

Merupakan pengujian tegangan terintegrasi yang dapat diuji di beberapa sistem yang berjalan pada server yang sama. Pengujian ini digunakan untuk menemukan bug dimana satu data aplikasi memblokir aplikasi lain.

5. Pengujian stress eksperimental

Merupakan salah satu jenis pengujian yang digunakan untuk menguji sistem dengan parameter atau kondisi yang tidak biasa terjadi dalam skenario nyata.

PERBEDAAN USABILITY DAN USER TESTING

Usability Testing	User Testing
Untuk mengetahui apakah user dapat menggunakan produk dengan efisien.	Untuk mengetahui apakah produk sudah sesuai dengan kebutuhan user.
Dilakukan setelah produk diluncurkan.	Dilakukan sebelum pembuatan produk selesai.
Berkaitan dengan kepuasan serta dorongan user untuk menggunakan produk tersebut.	Berkaitan dengan validasi roadmap pembuatan produk.

GERBANG LOGIKA

FR :

True and true	→	True
True and false	→	False
False and false	→	False
True or true	→	True
True or false	→	True
False or false	→	False

Mana dari antara pilihan diatas yang menghasilkan nilai true?

Cara menjawab:

True = 1

False = 0

And = perkalian

Or = penjumlahan

True And True = $1 \times 1 = 1$ (True)

True And False = $1 \times 0 = 0$ (False)

False And False = $0 \times 0 = 0$ (False)

True Or True = $1 + 1 = 1$ (karena bilangan biner hanya ada 0 dan 1) (True)

True Or False = $1 + 0 = 1$ (True)

False Or False = $0 + 0 = 0$ (False)

Sistem Informasi Bisnis

Sistem Informasi Bisnis, yaitu merupakan kumpulan dari berbagai informasi yang memiliki kesatuan antara satu dan yang lainnya yang ditujukan untuk kepentingan bisnis

1. Sistem Penjualan Dan Pemasaran

Fungsi penjualan dan pemasaran bertanggung jawab dalam menjual produk atau jasa organisasi. Pemasaran memperhatikan mengenai pengenalan pelanggan produk atau jasa perusahaan, menentukan kebutuhan dan keinginan pelanggan, merencanakan dan mengembangkan produk atau jasa untuk memenuhi kebutuhan dan mengiklankan serta mempromosikan produk atau jasa.

2. Sistem Manufaktur Dan Produksi

Fungsi manufaktur dan produksi bertanggung jawab untuk benar-benar memproduksi barang dan jasa perusahaan. Sistem manufaktur dan produksi berhubungan dengan perencanaan, pengembangan dan pemeliharaan fasilitas produksi ; penetapan sasaran produksi ; pengadaan, penyimpanan, dan ketersediaan bahan produksi ; dan penjadwalan peralatan, fasilitas, bahan baku, dan tenaga kerja yang dibutuhkan untuk membentuk produk akhir

3. Sistem Keuangan Dan Akuntansi

Fungsi keuangan bertanggung jawab mengelola aset keuangan perusahaan, seperti uang tunai, saham, obligasi, dan investasi lainnya, untuk memaksimalkan pengembalian atas aset keuangan ini. Fungsi keuangan juga bertanggung jawab dalam mengelola kapitalisasi perusahaan (menemukan aset keuangan baru pada saham, obligasi, atau bentuk utang lainnya). Fungsi akuntansi bertanggung jawab menjaga dan mengelola catatan keuangan perusahaan... penerimaan, pembayaran, depresiasi, penggajian untuk menghitung arus dana dalam perusahaan

4. Sistem Sumber Daya Manusia

Fungsi sumber daya manusia bertanggung jawab untuk menarik, mengembangkan, dan mempertahankan tenaga kerja perusahaan. Sistem informasi sumber daya manusia (human resources information system) mendukung aktivitas seperti mengenali karyawan potensial, menjaga catatan lengkap mengenai karyawan yang ada, dan menciptakan program untuk mengembangkan bakat dan keahlian karyawan.

FR Informasi Bisnis

1. Jika provinsi adalah sistem maka :

(Pada sistem pemerintahan di Indonesia, pemerintah nasional adalah sebuah suprasistem, yang dibawahnya ada sistem pemerintahan provinsi, subsistem pemerintahan kabupaten/kota serta sub-subsistem pemerintahan desa. Masing-masing merupakan sebuah sistem yang bulat dan utuh)

2. Proses cuci input proses output mobil. mana yang termasuk dalam proses (pilihan A. Air, mobil kotor, sabun cuci, B. Air, mobil kotor, sabun cuci pemilik mobil memilih jenis pembersihan, dll

FR Pemrograman

FR

• Jika $X=Y$ maka nilai $Y=Y$

•

• Output dari perulangan. Yang

• $a=1$; $a<10$; $a++$

• $B = a + 1$

• Output B apa?

• Hasil 2,3,4,5,6,7,8,9

• $X=1$

• Do

• $X=x+2$;

• While($x<10$)

• End

• Hasil 1,3,5,7,9

jelaskan singkat jenis pemeliharaan sistem

Tentu, berikut adalah beberapa jenis pemeliharaan sistem secara singkat:

1. Pemeliharaan preventif: jenis pemeliharaan yang dilakukan secara berkala untuk mencegah kerusakan dan mempertahankan kinerja sistem yang baik. Contohnya adalah membersihkan debu pada komponen sistem, mengganti suku cadang yang sudah aus, serta melakukan pengujian rutin pada sistem.
2. Pemeliharaan korektif: jenis pemeliharaan yang dilakukan setelah terjadi kerusakan pada sistem. Pemeliharaan ini bertujuan untuk memperbaiki sistem yang rusak dan mengembalikan fungsinya seperti semula. Contohnya adalah mengganti komponen yang rusak atau melakukan perbaikan pada program software.
3. Pemeliharaan adaptif: jenis pemeliharaan yang dilakukan ketika ada perubahan pada kebutuhan sistem atau lingkungan sekitar. Pemeliharaan ini bertujuan untuk menyesuaikan sistem dengan perubahan yang terjadi. Contohnya adalah menambahkan fitur baru pada sistem atau mengganti perangkat keras yang sudah ketinggalan zaman.
4. Pemeliharaan perfektif: jenis pemeliharaan yang dilakukan untuk meningkatkan kinerja sistem yang sudah berjalan dengan baik. Pemeliharaan ini bertujuan untuk meningkatkan efisiensi dan efektivitas sistem. Contohnya adalah meningkatkan kecepatan respon sistem atau mengoptimalkan penggunaan memori.



Fungsi Array

Array adalah suatu cara menyimpan banyak data dalam satu variabel. Biasanya dalam bahasa pemrograman lain, data-data yang disimpan pada array haruslah data yang sejenis. Namun, pada PHP anda bebas menyimpan data dengan tipe apa saja misalkan string dan integer. Kemudian, data-data yang disimpan dalam variabel tersebut bisa dibedakan dengan menggunakan index.

Jenis-jenis tipe array di php

1. Numeric Array

Numeric array merupakan jenis array dimana index elemen yang terdapat didalamnya di definisikan dengan angka, biasanya index pertama dimulai dengan angka "0" dan seterusnya.

2. Associative Array (array asosiatif)

Array associative berbeda dengan jenis array numeric, array jenis ini mendefinisikan index-nya menggunakan name atau nama untuk dapat mengakses nilai dari elemen array tersebut.

3. Multidimensional Array

Array multidimensional adalah metode penyimpanan nilai data array lebih dari satu array, ini artinya array didalam array, array multidimensi sangat berguna untuk menyimpan nilai variabel dengan banyak data meskipun lebih sulit untuk menggunakannya tetapi array jenis ini lebih efisien dalam pemakai karena mampu menampung banyak data.

UML (UNIFIED MODELLING LANGUAGE)

ANALISA & PERANCANGAN
SISTEM INFORMASI | 39

UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh Object Management Group dengan versi awal 1.0 pada bulan Januari 1997.

Adapun tujuan dan fungsi perlu adanya UML yaitu sebagai berikut:

1. Dapat memberikan bahasa pemodelan visual atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.
2. Menyatukan informasi-informasi terbaik yang ada dalam pemodelan.
3. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
4. Tidak hanya menggambarkan model sistem software saja, namun dapat memodelkan sistem berorientasi objek.
5. Mempermudah pengguna untuk membaca suatu sistem.
6. Berguna sebagai blueprint, jelas ini nantinya menjelaskan informasi yang lebih detail dalam perancangan berupa coding suatu program.

-Uml adalah satu jenis dari diagram UML (*unified modeling language) yang menggambarkan hubungan interaksi antara sistem dan aktor

-activity diagram adalah sebuah diagram yang dapat memodelkan berbagai proses yang terjadi pada sistem

- sequence diagram adalah diagram yang menjelaskan interaksi objek berdasarkan urutan waktu.

- class diagram adalah suatu diagram yang digunakan untuk menampilkan kelas-kelas berupa paket-paket untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya.

- statemachine diagram adalah salah satu jenis UML yang berfungsi untuk menggambarkan transisi serta perubahan pada suatu objek pada sistem

-component diagram adalah untuk menggambarkan software pada sistem



Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk merancang, mendokumentasikan, dan membangun sistem perangkat lunak. UML menggunakan beberapa jenis diagram untuk mewakili berbagai aspek sistem. Berikut adalah beberapa contoh diagram UML yang umum digunakan:

1. Diagram Use Case: Digunakan untuk menunjukkan interaksi antara pengguna dan sistem. Diagram ini menunjukkan bagaimana pengguna menggunakan sistem dan apa yang diharapkan dari sistem tersebut.
2. Diagram Kelas: Digunakan untuk menggambarkan kelas, objek, atribut, dan relasi antar kelas dalam sistem. Diagram ini membantu pengembang dalam merencanakan struktur kelas dan memahami bagaimana kelas saling berinteraksi.
3. Diagram Sekuensi: Digunakan untuk menggambarkan interaksi antara objek dalam sistem. Diagram ini menunjukkan urutan peristiwa dalam suatu skenario dan cara objek berinteraksi satu sama lain.
4. Diagram Aktivitas: Digunakan untuk menggambarkan aktivitas dalam suatu proses. Diagram ini menunjukkan urutan aktivitas dan kondisi yang perlu dipenuhi untuk menyelesaikan suatu tugas.
5. Diagram Komponen: Digunakan untuk menggambarkan komponen dalam sistem dan bagaimana komponen tersebut terhubung. Diagram ini menunjukkan bagaimana bagian-bagian sistem terhubung dan bekerja sama.
6. Diagram Pemakaian: Digunakan untuk menggambarkan penggunaan artefak, seperti kelas atau komponen, dalam sistem. Diagram ini menunjukkan bagaimana artefak digunakan dalam berbagai skenario.

Setiap diagram UML memiliki kegunaan dan tujuannya sendiri, dan dapat digunakan dalam berbagai tahap pengembangan sistem perangkat lunak.