

DAFTAR ISTILAH

<i>Engineer</i>	: Insinyur. Dalam konteks pengembangan perangkat lunak, dapat disebut sebagai pengembang.
<i>Sociotechnical</i>	: Sistem yang menggabungkan unsur manusia dan teknologi.
<i>On request</i>	: Sistem yang memberikan keluaran hanya pada saat diinginkan.
<i>Hyperlink</i>	: Tautan. Serangkaian teks yang mengarahkan ke suatu halaman <i>web</i>
<i>Website</i>	: Situs <i>web</i> yang diakses menggunakan internet dan jendela <i>web browser</i> .
<i>Browser / Web browser</i>	: Perangkat lunak pada komputer yang memungkinkan pengguna untuk mengakses situs <i>web</i> di internet.
<i>Web</i>	: Kependekan dari <i>World Wide Web</i> , jaringan yang menghubungkan seluruh <i>website</i> yang ada.
<i>Webpage</i>	: Satu halaman dari <i>website</i> .
<i>Web element</i>	: Elemen seperti teks atau gambar dalam sebuah halaman <i>website</i> .
<i>Tool(s)</i>	: Sebuah alat. Dalam konteks pengembangan perangkat lunak, <i>tool</i> adalah aplikasi atau komponen yang dapat membantu sebuah proses.
<i>Frontend</i>	: Konsep dalam pengembangan <i>website</i> yang merujuk kepada bagian dari aplikasi yang ditampilkan kepada pengguna.
<i>Backend</i>	: Konsep dalam pengembangan <i>website</i> yang merujuk kepada bagian dari aplikasi yang diproses dalam server.
<i>Database</i>	: Sebuah sistem untuk menyimpan data pada sebuah aplikasi.
<i>Markup</i>	: Instruksi atau tanda dalam teks yang menunjukkan struktur atau cara penyajian isi dalam <i>markup language</i> .

<i>White-box testing</i>	: Metode pengetesan aplikasi Dimana penguji memiliki akses ke dalam kode pemrograman
<i>Black-box testing</i>	: Metode pengetesan aplikasi Dimana penguji tidak memiliki akses ke dalam kode pemrograman (hanya tampilan luar saja)
<i>Unified Modeling Language</i>	: Sebuah <i>framework</i> untuk menggambarkan rancangan sebuah aplikasi.
<i>Software</i>	: Perangkat lunak. Aplikasi yang diakses menggunakan komputer.
<i>Class</i>	: Sebuah konsep pada pemrograman yang menggambarkan cetak biru yang mendefinisikan objek dalam paradigma pemrograman berbasis objek.
<i>Flow</i>	: Alur dari penggunaan aplikasi.
<i>Use Case</i>	: Skenario penggunaan dari suatu hal. Dalam konteks pengembangan perangkat lunak, merujuk kepada skenario penggunaan aplikasi.
<i>Sequence</i>	: Urutan.
<i>Activity</i>	: Aktivitas.
<i>Component</i>	: Komponen.
<i>Deployment</i>	: Peluncuran. Dalam konteks pengembangan perangkat lunak, peluncuran bermakna membuat sebuah aplikasi dapat digunakan oleh pengguna akhir, setelah selesai dikembangkan.
<i>Layer</i>	: Lapisan. Dalam konteks pengembangan perangkat lunak, merujuk kepada abstraksi bagian-bagian dari aplikasi.
<i>Hypertext Markup Language</i>	: Bahasa untuk menuliskan konten dari <i>webpage</i> .
<i>Tag</i>	: Tanda. Dalam konteks HTML, adalah bagian untuk menandai elemen HTML.
<i>Cascading Style Sheet</i>	: Bahasa untuk menuliskan peraturan-peraturan untuk mengubah tampilan dari HTML.

<i>Script</i>	:	Kode yang dibuat untuk menginstruksikan aplikasi untuk melakukan sesuatu.
<i>Library</i>	:	Sekumpulan kode untuk sebuah fungsionalitas yang dibuat untuk memudahkan orang lain agar tidak harus menulis kodenya sendiri.
<i>Framework</i>	:	Kerangka kerja. Suatu kerangka yang dapat diikuti sehingga pekerjaannya lebih terstruktur.
<i>Object Relational Mapping</i>	:	Sebuah fitur dalam <i>framework</i> Laravel yang memungkinkan pengembang untuk mengakses <i>database</i> tanpa harus menulis <i>query</i> satu-persatu.
<i>Query</i>	:	Perintah yang dapat dimasukkan ke <i>database</i> untuk mengakses <i>database</i> tersebut, seperti mengambil atau menyimpan data.
<i>Routing</i>	:	Konsep dalam pemrograman <i>website</i> yang mengarahkan pengguna ke halaman tertentu dengan suatu aksi.
<i>Relational Database Management System</i>	:	Sebuah sistem <i>database</i> yang mempunyai tabel objek yang saling berelasi.
<i>Software Development Life Cycle</i>	:	Sebuah kerangka pengembangan aplikasi yang mempunyai aturan-aturan pengembangan.
<i>Stakeholder</i>	:	Pemangku kepentingan. Orang-orang yang mempunyai kepentingan dalam suatu hal.
<i>Waterfall</i>	:	Air terjun. Dalam konteks SDLC, merupakan sebuah model yang mengalir seperti air terjun, yang artinya ketika sebuah proses sudah dilewati, tidak akan bisa kembali ke proses tersebut lagi.
<i>Agile</i>	:	Tangkas. Dalam konteks SDLC, merupakan model yang cepat dan efisien, serta tanggap perubahan.
<i>Rapid application development</i>	:	Model SDLC yang mengedepankan kecepatan dengan cara mempersingkat proses desain.

<i>Extreme Programming</i>	: Model SDLC yang lebih cepat daripada <i>Rapid application development</i> dengan meniadakan proses desain.
<i>Object Diagram</i>	: Diagram yang menunjukkan instansiasi nyata dari kelas, termasuk objek, atribut, dan hubungan antarobjek dalam sistem.
<i>Class Diagram</i>	: Diagram yang memodelkan struktur statis dari sistem, menunjukkan kelas-kelas, atribut, metode, dan hubungan antar kelas dalam pengembangan berbasis objek.
<i>Component Diagram</i>	: Diagram yang menggambarkan komponen-komponen perangkat lunak dalam sistem serta hubungan antar komponen tersebut.
<i>Use Case Diagram</i>	: Diagram yang menggambarkan actor dan fitur-fitur yang dapat digunakannya.
<i>Sequence diagram</i>	: Diagram yang memodelkan pesan-pesan antarlapisan sistem.
<i>Activity Diagram</i>	: Diagram yang menggambarkan alur sebuah proses dalam aplikasi.
<i>Rational Unified Process</i>	: Metodologi pengembangan perangkat lunak yang bersifat iteratif dan berorientasi objek.
<i>Property(ies)</i>	: Atribut yang dimiliki sebuah <i>class</i> .
<i>Method(s)</i>	: Perilaku atau aksi yang dapat dilakukan sebuah <i>class</i> .
<i>Inheritance</i>	: <i>Sub-class</i> yang mewarisi <i>property</i> dan <i>method</i> dari <i>class</i> induk.
<i>Agregasi</i>	: Hubungan antara dua <i>class</i> di mana satu <i>class</i> terdiri dari objek-objek <i>class</i> lain, namun dengan kepemilikan yang lemah
<i>Komposisi</i>	: Hubungan kuat antara dua <i>class</i> di mana satu <i>class</i> sepenuhnya memiliki objek dari <i>class</i> lain

<i>Public</i>	: Publik. Dalam konteks pemrograman berbasis objek, menjadi penanda bahwa objek tersebut dapat diakses dari <i>class</i> manapun.
<i>Private</i>	: Privat. Dalam konteks pemrograman berbasis objek, menjadi penanda bahwa objek tersebut hanya dapat diakses dalam <i>class</i> itu sendiri.
<i>Protected</i>	: Diproteksi. Dalam konteks pemrograman berbasis objek, menjadi penanda bahwa objek tersebut hanya dapat diakses dari <i>class</i> itu sendiri dan <i>class</i> turunannya.
<i>Enkapsulasi</i>	: Prinsip dalam pemrograman berorientasi objek yang menyembunyikan detail internal suatu kelas dan hanya menyediakan antarmuka publik untuk berinteraksi
<i>Modularitas</i>	: Prinsip dalam rekayasa perangkat lunak yang membagi sistem menjadi bagian-bagian terpisah (modul) yang independen namun saling terkait.
<i>Coupling</i>	: Prinsip dalam pemrograman berbasis objek mengenai objek yang saling terikat.
<i>Tight Coupling</i>	: <i>Coupling</i> yang sangat erat dan berlapis hingga objek-objek tak bisa dipisahkan sama sekali.
<i>Include</i>	: Istilah dalam <i>Use Case Diagram</i> yang menunjukkan bahwa satu <i>Use Case</i> bergantung pada <i>Use Case</i> lainnya.
<i>Extend</i>	: Istilah dalam <i>Use Case Diagram</i> yang menunjukkan bahwa satu <i>Use Case</i> sebagai ekstensi dari <i>Use Case</i> lain.
<i>Lifeline</i>	: Elemen dalam <i>Sequence Diagram</i> yang merepresentasikan satu lapisan aplikasi.
<i>Debugging</i>	: Proses membetulkan suatu kode yang salah dalam pemrograman.
<i>Decision node</i>	: Elemen dalam <i>Activity Diagram</i> yang menggambarkan alur terpecah menjadi pilihan kondisional.

<i>Merge node</i>	: Elemen dalam <i>Activity Diagram</i> yang menggabungkan alur yang sudah terpecah menjadi satu garis lagi.
<i>Fork node</i>	: Elemen dalam <i>Activity Diagram</i> yang menggambarkan alur terpecah yang berjalan bersamaan.
<i>Join node</i>	: Elemen dalam <i>Activity Diagram</i> yang menyambungkan alur yang terpecah karena <i>fork node</i> menjadi satu lagi.
<i>Sintaks</i>	: Istilah. Kata-kata yang mempunyai fungsi tersendiri dalam pemrograman.
<i>Model-View-Controller</i>	: Konsep dalam pemrograman yang membagi aplikasi menjadi tiga bagian, Model yang membungkus kelas dari objek-objek yang dibutuhkan, View yang merupakan tampilan yang dilihat oleh pengguna, dan Controller yang mengatur logika aplikasi.
<i>Middleware</i>	: Sebuah “jalan” yang dilalui alur kode saat dijalankan, yang berfungsi memodifikasi kode.
<i>Sesi</i>	: Satu sesi pengguna berinteraksi dengan sistem. Tampilan data dapat diasosiasikan dengan sesi tersebut.
<i>Autentikasi</i>	: Proses verifikasi identitas pengguna atau entitas sebelum diberikan akses ke sistem.
<i>Just-in-time compilation</i>	: Program yang dikompilasi hanya saat dijalankan.
<i>Form</i>	: Sejumlah bidang dalam <i>frontend</i> yang dapat diisi pengguna dan mengirimkan datanya ke <i>backend</i> .
<i>Task scheduling</i>	: Pengaturan jadwal.
<i>Service container</i>	: Komponen dalam arsitektur perangkat lunak yang bertanggung jawab untuk mengelola dependensi dan instansiasi objek secara otomatis.
<i>Service Provider</i>	: Komponen yang digunakan untuk mendaftarkan layanan atau konfigurasi ke dalam <i>service container</i> .
<i>Template engine</i>	: Sebuah tampilan antarmuka sederhana dengan fungsional tambahan.

<i>Command Line Interface</i>	: Antarmuka pengguna dengan aplikasi berbasis tulisan baris per baris.
<i>Scalable / Skalabel</i>	: Aplikasi yang dikembangkan sedemikian rupa sehingga mampu menyesuaikan dengan beban kerja.
<i>Array</i>	: Kumpulan atau sekelompok data.
<i>Stored procedure</i>	: Sekumpulan aksi yang dijalankan secara otomatis ketika variabel pembungkusnya dipanggil.
<i>Trigger</i>	: Aksi yang dijalankan otomatis ketika suatu kondisi terpenuhi
<i>Views</i>	: Variabel yang membungkus perintah untuk mengambil data dengan spesifikasi tertentu.
<i>Atomicity</i>	: Menjamin bahwa setiap transaksi dalam basis data dilakukan sepenuhnya atau tidak sama sekali.
<i>Consistency</i>	: Menjamin bahwa titik mula dan tujuan akhir dari transaksi keduanya valid.
<i>Isolation</i>	: Menjamin bahwa transaksi yang dijalankan secara bersamaan tidak akan saling mengganggu.
<i>Durability</i>	: Menjamin bahwa hasil dari transaksi yang telah selesai dan dikonfirmasi akan tetap tersimpan secara permanen
<i>Pair programming</i>	: Memrogram secara berpasangan.
<i>Test-driven development</i>	: Pengembangan berbasis <i>testing</i> . Artinya <i>testing</i> menjadi bagian krusial dalam pengembangan dan dapat dilakukan secara sering.
<i>Continuous Integration</i>	: Praktik dalam pengembangan perangkat lunak di mana kode yang dibuat oleh pengembang secara rutin digabungkan ke dalam repositori utama dan langsung diuji secara otomatis.
<i>Refactoring</i>	: Pengubahan kode, biasanya untuk mencari cara yang lebih efisien.
<i>Feature testing</i>	: Proses pengujian aplikasi secara mandiri dengan menentukan skenario tes dan membandingkan hasil uji.
<i>Unit testing</i>	: Pengujian terhadap unit terkecil dari kode program.

<i>Integration testing</i>	: Proses pengujian kesinambungan <i>frontend</i> dengan <i>backend</i> .
<i>User acceptance testing</i>	: Proses pengujian aplikasi dengan memberikan aplikasi kepada pengguna untuk dicoba secara langsung.
<i>Systematic Literature Review</i>	: Sebuah proses untuk meninjau literatur yang relevan terhadap penelitian.
<i>Preferred Reporting Items for Systematic Reviews and Meta-Analyses</i>	: Sebuah metodologi untuk menyortir literatur yang didapatkan sehingga mendapatkan sejumlah kecil literatur yang paling relevan.
<i>Input</i>	: Memasukkan. Dalam konteks perangkat lunak, memasukkan data.
<i>Et al.</i>	: Bahasa Latin untuk “dan lainnya.”
<i>Android</i>	: Sistem operasi untuk ponsel pintar.
<i>Mobile</i>	: Istilah lain untuk ponsel pintar. Dari Bahasa Inggris “ <i>mobile phone</i> .”
<i>Plan of attack</i>	: Rencana serangan. Dalam konteks penelitian, rencana untuk menyelesaikan masalah.
<i>Design Science Research</i>	: Sebuah kerangka penelitian dalam bidang sains.
<i>Instansiasi</i>	: Penerapan atau perwujudan suatu konsep abstrak menjadi bentuk nyata atau konkret.
<i>Environment</i>	: Lingkungan.
<i>People</i>	: Orang, manusia.
<i>Entity Relationship Diagram</i>	: Diagram yang menggambarkan relasi antartabel.
<i>Templating</i>	: Menyusun berdasarkan <i>format</i> tertentu.
<i>Timeline</i>	: Garis waktu. Urutan waktu kejadian.
<i>Kick-off-meeting</i>	: Pertemuan awal.
<i>Existing</i>	: Kondisi saat ini.
<i>Product owner</i>	: Seseorang yang ditunjuk oleh pihak klien sebagai penanggung jawab komunikasi terhadap tim pengembang.
<i>Spreadsheet</i>	: Dokumen digital berbentuk lembar kerja yang terdiri dari baris dan kolom untuk menyimpan data.

- Infografis* : Representasi visual dari suatu informasi yang dikemas agar enak dilihat dan mudah dibaca dan dimengerti.
- Targeting* : Kondisi yang ingin dicapai.
- Chatbot* : Fitur di mana pengguna bisa berbincang dengan AI.
- Endpoint* : Ujung URL di mana lewat URL tersebut dapat berinteraksi langsung dengan *backend*.
- Sticky* : Lengket. Dalam konteks *frontend*, *sticky* adalah atribut elemen yang menempel pada satu bagian dari layar.
- Dashboard* : Halaman di mana seluruh informasi dapat diakses dalam sekali lihat.