# Mengenal Program Komputer

## Apa itu Pemrograman?

Saat ini, kebanyakan orang tidak perlu tahu bagaimana komputer bekerja. Mereka hanya bisa menyalakan komputer ataupun smartphone dan menggunakannya sesuai keinginannya tanpa pernah tahu cara kerja dari semua itu.

Berinteraksi dengan komputer dapat dilakukan oleh siapapun, tapi memberikan instruksi kepada komputer sesuai keinginan hanya dapat dilakukan oleh **programmer**. Sehingga ketika Kamu belajar pemrograman maka Kamu akan tahu bagaimana komputer bekerja dan memberikan instruksi untuk melakukan sesuatu sesuai yang kita inginkan. Bagaimana keren bukan?

### Pengertian Pemrograman

**Pemrograman** adalah sebuah proses yang mencakup kegiatan seperti **analisis**, **algoritma**, yang diterjemahkan ke dalam bahasa pemrograman sehingga dapat dijalankan oleh komputer.

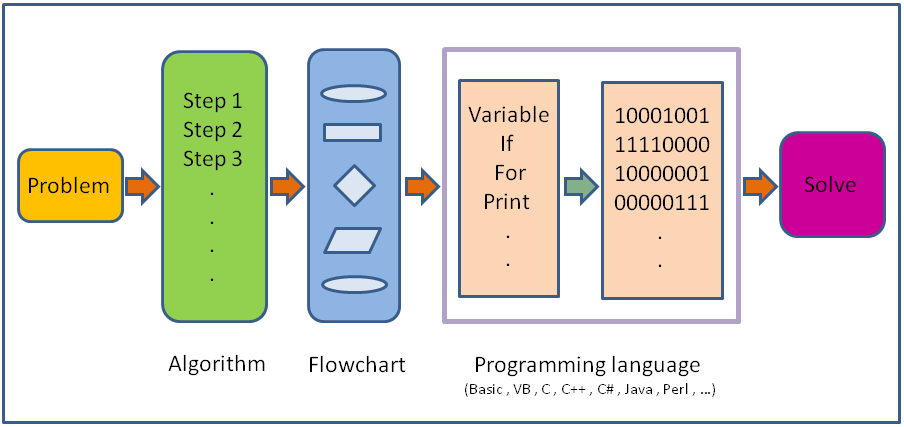
Atau secara sederhana, **pemrograman adalah proses kreatif yang menginstruksikan komputer bagaimana menyelesaikan suatu tugas**.

Meskipun saat ini telah banyak bahasa pemrograman yang telah tersedia, namun hal penting yang harus dimiliki adalah kemampuan menganalisa dan menemukan solusi karena tanpa **algoritma** maka tidak ada program.

Pemrograman merupakan cara kita membuat representasi untuk solusi yang kita miliki. Oleh karena itu, representasi dan proses pembuatannya menjadi bagian yang mendasar.

Disisi lain **algoritma** menggambarkan masalah yang dimiliki dengan serangkaian proses untuk mencapai solusi yang diinginkan.

Perhatikan skema berikut:

 Gambar 1.3.1 - Skema alur pembuatan program

Skema di atas memberikan kita gambaran bahwa dengan pemrograman sebuah masalah yang kita jumpai dapat dikonversi ke dalam dunia teknologi sehingga menjadi sebuah solusi.

### Membuat program

Membuat program adalah memberikan instruksi untuk memanipulasi berbagai jenis objek, misalnya angka, kata, gambar, suara, dll.

Membuat program komputer serupa dengan membuat musik, merancang rumah atau menciptakan hal lain. Sehingga sebagian pendapat mengatakan bahwa pemrograman adalah seni, bukan rekayasa.

### Pemrograman berdasarkan sudut pandang

Ketika ditanya apa itu pemrograman? mari kita jabarkan dari berbagai sudut pandang.

* **Pemrograman adalah ilmu pengetahuan**

Karena menggunakan algoritma yang digambarkan dengan matematika dan sains.

* **Pemrograman adalah skill**

Karena membutuhkan keterampilan dan perancangan.

* **Pemrograman adalah rekayasa**

Karena membutuhkan perpaduan antara ukuran program, kecepatan, waktu (termasuk development dan debugging).

* **Pemrograman adalah seni**

Karena melibatkan kreatifitas dan imajinasi.

## Keuntungan Mengusai Pemrograman

Setiap orang harus belajar pemrograman komputer, karena pemrograman akan mengajarkanmu bagaimana cara berpikir. **Steve Jobs**

Perkembangan teknologi yang sangat pesat membuat pemrograman memiliki peran sangat penting. Belajar pemrograman menawarkan peluang kepada siapapun untuk terlibat langsung dalam perkembangan teknologi.

### Keuntungan belajar pemrograman

Adapun keuntungan yang akan kita dapatkan ketika mempelajari pemrograman antara lain :

* **Berpikir dengan cara yang berbeda**

Ketika belajar pemrograman, Kamu belajar bagaimana memecah masalah menjadi bagian-bagian kecil agar lebih mudah diselesaikan.

Potongan-potongan kecil tersebut kemudian dibuatkan programnya dengan menggunakan bahasa yang dipahami komputer, sehingga komputer dapat menyelesaikan solusi dengan lebih cepat dan berulang.

* **Berdampak positif pada karir**

Salah satu pencapaian dalam hidup yang harus dicapai adalah pertumbuhan karir yang terus menanjak.

Belajar pemrograman membantu Kamu untuk mencapainya melalui banyak jalan dan peluang.

* **Menghasilkan lebih banyak uang**

Kesuksesan finansial merupakan tujuan penting bagi banyak orang, dan belajar pemrograman bisa memiliki dampak yang signifikan dengan munculnya banyak peluang yang sebelumnya tidak terlihat.

* **Membawa idemu ke kehidupan nyata**

Terkadang kita memiliki gagasan tentang perbaikan hidup dari aspek manapun, misalnya kehidupan pribadi, lingkungan kerja atau dari sudut pandang global. Akan tetapi untuk mewujudkan hal ini, kita kekurangan dukungan teknologi untuk mentransformasikan gagasan tersebut secara akurat menjadi proyek nyata yang dapat dilihat. Maka dengan belajar pemrograman Kamu dapat mengaplikasikan sendiri gagasan Kamu ke dalam bentuk aplikasi.

* **Menjadi lebih percaya diri**

Salah satu bentuk perasaan terbaik yang diharapkan adalah dengan mengetahui bahwa Kamu telah mencapai sesuatu yang hebat atau memiliki kemampuan untuk menyelesaikan sesuatu yang sangat kompleks. Dengan belajar pemrograman memberikan Kamu peluang untuk mendapatkan hal tersebut.

Dengan belajar pemrograman, Kamu akan belajar tentang konsep yang sangat berharga, terlepas apakah Kamu akan terus berkarir di bidang ini atau tidak.

Satu hal yang akan Kamu temukan nanti adalah fakta bahwa komputer itu sangat bodoh, tapi taat. Komputer tidaklah pintar, ia hanya dapat mengerjakan sesuatu yang berulang dengan sangat cepat. Ia akan melakukan apapun yang Kamu instruksikan sesuai keinginan Kamu.

Tapi hal yang paling penting untuk Kamu tanamkan dalam diri saat belajar pemrograman adalah bahwa **belajar pemrograman itu menyenangkan**.

Dengan sugesti positif, Kamu akan dapat menikmati proses belajar dan hasilnya pun akan lebih memuaskan.

## Apa yang Dimaksud dengan Bahasa Pemrograman?

**Bahasa pemrograman** adalah sekumpulan perintah, instruksi dan penggunaan sintaks lainnya untuk membuat sebuah program.

Bahasa pemrograman menjadi elemen penting yang harus diketahui, sebab bahasa pemrograman menjadi perantara antara programmer untuk berkomunikasi dengan komputer.

Bahasa pemrograman itu sendiri digolongkan menjadi beberapa bagian, antara lain machine language atau bahasa mesin, assembly language, dan high-level language atau bahasa tingkat tinggi.

Machine Language

Bahasa mesin adalah bahasa yang dapat dipahami oleh komputer secara langsung, direpresentasikan dalam bentuk kode biner atau 0 dan 1.

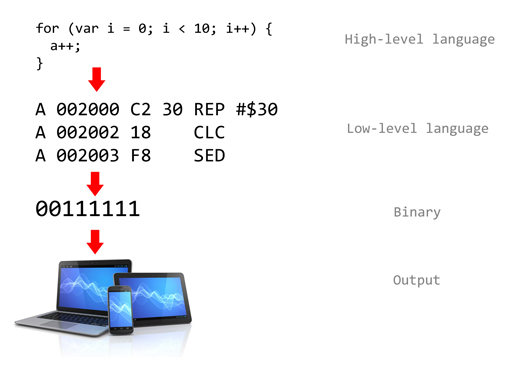
Assembly language

Assembly language atau bahasa assembly adalah bahasa pemrograman tingkat rendah (low-level language) yang dapat digunakan oleh programmer untuk menuliskan instruksi pada komputer.

High-level language

High-level language atau bahasa tingkat tinggi adalah bahasa pemrograman yang lebih dekat ke bahasa manusia dibanding bahasa pemrograman tingkat rendah, sehingga lebih mudah dipelajari, ditulis dan dibaca ulang.

Baik high-level language maupun low-level language perlu dikompilasi terlebih dahulu ke dalam bahasa mesin sehingga dapat dipahami oleh komputer.

 Gambar 1.3.1 - Contoh bentuk kode program

High-level language memungkinkan programmer untuk menulis kode secara lebih alami dengan menggunakan kata dan simbol yang logis. Misalnya, untuk kata seperti function, while, if, dan lain sebagainya yang banyak digunakan dalam bahasa pemrograman.

## Bahasa Pemrograman Apa yang Harus Dipelajari Terlebih Dahulu

Umumnya orang yang baru mau belajar pemrograman akan dihadapkan pada pertanyaan "bahasa pemrograman apa yang harus dipelajari pertama kali?".

Terlebih lagi karena bahasa pemrograman yang tersedia sudah sangat banyak.

Hal pertama yang harus dilakukan adalah dengan menentukan aplikasi jenis apa yang hendak dibuat ?

Semisal Kamu hendak membuat jenis aplikasi web, desktop, atau mobile.

Kamu kemudian mesti mencari informasi tentang bahasa pemrograman apa saja yang tersedia untuk jenis yang dipilih.

Namun bila Kamu masih bingung menentukan hendak membuat aplikasi apa, karena Kamu benar-benar pemula di pemrograman, maka ada beberapa pertimbangan yang dapat menjadi referensi sebelum Kamu memutuskan hendak mempelajari bahasa pemrograman yang mana dulu.

### Referensi bahasa pemrograman

* **Python**

Bila kita menimbang dari sudut pandang kemudahan dalam penulisan sintaks, Kamu dapat memilih bahasa pemrograman **Python**. Selain karena Python juga dapat digunakan untuk membuat beragam aplikasi diantaranya web, desktop, game, mobile Android, dan untuk kebutuhan komputasi berat seperti analisis data dan machine learning. Python juga dapat digunakan untuk memprogram hardware atau mikrokontroller.

* **JavaScript**

Bila kita menimbang dari sudut pandang tren penggunaan, maka yang paling marak digunakan sekarang adalah **JavaScript**. Dahulu JavaScript hanya tersedia sebagai bahasa pemrograman di sisi client (hanya berjalan di browser). Namun sekarang JavaScript sudah berevolusi dan hadir di berbagai platform. Kamu dapat membuat aplikasi web untuk sisi klien dan server sekaligus, aplikasi desktop, game, dan juga mobile.

* **PHP**

Bila kita menimbang kemudahan infrastruktur, maka **PHP** masih menjuarai lahan ini. PHP dipercaya masih menjadi bahasa pemrograman yang paling banyak digunakan di dunia. PHP didesain khusus untuk membuat aplikasi web. Bila Kamu menggunakan PHP, Kamu tidak akan kesulitan untuk mendapatkan hosting dengan biaya yang sangat terjangkau.

* **Java**

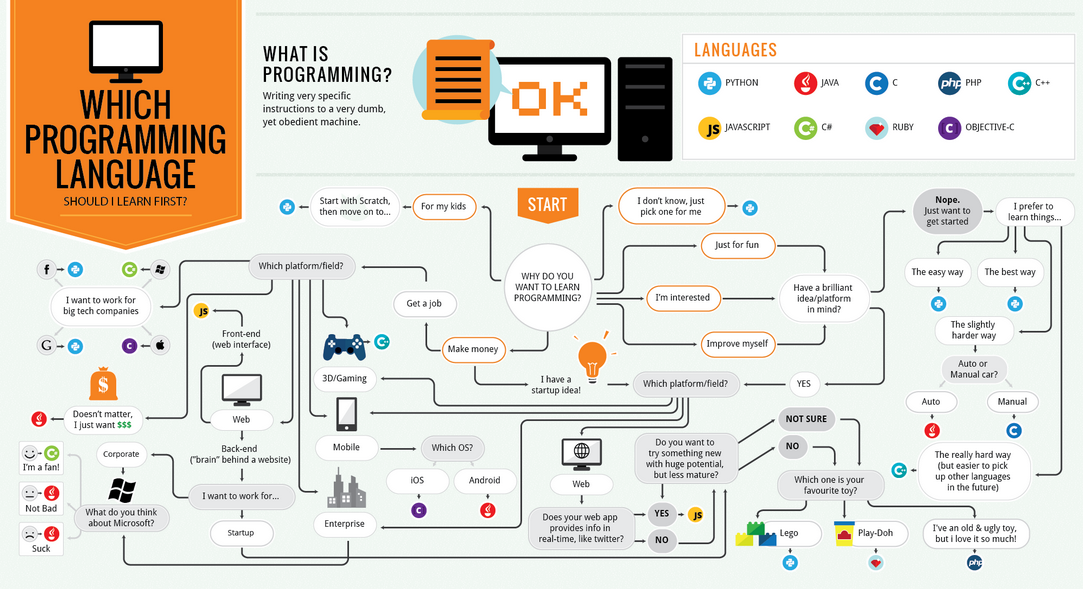
Bila Kamu menyasar pasar enterprise, **Java** masih cenderung dominan. Java dikenal sangat kuat untuk membuat aplikasi skala besar seperti ERP dan sistem informasi dan umumnya digunakan di perusahaan-perusahaan besar. Selain itu Java juga menjadi bahasa pemrograman utama untuk membangun aplikasi mobile Android.

* **C**

Bila kita menimbang konseptual pemrograman, maka **C** adalah pilihan bijak. Semua konsep pemrograman dari yang paling dasar akan Kamu temui di bahasa C. C merupakan bahasa tingkat tinggi yang paling cepat dan paling dekat ke bahasa mesin. Dari sisi optimasi produk akhir, aplikasi yang dikembangkan langsung dari bahasa C akan sangat efisien. Kampus-kampus IT umumnya menggunakan C untuk mengenalkan pemrograman. Hanya saja untuk mencapai produk akhir, C tidak lebih mudah dibanding bahasa yang disebutkan sebelum ini.

Kesimpulannya, untuk pilihan paling mudah tanpa banyak pertimbangan, kami sarankan untuk mulai belajar dari bahasa pemrograman **JavaScript** atau **Python**.

Sebagai tambahan referensi, Kamu dapat menelusuri infografis di bawah ini untuk gambaran umum tentang tema yang sedang kita bahas ini.

 Gambar 1.4.1 - Ilustrasi cara memilih bahasa pemrograman untuk dipelajari ([*kredit*](http://carlcheo.com/startcoding))

## Jenis-jenis Programmer

**Programmer** adalah orang yang membuat software komputer.

Istilah programmer bisa merujuk ke spesialis di satu bidang pemrograman komputer atau secara general yang menulis kode untuk berbagai jenis perangkat lunak.

Adapun programmer dapat digolongkan menjadi beberapa jenis:

* **Web Programmer**
* **Mobile Programmer**
* **Desktop Programmer**
* **Hardware Programmer**
* **Game Programmer**

Lebih jauh tentang detail masing-masing jenis akan Kamu temukan di bagian kedua kelas ini.

## Beragam Peluang Menjadi Programmer

Teknologi sudah menjadi bagian dari setiap lini kehidupan manusia. Imbasnya adalah kebutuhan akan tenaga programmer akan terus meningkat. Tidak hanya perusahaan besar yang memerlukan programmer. Perusahaan yang baru bertumbuh pun umumnya akan memerlukan dukungan teknis untuk membangun sistem berbasis IT. Dan disitulah peran programmer.

Meski demikian, Kamu tidak mesti bekerja di perusahaan ketika menjadi programmer. Ada banyak peluang yang dapat ditangkap oleh programmer, diantaranya:

* **Kesempatan bekerja di hampir semua industri**

Hampir setiap jenis bisnis telah melibatkan teknologi, mulai dari membuat website, aplikasi mobile hingga sistem informasi untuk kebutuhan internal perusahaan.

* **Freelancer**

Kamu dapat memberikan jasa pengembangan aplikasi secara mandiri sebagai programmer lepas untuk beragam segmen kehidupan dan bidang pekerjaan. Saat ini sedang ramai dilirik profesi sebagai Remote Worker yang memungkinkan kita untuk bekerja dari berbagai perusahaan manapun di seluruh dunia tanpa harus berpindah tempat atau dengan kata lain bekerja secara remote dari rumah. Cukup banyak situs yang menghubungkan antara programmer dengan pihak yang membutuhkan jasa programmer, seperti [Freelancer.com](http://freelancer.com/), [Upwork.com](http://upwork.com/), [Sribulancer.com](http://sribulancer.com/) dan [Projects.co.id](http://projects.co.id/).

* **Membuat produk aplikasi**

Kamu dapat menjadi klien untuk dirimu sendiri, dengan membuat solusi aplikasi dari permasalahan yang Kamu hadapi di masyarakat. Aplikasi yang Kamu buat dapat Kamu sebarkan secara cuma-cuma, atau Kamu jual dengan harga tertentu. Proses monetisasi aplikasi tidak melulu harus selalu menawarkan aplikasi berbayar. Seperti contoh umumnya, kita dapat membuat aplikasi mobile gratis yang didalamnya kita pasangi iklan sponsor. Lebih jauh lagi, dengan bekal solusi yang Kamu telah buat, Kamu dapat melanjutkan usahamu menjadi perusahaan rintisan atau startup dan berkembang lebih besar lagi.

* **Menjadi penulis**

Perihal menulis tentu saja bukan hanya terpaku pada tulisan yang berbentuk buku, akan tetapi kita dapat membuat sebuah tulisan berupa panduan ataupun dokumentasi yang dapat membantu banyak orang dalam belajar pemrograman. Tentu saja hal ini juga bisa membantu Kamu untuk mendatangkan benefit dan profit baik secara langsung maupun tidak langsung.

## Siapa Saja yang Bisa Menjadi Programmer?

Ketika menyebutkan kata programmer, maka yang terpikirkan pertama kali adalah seorang yang ahli dalam bidang komputer, sehingga muncul persepsi bahwa untuk menjadi seorang programmer harus memiliki latar belakang **IT** atau paling tidak menguasai segala macam persoalan terkait komputer.

Pernyataan ini tidak sepenuhnya salah. Sebab sering kita jumpai umumnya para programmer adalah orang-orang dari lulusan **IT**. Namun tidak sedikit pula programmer yang berasal dari **non-IT**.

Intinya siapapun dapat menjadi programmer. Modal dasarnya hanyalah pengetahuan dasar tentang komputer, minimal mampu mengoperasikan komputer. Sisanya adalah dengan proses belajar yang baik dan tekun.

Di era ini, telah menjamur panduan (tutorial), dokumentasi, bahkan pelatihan-pelatihan baik secara gratis maupun berbayar yang akan membantu kita untuk belajar bahasa pemrograman. Sehingga hampir setiap orang mempunyai peluang untuk belajar bahasa pemrograman. Hal yang perlu ditanamkan sebelum memutuskan untuk belajar bahasa pemrograman adalah keinginan kuat untuk belajar, karena belajar pemrograman tidak sesulit yang Kamu bayangkan tapi juga tidak semudah membalikkan telapak tangan.

Selain tentang latar pendidikan, umumnya usia juga kadang menjadi isu bagi pembelajar pemrograman. Jika kita berpikir bahwa seorang programmer pastilah seorang sarjana komputer atau seorang insinyur, atau orang dewasa yang sudah cukup matang pemikirannya, kenyataannya ternyata tidak mesti demikian. Di internet dan media sosial bertebaran informasi tentang anak-anak yang berprestasi menghasilkan aplikasi buatannya sendiri, hingga orang tua yang beralih profesi dari petani menjadi programmer.

Ada juga bahkan kisah tentang pria homeless yang diajari pemrograman dan akhirnya menjadi seorang programmer mobile.

Artinya siapapun dapat mempelajari pemrograman dan berkarir di bidang ini.

Pembahasan selanjutnya tinggal bagaimana dia mau belajar dengan sungguh-sungguh dan fokus hingga hasil belajarnya dapat dipetik.

## Apakah Saya Harus Menguasai Matematika untuk Menguasai Pemrograman?

Ada pertanyaan yang menarik perhatian ketika akan memulai belajar bahasa pemrograman, **haruskah menguasai matematika**?. Jawaban singkatnya tentu tidak harus. Akan tetapi untuk mempelajari pemrograman, Kamu harus berupaya agar memiliki **pola pikir matematis.**

Kebanyakan orang menganggap matematika sebagai sebuah proses aritmatika atau menghafalkan formula yang kompleks. Ini sama halnya menanyakan apakah Kamu memerlukan ejaan yang sempurna atau memiliki kosa kata yang tinggi untuk menjadi penulis yang baik?

Menulis adalah tentang komunikasi, dan matematika / pemrograman adalah tentang proses berpikir logis (kemampuan memecahkan masalah secara sistematis).

Peran menguasai matematika berada pada dua sisi tergantung peruntukannya. Sebab Kamu bisa menjadi programmer yang hebat tanpa perlu mengetahui matematika tingkat lanjut. Akan tetapi berbeda halnya jika Kamu ingin menjadi programmer low-level language, membuat aplikasi pemrosesan gambar, membuat game 3D, atau analisis spesifik tentang suatu penelitian, maka matematika akan memainkan peran yang besar.

Sebagai seorang programmer, yang terpenting yang harus Kamu miliki adalah logika yang baik dan kemampuan memecahkan masalah.

Namun orang pandai dalam matematika umumnya lebih mudah dalam mempelajari pemrograman. Bukan berarti belajar matematika tidak penting, akan tetapi bukan juga menjadi halangan untuk Kamu mulai belajar pemrograman.

## Perangkat yang Dibutuhkan untuk Belajar Pemrograman

Ketika akan memulai belajar pemrograman, kita harus memiliki perangkat untuk memprogram. Kita akan perlu menjabarkan secara detail berdasarkan jenis pemrograman yang akan Kamu geluti, namun secara garis besar hal yang paling diperlukan adalah komputer atau laptop.

Bila Kamu hendak belajar pemrograman **web**, komputer atau laptop yang terpasang aplikasi pendukung **pemrograman web** seperti **web server**, **code editor** dan **database server** sudah cukup untuk memulai belajar memprogram.

Spesifikasi komputernya pun tidak perlu terlalu besar, **processor dual core** dan **memori RAM 2GB** sudah cukup. Bila Kamu punya spesifikasi yang lebih tinggi itu akan meningkatkan kecepatan pengembangan aplikasimu.

Selain itu bila Kamu sudah masuk pada tahap belajar deployment atau memasang aplikasi webmu di internet, Kamu akan memerlukan **layanan hosting** yang bisa Kamu sewa mulai dari 10 ribu per bulannya.

Bila Kamu hendak membuat aplikasi **desktop**, perangkat yang kamu perlukan tidak jauh dengan kebutuhan pemrograman web. Termasuk bila Kamu hendak membuat aplikasi game untuk desktop. Hanya saja bila Kamu hendak belajar pemrograman aplikasi game, Kamu akan memerlukan memori **RAM** yang sedikit lebih tinggi, serta VGA card yang bagus di komputermu.

Bila Kamu hendak membuat aplikasi **mobile Android**, Kamu bakal perlu memori **RAM minimal 4GB** supaya proses belajarnya nyaman (tidak perlu menunggu proses loading Android Studio dan proses compiling terlalu lama). Dan tentu saja Kamu akan memerlukan smartphone Android untuk mencoba aplikasimu langsung di perangkat mobile.

Khusus untuk pengembangan aplikasi iOS (mobile maupun desktop), Kamu harus menggunakan perangkat **Apple** (iMac, Macbook, iPhone dan lain-lain).

## Istilah-istilah Umum dalam Pemrograman

Setiap keilmuan pasti memiliki istilah-istilah yang nampaknya keren, akan tetapi menjadi hal yang membingungkan bagi orang yang tidak berada dalam bidang tersebut, dalam hal ini bidang programming. Oleh karena itu kita perlu mengetahui agar kita paham ketika dihadapkan dengan istilah tersebut, misal ketika berbincang, bertanya dalam forum, dan lain sebagainya.

Berikut ini beberapa istilah yang umum digunakan di dunia pemrograman :

* **Server**

Perangkat yang melayani permintaan data atau kebutuhan aplikasi di sisi klien.

* **Browser**

Perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi di Internet

* **Back-end Developer**

Seorang programmer yang bertugas untuk mengurusi sisi lain dari sebuah website yang tidak tampak, meliputi server, aplikasi, database, dll. Kode yang dituliskan oleh back-end developer inilah yang menjembatani informasi dari database ke browser.

* **Front-end Developer**

Seorang programmer yang bertugas membuat tampilan yang meliputi layout, feedback, dll dari sebuah website, sehingga pengguna dapat melihat dan berinteraksi dengan website tersebut secara langsung.

* **Fullstack Developer**

Seorang programmer yang mampu bekerja sebagai Front-end dan Back-end sekaligus.

* **Debug**

Proses mencari, menemukan dan mengatasi masalah yang terdapat dari sebuah program yang dapat mengganggu kelancaran penggunaan aplikasi tersebut dari berbagai sisi.

* **Bug**

Sebuah error atau kesalahan yang terdapat dalam sebuah program aplikasi sehingga mengganggu kelancaran penggunaan aplikasi.

* **Syntax**

Bisa diartikan sebagai tata bahasanya dalam sebuah bahasa pemrograman.

* **Method/Function**

Sebuah kode blok yang dapat digunakan kembali dalam proses tertentu sesuai kebutuhan.

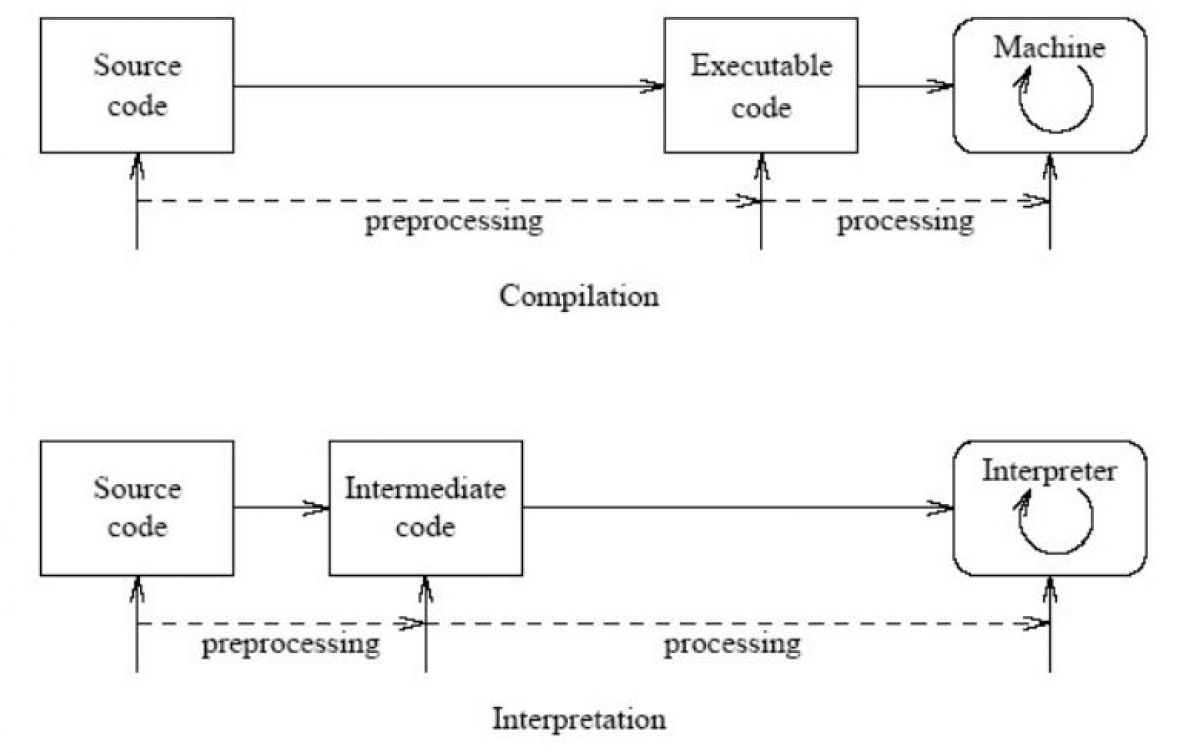
* **API** (Aplication Programming Interface)

Antarmuka aplikasi yang memungkinkan dua atau lebih program aplikasi untuk saling berkomunikasi satu sama lain.

Sebenarnya banyak sekali istilah di dunia pemrograman. Yang perlu Kamu lakukan bila menemukan istilah baru yang belum Kamu ketahui adalah dengan mencari definisinya di kamus istilah atau mesin pencari. Semakin lama Kamu bergelut dengan pemrograman akan semakin akrab dengan istilah-istilah lainnya.

## Bagaimana Proses Sebuah Program Dibuat?

Di bawah ini bagan ilustrasi proses kompilasi dan interpretasi kode program:



Beberapa bahasa pemrograman terkenal yang berbasis interpreted diantaranya adalah J, Julia, JavaScript, Lua, MATLAB, Perl, PHP, Python, R, Ruby, VBScript, dan Wolfram Language.

Beberapa bahasa pemrograman terkenal yang berbasis compiled diantaranya adalah ActionScript, C, C++, C#, COBOL, JavaScript, Delphi, Erlang, Fortran, Java, Objective-C, Pascal, Scala, Visual Basic, Visual FoxPro.

Compile membutuhkan Compiler

Interpretasi membutuhkan interpreter ( menjalankan interpreter sekaligus menjalankan programnya / interpreter berjalan secara realtime pada saat menterjemahkan source code baris perbaris)

## Bagaimana Cara Memulai Belajar Pemrograman?

Setelah Kamu mengetahui apa itu pemrograman, keuntungan belajar pemrograman, apa itu bahasa pemrograman dan lain sebagainya. Maka tiba saatnya untuk mulai belajar pemrograman, namun kerap kali hal ini akan menimbulkan pertanyaan baru, *Bagaimana cara memulai belajar pemrograman?*.

Saat ini telah banyak media sebagai alternatif untuk belajar pemrograman baik yang bersifat offline maupun online. Untuk metode belajar baik secara offline maupun online kembali kepada pribadi masing-masing. Adapun yang akan kami jabarkan dibawah ini langkah memulai belajar meskipun setiap orang berbeda, tapi secara garis besar dapat dijadikan rujukan.

* Tetapkan bahasa pemrograman yang ingin Kamu pelajari, untuk memulai cukup satu bahasa pemrograman saja agar Kamu menjadi lebih fokus dalam belajar.
* Beli buku pemrograman (baik bersifat fisik maupun elektronik) dan ikuti panduannya. Apabila Kamu mempunyai kemampuan berbahasa inggris dan daya tangkap yang cepat maka sumber belajar primer dari masing-masing bahasa pemrograman umumnya sudah sangat cukup untuk dijadikan referensi belajar.
* Mulailah belajar dasar dari bahasa pemrograman, sebab dasar yang kuat akan menjadikan Kamu lebih mudah dalam mengatasi masalah yang akan Kamu temukan kedepannya.
* Mulai membuat project sederhana untuk melatih kemampuan Kamu dalam memecahkan masalah yang dihadapi.
* Manfaatkan mesin pencari untuk menemukan solusi dari masalah yang Kamu hadapi selama belajar pemrograman. Untuk belajar pemrograman dasar, Kamu nyaris akan selalu menemukan solusinya di internet.
* Bergabung di forum pemrograman sehingga Kamu bisa bertanya, belajar dari pengalaman teman-teman seperjuangan dalam belajar bahasa pemrograman. Perhatikan etika dan tata cara bertanya di forum.
* Berkontribusi di dunia open source atau mulai bekerja secara tim meskipun dalam ruang lingkup kecil. Karena lebih dari satu kepala jauh lebih baik dan Kamu bisa mendapatkan sudut pandang baru jika saling bertukar pendapat maupun informasi dengan rekan setim.
* Ikuti informasi terbaru seputar pemrograman, karena pemrograman adalah salah satu bagian teknologi yang berkembang sangat cepat. Kamu dapat mengikuti artikel-artikel yang dirilis di website **CodePolitan** untuk mendapatkan pembaharuan seputar pemrograman.

Lebih dalam lagi pembahasan tentang ini dapat kamu simak di artikel [Cara Enjoy Memulai Belajar Pemrograman](https://www.codepolitan.com/cara-enjoy-memulai-belajar-pemrograman)

## Bagaimana Caranya Supaya Jadi Programmer Hebat?

Mendapatkan predikat sebagai seorang programmer bukanlah perkara mudah, meskipun untuk mendapatkan predikat tersebut tidak harus melalui upacara khusus pemberian gelar akan tetapi mengklaim diri sebagai seorang programmer harus diikuti dengan bukti kemampuan membuat sebuah program aplikasi. Kamu tidak akan dapat menjadi programmer hebat kecuali dengan melalui proses yang panjang dari satu aplikasi ke aplikasi lainnya. Semakin lama pengalamanmu memprogram, semakin tinggi kemampuanmu.

Perkembangan dunia teknologi yang sangat cepat menuntut seorang programmer untuk mampu belajar cepat juga. *Bertumbuh atau mati*, meski programmer tidak dapat dipisahkan dengan teknologi, akan tapi jika seorang progr ammer tidak mampu mengikuti perkembangan teknologi akan tergerus oleh teknologi itu sendiri. Sebagai programmer Kamu harus menyukai proses belajar.

Berikut beberapa tips yang dapat kami bagikan agar dapat menjadi seorang programmer yang hebat.

* Tanamkan dalam diri Kamu untuk lebih banyak belajar. Langkah pertama dalam mempelajari sesuatu adalah dengan menyadari bahwa Kamu tidak mengetahuinya. Kebanyakan orang terlalu cepat berpuas diri dengan hasil pembelajaran yang mereka lakukan, dan mengklaim diri lebih tahu daripada yang lain. Hal ini akan menimbulkan sekat untuk belajar hal-hal baru yang dibagikan orang lain baik secara langsung maupun tidak.
* Berhentilah untuk menganggap diri paling benar. Untuk menjadi hebat Kamu harus belajar dari pengalaman. Pengalaman bisa mengajari kita untuk tidak mengulagi perilaku buruk dan kebiasaan buruk. Maka setiap proses yang kita lalui harus diikuti dengan pertanyaan, "**Bagaimana saya bisa lebih baik dari sebelumnya ?**". Programmer pemula biasaya melihat kode program mereka untuk mengagumi kehebatannya. Berpuas diri dengan hasil yang telah dicapai menjadikannya luput untuk mencari kesalahan dari program yang telah dibuat, sehingga merasa diri bahwa programnya telah sempurna. Padahal programmer yang benar-benar hebat, secara aktif mencari dimana letak kesalahannya karena mereka tahu bahwa pada akhirnya pengguna akan menemukan kekurangan yang mereka lewatkan.
* "Program Kamu bekerja", bukan berarti Kamu harus berhenti. Akan tetapi disitulah Kamu baru memulai. Langkah pertama yang harus Kamu lakukan adalah dengan selalu menanamkan pada diri sendiri untuk selalu menulis program yang berkualitas yang memenuhi spesifikasi.

Rata-rata programmer berhenti saat programnya telah berjalan dan beralih ke hal lainnya. Programmer yang berpuas diri hanya karena programnya "bisa berjalan", seperti mengambil snapshot dan mengharapkannya menjadi sebuah karya seni.

Programmer hebat tahu, bahwa programnya sudah berjalan dengan baik, akan tetapi dia juga sadar untuk membuatnya jadi lebih baik lagi. Aspek "lebih baik" dapat ditinjau dari berbagai hal, seperti menjadikannya lebih cepat, lebih mudah dibaca, memiliki dokumentasi yang rapi, dapat digunakan kembali, mudah dikembangkan, dan lain-lain.

* Baca kode anda, dan baca lebih banyak kode orang lain, Hal yang penting untuk Kamu lakukan ketika menuliskan kode program adalah dengan banyak membacanya kembali dan sering membaca kode program milik orang lain. Sebab ketika Kamu membaca kode orang lain, Kamu akan melihat bagaimana orang lain memecahkan masalah pemrograman. Sehingga dapat Kamu jadikan referensi pembelajaran sekaligus tantangan untuk menjadi lebih baik.
* Jadikan sebuah kesenangan dalam menulis kode program, bukan hanya sebagai sebuah kewajiban yang harus ditunaikan.
* Belajar teknik, bukan tools.

Bahasa pemrograman, tools dan metodologi datang silih berganti sesuai perkembangan. Itulah sebabnya Kamu harus belajar dasar-dasar pemrograman, sebab dasar pemrograman cenderung tidak berubah. Jika Kamu merasa yakin bahwa hanya ada satu cara yang benar untuk melakukan sesuatu, mungkin ini saatnya untuk Kamu mulai belajar hal baru.

# Jalur Menjadi Programmer

## Web Programming

Web programming adalah serangkaian proses yang dilakukan untuk membuat website baik dari skala kecil seperti halaman yang sifatnya statis berupa plain text, hingga ke skala yang sangat kompleks bersifat dinamis yang telah berinteraksi dengan database atau berkomunikasi antar aplikasi. Beberapa kompetensi yang bersanding dengan web programming diantaranya: web design, database administrator, web server, konfigurasi keamanan server, dan lain-lain.

Beberapa hal mendasar yang perlu kita ketahui untuk menjadi seorang Web Programmer diantaranya:

* HTML, digunakan untuk membuat struktur atau kerangka dari halaman web. Kamu juga akan perlu untuk mempelajari CSS yang berfungsi untuk menata tampilan halaman website.
* Bahasa pemrograman web, digunakan untuk memproses sejumlah operasi yang diperlukan pada aplikasi web seperti menjalankan fungsi logika, berinteraksi dengan database, menampilkan antarmuka yang sesuai dan sebagainya. Beberapa bahasa pemrograman web diantaranya Javascript, PHP, Ruby, Python, Perl dan ASP.NET.
* Web Browser, adalah perangkat lunak yang berfungsi untuk menyajikan halaman website yang kita inginkan. Beberapa web browser populer diantaranya: Mozilla Firefox, Google Chrome, Internet Explorer, Microsoft Edge, Opera, Safari dan sebagainya.
* Code Editor, adalah perangkat lunak yang digunakan untuk menuliskan kode program
* Web Server, adalah perangkat lunak yang digunakan untuk menjalankan aplikasi web
* Domain, baik di lokal maupun di jaringan internet kita butuhkan sebagai alamat identitas aplikasi web kita. Jika dalam mode offline kita mengenal localhost / 127.0.0.1, maka di online kita bisa memiliki nama domain secara custom dengan sistem penyewaan domain tersebut seperti: codepolitan.com, facebook.com atau bisa juga namakamu.com.

Web programming menjadi daya tarik banyak orang untuk dipelajari karena potensi pekerjaan sebagai web prorammer masih sangat luas. Seiring dengan terus berkembangnya teknologi internet berkembang dan meningkat pula kebutuhan akan pengembangan aplikasi web. Bila dahulu orang mengenal web programming sebatas untuk membuat website saja, kini pemanfaatan aplikasi berbasis website menjadi lebih luas lagi, seperti pengembangan sistem informasi, media sosial, ecommerce dan transaksi keuangan. Kita bahkan dapat membuat aplikasi desktop dan mobile menggunakan teknologi web. Belajar pemrograman web menjadi pilihan yang tepat dan prospektif

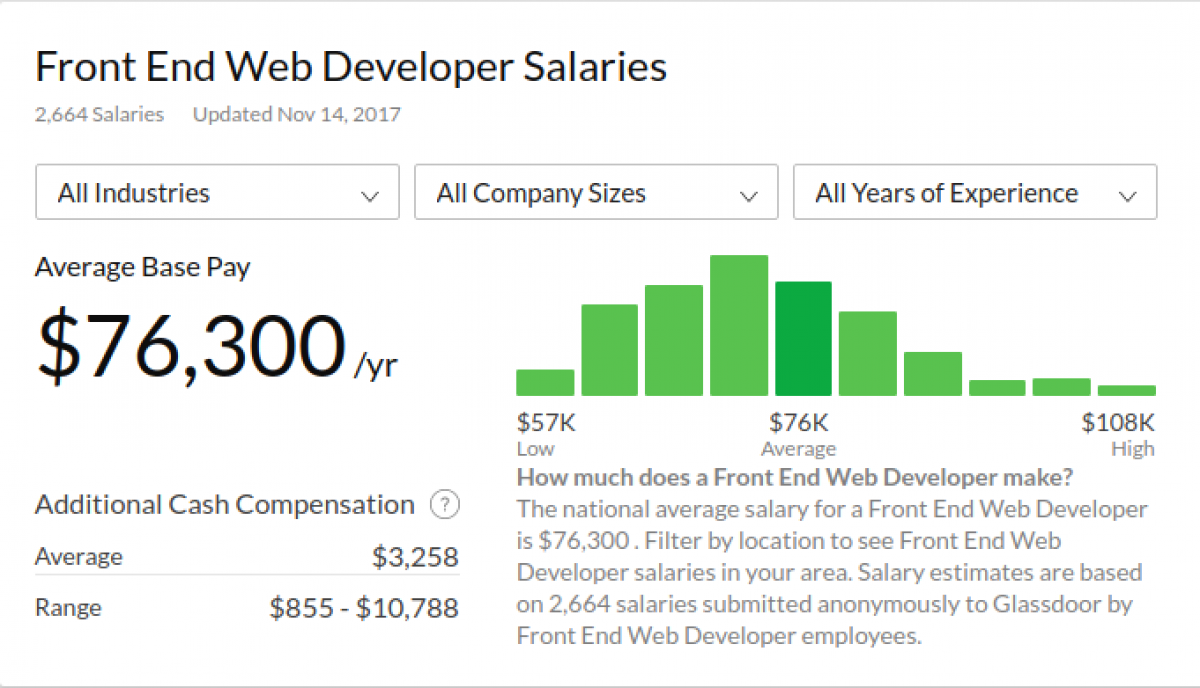
## Frontend Developer

Ketika kita mengakses sebuah website maka yang akan disuguhkan pertama kali adalah layout atau tampilan dari website tersebut dengan beragam style agar terlihat memikat namun tetap informatif. Keilmuan yang menangani hal ini dalam dunia web programming disebut dengan istilah Frontend Programming.

Frontend programming memiliki tugas untuk membuat layout sebuah website yang interaktif, responsive, sehingga memberikan kesan kepada user bahwa website tersebut mudah digunakan. Adapun bahasa yang digunakan untuk menjadi Frontend programmer meliputi: HTML, CSS, Javascript. Sebab apapun yang Kamu lihat ketika mengakses sebuah halaman web merupakan kombinasi antara HTML, CSS dan Javascript yang diterjemahkan oleh browser, hal ini termasuk font, menu drop-down, button, transition, slide, form, dan lain-lain. Kini bahkan sudah banyak bertebaran framework untuk frontend, yakni kumpulan library CSS dan JavaScript yang dapat digunakan untuk mempercepat proses pengembangan web.

Tak hanya berada dalam ruang lingkup layout atau tampilan dari sebuah website, frontend developer juga memikirkan hal yang lebih kompleks lagi. Google telah memperkenalkan metode baru dalam dunia frontend yakni Progressive Web App (PWA) dimana kita dapat melakukan optimasi agar website dapat diakses dengan lebih cepat dan dapat diakses meskipun dalam keadaan tidak terhubung ke internet.

Frontend programming adalah salah satu bagian dari web programming yang mengkhususkan diri di ranah frontend saja. Frontend programmer tidak menangani managemen data dan infrastruktur. Ia hanya fokus membuat penyajian konten di halaman depan sebuah aplikasi web. Ada banyak permintaan sumberdaya khusus sebagai frontend programmer, dengan rentang salary yang menarik.

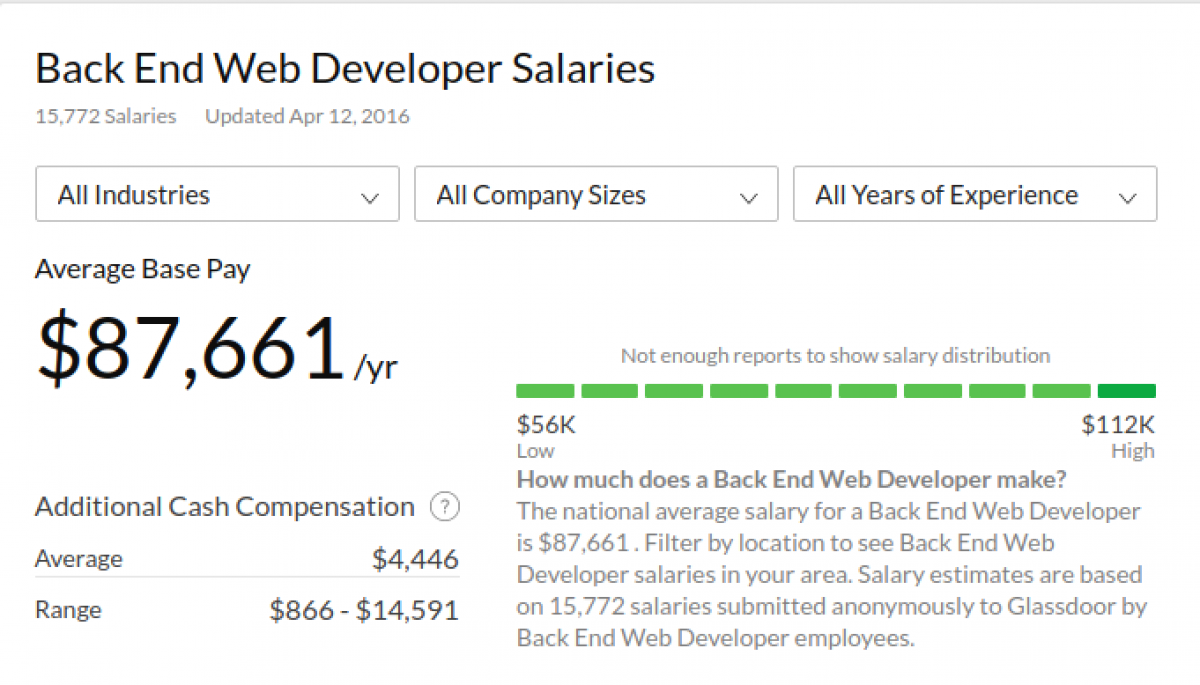
 Ilustrasi pendapatan rata-rata per tahun programmer frontend ([*sumber*](http://www.glassdoor.com/Salaries/front-end-web-developer-salary-SRCH_KO0,23.htm))

## Backend Developer

Berbeda halnya dengan Frontend programming, Backend programming bekerja di belakang layar atau biasa disebut dengan istilah *server side*. Backend biasanya terdiri dari tiga bagian: aplikasi, server, dan database. Analogi sederhananya adalah ketika Kamu memesan barang di salah satu website jual beli, maka yang Kamu lihat atau apa yang disuguhkan kepada Kamu adalah frontend. Ketika Kamu menginput pesanan, maka backend bekerja untuk menyimpannya ke dalam database. Proses backend nyaris tidakt terlihat layaknya frontend, akan tetapi backend bekerja untuk memenuhi kebutuhan proses untuk ditampilkan di frontend.

Untuk menjadi seorang backend programmer, berikut hal yang perlu kamu ketahui:

* Bahasa pemrograman server-side seperti Ruby, PHP, Phyton, JavaScript, dan lain-lain. Sama halnya dengan frontend programming, backend programming juga telah memiliki banyak framework yang dapat memudahkan dan mempercepat proses pengembangan web.
* Paham teknologi yang digunakan dalam frontend programming seperti HTML, CSS dan Javascript. Hal ini perlu dilakukan agar dapat berkomunikasi dengan tim frontend. Sebab backend dan frontend adalah satu kesatuan dalam web programming yang harus saling berinteraksi untuk membangun aplikasi.
* Memiliki kemampuan untuk menangani server, termasuk database administrasi. Meskipun ada profesi tersendiri untuk itu yakni system administrator dan database administrator.
* Baik bekerja secara individu maupun kelompok. Menggunakan Version Control System (VCS) adalah hal yang wajib dilakukan dalam proses kolaborasi.
* Memahami konsep pengembangan API atau application programming interface. API akan sangat diperlukan terutama pada proyek pengembangan aplikasi untuk banyak platform seperti web, mobile, atau keduanya sekaligus.

 Ilustrasi pendapatan rata-rata per tahun programmer backend ([*sumber*](http://www.glassdoor.com/Salaries/back-end-web-developer-salary-SRCH_KO0,22.htm))

## Mobile Programming

Berbicara tentang mobile programming, maka bukan hanya berbicara tentang bagaimana membuat aplikasi untuk perangkat mobile meskipun hal tersebut menjadi tujuan utama. Mobile programming adalah proses yang dilakukan untuk membuat aplikasi berbagai jenis perangkat mobile. Maksud dari pernyataan ini, bahwa mobile programming adalah proses development perangkat lunak yang mencakup aplikasi ponsel, tablet, smartwatch dan semua jenis perangkat mobile lainnya.

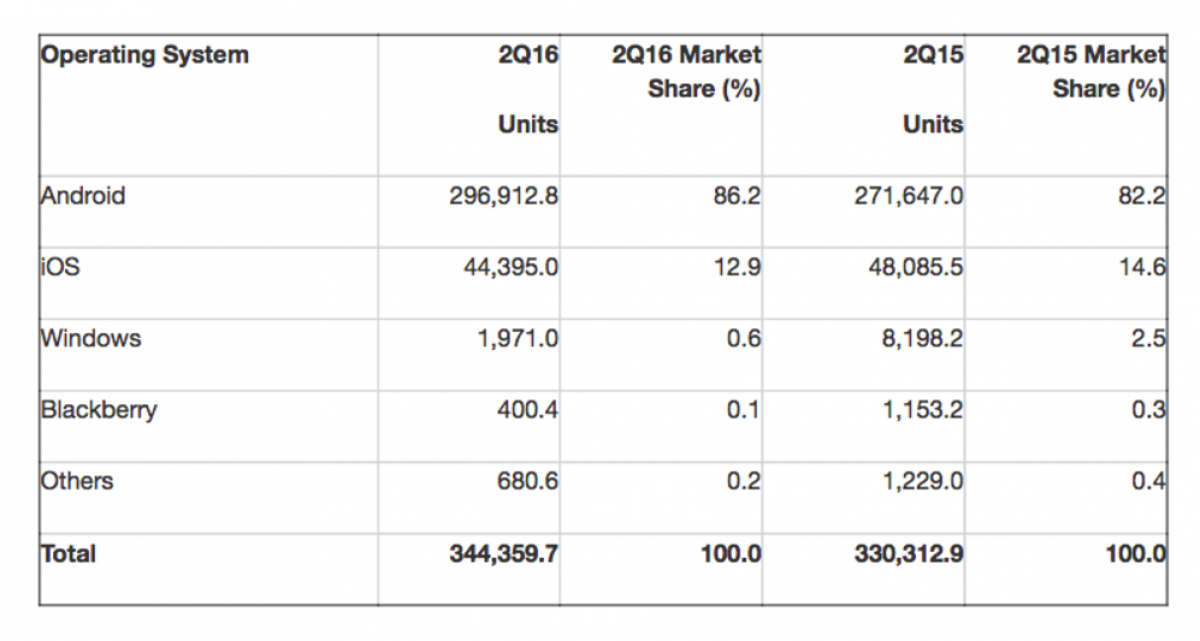
Terdapat dua sistem operasi mobile yang paling banyak digunakan saat ini, yaitu Android dan iOS. Masing-masing sistem operasi tersebut memiliki kelebihan tersendiri dan membutuhkan perangkat sesuai spesifikasinya. Termasuk dalam membuat aplikasinya terkadang kita harus membuat dua versi aplikasi agar dapat berjalan di kedua sistem operasi tersebut. Meskipun belakangan ini ada teknologi yang dapat mencakup keduanya sekaligus, artinya kita hanya perlu menuliskan satu kode program yang nantinya dapat dicompile ke dua jenis sistem operasi ini.

Mobile programmer kini menjadi salah satu profesi yang paling banyak dicari. Hal ini muncul karena teknologi mobile sudah berkembang dan disinyalir akan menjadi perkembangan teknologi paling pesat di masa depan. Kalau dahulu internet hanya bisa dinikmati melalui komputer atau laptop dan sebagian perangkat telepon genggam yang terkoneksi ke internet, kini smartphone sudah dimiliki oleh mayoritas orang. Berdasarkan statistik, dari 5 milyar penduduk bumi tercatat ada lebih dari 3,4 milyar pengguna aktif internet melalui smartphone. Ditambah lagi perkembangan teknologi mobile ke banyak bentuk seperti smartwatch dan internet of things yang terintegrasi ke smartphone.

## Android Programming

Android adalah sistem operasi terpopuler di dunia untuk perangkat seluler dan tablet. Android adalah sistem operasi open source yang dibuat oleh Google. Belakangan Android juga dikembangkan untuk dapat berjalan di smartwatch, smart TV dan juga laptop/PC.

Hingga saat ini, Android adalah pemimpin pasar smartphone secara global yang tak terbantahkan. Karena pertumbuhan di pasar negara berkembang seperti Meksiko, Turki dan Brasil, dominasi ini tidak akan lagi mengalami penurunan secara tajam dalam waktu dekat.

 Sumber: [*9to5Mac*](https://9to5mac.com/2016/08/18/android-ios-smartphone-market-share/)

Lalu apa saja yang perlu kamu ketahui untuk menjadi seorang android programmer?

* XML (Extensible Markup Language), digunakan untuk membuat tampilan antarmuka aplikasi Android
* Java, untuk menjadi seorang android programmer Kamu harus terbiasa dengan bahasa pemrograman Java. Java adalah salah satu bahasa pemrograman terpopuler yang digunakan software developers saat ini.
* SQL, Kamu perlu untuk mempelajari dasar-dasar SQL untuk mengatur database dalam aplikasi android Kamu.
* Android software development kit (SDK) dan Android studio

## IOS Programming

Salah satu bidang lain yang termasuk kategori Mobile programming adalah iOS programming. iOS programming diperutukkan untuk membuat aplikasi berbasis iOS yang dapat dijalankan pada perangkat yang mendukungnya. Orang yang menggelutinya disebut sebagai iOS Programmer.

Sebelum belajar membuat aplikasi iOS, maka berikut beberapa hal yang perlu kamu ketahui

* Swift, adalah bahasa pemrograman yang dibuat Apple khusus untuk developer aplikasi mereka. Swift adalah bahasa yang banyak digunakan dalam developmen aplikasi iOS.
* Design Guidelines, yang membuat aplikasi iOS menarik perhatian karena sangat intiutif. Alasannya adalah karena aplikasi ini mengikuti aturan design guidelines yang sangat ketat. Tentang bagaimana data disimpan dan ditampilkan kepada pengguna ditangani dengan cara yang sangat spesifik. Untuk memahami itu setidaknya Kamu harus paham tentang konsep MVC (Model View Controller), Delegate Pattern, Notification, dan lain sebagainya.
* Xcode, adalah IDE (Integrated Development Environment) untuk membuat aplikasi iOS. Dengan menggunakan Xcode Kamu dapat melakukan penulisan kode program, pengeditan, dan debugging untuk aplikasi iOS. Kamu juga dapat menggunakan Xcode untuk mengupload aplikasi Kamu ke App Store.

## Desktop Programming

Aplikasi desktop adalah aplikasi yang dapat diinstal dan berjalan di komputer desktop atau laptop tanpa dependensi dari aplikasi lain. Berbeda halnya dengan aplikasi berbasis web yang membutuhkan browser agar dapat diakses.

Lalu mengapa aplikasi berbasis desktop masih digunakan?

* Ada beberapa jenis aplikasi yang belum memungkinkan atau tidak efisien bila dibuat dalam basis web atau mobile, seperti aplikasi pengolah citra, image editor, dan aplikasi atau sistem informasi yang tidak memerlukan koneksi internet.
* Dalam hal keamanan merupakan masalah yang tidak terlalu kompleks, karena aplikasi berbasis desktop dapat dispesifikkan untuk pribadi sehingga tidak terbuka untuk umum dalam hal mengaksesnya sehingga Kamu lebih mudah melindungi data dan program Kamu.
* Aplikasi desktop karena di-compile kedalam bahasa mesin secara langsung, dapat berjalan dengan sangat cepat ketimbang aplikasi berbasis web
* Aplikasi desktop tidak harus bersaing dengan keterbatasan yang dimiliki oleh browser sehingga dapat menjadi lebih efisien dalam desain dan dan performa
* Ada kebutuhan dimana pengguna hanya ingin sebuah aplikasi hanya dapat diakses di komputer tertentu saja, seperti aplikasi kasir supermarket dan sistem informasi internal sebuah instansi (rumah sakit, apotik, dan perusahaan manufaktur).

Beberapa bahasa pemrograman yang umum digunakan untuk membuat aplikasi desktop diantaranya adalah Java, Python, Delphi, Qt, C#, C/C++ dan Visual Basic.

## Hardware/Embedded System Programming

Hardware programming adalah bagian dari hardware engineering. Hardware programmer adalah orang yang melakukan programming secara low level terhadap hardware, misalnya mikrokontroler, embedded system, PLC atau device lainnya. Pada awal diciptakannya komputer, programmer jenis ini lebih dominan karena cara memprogram komputer waktu itu mirip dengan cara memprogram mikrokontroller saat ini. Bahasa yang digunakan dulunya adalah bahasa mesin tetapi saat ini yang umum digunakan bahasa Assembly dan C. Selain dua bahasa tersebut, sekarang kita sudah dapat memprogram mikrokontroller menggunakan bahasa Python dan JavaScript.

Beberapa hal dasar yang perlu Kamu ketahui untuk mulai belajar Hardware/Embedded System programming:

* Memilih bahasa pemrograman
  1. Embedded System dapat diprogram menggunakan bahasa pemrograman tingkat rendah dan bahasa pemrograman tingkat tinggi
  2. Kedua bahasa memiliki kelebihan dan cara penggunannya tersendiri. Bahasa assembly digunakan untuk pemrograman tingkat rendah. Untuk pemrograman tingkat tinggi, Kamu dapat menggunakan C.
  3. Sistem yang memerlukan efisiensi tinggi untuk sumberdaya pada alat umumnya diprogram menggunakan menggunakan bahasa tingkat rendah. Pengguna memiliki kontrol lebih pada hardware dan memori menggunakan bahasa pemrograman tingkat rendah.
  4. Sistem yang tidak terlalu kompleks menggunakan bahasa pemrograman tingkat tinggi. Bahasa semacam ini mudah dimengerti dan programnya mudah dibaca.
  5. Sebagai seorang pemula, Kamu bisa memulai mempelajari pemrograman hardware menggunakan bahasa pemrograman C.
* Belajar C / C++
  1. Hal kedua yang harus dilakukan untuk membuat embedded system adalah belajar bahasa pemrograman. Bahasa yang paling umum digunakan adalah C atau C++.
  2. Sebagian besar produk embedded system dirancang sedemikian rupa sehingga mendukung bahasa C
  3. Program yang dibangun dengan bahasa C dapat dibaca dan mudah untuk melakukan debugging. Bahasa ini efisien dan memberikan dukungan untuk perangkat input dan output.
* Microcontroller, adalah bagian penting dari \*embedded system\*. Sebuah microcontroller akan bertindak sebagai CPU dari sebuah hardware. Selain itu microcontroller umumnya sudah memiliki RAM, ROM dan beberapa periferal lainnya seperti time, counter, dan lain-lain. Microcontroller yang paling umum digunakan adalah ATMega, ARM, PIC Microcontroller atau 8051 Microcontroller, dan lain-lain.
* Memiliki Dasar Pengetahuan tentang elektronik. Untuk mempelajari \*embedded system\*, Kamu perlu tahu tentang perangkat elektronik. Jika Kamu tidak memiliki dasar tentang elektronika, Kamu akan kesulitan dalam merancang program embedded system. Embedded system bukan hanya sekedar menuliskan kode program. karena tujuan Kamu adalah membuat sistem yang bisa mempengaruhi hal-hal fisik, maka Kamu akan berhubungan langsung dengan perangkat sensor dan aktuator.

# Hal yang Wajib Dikuasai Programmer

## Algoritma dan Struktur Data

Dalam pemrograman, algoritma adalah himpunan instruksi yang didefinisikan dengan baik secara berurutan untuk menyelesaikan sebuah program. Algoritma harus selalu memiliki titik penyelesaian masalah sehingga apa yang kita inginkan dapat dibuat sebagaimana mestinya. Dalam matematika dan ilmu komputer, algoritma adalah prosedur untuk memecahkan masalah menjadi bagian-bagian kecil lalu menyelesaikannya satu persatu secara berurutan. Algoritma sangat penting dalam membuat program, sebab dengan algoritma kita bisa menggambarkan langkah apa saja yang harus dilakukan oleh sebuah program.

Ciri-ciri algoritma yang bagus diantaranya:

* Input dan Output harus didefinisikan secara tepat
* Setiap langkah dalam algoritma harus jelas dan tidak ambigu
* Algoritma yang digunakan harus paling efektif diantara banyak cara yang dapat digunakan untuk memecahkan suatu masalah spesifik
* Algoritma seharusnya dibuat tanpa menggunakan kode komputer. Akan tetapi algoritma harus ditulis sedemikian rupa sehingga dapat dipahami dan digunakan oleh orang lain dalam bahasa pemrograman lain

Dalam banyak kasus, ada beberapa cara untuk melakukan operasi tertentu dalam sebuah program. Oleh karena itu, programmer biasanya menciptakan algoritma yang paling efisien. Dengan menggunakan algoritma yang efisien, programmer dapat memastikan program mereka dapat berjalan secepat mungkin dan menggunakan sumber daya seminimal mungkin. Tentu saja, tidak semua algoritma diciptakan dengan sempurna untuk pertama kalinya. Oleh karena itu, programmer sering kali memperbaiki algoritma yang digunakannya dan melakukan pembaharuan pada program yang dimilikinya. Jika Kamu melihat versi baru dari sebuah program telah dioptimasi atau memiliki kinerja lebih cepat, itu berarti telah dilakukan pembaharuan algoritma menjadi lebih efisien, selain juga mencakup pembaharuan teknologi yang digunakan.

Bentuk Struktur Data

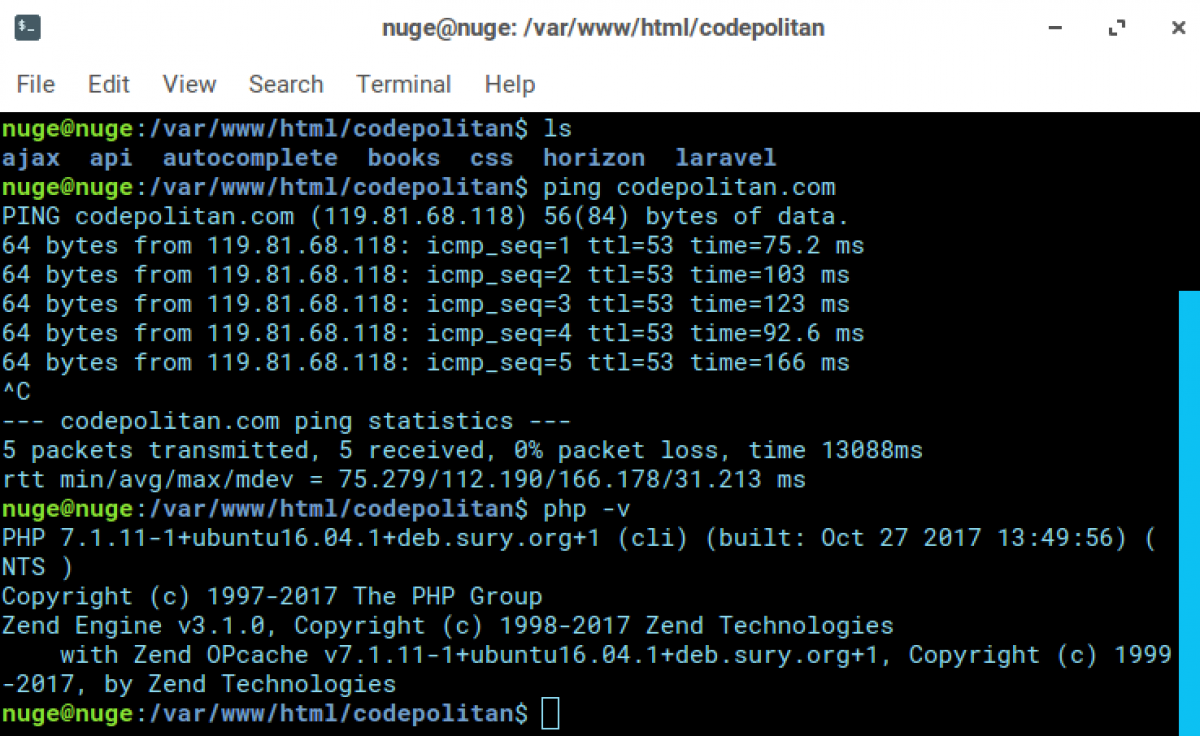
* Array/Larik
* List/Senarai
* Stack/Tumpukan
* Queue/Antrian
* Tree/Data berbentuk Pohon

## Pentingnya Bahasa Inggris

## Command Line

CLI (Command Line Interface) adalah antarmuka berbasis teks yang digunakan untuk mengoperasikan perangkat lunak dan sistem operasi dengan menggunakan perintah berupa teks. CLI adalah metode lama untuk berinteraksi dengan aplikasi dan sistem operasi. CLI umumnya digunakan untuk melakukan tugas spesifik sesuai yang diinginkan oleh pengguna.

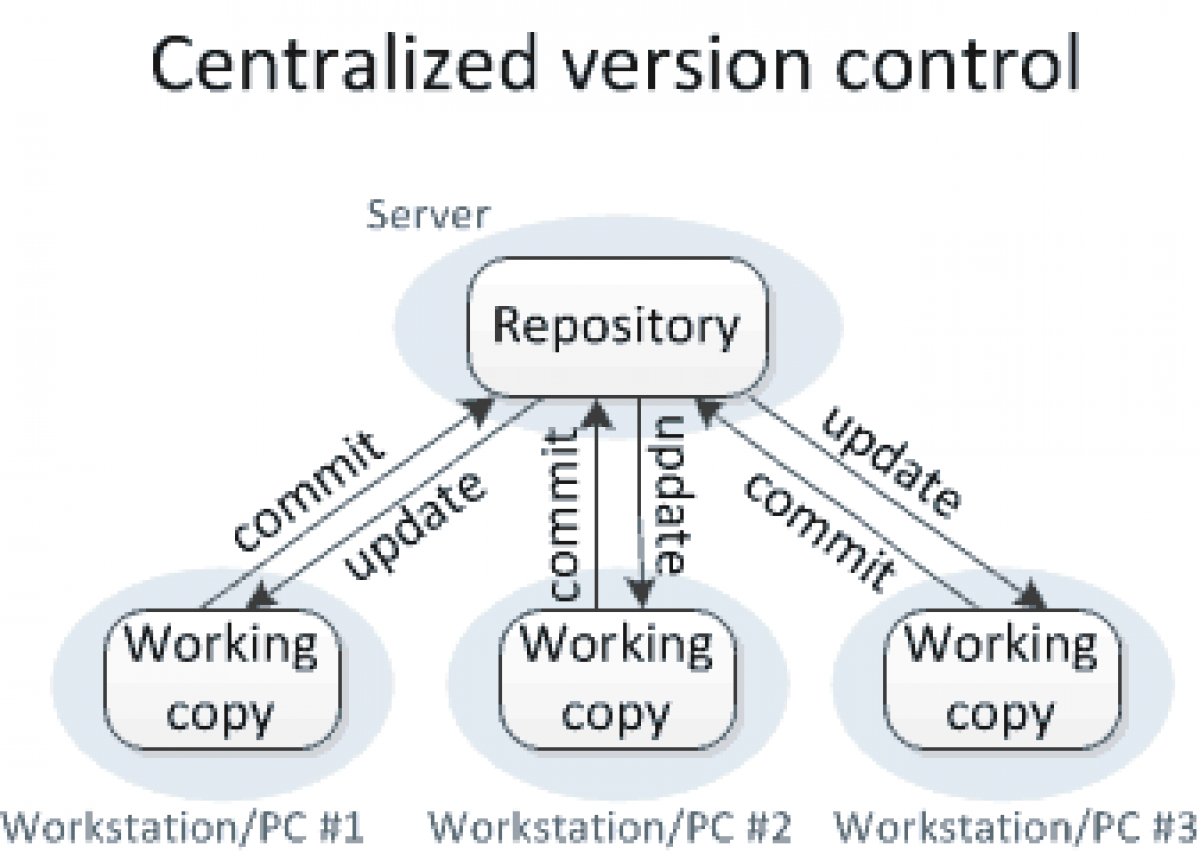
CLI berbeda dengan GUI (Graphical User Interface) yang menyediakan antarmuka grafis sehingga setiap orang dapat mengoperasikannya dengan mudah. CLI hadir dengan kumpulan perintahnya sesuai peruntukannya sehingga kebanyakan orang menghindari untuk menggunakannya karena enggan untuk menghapalkan perintah-perintah tersebut. Namun disisi lain dari kesan yang 'tidak menarik' itu ternyata menyimpan daya tarik tersendiri, terlebih bagi programmer. Karena dengan CLI kita justru dapat lebih mudah untuk melakukan pekerjaan yang sederhana dan berulang dengan cepat tanpa dibebani pemuatan antarmuka grafis. Selain itu membuat program berbasis CLI lebih mudah daripada membuat aplikasi GUI, terutama bila program yang diinginkan hanya melakukan proses sederhana tapi sering digunakan.

 Contoh tampilan command line interface

## Version Control System

Version control system atau disingkat VCS adalah kategori perangkat lunak yang dapat membantu Kamu atau tim Kamu untuk mengelola setiap perubahan yang dilakukan dari waktu ke waktu. VCS merekam setiap perubahan kode yang dilakukan sehingga jika terjadi kesalahan kita dapat melakukan rollback untuk membanding code sebelum dan sesudah perubahan dilakukan sehingga dapat membantu kita dalam meminimalisir kesalahan.

Bagi seorang programmer source yang telah dibuat adalah aset yang berharga karena berisi kumpulan solusi dari masalah yang telah dipecahkan dan diselesaikan. Dengan adanya VCS dapat melindungi source code dari kesalahan yang dilakukan oleh manusia dan dari hal-hal lain yang tidak diinginkan. Tanpa VCS Kamu akan kesulitan dalam mengelola dan berkordinasi dengan tim terkait perubahan yang telah dilakukan sehingga tidak saling timpa-menimpa satu sama lain. Dalam mengembangkan perangkat lunak tanpa menggunakan VCS sangat beresiko, seperti tidak memiliki backup. VCS juga dapat membantu programmer untuk bekerja lebih cepat tanpa harus memikirkan file mana saja yang telah diperbaharui.

 Ilustrasi alur managemen versi kode dengan VCS

VCS dikenal juga dengan istilah SCM (Source Code Management) tools atau RCS (Revision Controler System). Salah satu tools VCS yang paling populer dan banyak digunakan adalah Git. Git adalah VCS terdistribusi, bersifat open source dan gratis untuk digunakan. Mengingat pentingnya VCS dalam pengembangan perangkat lunak, menjadikannya salah satu hal yang wajib Kamu pelajari sebagai programmer

## Package Manager

Package manager adalah perangkat lunak atau tools yang digunakan untuk proses otomatisasi melakukan penginstalan, upgrading, konfigurasi dan menghapus program komputer untuk sistem operasi atau library dengan tuntas dengan menggunakan command tertentu. Package manajer menangani package, mendistribusikan perangkat lunak dan data dalam file arsip. Package berisi metadata, seperti nama software, deskripsi, version number, vendor, checksum dan daftar dependensi yang diperlukan agar software dapat berjalan dengan baik

Dalam dunia pemrograman juga terdapat berbagai macam package manager yang dapat digunakan guna menunjang proses development agar menjadi lebih cepat. Sebab kerap kali fitur yang kita inginkan sudah dibuat oleh orang lain menjadi sebuah library, sehingga kita tidak perlu membuat fitur yang serupa dan dapat menggunakan library yang sudah ada untuk diintegrasikan pada project kita.

Berikut beberapa nama package manager dan bahasa pemrograman yang menggunakannya, diantaranya:

* Cabal (Haskell) : http://www.haskell.org/cabal
* Composer (PHP) : https://getcomposer.org
* Go Nuts (Go) : http://www.gonuts.io
* Gradle (Groovy dan JVM language) : http://www.gradle.org
* CPAN (Perl) : http://www.cpan.org
* PIP / PyPI (Python) : https://pip.pypa.io
* Bower (Js Css) : http://bower.io
* NuGet (.NET Framework) : https://www.nuget.org
* CocoaPods (Objective-C dan RubyMotion projects) : http://cocoapods.org
* LuaRocks (Lua) : http://luarocks.org
* Maven (Java) : http://maven.apache.org
* npm (Node.js) : https://www.npmjs.org
* RubyGems (Ruby) : https://rubygems.org
* sbt (Scala) : http://www.scala-sbt.org
* leiningen (Clojure) : http://leiningen.org

Cara menggunakan package manager di masing-masing bahasa pemrograman hampir serupa. Umumnya dengan membuat sebuah file berisi penentuan library apa saja yang dibutuhkan untuk aplikasi yang dikembangkan. File tersebut lantas di simpan bersamaan dengan project kita dan menjadi acuan bila hendak melakukan pembaharuan library.

## Security

Security dalam dunia teknologi adalah ide yang diterapkan untuk melindungi software dari serangan berbahaya dan resiko lainnya sehingga software tetap berfungsi sebagaimana mestinya. Security diperlukan untuk memberikan integritas, otentikasi dan availability dari aplikasi Kamu. Sebagai seorang programmer kita harus memikirkan keamanan dari aplikasi yang dibuat, dan seiring berjalannya waktu kita harus memperbaiki celah keamanan yang ditemukan. Celah keamanan dapat mengancam integritas data pengguna, baik terjadi secara tidak sengaja maupun dilakukan oleh orang-orang yang tidak bertanggung jawab untuk mencuri informasi, memantau konten, memanfaatkan celah lainnya bahkan pada tingkat paling parah bisa merusak aplikasi dan data di dalamnya.

Buffer overflow, command injection, SQL Injection adalah beberapa istilah untuk serangan yang paling umum pada perangkat lunak. Seorang IT profesional memahami bahwa setiap jenis perangkat teknologi dan sistem operasi pasti memiliki kerentanan terhadap berbagai masalah keamanan, baik dari internal maupun eksternal. Dengan pemikiran tersebut, prinsip secure coding membantu programmer untuk merancang dan mengembangkan software untuk mencegah berbagai celah keamanan.

## Automatic Testing

Testing dalam sebuah aplikasi merupakan hal yang wajib dilakukan untuk mencari kesalahan yang terdapat dari aplikasi yang telah kita buat, sehingga error yang didapatkan bisa segera diperbaiki. Dalam ruang lingkup aplikasi yang masih sederhana, testing secara manual bukanlah perkara yang menyulitkan meski kadang merepotkan karena harus dilakukan berulang kali. Namun, dalam ruang lingkup aplikasi yang sudah cukup kompleks, testing menggunakan cara manual adalah cara yang tidak efektif karena akan menghabiskan banyak waktu dan tenaga.

Automate testing adalah proses pengujian aplikasi secara otomatis dengan menggunakan tools untuk mencari kesalahan pada code program sebelum berlanjut ke tahap rilis atau production. Berikut beberapa alasan kenapa harus menggunakan automated testing:

* Sebab pengujian secara manual terhadap alur kerja, semua field, dan kemungkinan negatif yang dapat ditemukan memberlukan waktu dan biaya
* Sangat merepotkan untuk menguji aplikasi secara manual karena melakukan hal yang sama secara berulang-ulang
* Automated testing tidak memerlukan campur tangan manusia, Kamu bisa menjalankannya secara otomatis lalu tinggalkan
* Automated testing meningkatkan kecepatan proses testing