

**APLIKASI KEHADIRAN DOSEN
MENGGUNAKAN OOP PADA PHP**

KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh. Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga dapat menyelesaikan laporan Membuat Aplikasi Proyek II ini, tanpa pertolongan-Nya mungkin penulis tidak akan sanggup menyelesaikannya dengan baik. Shalawat dan salam semoga terlimpah curahkan kepada Nabi Muhammad SAW beserta sahabat dan keluarga Beliau.

Laporan ini disusun untuk memenuhi kelulusan matakuliah Proyek II pada Program Studi DIV Teknik Informatika. Proses Proyek II ini juga tidak terlepas dari bantuan pihak Pembimbing. Oleh karena itu, pada kata pengantar ini penulis menyampaikan terimakasih kepada :

1. Roni Habibi, S.Kom., M.T. selaku Pembimbing Internal dan Penguji Utama dalam penyusunan laporan Proyek II ini;
2. selaku Penguji Pendamping dalam Sidang Proyek II ini;
3. Rolly Maulana Awangga, S.T., M.T. selaku Koordinator Proyek II Tahun Akademik 2019/2020;
4. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika Tahun Akademik 2019/2020;

5. Dr. Ir. Agus Purnomo, M.T. selaku Direktur Politeknik Pos Indonesia Tahun Akademik 2019/2020.

Penulis telah membuat laporan ini dengan sebaik-baiknya, diharapkan memberikan kritik dan saran dari semua pihak yang bersifat membangun, terimakasih.

Bandung, Desember 2019

Penulis

DAFTAR ISI

ABSTRAK	Error! Bookmark not defined
<i>ABSTRACT</i>	Error! Bookmark not defined
KATA PENGANTAR	
DAFTAR ISI	
DAFTAR GAMBAR	
DAFTAR TABEL	
DAFTAR SIMBOL	Error! Bookmark not defined
BAB I	
PENDAHULUAN	Error! Bookmark not defined
1.1 Latar Belakang	Error! Bookmark not defined
1.2 Identifikasi Masalah	Error! Bookmark not defined
1.3 Tujuan	Error! Bookmark not defined
1.4 Ruang Lingkup	Error! Bookmark not defined
1.5 Sistematika Penulisan	Error! Bookmark not defined
BAB II	Error! Bookmark not defined
LANDASAN TEORI	Error! Bookmark not defined
2.1 Kajian Pustaka	Error! Bookmark not defined
2.2 Analisis	Error! Bookmark not defined
2.3 Basis Data	Error! Bookmark not defined
2.4 Use Case Diagram	Error! Bookmark not defined
2.5 Class Diagram	Error! Bookmark not defined
2.6 Metode Penelitian	Error! Bookmark not defined
2.7 Hasil Penelitian	Error! Bookmark not defined
BAB III	Error! Bookmark not defined
ANALISIS DAN PERANCANGAN	Error! Bookmark not defined
3.1 Analisis	Error! Bookmark not defined
3.1.1 Analisis Sistem yang sedang Berjalan pada Registrasi PMB Poltekpos	Error! Bookmark not defined
3.1.2 Analisis Sistem yang akan Dibangun.....	Error! Bookmark not defined

3.2	Perancangan	Error! Bookmark not defined
3.2.1	Use Case Diagram.....	Error! Bookmark not defined
3.2.2	Class Diagram.....	Error! Bookmark not defined
BAB IV		Error! Bookmark not defined
IMPLEMENTASI.....		Error! Bookmark not defined
4.1	Lingkungan Implementasi.....	Error! Bookmark not defined
4.1.1	Lingkungan Perangkat Keras	Error! Bookmark not defined
4.1.2	Lingkungan Perangkat Lunak	Error! Bookmark not defined
4.2	Pembahasan Hasil Implementasi.....	Error! Bookmark not defined
BAB V		Error! Bookmark not defined
KESIMPULAN DAN SARAN.....		Error! Bookmark not defined
5.1	Kesimpulan.....	Error! Bookmark not defined
5.2	Saran.....	Error! Bookmark not defined
DAFTAR PUSTAKA		Error! Bookmark not defined

DAFTAR GAMBAR

Gambar 3.1 Flowmap sistem yang sedang berjalan.....	Error! Bookmark not defined
Gambar 3.2 Flowmap sistem yang akan dibangun	Error! Bookmark not defined
Gambar 3.3 Perancangan use case diagram	Error! Bookmark not defined
Gambar 3.4 Perancangan class diagram	Error! Bookmark not defined
Gambar 4.1 Tampilan awal website.....	Error! Bookmark not defined
Gambar 4.2 Tampilan menu untuk pendaftaran.....	Error! Bookmark not defined
Gambar 4.3 Tampilan jalur registrasi	Error! Bookmark not defined
Gambar 4.4 Tampilan jalur registrasi	Error! Bookmark not defined
Gambar 4. 5 Tampilan registrasi data calon mahasiswa (2)	Error! Bookmark not defined
Gambar 4.6 Tampilan registrasi data orang tua	Error! Bookmark not defined
Gambar 4.7 Tampilan registrasi data sekolah.....	Error! Bookmark not defined

DAFTAR TABEL

Tabel 3.1 Dokumen penginputan data calon mahasiswa	Error! Bookmark not defined
Tabel 3.2 Dokumen penginputan data orang tua	Error! Bookmark not defined
Tabel 3.3 Dokumen pembuatan laporan calon mahasiswa.....	Error! Bookmark not defined
Tabel 3.4 Dokumen pembuatan laporan orang tua	Error! Bookmark not defined
Tabel 3.5 Spesifikasi minimum perangkat keras server yang digunakan untuk menganalisis	Error! Bookmark not defined
Tabel 3.6 Spesifikasi minimum perangkat keras client ...	Error! Bookmark not defined
Tabel 3.7 Spesifikasi minimum perangkat lunak server yang digunakan untuk menganalisis	Error! Bookmark not defined
Tabel 3.8 Spesifikasi minimum perangkat lunak client...	Error! Bookmark not defined

BAB I

APLIKASI

1.1 Sejarah Aplikasi

Sebelum aplikasi pertama di dunia ditemukan, ada sejarah yang sangat panjang dalam proses ditemukannya aplikasi. Jauh sebelum aplikasi yang ada pada komputer dapat berjalan secanggih saat ini, semua perangkat lunak (*software*) selalu mengandalkan aljabar Boolean. Aljabar ini menggunakan kode binary digit (bit) yang terdiri dari dua angka yaitu 1 (benar/*on*) dan 0 (salah/*off*). Rumitnya (terlalu panjang) penggunaan kode binary digit ini membuat orang-orang pada saat itu mulai membuat kelompok-kelompok bit yang terdiri dari *nible* (4 bit), *byte* (8 bit), *word* (2 byte), dan *double word* (32 bit).

Hadirnya kelompok-kelompok bit ini terbilang cukup membantu dalam berbagai kegiatan yang dilakukan di perangkat lunak komputer. Dengan berbagai kelompok bit yang ada pada saat itu, orang-orang mulai merakit kode-kode tersebut menjadi berbagai struktur instruksi seperti transfer, operasi logika, dan penyimpanan, hingga pada akhirnya terbentuklah kode-kode baru yang kita kenal dengan nama *assembler*. Kode-kode *assembler* ini lah yang nantinya menjadi cikal-bakal dibuatnya berbagai macam aplikasi yang saat ini dapat mempermudah berbagai kegiatan manusia.

1.2 Pengertian Aplikasi

Aplikasi adalah suatu perangkat lunak atau dalam Bahasa Inggris adalah *Software* yang beroperasi pada computer yang berfungsi untuk membantu berbagai kegiatan yang dilakukan oleh seseorang atau *user*. Aplikasi dikelompokkan menjadi 3 kelompok, yaitu *Dekstop Application*, *Website Application*, dan *Mobile Application*.



Berikut ini adalah pengertian aplikasi menurut para ahli:

1. Ali Zaki dan Smitdev Community

Menurut Ali Zaki dan Smitdev Community, Aplikasi merupakan komponen yang bermanfaat sebagai media untuk menjalankan pengolahan data ataupun berbagai kegiatan lainnya seperti pembuatan ataupun pengolahan dokumen dan file.

2. Sri Widiani

Menurut Sri Widiani, Aplikasi merupakan sebuah *software* (perangkat lunak) yang bertugas sebagai *front end* pada sebuah sistem yang dipakai untuk mengelolah berbagai macam data sehingga menjadi sebuah informasi yang bermanfaat untuk penggunanya dan juga sistem yang berkaitan.

3. Harip Santoso

Menurut Harip Santoso, aplikasi merupakan sebuah kelompok file (*class, form, report*) yang ditujukan sebagai pengeksekusi aktivitas tertentu yang saling berkaitan seperti contohnya aplikasi *payroll* dan aplikasi *fixed asset*.

4. Yuhefizar

Menurut Yuhefizar, aplikasi adalah program yang sengaja dibuat dan dikembangkan sebagai pemenuh kebutuhan penggunanya dalam menjalankan suatu pekerjaan tertentu.

5. Hengky W. Pramana

Menurut Hengky W. Pramana, pengertian aplikasi adalah satu unit perangkat lunak yang sengaja dibuat untuk memenuhi kebutuhan akan berbagai aktivitas ataupun pekerjaan, seperti aktivitas perniagaan, periklanan, pelayanan masyarakat, *game*, dan berbagai aktivitas lainnya yang dilakukan oleh manusia.

1.3 Jenis-jenis Aplikasi

Berikut ini adalah beberapa jenis dari aplikasi yang biasa digunakan beberapa orang.

1.3.1 *Multimedia Software*

Multimedia software adalah aplikasi yang dibuat untuk mengelolah suatu objek yang dapat berupa gambar, teks, suara, dan grafik. Banyak sekali aplikasi multimedia yang telah dibuat saat ini, contohnya adalah *Windows Media Player*, *Adobe Photoshop*, dan lain-lain.

1.3.2 *Database Software*

Database software adalah perangkat lunak yang berfungsi untuk mengelolah kumpulan data dalam jumlah besar. Macam-macam Aplikasi *Database Management System (DBMS)*, yaitu *MySQL*, *Oracle*, *Microsoft SQL Server*, *IBM DB2*, dan masih banyak lagi beberapa Aplikasi *DBMS*.

1.3.3 *Presentation Software*

Presentation Software adalah suatu aplikasi yang menampilkan informasi dalam tayangan *slide show*, yang dapat dihias, dan dapat memasukkan gambar grafis. Contoh dari aplikasi *Presentation Software* selain *Microsoft Power Point* adalah *Slidedog*, *Visme*, *Google Slide*, dan lain-lain.

1.3.4 Word Processing Software

Word Processing Software adalah perangkat lunak atau *software* yang berfungsi untuk mengelolah kata atau *word* yang berhubungan dengan dokumen, seperti laporan, karya tulis, surat menyurat, artikel, dan brosur. Contoh aplikasinya adalah *Microsoft Word*, *Notepad*, dan *Google Docs*.

1.3.5 Spreadsheet Software

Spreadsheet Software adalah aplikasi yang menampilkan data dalam bentuk baris dan kolom. Biasanya aplikasi ini digunakan untuk membuat data keuangan, data seseorang, dll. Contoh dari *Spreadsheet Software* adalah *Microsoft Excel*, *StarCalc*, *Siag*, dan sebagainnya.

1.4 Fungsi Aplikasi

Aplikasi memiliki fungsi yang digunakan. Pada bidang pendidikan aplikasi memiliki fungsi sebagai bahan untuk mengajar. Contohnya pada saat siswa presentasi dapat menggunakan aplikasi *Microsoft Power Point*. Tujuannya adalah untuk membuat presentasi tersebut lebih menarik dengan menggunakan aplikasi *Microsoft Power Point*.

Tidak hanya di bidang pendidikan saja, aplikasi juga berfungsi pada bidang kedokteran, bisnis, dan militer. Pada bidang tersebut aplikasi digunakan untuk menginput data manusia, keuangan, dan masih banyak lagi.

BAB II

BAHASA PEMROGRAMAN

2.1 Latar Belakang

Sejak dulu hingga sekarang, tentu kita mengetahui bahwa di dunia komputer terdapat beraneka ragam bahasa pemrograman. Karena begitu banyaknya jenis-jenis bahasa pemrograman, bahasabahasa tersebut juga dikelompokan berdasarkan kriteria tertentu. Ada yang mengelompokkannya menjadi 3 level bahasa yaitu: *high level* (seperti *Pascal* dan *Basic*), *middle level* (seperti Bahasa C), dan *low level* (seperti bahasa *Assembly*). Ada juga yang mengelompokkannya menjadi *procedural/functional programming*, *object oriented programming*, dan sebagainya.

Namun pada dasarnya, bahasa-bahasa pemrograman tersebut memiliki bagian-bagian yang serupa. Yang membedakan hanyalah tata bahasa yang digunakan. Seperti halnya belajar bahasa Indonesia, bahasa Jawa, ataupun bahasa asing. Masing-masingnya memiliki bagian-bagian yang sama, yang membedakan hanyalah istilah/simbol yang digunakan dan dialek bahasa tersebut.

.Bahasa ini memungkinkan seorang *programmer* dapat menentukan mana yang data yang akan diproses oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan langkah-langkah apa yang persis jenis yang akan diambil dalam berbagai situasi.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari :

1. Bahasa mesin, yang memberikan perintah ke komputer dengan menggunakan kode bahasa biner, misalnya 01100101100110.
2. Bahasa tingkat rendah, atau dikenal sebagai bahasa *assembly* yang memberikan perintah ke komputer dengan menggunakan kode pendek (*kode mnemonic*), misalnya [kode_mesin | MOV], SUB, CMP, JMP, JGE, JL , LOOP, dll. Inggris *Intermediate*, yang merupakan bahasa komputer yang menggunakan campuran instruksi dalam kata-kata bahasa manusia lihat contoh di bawah Bahasa Tingkat Tinggi dan instruksi yang bersifat simbolik, misalnya, {,},?, <<, >> , &&, ||, dll.
3. Bahasa tingkat tinggi, yang merupakan bahasa komputer yang menggunakan instruksi berasal dari unsur kata-kata bahasa manusia, misalnya, mulai, akhir, jika, sementara, dan, atau, dll Komputer dapat memahami *compiler* bahasa manusia atau penerjemah program yang dibutuhkan.

2.2 Sejarah Bahasa Pemrograman

Penemuan bahasa pemrograman di mulai bersamaan dengan kemunculan komputer itu sendiri. Bahasa pemrograman dan komputer merupakan dwitunggal yang tidak terpisahkan karena sebuah komputer hanya bisa berfungsi jika ada program.

Sementara itu, sebuah program yang dibuat dengan bahasa pemrograman hanya bisa hidup dengan menggunakan komputer sebagai media. Bahasa pemrograman telah mengalami berbagai proses perkembangan. Kondisi bahasa pemrograman saat ini telah berbeda jauh dengan bahasa pemrograman di tahun 40-an. Dahulu *programmer* memerintah komputer secara fisik menggunakan metode pengkabelan / *wiring* secara manual.

Metode *wiring* kemudian berkembang hingga menjadi bahan pemrograman sederhana. Kualitas dan fitur berkembang dan bertambah dengan pesat. Bahasa pemrograman di awal kemunculannya hanya dapat digunakan untuk satu keperluan saja/*one purpose only*, tetapi kini dapat digunakan untuk bermacam-macam keperluan.

Tokoh pertama yang di anggap memperkenalkan bahasa pemrograman pada komputer adalah Charles Babbage. Charles Babbage adalah seorang ilmuwan yang hidup di Inggris pada abad pertengahan. Beliau adalah orang yang pertama kali mendesain mesin menyerupai komputer. Mesin tersebut diisi beberapa program yang dapat berjalan di dalamnya (identik seperti bahasa *assembly* sekarang ini). Pembuat program tersebut adalah Byron Countess of Lovelace, putri dari Lord Byron.

Komputer modern yang menggunakan tenaga listrik pertama kali diciptakan pada tahun 1940-an. Ketika itu, komputer masih menjadi barang mewah dan hanya dimiliki oleh pihak-pihak yang memerlukan perhitungan data yang sangat banyak, seperti pihak

militer dan universitas. Komputer pada waktu itu berukuran sangat besar sehingga memakan banyak tempat. Meskipun besar, komputer dianggap lambat menjalankan proses komputasi. Di awal penciptaannya hanya sedikit tokoh yang menyarankan untuk membuat program pada komputer.

Dunia pemrograman kemudian mengalami perkembangan yang sangat pesat setelah ditemukan komponen-komponen elektronika. Penggunaan komponen elektromika menyebabkan komputer menjadi komputer berukuran kecil karena ditemukannya teknologi *Integrated Circuit (IC)* dan teknologi pembuatan *chip* lainnya. Selain ukuran yang mengecil, kelebihan komputer setelah revolusi elektronika terdapat pada kemampuan komputasinya yang meningkat tajam. Oleh karena itu kemampuannya meningkat, secara otomatis komputer mampu menangani pemrosesan algoritma dengan lebih cepat. Hal tersebut mendorong dibuatnya berbagai macam bahasa pemrograman tingkat tinggi.

2.3 Pengertian Bahasa Pemrograman

Bahasa adalah suatu sistem untuk berkomunikasi. Bahasa tertulis menggunakan simbol (yaitu huruf) untuk membentuk kata. Dalam ilmu komputer, bahasa manusia disebut bahasa alamiah, dimana komputer tidak bisa memahaminya, sehingga diperlukan suatu bahasa komputer.

Program merupakan sekumpulan instruksi yang merupakan penyelesaian masalah. Program ‘dimasukkan’ ke dalam komputer, komputer mengerjakan instruksi-instruksi di dalam program

tersebut, lalu memberikan hasil atau keluaran yang diinginkan. Agar program dapat dilaksanakan oleh komputer, program tersebut harus ditulis dalam suatu bahasa yang dapat dimengerti komputer. Karena komputer adalah mesin, maka program harus ditulis dalam bahasa yang khusus dibuat untuk berkomunikasi dengan komputer. Bahasa komputer yang digunakan dalam menulis program dinamakan bahasa pemrograman.

Bahasa pemrograman adalah instruksi standar untuk memerintah komputer yang memiliki fungsi tertentu. Bahasa pemrograman ini adalah satu set aturan sintaks dan semantik yang digunakan untuk mendefinisikan program komputer.

Bahasa ini memungkinkan seorang programmer dapat menentukan mana yang data yang akan diproses oleh komputer, bagaimana data ini akan disimpan / diteruskan, dan langkah-langkah apa yang persis jenis yang akan diambil dalam berbagai situasi.

2.4 Fungsi Bahasa Pemrograman

Fungsi dari bahasa pemrograman yang memerintahkan komputer untuk mengolah data sesuai dengan logika yang kita inginkan. *Output* dari bahasa pemrograman dalam bentuk program / aplikasi. Contohnya adalah program yang digunakan oleh kasir di mall atau supermarket, penggunaan lampu lalu lintas di jalan raya, dll

Bahasa pemrograman yang kita tahu ada banyak orang di dunia, sekitar ilmu komputer dan teknologi saat ini. Pengembangannya mengikuti inovasi tinggi dalam dunia teknologi.

Contoh bahasa pemrograman yang kita ketahui, antara lain, adalah untuk membuat aplikasi permainan, anti-virus, web, dan teknologi lainnya.

Bahasa pemrograman komputer yang kita tahu termasuk *Java*, *Visual Basic*, *C ++*, *C*, *Cobol*, *PHP*, *Net*, dan ratusan bahasa lain. Tapi tentu saja, bahasa harus disesuaikan dengan fungsi dan perangkat yang menggunakannya.

2.5 Tingkat Bahasa Pemrograman

Bahasa pemrograman juga memiliki beberapa tingkatan, antara lain sebagai berikut.

2.5.1 Bahasa Tingkat Tinggi

Bahasa pemrograman ini masuk ke level sebagai bahasa dekat dengan bahasa manusia. Sebagai contoh, bahasa Basic, Visual Basic, Pascal, Java dan lainnya.ds dan tidak lagi pada pemula sulit dimengerti.

2.5.2 Bahasa Tingkat Menengah

Disebut tingkat menengah karena dapat masuk ke dalam bahasa tingkat rendah tinggi. Misalnya dalam bahasa program C.

2.5.3 Bahasa Tingkat Rendah

Bahasa pemrograman masuk ke level ini karena bahasa masih jauh dari bahasa manusia. Contoh bahasa *Assembly*.

2.6 Macam Bahasa Pemrograman

Bahasa pemrograman memiliki banyak macam. Pada buku ini hanya akan menjelaskan beberapa macam saja dari Bahasa pemrograman.

2.6.1 *HyperText Markup Language (HTML)*



HyperText Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser Internet. *HTML* saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium (W3C)*.

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file *HTML* dapat dibuka dengan menggunakan browser web seperti *Mozilla Firefox* atau *Microsoft Internet Explorer*.

2.6.2 PHP: Hypertext Preprocessor



PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *FI (Form Interpreted)*, yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari web.

PHP banyak dipakai untuk membuat situs web yang dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. *PHP* biasanya berjalan pada sistem operasi *linux* (*PHP* juga bisa dijalankan dengan hosting windows).

2.6.3 Javascript



Javascript adalah bahasa *scripting* yang handal yang berjalan pada sisi *client*. *JavaScript* merupakan sebuah bahasa *scripting* yang

dikembangkan oleh *Netscape*. Untuk menjalankan *script* yang ditulis dengan *JavaScript* kita membutuhkan *JavaScript-enabled browser* yaitu browser yang mampu menjalankan *JavaScript*.

2.6.4 CSS

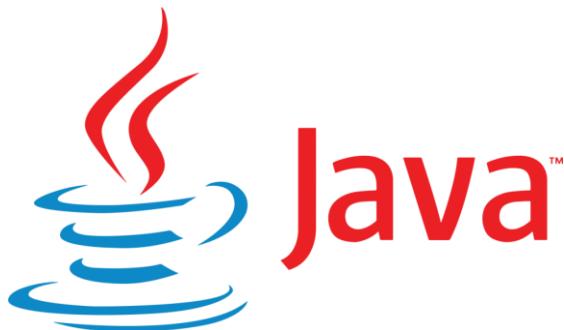


Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan *HTML*. Spesifikasi CSS diatur oleh *World Wide Web Consortium (W3C)*.

2.6.5 Java

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, *Java* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Kita lebih menyukai menyebut *Java* sebagai sebuah teknologi dibanding hanya sebuah bahasa

pemrograman, karena *Java* lebih lengkap dibanding sebuah bahasa pemrograman konvensional.



Bahasa *Java* dapat dikategorikan sebagai sebuah bahasa pemrograman berorientasi objek, pemrograman terdistribusi dan bahasa pemrograman multithreaded .Objek *Java* dispesifikasi dengan membentuk kelas. Untuk masing-masing kelas *Java*, kompiler *Java* memproduksi sebuah file keluaran arsitektur netral yang akan jalan pada berbagai implementasi dari *Java Virtual Machine (JVM)*. Awalnya *Java* sangat digemari oleh komunitas pemrograman internet, karena *Java* mendukung untuk applets, dimana program dengan akses sumber daya terbatas yang jalan dalam sebuah web browser.

Java juga menyediakan dukungan level tinggi untuk *networking* dan objek terdistribusi. *Java* juga dianggap sebagai sebuah bahasa yang aman. Tampilan ini pada khususnya penting menganggap bahwa sebuah program *Java* boleh mengeksekusi silang sebuah jaringan terdistribusi. Sehingga bahasa *Java* saat ini termasuk bahasa pemrograman yang relatif mudah untuk dipelajari karena banyaknya contoh dan konsep yang beredar luas, baik berupa

buku maupun di internet. Tetapi program yang dibuat dalam bahasa *Java* juga relatif lebih banyak membutuhkan waktu saat di eksekusi (lebih lama) dikarenakan untuk menjalankan programnya dibutuhkan *JVM* (perantara antar program dan sistem operasi).

2.6.6 Bahasa C



Akar dari bahasa *C* adalah bahasa *BCPL* yang dikembangkan oleh Martin Richard pada tahun 1967. Bahasa ini memberikan ide kepada Ken Thompson yang kemudian mengembangkan bahasa yang disebut dengan *B* pada tahun 1970. Perkembangan selanjutnya dari bahasa *B* adalah bahasa *C* oleh Dennis Ritchie sekitar tahun 1972-an di *Bell Telephone Laboratories Inc.*(sekarang adalah *AT&T Bell Laboratories*).

Kelebihan Bahasa *C* adalah Bahasa *C* tersedia hampir di semua jenis komputer, kode Bahasa *C* sifatnya adalah *portable*, aplikasi yang ditulis dengan Bahasa *C* untuk suatu komputer tertentu dapat digunakan di komputer lain hanya dengan sedikit modifikasi, dan Bahasa *C* hanya menyediakan sedikit kata-kata kunci.

Bahasa *C* mempunyai struktur yang baik sehingga mudah untuk dipahami. Bahasa *C* mempunyai fungsi-fungsi sebagai program bagiannya. Selain bahasa tingkat tinggi, *C* juga dianggap sebagai bahasa tingkat menengah. Bahasa *C* mampu menggabungkan kemampuan bahasa tingkat tinggi dengan bahasa tingkat tingkat rendah. Karena *C* sifatnya adalah kompiler, maka akan menghasilkan *executable program* yang banyak dibutuhkan oleh program-program komersial.

Kekurangan Bahasa *C* adalah bahasa pemrograman yang memiliki portabilitas tinggi. Program *C* yang kita tulis untuk satu jenis *platform*, bisa kita kompile dan jalankan di *platform* lain dengan tanpa ataupun hanya sedikit perubahan. Ini bisa diwujudkan dengan adanya standarisasi ANSI untuk *C*.

C adalah bahasa pemrograman dengan kata kunci (*keyword*) sedikit. Kata kunci disini adalah merupakan fungsi ataupun kata dasar yang disediakan oleh kompiler suatu bahasa pemrograman. Hal ini membawa pengaruh semakin mudahnya kita menulis program dengan bahasa tingkat rendah. Pengaruh lain dari sedikitnya kata kunci ini adalah proses eksekusi program *C* yang sangat cepat.

2.6.7 Python



Python merupakan bahasa pemrograman tingkat tinggi yang diracik oleh Guido van Rossum. *Python* banyak digunakan untuk membuat berbagai macam program, seperti: program *CLI*, *Program GUI (desktop)*, Aplikasi *Mobile*, *Web*, *IoT*, *Game*, Program untuk *Hacking*, dsb.

Python juga dikenal dengan bahasa pemrograman yang mudah dipelajari, karena struktur sintaknya rapi dan mudah dipahami. *Python* memang sangat sederhana dibandingkan bahasa yang lainnya. Tidak perlu ini dan itu untuk membuat program *Hello World!*. Bahkan *tagline* di *website*-nya menjelaskan, kalau *python* akan membuatmu bekerja lebih cepat dan efektif.

BAB III

SOFTWARE UNTUK MEMBUAT APLIKASI

3.1 Software untuk membuat Aplikasi Website

Untuk membuat sebuah aplikasi dibutuhkan *software* pendukung agar aplikasi yang dibuat mendapatkan hasil yang maksimal. Membuat sebuah Aplikasi *Website* dibutuhkan *text editor software*. Berikut ini adalah beberapa contoh *text editor software*.

3.1.1 Notepad++

Notepad++ adalah sebuah penyunting teks dan penyunting kode sumber yang berjalan di sistem operasi *Windows*. *Notepad++* menggunakan komponen *Scintilla* untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman.



Notepad++ didistribusikan sebagai perangkat lunak bebas. Proyek ini dilayani oleh *Sourceforge.net* dengan telah diunduh lebih dari 27 juta kali dan dua kali memenangkan

penghargaan *SourceForge Community Choice Award for Best Developer Tool*.

Didukung dalam hal ini adalah dimengerti dan diterjemahkan menjadi teks oleh *Notepad++*. Misalnya pada *C++*, fungsi-fungsinya akan di masukan kedalam daftar fungsi dan kata-katanya akan berubah warna sesuai dengan makna kata tersebut *di C++*. Berikut ini adalah beberapa Bahasa pemrograman yang didukung oleh *Notepad++* sejak versi 5.3.9.

1. *ActionScript, Ada, ASP, Assembler, autoIt*
2. *C, C++, C#, Caml, Cmake, COBOL, CSS*
3. *Flash ActionScript, Fortran*
4. *Haskell, HTML*
5. *Java, Javascript, JSP*
6. *KiXtart*
7. *LISP, Lua*
8. *Makefile, Matlab, MS-DOS*
9. *NSIS*
10. *Objective-C*
11. *Pascal, Perl, PHP, Postscript, PowerShell, Properties file, Python*
12. *R, Resource file, Ruby*
13. *Shell, Scheme, Smalltalk, SQL*
14. *TCL, TeX*
15. *Pascal, Perl, PHP, Postscript, PowerShell, Properties file, Python*
16. *XML*

17. YAML

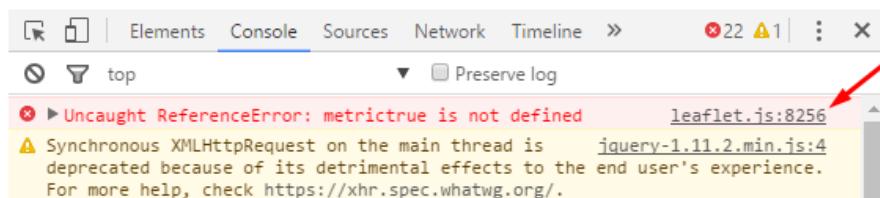
Berikut ini adalah fitur – fitur yang ada pada aplikasi *NotePad++*.

1. Go To Line

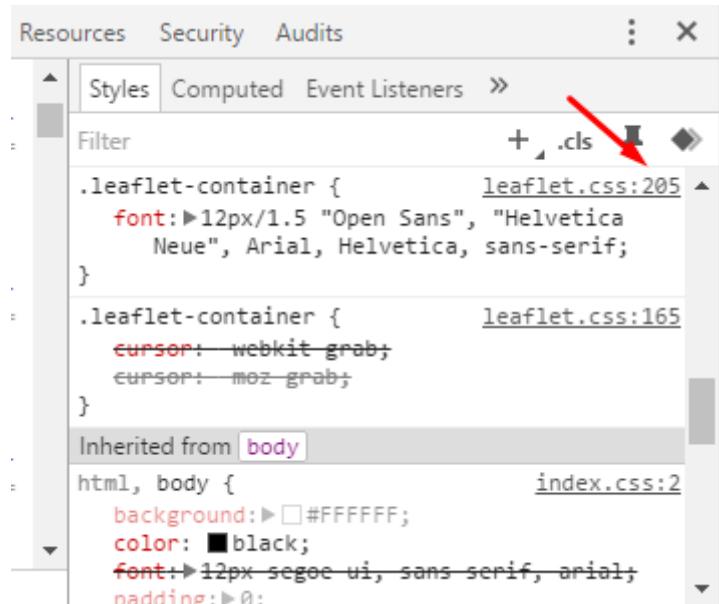
Bagi *programer*, pada kondisi tertentu nomor baris suatu kode akan menjadi begitu penting, terutama ketika melakukan *debugging* program seperti itu *PHP* maupun *JavaScript*, karena, jika terjadi *error*, biasanya program tersebut akan memberi tahu baris berapa yang bermasalah. seperti ini:



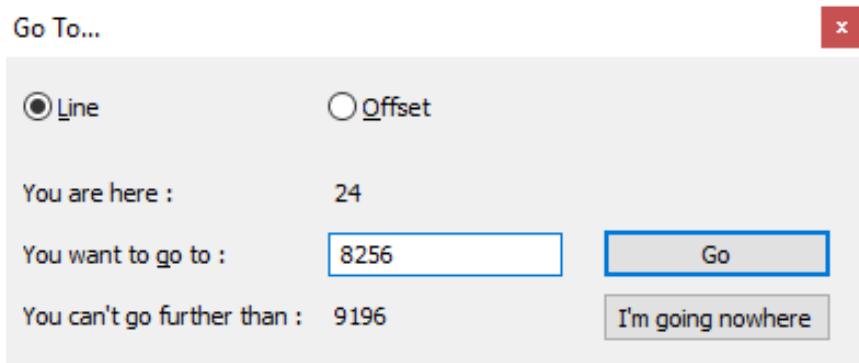
atau ini:



Tidak hanya itu saja, ketika kita meng-*inspect* suatu *element html* dan ingin merubah *style element* tersebut, kita dapat menuju ke nomor baris yang diinginkan.



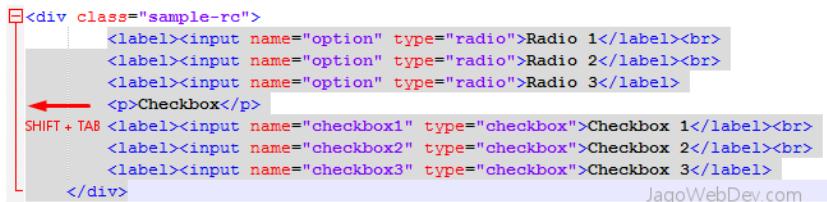
Penting bukan? Untuk itu manfaatkan fitur ini, caranya dengan menekan tombol *Ctrl + G* dan isikan nomor baris yang ingin dituju kemudian klik *OK*.



2. Reverse Indent / Back Tab

Agar kode tampak rapi tentu kita akan sering membuat *indent* -menggeser kode ke kanan, yang biasanya dilakukan menggunakan tombol *Tab*,

bagaimana jika ingin mengembalikannya -menggeser indent ke arah kiri-?, caranya mudah, gunakan shortcut *Shift+Tab*



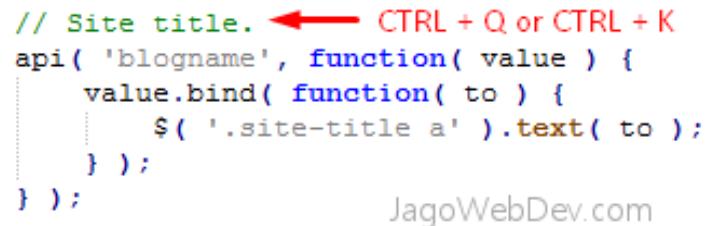
```
<div class="sample-rc">
    <label><input name="option" type="radio">Radio 1</label><br>
    <label><input name="option" type="radio">Radio 2</label><br>
    <label><input name="option" type="radio">Radio 3</label>
    ← <p>checkbox</p>
    SHIFT + TAB <label><input name="checkbox1" type="checkbox">Checkbox 1</label><br>
    <label><input name="checkbox2" type="checkbox">Checkbox 2</label><br>
    <label><input name="checkbox3" type="checkbox">Checkbox 3</label>
</div>
```

JagoWebDev.com

3. Single Line Comment / Uncomment

Untuk menjadikan suatu baris menjadi komentar, biasanya kita tambahkan diawal baris tersebut tanda *double slash* (//), secara umum hal tersebut dilakukan dengan mengarahkan cursor ke awal baris kemudian menekan tombol *slash* dua kali, namun hal ini dapat dipercepat dengan menekan tombol *Ctrl+K* atau *Ctrl+Q*.

Saya sendiri lebih nyaman menggunakan *Ctrl+Q* karena dapat dikerjakan dengan satu tangan kiri, sehingga lebih mudah. Untuk mengembalikannya, tekan kembali tombol *Ctrl+K* atau *Ctrl+Q*.



```
// Site title. ← CTRL + Q or CTRL + K
api( 'blogname', function( value ) {
    value.bind( function( to ) {
        $( '.site-title a' ).text( to );
    } );
});
```

JagoWebDev.com

4. Block Comment / Uncomment

Block comment biasanya dilakukan dengan menambahkan tanda /* di awal block dan ditutup dengan */, pada Dokumen HTML, tanda tersebut berupa <!-- dan -->, dengan notepad++ pekerjaan tersebut dapat dipersingkat, caranya seleksi baris yang akan di comment kemudian tekan tombol **Ctrl+Shift+Q**.

Notepad++ akan menyesuaikan tanda yang digunakan sesuai dengan tipe file yang ada.

```
/* api( 'blogname', function( value ) {
    value.bind( function( to ) {
        $('.site-title a').text( to );
    } );
} );

```

CTRL + SHIFT + Q

```
api( 'blogdescription', function( value ) {
    value.bind( function( to ) {
        $('.site-description').text( to );
    } );
} ); */

```

JagoWebDev.com

Untuk mengembalikannya kita tidak dapat menekan kembali shortcut tersebut seperti yang kita lakukan pada single line comment, melainkan kita harus mendefinisikannya sendiri, caranya buka menu **Run > Modify Shortcut/Delete Command...**

Selanjutnya pada tab Main Menu, cari baris nomor 46. Block Uncomment, selanjutnya isikan shortcut yang diinginkan, saya sendiri memiliki **Ctrl+Shift+W** agar mudah dioperasikan dengan satu tangan.

Shortcut mapper

	Name	Shortcut
42	Toggle Single Line Comment	Ctrl+Q
43	Single Line Comment	Ctrl+K
44	Single Line Uncomment	Ctrl+Shift+K
45	Block Comment	Ctrl+Shift+Q
46	Block Uncomment	Ctrl+Shift+W
47	Function Completion	Ctrl+Spacebar
48	Path Completion	Ctrl+Alt+Spacebar

5. Seleksi Berdasarkan Kolom

Selain menyeleksi dan mengedit baris tertentu, dengan notepad++ kita dapat melakukannya untuk beberapa baris sekaligus, caranya **tekan dan tahan tombol alt** kemudian seleksi baris yang

ingin diedit. Fitur ini bermanfaat salah satunya ketika kita mengedit file csv atau mengubah tabel menjadi file csv.

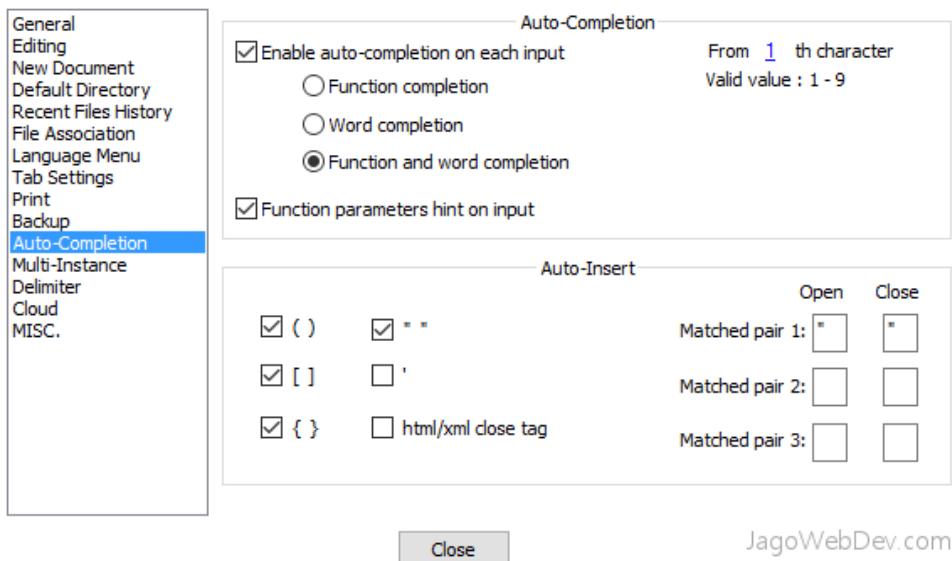
1	01/01/2016;11;10
2	01/01/2016;12;5
3	11/01/2016;11;2
4	10/02/2016;11;22
5	10/02/2016;12;12
6	01/01/2016;21;10
7	01/01/2016;22;5
8	11/01/2016;22;2
9	10/02/2016;21;22
10	10/02/2016;22;12
11	01/01/2016;31;10
12	01/01/2016;33;5
13	11/01/2016;33;2
14	10/02/2016;31;22
15	10/02/2016;33;12

TIPS

Untuk menseleksi teks yang panjang, (1) tekan dan tahan tombol ALT kemudian klik pada teks yang ingin dijadikan awal seleksi, (2) gunakan mouse untuk menscroll jendela ke bawah, (3) tekan tombol SHIFT (tombol ALT tetap ditekan) kemudian klik pada teks yang akan dijadikan akhir seleksi.

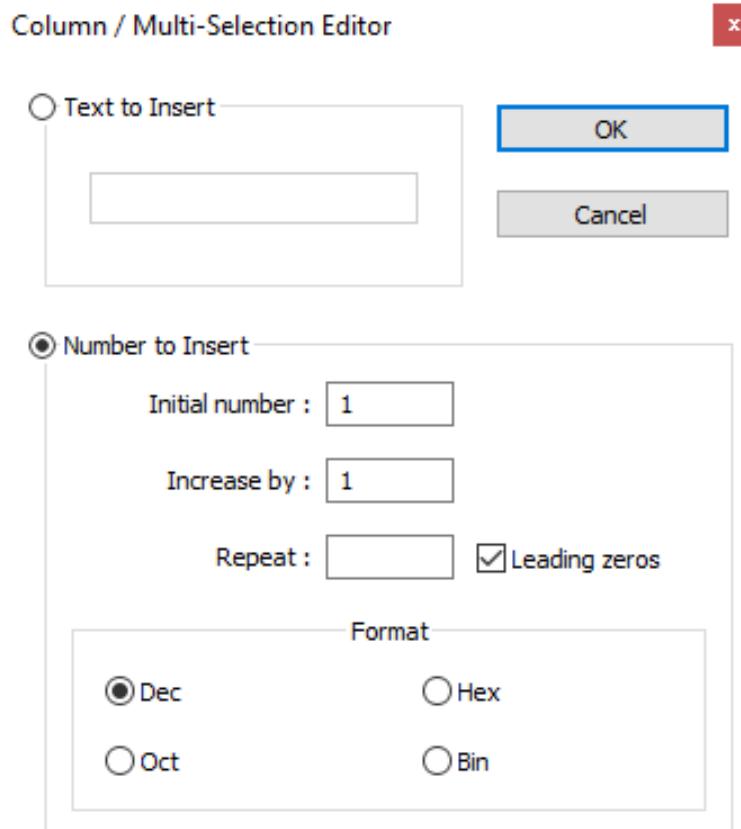
6. Auto Completion

Fitur ini bermanfaat untuk memberikan tanda penutup untuk karakter tertentu, secara default Notepad++ akan memberikan penutup untuk tanda kurung (), kurung siku [], dan kurung kurawal {}, disamping itu sobat juga dapat menambahkan tanda lain misal dua tanda kutip “” atau satu tanda kutip ”, caranya buka menu **Setting > Preferences**, kemudian pilih **Tab Auto Completion**



7. Menambahkan dan Menghapus Nomor Urut

Misalkan kita memiliki kode atau daftar list dan ingin menambahkan nomor urut di depannya, untuk keperluan tersebut, buka jendela “Column / Multi-Selection Editor” melalui menu **Edit > Column Editor...** atau dengan shortcut **Alt + C**, kemudian isikan parameter yang ada.



Menghapus nomor urut

Ketika mengcopy – paste dari media lain baik berupa kode maupun artikel, terkadang terdapat nomor urut di depannya, Notepad++ telah menyediakan fitur untuk menghilangkannya melalui **TextFX > TextFX Tools > Delete Line Numbers or First Word**

```
1   1 // Site title.  
2   2 api( 'blogname', function( value ) {  
3     3   value.bind( function( to ) {  
4       4     $( '.site-title a' ).text( to );  
5     5   } );  
6   6 } );  
7  
8   8 // Site tagline.  
9   9 api( 'blogdescription', function( value ) {  
10 10   value.bind( function( to ) {  
11 11     $( '.site-description' ).text( to );  
12 12   } );  
13 13 } );
```



```
1   // Site title.  
2   api( 'blogname', function( value ) {  
3     value.bind( function( to ) {  
4       $( '.site-title a' ).text( to );  
5     } );  
6   } );  
7  
8   // Site tagline.  
9   api( 'blogdescription', function( value ) {  
10  value.bind( function( to ) {  
11  $( '.site-description' ).text( to );  
12  } );  
13 } );
```

8. Menghapus Baris Kosong

Seperti sebelumnya, ketika mengcopy -paste teks dari media lain, terkadang terdapat baris kosong diantara teks tersebut, disamping itu juga terkadang terdapat tambahan spasi di akhir tiap baris, untuk menghilangkan nya gunakan menu **TextFX > TextFX Edit > Trim Trailing Spaces** dan **TextFX > TextFX Edit > Delete Blank Lines**

```
1 Auto-completion for a subset of the API of some programming language
2
3 Bookmarks
4
5 Syntax highlighting and syntax folding
6
7 Indent guidelines for tab-indented files
8
9 Matching brace highlighting (for curly braces and square brackets)
10 Matching tag highlighting (for HTML and XML tags)
```

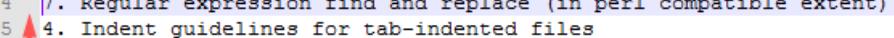
TextFX > TextFX Edit > Trim Trailing Spaces  TextFX > TextFX Edit > Delete Blank Lines

```
1 Auto-completion for a subset of the API of some programming language
2 Bookmarks
3 Syntax highlighting and syntax folding
4 Indent guidelines for tab-indented files
5 Matching brace highlighting (for curly braces and square brackets)
6 Matching tag highlighting (for HTML and XML tags)
```

JagoWebDev.com

9. Mengubah posisi baris

Untuk menggeser baris ke atas, gunakan kombinasi **Ctrl + Shift + Up** sedangkan untuk menggeser ke bawah gunakan kombinasi **Ctrl + Shift + Down**

```
1 1. Auto-completion for a subset of the API of some programming language
2 2. Bookmarks
3 3. Syntax highlighting and syntax folding
4 4. Regular expression find and replace (in perl compatible extent) 
5 5. Indent guidelines for tab-indented files
6 6. Matching brace highlighting (for curly braces and square brackets)
7 7. Matching tag highlighting (for HTML and XML tags)
8 CTRL + SHIFT + UP / CTRL + SHIFT + DOWN
```

JagoWebDev.com

10. Line up lines

Ketika menulis atau meng copy-paste banyak variabel dengan panjang nama variabel yang berbeda beda, kemungkinan besar posisi tanda sama dengan tersebut tidak rata / tidak rapi, untuk merapikannya gunakan menu **TextFX > TextFX Edit > Line up multiple lines by (=)**

```
53 // Sidebar scrolling.  
54 function resize() {  
55     windowHeight = $window.height();  
56     bodyWidth = $body.width();  
57     bodyHeight = $body.height();  
58     sidebarWidth = $sidebar.width();  
59     sidebarHeight = $sidebar.height();  
60  
61     if ( 955 > windowHeight ) {  
62         top = bottom = false;  
63         $sidebarremoveAttr( 'style' );  
64     }  
65 }  
66 }
```

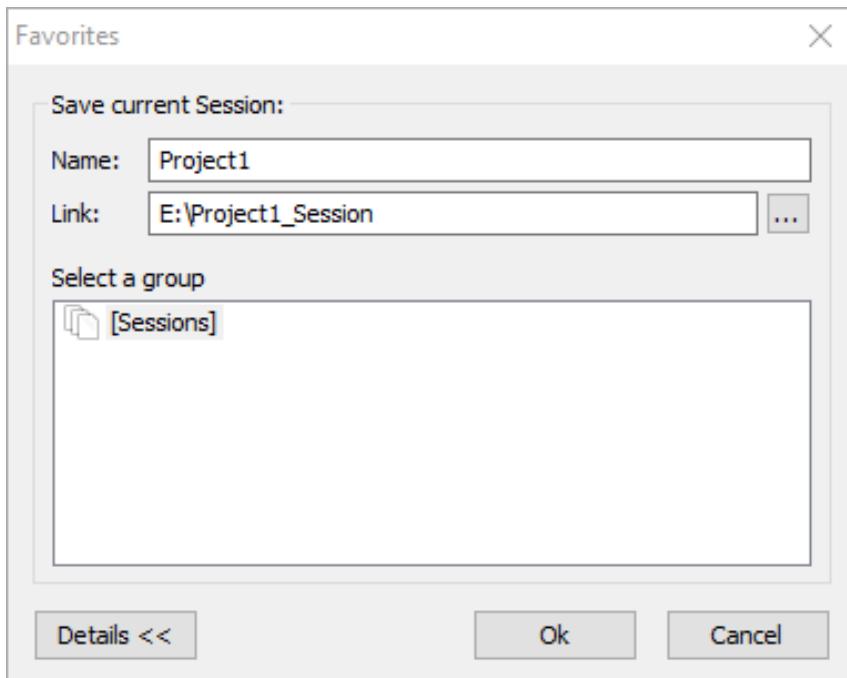


```
53 // Sidebar scrolling.  
54 function resize() {  
55     windowHeight = $window.height();  
56     bodyWidth = $body.width();  
57     bodyHeight = $body.height();  
58     sidebarWidth = $sidebar.width();  
59     sidebarHeight = $sidebar.height();  
60  
61     if ( 955 > windowHeight ) {  
62         top = bottom = false;  
63         $sidebarremoveAttr( 'style' );  
64     }  
65 }  
66 }
```

Jika ingin pilihan opsi yang lebih lengkap, dapat menggunakan [Plugin Code Alignment](#)

11. Membuat Session

Ketika bekerja dengan beberapa file sekaligus misal file html + JavaScript + css, maka akan merepotkan jika kita harus membuka dan menutupnya secara manual, Notepad++ menyediakan fitur bernama Save Session dan Load Session, yang memungkinkan kita untuk menyimpan dan membuka beberapa file sekaligus sesuai yang telah kita tentukan. Fitur tersebut dapat diakses melalui menu File > Load Session atau File > Save Session



12. Membandingkan File – Sinkronisasi Scrolling

Seperti telah kita bahas sebelumnya, notepad++ memiliki plugin compare yang akan membandingkan dua file dan memberikan warna tertentu pada baris yang berbeda, sebenarnya Notepad++ memiliki fitur bawaan yang fungsinya sama persis dengan plugin tersebut.

Bagaimana caranya?

Caranya, klik salah tab tertentu dan pilih Move to Other View, kemudian pada menu View, pilih Synchronize Vertical Scrolling dan Synchronize Horizontal Scrolling.

```
admin.php
```

```
7 */  
8 /**  
9 * In WordPress Administration Screens  
*  
* @since 2.3.2  
*/  
10 if ( ! defined('WP_ADMIN') )  
11     define('WP_ADMIN', true);  
12  
13 if ( ! defined('WP_NETWORK_ADMIN') )  
14     define('WP_NETWORK_ADMIN', false);  
15  
16 if ( ! defined('WP_USER_ADMIN') )  
17     define('WP_USER_ADMIN', false);  
18  
19  
20  
21
```



```
admin.php
```

```
7 */  
8 /**  
9 SINKRONISASI SCROLLING  
10 * In WordPress Administration Screens  
*  
* @since 2.3.2  
*/  
11 if ( ! defined( 'WP_ADMIN' ) ) {  
12     define( 'WP_ADMIN', true );  
13 }  
14  
15 if ( ! defined('WP_NETWORK_ADMIN') )  
16     define('WP_NETWORK_ADMIN', false);  
17  
18  
19  
20
```

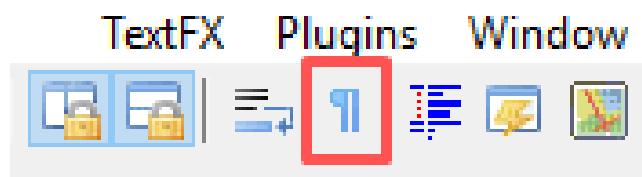
Untuk mengembalikan ke posisi semula klik dan drag tab kembali ke posisi semula.

13. Menampilkan Tab, Spasi, dan Enter Pada Notepad++

Pernahkah sobat bingung membedakan antara spasi dan tab? atau mungkin dengan enter juga? jika ya maka “Show All Characters” adalah solusinya, fitur ini akan menampilkan karakter simbol, diantaranya **TAB** (Tanda Panah), **SPASI** (Tanda Titik), dan **ENTER** (LF-Line Feed) dan (CR-Carriage Return).

CRLF dan **LF** berlaku untuk sistem operasi windows, sedangkan **CR** untuk sistem operasi Linux.

Untuk mengaktifkan fitur ini, buka menu **View > Show Symbol > Show All Character** atau melalui icon pada toolbar.



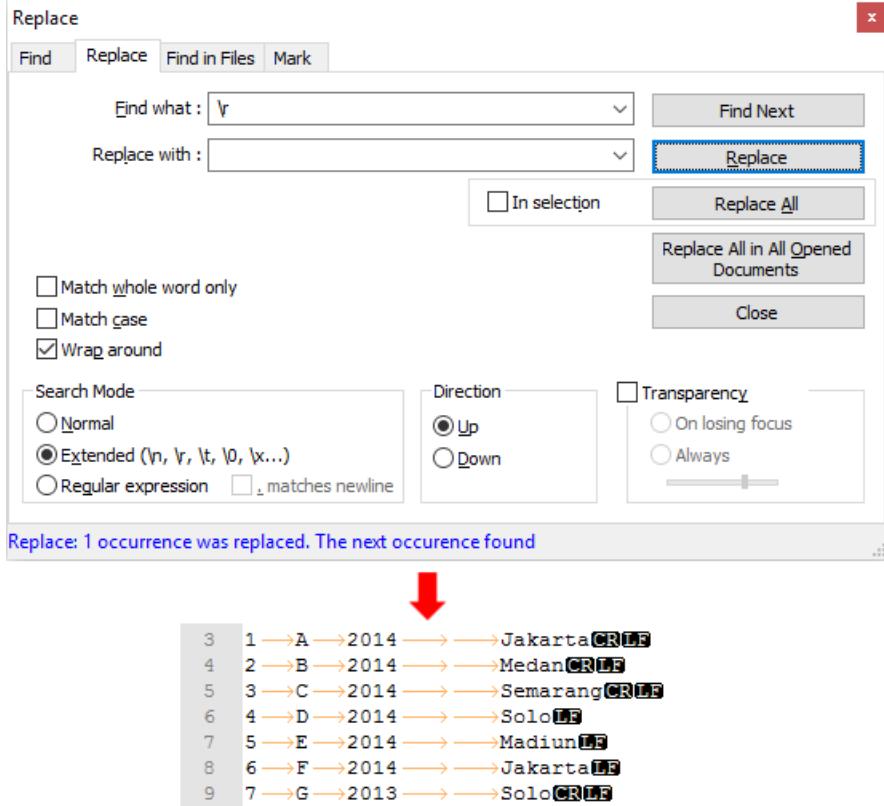
14. Menghapus Tab dan Enter Melalui Extended Search

Selain dapat melakukan Find & Replace karakter umum, notepad++ juga dapat melakukan Find & Replace karakter khusus seperti tab (\t) dan enter (\n), dengan fitur ini kita dapat merapikan berbagai file seperti csv, php, js, html, dll atau mengecilkan nya dengan menghilangkan spasi, tab dan enter.

Untuk menggunakan fitur ini, buka menu “Find & Replace” atau **CTRL+H**, kemudian pada bagian Search Mode, pilih Extended (\n,\r,\t,\0,\x,...)

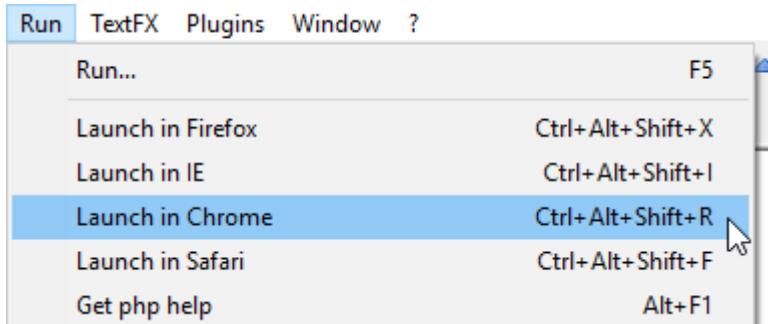
PENTING

Penting untuk diperhatikan bahwa jika ingin menghilangkan spasi, tab, atau enter (CR dan LF), silakan aktifkan fitur **Show All Character** yang telah kita bahas sebelumnya sehingga terlihat mana **SPASI**, **TAB**, **CR** dan **LF** nya, untuk menghilangkan enter, **gunakan \r untuk CR dan \n untuk LF**.



15. Launch In Browser

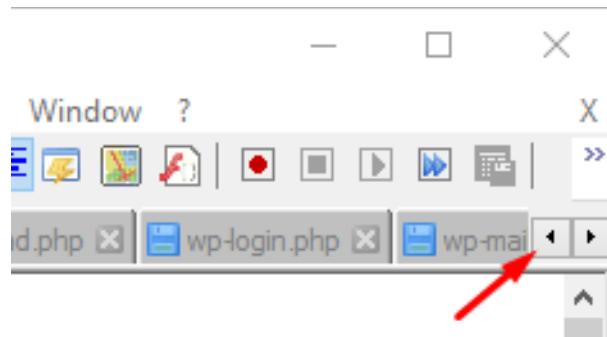
Fitur lain yang bermanfaat yaitu Launch in Browser, dimana secara otomatis kita dapat menjalankan browser untuk menampilkan kode yang sedang kita buat, caranya klik menu Run, selanjutnya pilih browser yang diinginkan,



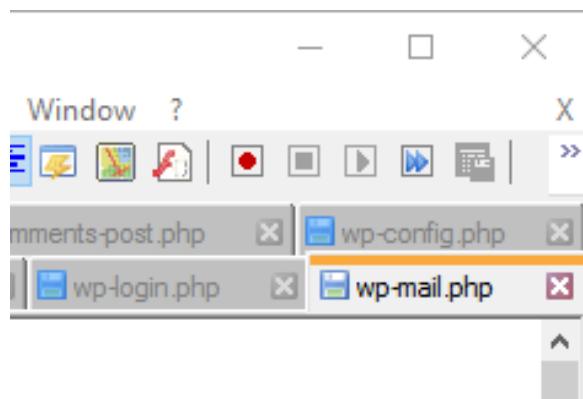
Jika browser tersebut tidak ada dalam daftar sobat dapat mendefinisikannya dengan memilih sub menu Run.., kemudian pilih file exe browser tersebut, selanjutnya pilihan tersebut dapat disimpan pada shortcut dengan mengklik tombol Save..

16. Manajemen Tab

Pada Notepad++, dokumen yang terbuka disusun menjadi tab yang secara default diletakkan di sebelah atas dan di bagian bawah toolbar, jika file yang dibuka banyak hingga melebihi area yang ada, maka akan menyulitkan karena kita harus mencarinya dengan mengklik tanda panah ke kanan dan ke kiri.



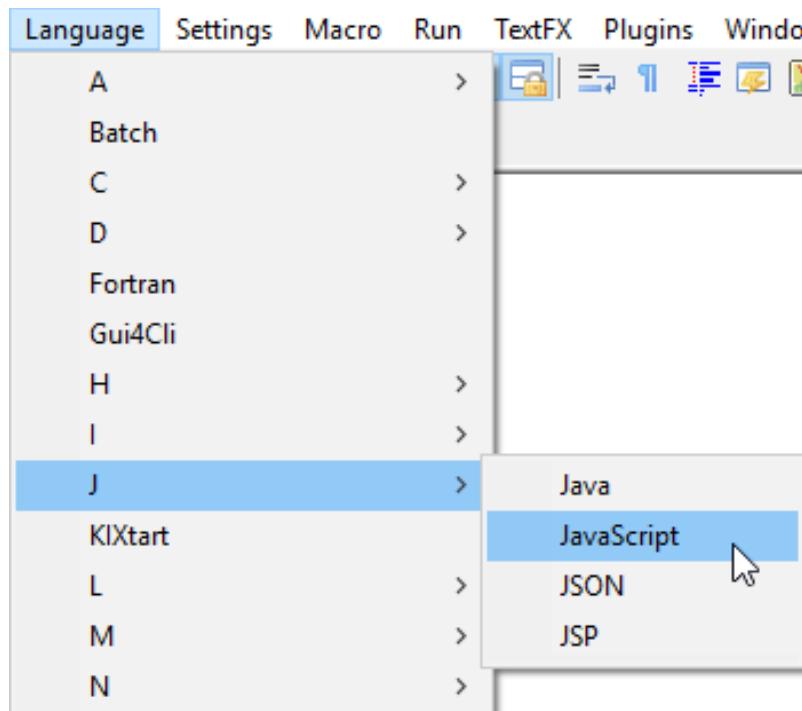
Hal tersebut dapat diatasi dengan membuat Tab tersebut menjadi multi line, caranya, buka menu **Settings > Preferences**, kemudian pilih tab **General**, pada bagian Tab Bar, centang pilihan **Multi-line**. Selain itu, pada bagian ini terdapat berbagai pilihan menarik lainnya seperti memunculkan tombol close pada tab atau menutup tab dengan dobel klik.



Terdapat alternatif lain yaitu menampilkan daftar file yang terbuka pada window tersendiri, hal ini dapat dilakukan dengan bantuan plugin Window Manager.

17. Select Language

Notepad++ akan memberikan warna khusus untuk bagian tertentu dari kode seperti warna khusus untuk class, fungsi, string, variabel, numeric, dll, yang biasanya disebut dengan Syntax Highlight. Notepad++ akan otomatis menyesuaikannya dengan jenis file yang kita buka (php, js, css, html, dll), namun demikian sobat dapat mendefinisikannya sendiri melalui menu Language.



18. Plugins

Banyak sekali plugin yang dapat kita gunakan untuk mempermudah dan mempercepat pekerjaan kita, seperti Code Alignment, NppFTP, Finger Text, NppExport, dan JSTools,

3.1.2 Microsoft Visual Studio 2010



Visual Studio 2010 pada dasarnya adalah sebuah Bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu.

Visual Studio 2010 (yang sering juga disebut dengan *VB .Net 2010*) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasiskan *windows*. Beberapa kemampuan atau manfaat dari *Visual Studio 2010* diantaranya seperti :

1. Untuk membuat program aplikasi berbasiskan *windows*.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol *ActiveX*, *file Help*, aplikasi Internet dan sebagainya.

3. Menguji program (*debugging*) dan menghasilkan program berakhiran *EXE* yang bersifat *executable* atau dapat langsung dijalankan.

Visual Studio 2010 adalah bahasa yang cukup mudah untuk dipelajari. Bagi programer pemula yang baru ingin belajar program, lingkungan *Visual Studio* dapat membantu membuat program dalam sekejap mata. Sedang bagi *programmer* tingkat lanjut, kemampuan yang besar dapat digunakan untuk membuat program-program yang kompleks, misalnya lingkungan *networking* atau *client server*.

Bahasa *Visual Studio* cukup sederhana dan menggunakan kata-kata bahasa Inggris yang umum digunakan. Kita tidak perlu lagi menghafalkan sintaks-sintaks maupun format-format bahasa yang bermacam-macam, di dalam *Visual Basic* semuanya sudah disediakan dalam pilihan-pilihan yang tinggal diambil sesuai dengan kebutuhan. Selain itu, sarana pengembangannya yang bersifat visual memudahkan kita untuk mengembangkan aplikasi berbasiskan *windows*, bersifat *mouse-driven* (digerakkan dengan *mouse*) dan berdaya guna tinggi.

Sejarah Singkat Visual Studio 2010

Berikut ini beberapa point penting dalam sejarah perkembangannya :

1. Pertama kali di *release* dengan nama *Visual Basic* yang dikeluarkan pada tahun 1991, yaitu *Visual Basic* yang masih berbasis *DOS* dan untuk *Windows*

2. *Visual Basic 3.0* dirilis pada tahun 1993
3. *Visual Basic 4.0* dirilis pada tahun 1994 dengan dukungan untuk aplikasi 32 bit
4. *Visual basic 6.0* dirilis pada akhir tahun 1998
5. *Visual basic* untuk selanjutnya yaitu versi .Net yang dirilis awal 2002
6. Pada tahun 2003 *Visual Studio 2003* dirilis untuk memperbaiki kinerja dari *Visual Studio 2002* dengan meluncurkan *.NET Framework* versi 1.1.
7. Pada tahun 2005 Microsoft mengeluarkan *Visual Basic Versi 8.0* atau biasa di sebut *Microsoft Visual Studio 2005*.
8. Pada tahun 2008 Microsoft juga mengeluarkan versi 9.0 atau biasa di sebut *Microsoft Visual Studio 2008*.
9. Hingga pada tahun 2010 *Visual Studio* sudah pada versi 10.0 atau biasa di sebut *Microsoft Visual Studio 2010*, dimana didalamnya sudah ditambahkan bahasa pemprograman yang baru yaitu *F#* yang merupakan penyempurnaan dari versi-versi sebelumnya.

Keistimewaan *Visual Studio 2010*

Beberapa keistimewaan *Visual Studio 2010* ini diantaranya seperti :

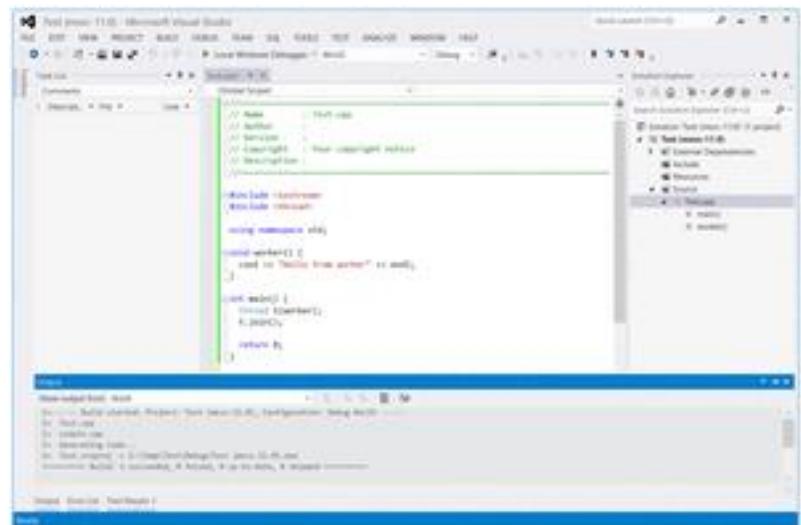
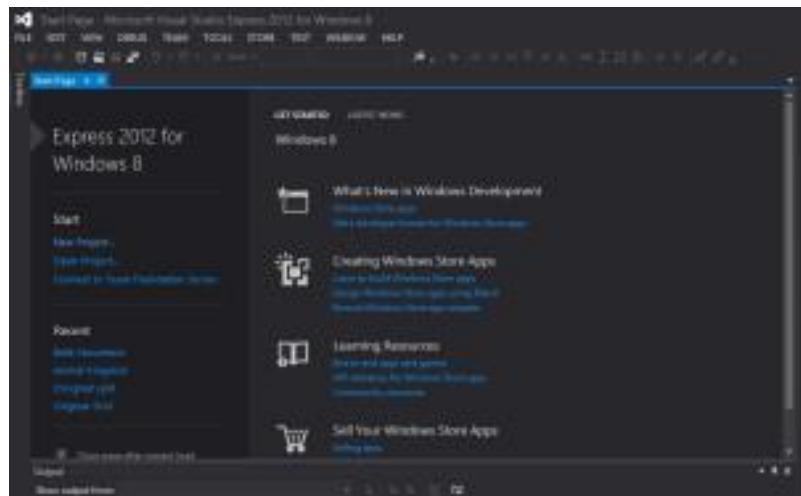
1. Menggunakan *platform* pembuatan program yang dinamakan *developer studio*, yang memiliki tampilan dan

sarana yang sama dengan *Visual C++* dan *Visual J++*. Dengan begitu Anda dapat bermigrasi atau belajar bahasa pemrograman lainnya dengan mudah dan cepat.

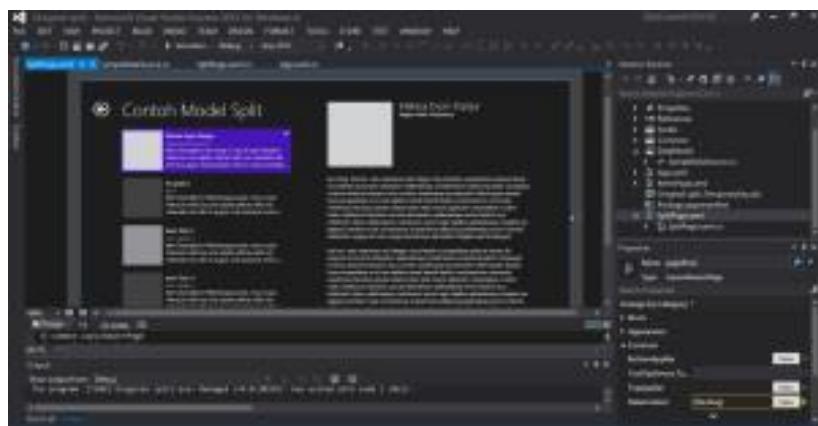
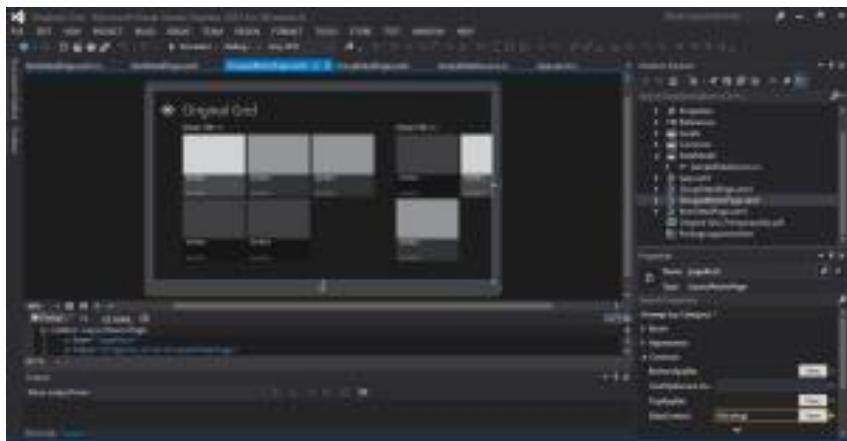
2. Memiliki *compiler* handal yang dapat menghasilkan *file executable* yang lebih cepat dan lebih efisien dari yang sebelumnya.
3. Memiliki beberapa tambahan *wizard* yang baru. *Wizard* adalah sarana yang mempermudah di dalam pembuatan aplikasi dengan mengotomatisasi tugas-tugas tertentu.
4. *Visual Studio 2010* mempunyai beberapa fitur untuk pengembangan berbagai macam aplikasi yang diantaranya; *Windows Development*, *Web Development*, *Office Development*, *Sharepoint Development*, *Cloud Development* (*Windows Azure*), *Silverlight Tooling*, *Multi-Core Development*, *Customizable IDE*.

Nah Fitur-Fitur baru yang saya rasakan diantaranya :

1. Untuk User interface yang jelas sudah baik sekali , tampilanya simple tapi mengandung banyak fitur bagus, ada 2 tema yang dapat kita pilih yaitu Dark dan Light.

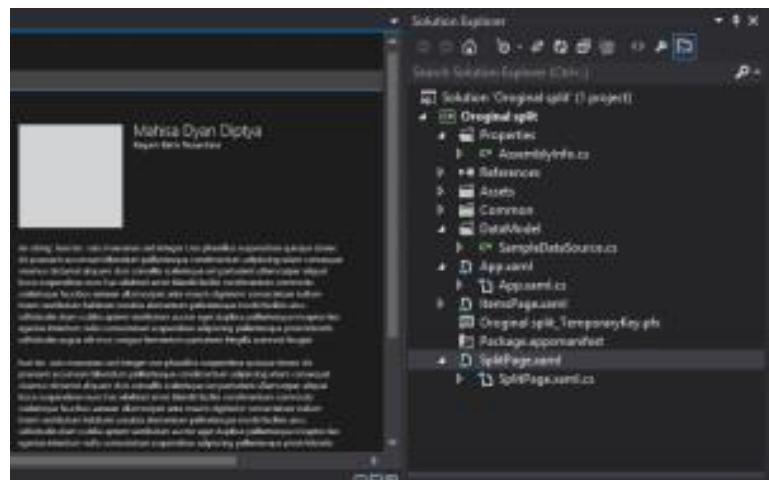


2. Kemampuanya membangun aplikasi Metro Style , dengan menggunakan c++ , c#, visual basic, HTML, JavaScript, XAML Kita sudah dapat membuat aplikasi Metro-Style untuk windows 8, tetapi kalau gak mau ribet Visual studio juga menyediakan 2 template Metro Style nya, Diantaranya model Grid dan Split



3. LightSwitch, dengan LightSwitch kita akan dapat mempublish aplikasi yang sudah kita buat ke Azure dengan cara yang lebih gampang dan simple, selain itu membuat peforma menjadi lebih baik misalnya peningkatan proses loading pada saat membuka *project*, membuat screens dan tables, juga ketika kita Building Projects. oh iya sekedar info aja, pada versi sebelumnya LightSwitch ini terpisah dengan Visual Studio tetapi sekarang sudah termasuk/include di dalam visual Studio 2012.

4.Solution Explorer yang berbeda, Solution Explorer itu merupakan semacam jendela yang mengandung semua file aplikasi Visual Studio dan dimana setiap proyek dapat mengandung lebih dari satu file. Jadi, kita dapat melakukan penelusuran ke methods dan properties pada proyek kita dan juga memungkinkan kita untuk melakukan pencarian file dan yang paling saya suka karena dapat melakukan preview pada objek-objek dan item-item eksternal.



5. Oh iya yang terakhir yang terbaru juga, dukungan dari .NET Framework 4.5

Nah untuk versi dari Visual Studio 2012, terdapat pada gambar di bawah ini :



3.1.3 Sublime Text



Sublime text adalah salah satu *text editor* yang kini cukup banyak peminatnya, dan penggunaan *software* ini bisa digunakan

juga oleh berbagai macam *platform OS (Operating System)*. *Sublime text* juga banyak sekali mendukung banyak bahasa pemrograman dan bahasa markup.

Sublime text diresmikan pada tahun 2008 dan sudah memiliki beberapa versi dan kini sudah sampai ke versi *Sublime text 3* juga. Penggunaan *sublime text* banyak sekali membantu pekerjaan seorang *web developer* atau *programer* lainnya seperti memudahkan *programer* dalam membedakan syntak dengan warna yang dimilikinya dan banyak juga kelebihan lainnya yang dapat dilakukan oleh *text editor* ini.

Cara menggunakan Sublime Text

Cara dasar yang dapat kalian lakukan pertama sekali dalam menggunakan aplikasi ini yaitu silahkan buka Aplikasi *sublime text* dengan cara klik 2x atau juga bisa klik kanan dan pilih *open*. Selain itu kalian juga bisa memasukan folder *project* kalian kedalam *sublime text* agar nantinya dengan mudah untuk menemukan *file* lainnya yang terdapat pada folder *project* tanpa harus menggunakan *open file*, cara sangat mudah yaitu saat *sublime text* kalian sudah terbuka silahkan tarik folder *project* kalian kedalam *sublime text* maka akan otomatis *sublime text* akan membaca folder tersebut dan meletakannya pada *side* kiri, atau juga bisa kalian menggunakan klik menu file pada bar dan pilih *open folder*.

```
halaman1.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Belajar HTML</title>
5 </head>
6 <body>
7   <p>Hello World</p>
8   <p>Selamat beraktivitas</p>
9 </body>
10 </html>
```

Kelebihan *Sublime Text*

Banyak kelebihan yang dimiliki oleh text editor ini, dan berikut saya akan memaparkan beberapa kelebihan dalam penggunaan sublime text diantara:

4. *Software* yang ringan

Kelebihan pertama yang dimiliki oleh *text editor* satu ini diantaranya yaitu *software* yang ringan dan tidak banyak memakan *RAM* pada komputer kalian.

5. *Support Platform*

Yang kedua yaitu *support* dibanyak *platform OS* seperti *Windows* dan lainnya.

6. Mini Map

Selain itu sublime text juga memiliki *mini map* disisi kanan atasnya untuk mempermudahkan kita dalam menemukan kode.

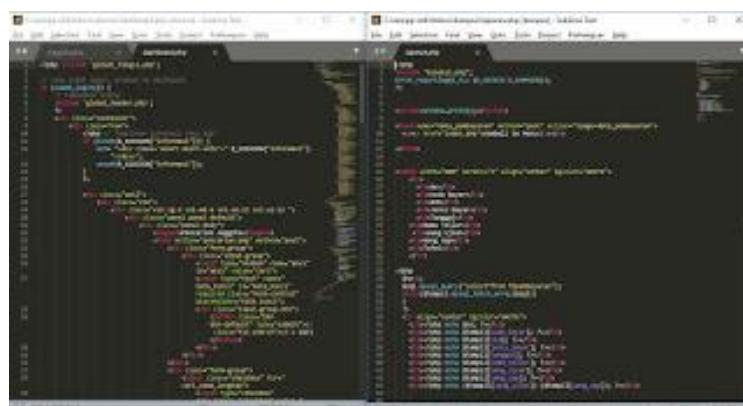
7. Mengganti Background.

Kelebihan lainnya yaitu kalian juga bisa menganti *background* atau *scheme text editor* satu ini jika kalian bosan dengan warna bawaannya yaitu hitam dengan cara pilih menu *preferences* pada menu bar dan pilih *color scheme*.

Fitur dimiliki Sublime Text

Selain kelebihan diatas sebenar fitur-fitur yang dimiliki *sublime text* ini juga merupakan kelebihan dari *text editor* ini, dan *sublime text* memiliki banyak sekali fitur-fitur yang menarik dan sangat mempermudah bagi seorang yang menuliskan *script* pada *text editor* ini. diantaranya yaitu:

1. Split Editing



Penggunaan *split mode* bisa dilakukan bagi kalian ingin membuka beberapa *file* atau beberapa *project* sekaligus tanpa harus menutup *project* lainnya.

2. Multi Selection



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5  </head>
6  <body>
7
8  </body>
9  </html>
```

Multi Selection yaitu fitur yang digunakan untuk dapat menyeleksi beberapa baris *text* atau kode pada *script*, cara menggunakannya hanya dengan (*CTRL+Klik*) dengan fungsi ini sangat memudahkan *programer* dalam mengedit dalam banyak baris.

3.2 Software untuk Membuat Aplikasi Mobile

Tidak hanya dalam pembuatan aplikasi *website* yang membutuhkan *software* pendukung, untuk membuat Aplikasi *Mobile* juga membutuhkan *software* pendukung. Berikut ini adalah beberapa contoh *software* untuk membuat Aplikasi *Mobile*.

3.2.1 Android Studio



Android Studio merupakan Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Android berubah menjadi platform yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembangan utama dibelakangnya, yaitu Google. Googlelah yang mengakuisisi Android dan kemudian membuatkan sebuah platform.

Platform android terdiri dari Sistem Operasi berbasis Linux, sebuah GUI (Graphic User Interface), sebuah web browser dan Aplikasi Studio End-User yang dapat di download dan juga para pengembang bisa dengan leluasa berkarya serta menciptakan aplikasi yang terbaik dan terbuka untuk digunakan oleh berbagai macam perangkat.

Sejarah Android

Pada awalnya mulanya, Android Inc merupakan sebuah perusahaan software kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Didirikan oleh beberapa senior di beberapa perusahaan yang berbasis IT & Communication, Andy Rubin, Rich Miner, Nick Sears dan Chris White.

Menurut Rubin, Android Inc Didirikan untuk mewujudkan mobile device yang lebih peka terhadap lokasi dan preferensi pemilik. Dengan kata lain, Android Inc ingin mewujudkan mobile device yang lebih mengerti pemiliknya.

Konsep yang dimiliki Android Inc ternyata menggugah minat Google untuk memilikinya. Pada bulan Agustus 2005, Akhirnya Android Inc diakuisisi oleh Google Inc. seluruh sahamnya dibeli oleh Google. Banyak yang memperkirakan nilai pembelian Android Inch Oleh Google adalah sebesar USD 50 juta. saat itu banyak yang berspekulasi bahwa akuisisi ini adalah langkah awal yang dilakukan Google untuk masuk kepasar mobile phone.



Andy Rubin, Rich Miner, Nick Sears dan Chris White tetap di Android Inc yang dibeli Google, sehingga akhirnya mereka semua menjadi bagian dari raksasa Google dan sejarah Android. Saat itulah mereka mulai menggunakan platform Linux untuk membuat sistem operasi bagi mobile phone.

Di atas alat editor dan pengembang hebat IntelliJ, Android Studio menawarkan lebih banyak fitur yang meningkatkan produktivitas Anda saat membangun aplikasi Android, seperti:

Sistem build berbasis Gradle yang fleksibel

- 1).** *Emulator cepat dan kaya fitur*
- 2).** *Lingkungan terpadu yang dapat Anda kembangkan untuk semua perangkat*
- 3).** *Instan Berjalan untuk mendorong perubahan pada aplikasi yang sedang berjalan tanpa membangun APK Baru*
- 4).** *Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan kode contoh import*
- 5).** *Alat pengujian ekstensif dan kerangka kerja*
- 6).** *Alat lint untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya*
- 7).** *Dukungan C++ dan NDK*
- 8).** *Dukungan terintegrasi untuk Google Cloud Platform, sehingga mudah mengintegrasikan Google Cloud Messaging dan App Engine*
- 9).** *Memberikan pengenalan fitur dasar Android Studio. Untuk ringkasan perubahan terbaru, lihat Android Studio Release*
- 10).** *Struktur Proyek*

Setiap proyek di Android Studio berisi satu atau lebih modul dengan file kode sumber dan file sumber daya. Jenis modul meliputi:

- a). Modul aplikasi Android
 - b). Modul perpustakaan
 - c). Modul Google App Engine
- Secara default,

Android Studio menampilkan file proyek Anda dalam tampilan proyek Android. Tampilan ini disusun oleh modul untuk menyediakan akses cepat ke file sumber utama proyek Anda. Semua file build terlihat di tingkat atas di bawah Gradle Scripts dan setiap modul aplikasi berisi folder berikut:

Bermanifestasi: Berisi file `AndroidManifest.xml`.

Java: Berisi file kode sumber Java, termasuk kode uji JUnit.

Res: Berisi semua sumber daya non-kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur proyek Android pada disk berbeda dari representasi yang rata ini. Untuk melihat struktur file proyek yang sebenarnya, pilih Project from the Project dropdown.

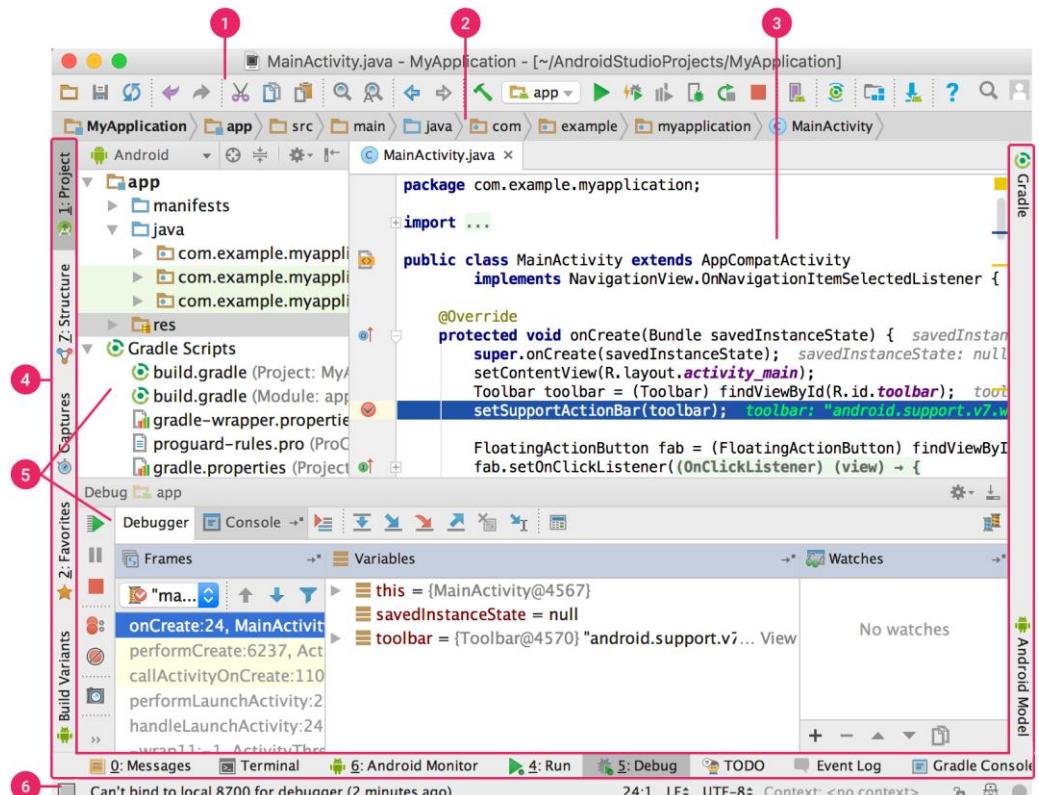
Anda juga dapat menyesuaikan tampilan file proyek agar fokus pada aspek spesifik pengembangan aplikasi Anda. Misalnya, memilih tampilan Masalah dari proyek Anda yang menampilkan tautan ke file

sumber yang berisi kesalahan pengkodean dan sintaks yang dikenali, seperti tag penutup elemen XML yang hilang dalam file tata letak.

File proyek dalam tampilan Masalah, menampilkan file tata letak dengan masalah.

Antarmuka Pengguna

Jendela utama Android Studio terdiri dari beberapa area logis yang diidentifikasi dalam gambar 3.



Gambar 3. Jendela utama Android Studio.

1. **Toolbar** memungkinkan Anda melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan fitur Android.
2. **Menu navigasi** membantu Anda menjelajah project dan membuka file untuk diedit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela **Project**.
3. **Jendela editor** adalah tempat Anda membuat dan memodifikasi kode. Tergantung jenis file yang ada, editor ini dapat berubah. Misalnya, saat menampilkan file tata letak, editor akan menampilkan Layout Editor.
4. **Panel jendela fitur** berada di sisi luar jendela IDE dan berisi tombol-tombol yang memungkinkan Anda memperluas atau menciatukan setiap jendela fitur.
5. **Jendela fitur** memberi Anda akses ke tugas tertentu seperti pengelolaan project, penelusuran, kontrol versi, dan banyak lagi. Anda dapat memperluas dan menciatukan jendela ini.
6. **Status bar** menampilkan status project Anda dan IDE itu sendiri, serta semua peringatan atau pesan.

Anda dapat mengatur jendela utama untuk memperluas ruang layar dengan menyembunyikan atau memindahkan toolbar dan jendela fitur. Anda juga dapat menggunakan pintasan keyboard untuk mengakses sebagian besar fitur IDE.

Anda dapat menelusuri kode sumber, database, tindakan, elemen antarmuka pengguna, dan sebagainya kapan saja, dengan menekan

tombol Shift dua kali, atau mengklik kaca pembesar di sudut kanan atas jendela Android Studio. Tips ini sangat berguna jika, misalnya, Anda mencoba menemukan tindakan IDE tertentu yang Anda lupa cara memicunya.

Jendela fitur

Sebagai ganti menggunakan perspektif preset, Android Studio mengikuti konteks Anda dan otomatis menampilkan jendela fitur yang relevan saat Anda bekerja. Secara default, jendela fitur yang paling umum digunakan disematkan ke panel jendela fitur di tepi jendela aplikasi.

- Untuk memperluas atau mencuitkan jendela fitur, klik nama fitur di panel jendela fitur. Anda juga dapat menarik, menyematkan, melepaskan sematan, memasang, dan melepas jendela fitur.
- Untuk kembali ke tata letak jendela fitur default saat ini, klik **Window > Restore Default Layout** atau sesuaikan tata letak default dengan mengklik Window > Store Current Layout as Default.
- Untuk menampilkan atau menyembunyikan seluruh panel jendela alat, klik ikon jendela  di pojok kiri bawah jendela Android Studio.
- Untuk menemukan jendela alat tertentu, arahkan kursor ke atas ikon jendela dan pilih jendela alat tersebut dari menu.

Anda juga bisa menggunakan pintasan keyboard untuk membuka jendela alat. Tabel 1 mencantumkan pintasan jendela paling umum.

Tabel 1. Pintasan keyboard ke beberapa jendela alat yang berguna.

Jendela alat	Windows dan Linux
Project	Alt+1
Kontrol Versi	Alt+9
Run	Shift+F10
Debug	Shift+F9
Logcat	Alt+6
Kembali ke Editor	Esc
Menyembunyikan Semua Jendela Alat	Control+Shift+F12

Jika Anda ingin menyembunyikan semua toolbar, jendela alat, dan tab editor, klik **View > Enter Distraction Free Mode**. Langkah ini akan mengaktifkan *Distraction Free Mode*. Untuk keluar dari Distraction Free Mode, klik **View > Exit Distraction Free Mode**.

Anda dapat menggunakan *Speed Search* untuk menelusuri dan memfilter di dalam sebagian besar jendela fitur pada Android

Studio. Untuk menggunakan Speed Search, pilih jendela alat, lalu ketik kueri penelusuran Anda.

Untuk tips lainnya, lihat [Pintasan keyboard](#).

Pelengkapan kode

Android Studio memiliki tiga jenis pelengkapan kode, yang dapat Anda akses menggunakan pintasan keyboard.

Tabel 2. Pintasan keyboard untuk pelengkapan kode.

Jenis	Deskripsi	Windows	Mac OS X
Basic Completion	Menampilkan saran dasar untuk variabel, jenis, metode, ekspresi, dan sebagainya. Jika memanggil Basic Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk anggota pribadi dan anggota statis yang tidak diimpor.	Control+Space	Control+Space
Smart Completion	Menampilkan opsi yang relevan berdasarkan konteks. Smart Completion mengetahui jenis dan alur data yang diharapkan. Jika Anda memanggil Smart Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk chain.	Control+Shift+Space	Control+Shift+Space
Statement Completion	Melengkapi pernyataan saat ini secara otomatis, seperti menambahkan tanda kurung, tanda kurung siku, tanda	Control+Shift+Enter	Control+Shift+Enter

Completion kurung kurawal, pemformatan, dll. yang tidak lengkap.

Anda juga dapat melakukan perbaikan cepat dan menampilkan tindakan maksud dengan menekan **Alt+Enter**.

Menemukan kode contoh

Code Sample Browser di Android Studio membantu Anda menemukan contoh kode Android berkualitas tinggi yang disediakan Google berdasarkan simbol yang saat ini ditandai dalam project Anda. Untuk informasi selengkapnya, lihat [Menemukan kode contoh](#).

Navigasi

Berikut ini beberapa tips untuk membantu Anda menjelajah di dalam Android Studio.

- Beralih antar file yang baru saja diakses menggunakan tindakan *Recent Files*. Tekan **Control+E** (**Command+E** pada Mac) untuk memunculkan tindakan Recent Files. Secara default, file yang terakhir diakses akan dipilih. Anda juga dapat mengakses jendela fitur mana saja melalui kolom kiri dalam tindakan ini.
- Lihat struktur file saat ini menggunakan tindakan *File Structure*. Munculkan tindakan File Structure dengan menekan **Control+F12** (**Command+F12** pada Mac). Dengan

tindakan ini, Anda dapat membuka bagian mana pun dari file saat ini dengan cepat.

- Telusuri dan buka class tertentu dalam project menggunakan tindakan *Navigate to Class*. Munculkan tindakan ini dengan menekan **Control+N** (**Command+O** pada Mac). *Navigate to Class* mendukung ekspresi canggih, termasuk camel humps, jalur, baris navigasi ke, pencocokan nama tengah, dan banyak lagi. Jika Anda memanggilnya dua kali berturut-turut, hasil dari class project akan ditampilkan.
- Buka file atau folder menggunakan tindakan *Navigate to File*. Munculkan tindakan *Navigate to File* dengan menekan **Control+Shift+N** (**Command+Shift+O** pada Mac). Untuk menelusuri folder dan bukan file, tambahkan / (garis miring) di akhir ekspresi Anda.
- Buka metode atau kolom menurut nama menggunakan tindakan *Navigate to Symbol*. Munculkan tindakan *Navigate to Symbol* dengan menekan **Control+Shift+Alt+N** (**Command+Option+O** pada Mac).
- Temukan semua bagian kode yang merujuk ke class, metode, kolom, parameter, atau pernyataan di posisi kursor saat ini dengan menekan **Alt+F7** (**Option+F7** pada Mac).

Gaya dan pemformatan

Saat Anda mengedit, Android Studio otomatis menerapkan pemformatan dan gaya seperti yang ditentukan dalam setelan gaya kode Anda. Anda dapat menyesuaikan setelan gaya kode menurut bahasa pemrograman, termasuk menentukan konvensi untuk tab dan indentasi, spasi, penggabungan, tanda kurung kurawal, dan baris kosong. Untuk menyesuaikan setelan gaya kode Anda, klik **File > Settings > Editor > Code Style (Preferences > Editor > Code Style pada Mac.)**

Meskipun IDE otomatis menerapkan pemformatan selagi Anda bekerja, Anda juga bisa memanggil tindakan *Reformat Code* secara eksplisit dengan menekan **Control+Alt+L** (**Opt+Command+L** pada Mac), atau otomatis mengindentasi semua baris dengan menekan **Control+Alt+I** (**Control+Option+I** pada Mac).

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mActionBar = getSupportActionBar();  
    mActionBar.setDisplayHomeAsUpEnabled(true);
```

Gambar 4. Kode sebelum pemformatan.

```
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        mActionBar = getSupportActionBar();  
        mActionBar.setDisplayHomeAsUpEnabled(true);  
        // Get reference to the drawer layout and set event listener
```

Formatted 7 lines
Show reformat dialog: ⌘L

Gambar 5. Kode setelah pemformatan.

Dasar-dasar kontrol versi

Android Studio mendukung berbagai sistem kontrol versi (VCS), termasuk Git, GitHub, CVS, Mercurial, Subversion, dan Google Cloud Source Repositories.

Setelah mengimpor aplikasi Anda ke Android Studio, gunakan opsi menu VCS pada Android Studio untuk mengaktifkan dukungan VCS bagi sistem kontrol versi yang diinginkan, membuat repositori, mengimpor file baru ke kontrol versi, dan menjalankan operasi kontrol versi lainnya:

1. Dari menu **VCS** Android Studio, klik Enable Version Control Integration.
2. Dari menu drop-down, pilih sistem kontrol versi yang ingin dikaitkan dengan root project, lalu klik **OK**.

Menu VCS sekarang menampilkan sejumlah opsi kontrol versi berdasarkan sistem yang Anda pilih.

Sistem build Gradle

Android Studio menggunakan Gradle sebagai dasar dari sistem build, dengan lebih banyak kemampuan khusus Android yang disediakan oleh [plugin Android untuk Gradle](#). Sistem build ini berjalan sebagai fitur terintegrasi dari menu Android Studio, dan terpisah dari command line. Anda dapat menggunakan fitur-fitur sistem build untuk:

- Menyesuaikan, mengonfigurasi, dan memperluas proses pembuatan build.
- Membuat banyak APK untuk aplikasi Anda, dengan berbagai fitur yang menggunakan project dan modul yang sama.
- Menggunakan kembali kode dan resource di seluruh set sumber.

Berkat fleksibilitas Gradle, Anda dapat mencapai semua ini tanpa mengubah file sumber inti aplikasi Anda. File build Android Studio diberi nama `build.gradle`. File tersebut adalah file teks biasa yang menggunakan sintaks [Groovy](#) untuk mengonfigurasi build dengan elemen yang disediakan oleh plugin Android untuk Gradle. Setiap project memiliki satu file build tingkat atas untuk seluruh project dan file build tingkat modul terpisah untuk setiap modul. Saat Anda mengimpor project yang ada, Android Studio akan otomatis menghasilkan file build yang diperlukan.

Untuk mempelajari sistem build dan cara mengonfigurasinya lebih lanjut, lihat [Mengonfigurasi build](#).

Varian build

Sistem build dapat membantu Anda membuat beberapa versi berbeda untuk aplikasi yang sama dari satu project. Hal ini berguna saat Anda menyediakan aplikasi dalam versi gratis dan berbayar, atau jika Anda ingin mendistribusikan beberapa APK untuk berbagai konfigurasi perangkat di Google Play.

Untuk informasi selengkapnya tentang cara mengonfigurasi varian build, lihat [Mengonfigurasi varian build](#).

Dukungan multi-APK

Dukungan multi-APK memungkinkan Anda membuat beberapa APK sekaligus secara efisien berdasarkan kepadatan layar atau ABI. Misalnya, Anda dapat membuat APK aplikasi terpisah untuk kepadatan layar hdpi dan mdpi, dengan tetap menganggapnya sebagai varian tunggal serta mengizinkannya berbagi setelan APK pengujian, javac, dx, dan ProGuard.

Untuk informasi selengkapnya tentang dukungan multi-APK, baca [Membuat beberapa APK](#).

Penyusutan resource

Penyusutan resource di Android Studio otomatis menghapus resource yang tidak digunakan dari aplikasi terpaket dan dependensi library Anda. Misalnya, jika aplikasi Anda menggunakan [layanan Google Play](#) untuk mengakses fungsi Google Drive, dan saat ini

Anda tidak menggunakan [Login dengan Google](#), penyusutan resource dapat menghapus beragam asset drawable untuk tombol SignInButton.

Mengelola dependensi

Dependensi untuk proyek Anda ditetapkan menurut nama dalam file `build.gradle`. Gradle menangani penemuan dependensi dan menyediakannya di build Anda. Anda dapat mendeklarasikan dependensi modul, dependensi biner jarak jauh, dan dependensi biner lokal dalam file `build.gradle` Anda. Android Studio mengonfigurasi project untuk menggunakan Maven Central Repository secara default. (Konfigurasi ini disertakan dalam file build tingkat atas untuk project.) Untuk informasi selengkapnya tentang mengonfigurasi dependensi, baca [Menambahkan dependensi build](#).

Fitur profil dan debug

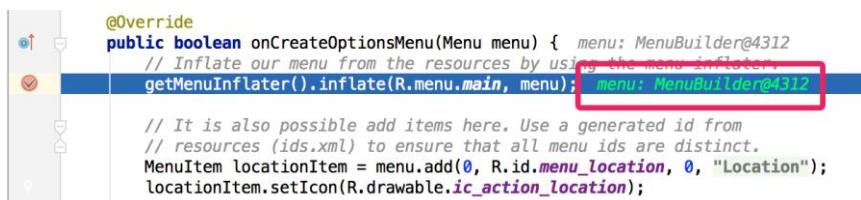
Android Studio membantu Anda menjalankan proses debug dan meningkatkan performa kode, termasuk proses debug inline dan fitur analisis performa.

Proses debug inline

Gunakan proses debug inline untuk menyempurnakan panduan kode Anda dalam tampilan debugger dengan verifikasi inline untuk

referensi, ekspresi, dan nilai variabel. Informasi debug inline meliputi:

- Nilai variabel inline
- Objek perujuk yang merujuk ke objek terpilih
- Nilai yang dihasilkan metode
- Ekspresi operator dan lambda
- Nilai tooltip



Gambar 6. Nilai variabel inline.

Untuk mengaktifkan proses debug inline, di jendela **Debug**, klik Settings dan centang kotak Show Values Inline.

Profiler performa

Android Studio menyediakan profiler performa agar Anda dapat melacak penggunaan memori dan CPU aplikasi, menemukan objek yang batal dialokasikan, menemukan kebocoran memori, mengoptimalkan performa grafis, dan menganalisis permintaan jaringan dengan mudah. Saat aplikasi Anda berjalan di perangkat atau emulator, buka tab **Android Profiler**.

Untuk informasi lebih lanjut tentang profiler performa, lihat [Alat profiling performa](#).

Heap dump

Saat memprofilkan penggunaan memori di Android Studio, Anda dapat sekaligus memulai pembersihan sampah memori dan membuang heap Java ke cuplikan heap dalam file format biner HPROF khusus Android. Penampil HPROF akan menampilkan class, instance setiap class, dan struktur referensi untuk membantu Anda melacak penggunaan memori serta menemukan kebocoran memori.

Untuk informasi selengkapnya tentang menangani heap dump, lihat [Memeriksa heap dan alokasi](#).

Memory Profiler

Anda dapat menggunakan Memory Profiler untuk melacak alokasi memori dan melihat di mana objek dialokasikan saat Anda melakukan tindakan tertentu. Dengan mengetahui alokasi ini, Anda dapat mengoptimalkan performa dan penggunaan memori aplikasi dengan menyesuaikan panggilan metode yang terkait dengan tindakan tersebut.

Untuk informasi tentang pelacakan dan analisis alokasi, lihat [Memeriksa heap dan alokasi](#).

Akses file data

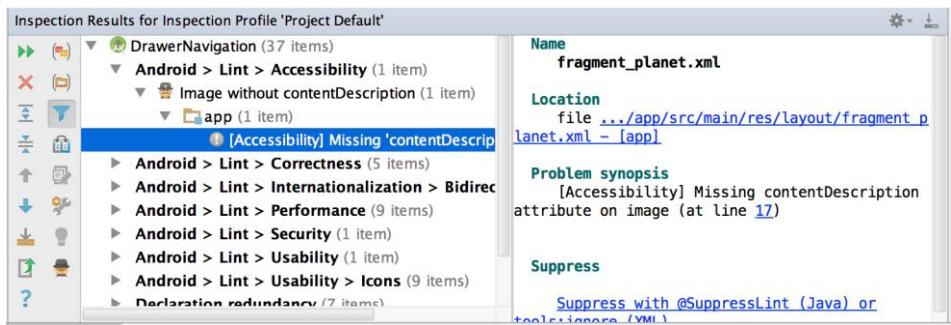
Android SDK Tools, seperti Systrace dan logcat, menghasilkan data performa dan proses debug untuk analisis aplikasi secara mendetail.

Untuk menampilkan file data yang dihasilkan, buka jendela alat Captures. Pada daftar file yang dihasilkan, klik dua kali file untuk melihat data. Klik kanan sembarang file .hprof untuk mengonversinya ke dalam format file standar Menyelidiki penggunaan RAM.

Pemeriksaan kode

Setiap kali Anda mengompilasi program, Android Studio akan otomatis menjalankan Lint yang telah dikonfigurasi dan pemeriksaan IDE lainnya untuk memudahkan Anda mengidentifikasi serta memperbaiki masalah kualitas struktur kode Anda.

Fitur Lint memeriksa file sumber project Android Anda untuk menemukan potensi bug dan peluang pengoptimalan guna mencapai ketepatan, keamanan, performa, kegunaan, aksesibilitas, serta internasionalisasi.



Gambar 7. Hasil pemeriksaan Lint di Android Studio.

Selain pemeriksaan Lint, Android Studio juga menjalankan pemeriksaan kode IntelliJ dan memvalidasi anotasi untuk menyederhanakan alur kerja coding Anda.

Untuk informasi selengkapnya, lihat [Menyempurnakan kode dengan pemeriksaan lint](#).

Anotasi di Android Studio

Android Studio mendukung anotasi variabel, parameter, dan nilai kembalian untuk membantu Anda merekam bug, seperti pengecualian pointer null dan konflik jenis resource. Android SDK Manager mengemas library Support-Annotations di Android Support Repository untuk digunakan dengan Android Studio. Android Studio memvalidasi anotasi yang sudah dikonfigurasi selama pemeriksaan kode.

Untuk penjelasan selengkapnya tentang anotasi Android, lihat [Menyempurnakan pemeriksaan kode dengan anotasi](#).

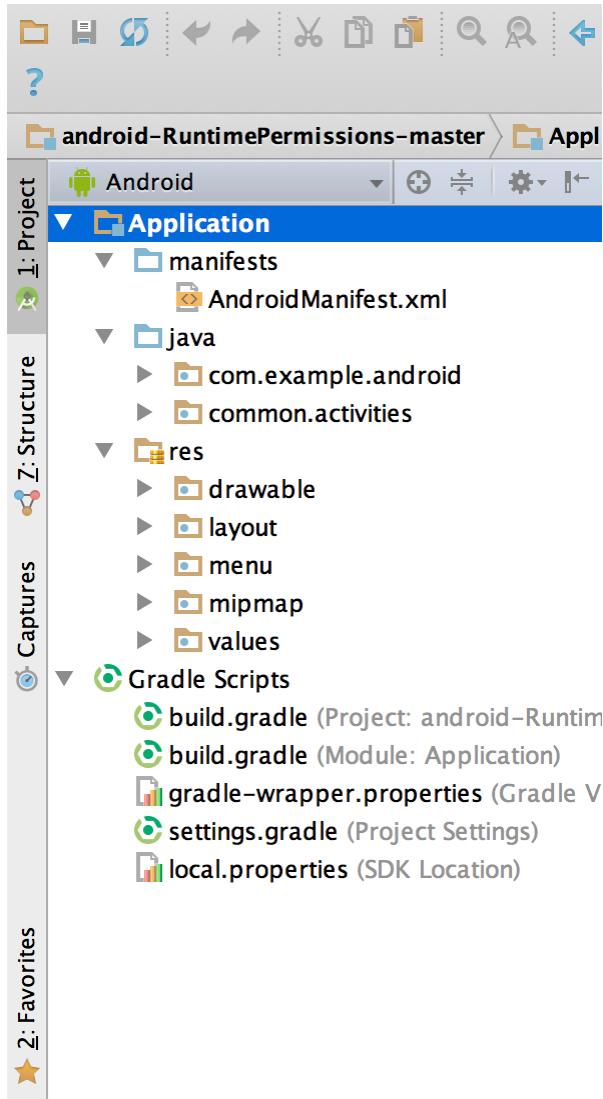
Pesan log

Pada saat membangun dan menjalankan aplikasi dengan Android Studio, Anda dapat melihat entri log dan log perangkat iklan (logcat) dengan mengeklik Android Monitor di bagian bawah jendela.

Apabila Anda ingin men-debug aplikasi Anda dengan Android Device Monitor, Anda dapat meluncurkan Monitor Perangkat dengan mengklik **Tools> Android > Android Device Monitor**.

Monitor Perangkat adalah tempat Anda dapat menemukan perangkat DDMS lengkap untuk membuat profil aplikasi Anda, mengendalikan perilaku perangkat, dan banyak lagi. Ini juga mencakup alat Hierarchy Viewer untuk membantu mengoptimalkan tata letak Anda.

Struktur project



Gambar 1. File project dalam tampilan Android.

Setiap project di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file resource. Jenis modul meliputi:

- Modul aplikasi Android
- Modul library
- Modul Google App Engine

Secara default, Android Studio menampilkan file project Anda dalam tampilan project Android, seperti yang ditunjukkan pada gambar 1. Tampilan ini disusun menurut modul untuk memberikan akses cepat ke file sumber utama project Anda.

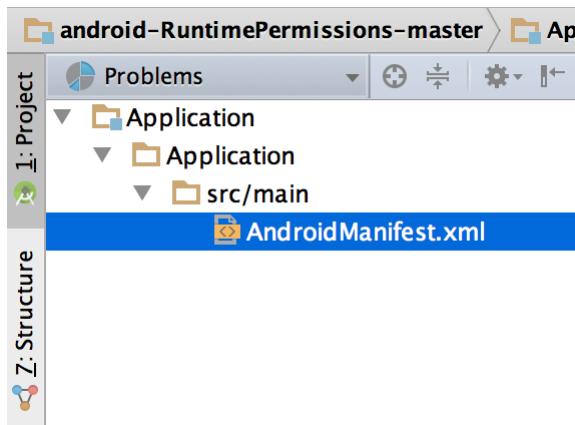
Semua file build terlihat di tingkat teratas di bagian **Gradle Script** dan setiap modul aplikasi berisi folder berikut:

- **manifes**: Berisi file `AndroidManifest.xml`.
- **java**: Berisi file kode sumber Java, termasuk kode pengujian JUnit.
- **res**: Berisi semua resource non-kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur project Android pada disk berbeda dengan representasi tersatukan ini. Untuk melihat struktur file project sebenarnya, pilih **Project** dari menu drop-down **Project** (pada gambar 1, ditampilkan sebagai **Android**).

Anda juga dapat menyesuaikan tampilan file project untuk berfokus pada aspek spesifik dari pengembangan aplikasi Anda. Misalnya, memilih tampilan **Problems** pada project Anda akan menampilkan link ke file sumber yang berisi error coding dan sintaks yang

dikenali, seperti tag penutup elemen XML yang tidak ada dalam file tata letak.



Gambar 2. File project dalam tampilan Problems, menunjukkan file tata letak yang memiliki masalah.

3.2.2 Eclipse

SEJARAH ECLIPSE



Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platformindependent). Berikut ini adalah sifat dari Eclipse: 1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X. 2. Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya. 3. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya. Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan open source. Open source berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut plugin. 1. Sejarah Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk Eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. IBM menginvestasikan US\$ 40 juta untuk pengembangannya. Sejak 5 November 2001, konsorsium Eclipse Foundation mengambil alih pengembangan Eclipse lebih lanjut.

1.6 Eclipse

Eclipse adalah sebuah IDE integrated Development Environment untuk mengembangkan perangkat lunak dan dapat dijalankan disemua platform platform independent. Berikut ini adalah sifat dari Eclipse:

- 1. Multi-platform: target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- 2. Multi-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti CC++, Cobol, Phthon, Perl, PHP dan lain sebagainya.
- 3. Multi-role: selain sebagai IDE untuk pengembangan aplikasi, eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikeranakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plugin

1.6.1 Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US 40 Juta untuk pengembangannya. Semenjak itu konsumen Eclipse foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

1.6.2 Arsitektur Eclipse

Sejak versi 3.0 eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan didalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform RCP. Berikut ini adalah komponen yang membentuk RCP:

1. Core platform
2. OSGi
3. SWT standar Widget Toolkit
4. Jface
5. Eclipse Workbench

Sejarah standar Eclipse selalu dilengkapi dengan JDT java Development Tools, Plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE Plug-in Development Environment untuk mengembangkan plug-in baru. Eclipse beserta plug-in nya diimplementasikan dalam bahasa pemrograman Java. Konsep Eclipse Adalah IDE yang terbuka open, mudah diperluas extensible untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, eclipse tidak saja untuk mengembangkan program java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin menembangkan program CC++ terdapat Plug-in CDT CC++ Development tools. Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat Plug-in sesuai dengan keinginannya.

1.6.3 Perkembangan Eclipse

Sejak tahun 2006, Eclipse foundation mengkoordinasikan peluncuran eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous release. Setiap versi peluncuran terdiri dari

eclipse platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah deployment dan maintenance untuk sistem enterprise. Adapun versi eclipse yang telah diluncurkan yaitu: 1. Eclipse 3.0 2. Eclipse 3.1 3. Callisto 4. Europa 5. Ganymede 6. Galileo 7. Helios 8. Juno

1.7 ADT

Arsitektur Sejak versi 3.0,

Eclipse pada dasarnya merupakan sebuah kernel. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah dipasang (diinstal). Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP:

- ♣ Core platform
- ♣ OSGi
- ♣ SWT (Standard Widget Toolkit)
- ♣ JFace
- ♣ Eclipse Workbench
- Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru.
- Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java.
- Konsep Eclipse adalah IDE
- adalah 1. terbuka (open), 2. mudah diperluas (extensible) untuk apa saja, dan 3. tidak untuk sesuatu yang spesifik.
- Eclipse tidak saja untuk mengembangkan program Java, tetapi juga untuk

berbagai macam keperluan. Perluasan apapun cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ maka telah terdapat plug-in CDT (C/C++ Development Tools) yang dapat dipasang di Eclipse untuk Eclipse menjadi perangkat untuk pengembangan C/C++.

Pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya. Salah satu situs yang menawarkan plug-in yang gratis adalah Eclipse downloads by project. 3. Histori (versi-versi) Eclipse Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah deployment dan maintenance untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

Instalasi Langkah-langkah instalasi Eclipse adalah sebagai berikut:

1. Memastikan JDK telah terpasang (terinstall) di komputer atau laptop. Homepage Eclipse adalah <http://www.eclipse.org>, dan pemrogram bisa mendownload Eclipse IDE (Integrated Development Environment).

2. File Eclipse yang telah diunduh (download) berbentuk (.zip). Pemrogram perlu mengekstrak menjadi folder. File hasil ekstrak disimpan di direktori C:/Program Files/ atau pada direktori lain. Eclipse Galileo bisa langsung digunakan tanpa memasang melalui proses instalasi terlebih dahulu. Eclipse Galileo tinggal menaruhnya. Kemudian Klik icon eclipse application.
3. Maka akan muncul tampilan awal eclipse seperti dibawah ini.
4. Kemudian muncul Workspace Launcher, yaitu direktori tempat menyimpan project yang dibuat. Maka untuk menyimpan di project di direktori tertentu, pilih Browse ◊ Direktory tertentu (Misal, E:\PROJECT\Project). Jika ingin direktorinya default maka, checklist Use this as default and not ask again
5. Selanjutnya akan tampil jendela seperti berikut ini.

Membuat Project Baru : Klik menu File ◊ New ◊ Project.
Gambar

6 Buat Project Baru

7. Maka akan tampil seperti ini, kemudian Pilih Java Project, klik Next.
8. Isi Project name dengan nama Z2S, Klik Next.
9. Muncul tampilan seperti ini, kemudian Klik Finish. Klik Yes ketika eclipse menanyakan apakah anda mau mengganti view ke Java Perspective.
- 10. Tampilan jendela kerja Eclipse**

- 11.** Selanjutnya pada Project Z2S, terlebih dahulu buat package, yaitu Klik kanan Z2S ◊ Klik New ◊ Klik Package.
- 12.** Pada Java Package, Name diisi dengan z2s.a000.u001.template ◊ Klik Finish
- 13.** Selanjutnya buat Kelas baru di package z2s.a000.u001.template, yaitu Klik kanan z2s.a000.u001.template ◊ New Class
- 14.** Pada Java Class, Name diisi dengan Template ◊ Klik Finish
- 15.**

3.2.3 Ionic Framework



Pengenalan Ionic Framework

Apa itu Ionic Framework ? pertanyaan ini mungkin yang terbesit dalam hati dan pikiran anda ketika mendengar Tentang Ionic Framework. ionic framework merupakan platform yang digunakan untuk mengembangkan aplikasi mobile hybrid. maksud dari mobile hybrid disini adalah aplikasi mobile yang sekali dikembangkan bisa digunakan untuk beberapa platform contohnya ios, dan android.

ionic ini dibangun menggunakan bahasa pemrograman basis web (HTML5, JavaScript). ionic sendiri di targetkan untuk programer web agar bisa mengembangkan aplikasi mobile menggunakan teknologi web.

ionic menggunakan node.js, Angular, HTML5, SASS, sebagai engine untuk membangun aplikasi mobile. berikut saya akan membahas beberapa kelebihan dari ionic framework. kelebihan ionic framework

1. opensource. ionic menggunakan lisensi opensource sehingga anda lebih mudah mengembangkan aplikasi mobile yang free atau komersial dengan menggunakan ionic ini
2. ionic framework ini dikenal lebih cepat dibandingkan dengan beberapa framework lain.
3. target ionic untuk android 4+ dan ios 7+. artinya bagi yang suka mengupdate perkembangan teknologi, anda tidak perlu khawatir akan kompatibilitas dengan sistem operasi yang lama. karena itu ionic sangat cocok untuk anda yang mengembangkan aplikasi mobile versi terbaru.

Kelebihan Ionic Framework

Jadi jika anda punya website atau WebApp dan ingin dibuatkan aplikasi mobilennya, Ionic Framework adalah pilihan yang tepat. Terus apa bedanya Ionic dengan Framework lain? cek disini

Ionic Menggunakan lisensi Opensource

Ionic platform menggunakan lisensi opensource, anda boleh membuat aplikasi free ataupun comercial dengan Ionic.

Menggunakan teknologi web terbaru

Ionic memanfaatkan AngularJS untuk implementasi logicnya. Mengapa tidak memakai jQuery? karena jQuery terkenal lambat di mobile sedangkan Angular menawarkan performa dan respon cepat serasa aplikasi native.

Target Hanya untuk Android 4 dan ios 7 Keatas

Anda bisa melihat ini sebagai kelebihan ataupun kekurangan. Bagi yang suka dengan hal baru dan ingin kode yang bersih tanpa memikirkan kompatibilitas dengan versi mobile OS lama, maka ionic adalah platform yang paling pas.

Berbasis Apache Cordova /Phonegap

Perhatikan point ini. Ionic hanya menyediakan frameworknya, untuk membungkusnya menjadi aplikasi Android atau iOS anda tetap pakai phonegap. Artinya bagi para programer phonegap dengan platform lain, keahliannya tetap bisa dipakai.

Tool Resmi Ionic

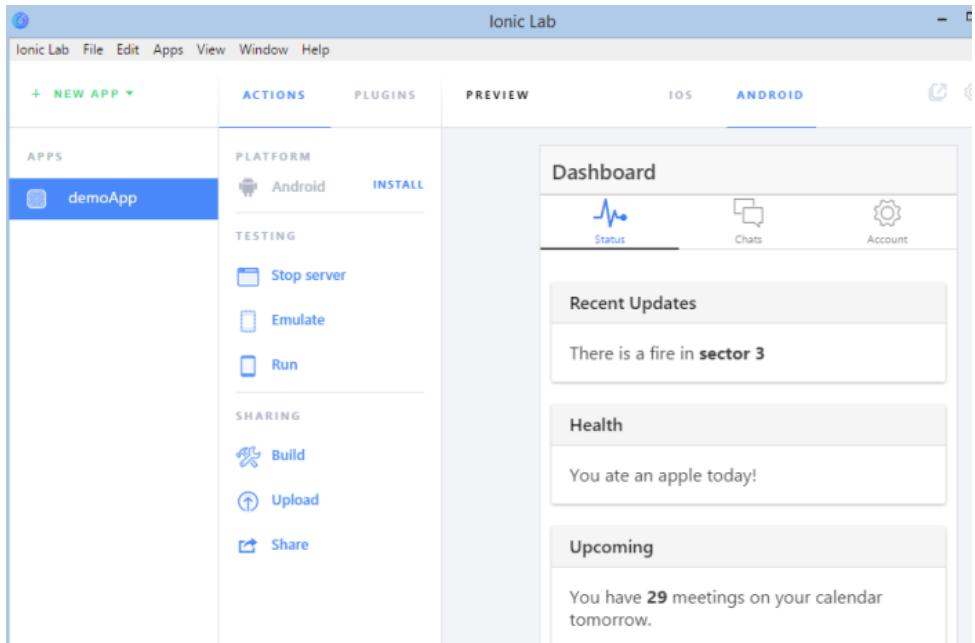
Ionic CLI

Ionic CLI adalah tool command line yang digunakan untuk membuat project dan menjalankan project. Masih banyak fitur dari

ionic CLI ini, silahkan mengacu ke <http://ionicframework.com/docs/cli/start.html>.

ionic Lab

IonicLab Adalah solusi bagi anda yang ingin kenyamanan. IonicLab adalah Versi GUI dari Ionic CLI, anda tinggal klik untuk melakukan operasi operasi yang semula harus mengetikan perintah di Console.



Ionic Lab

Ionic Creator

Berbeda dengan jQuery mobile atau platform lain yang memaksa kita coding manual untuk mendesain tampilan UInya. Ionic menyediakan Ionic Creator, produk free yang akan memudahkan anda mendesain tampilan UI dari aplikasi. Ionic Creator sendiri

adalah WebApp artinya anda harus mengaksesnya lewat Browser. Untuk Akun Free kita hanya bisa mendesain satu aplikasi saja, Jika ingin lebih belilah paket Premium. Anda bisa mencobanya di <https://creator.ionic.io/app/login>.

Ionic ViewApp

Aplikasi ini adalah Aplikasi Android. Fungsi utamanya adalah memreview program Ionic langsung dari Android. Aplikasi yang bisa di preview oleh ViewApp bisa aplikasi yang kita buat atau aplikasi orang lain yang dishare di Ionic Market.

Apa yang harus dipelajari terlebih dahulu

Setelah kita tahu teknologi kelebihan ionic diatas, ada beberapa teknologi yang harus ada kuasai, minimal anda tahu sebelum terjun ke Ionic. Ilmu ini harus anda pahami kalau ingin membuat aplikasi bagus. Kalau anda tidak pernah mencoba atau mendengar teknologi dibawah ini, jangan buru buru coba ionic, kalau dipaksakan, paling paling anda berhenti ditengah jalan.Teknologi itu adalah

HTML5/CSS/Javascript

Nama komponen visual di mobile App (android/iOS)

AngularJS

Dasar dasar NodeJS/NPM

Phonegap/Apache Cordova

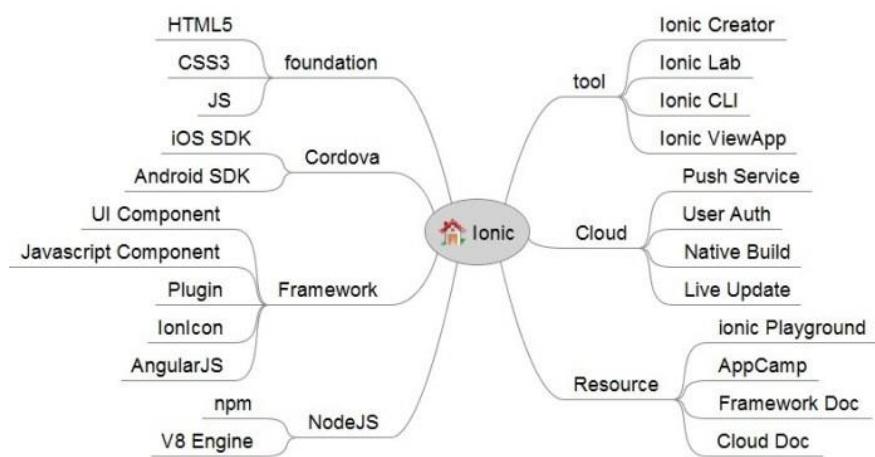
Command line (CMD/Bash/Terminal Linux)

Cara membuatApk atau membuild /compile iosApp.

Jadi 7 point itu harus anda pahami dulu, mengapa saya tekankan disini. Untuk point 7, jika anda hanya ingin membuat versi androidnya, maka anda tidak perlu belajar XCode. karena banyak programer yang berusaha memakai teknologi baru tanpa memperhatikan ilmu dasarnya, makanya jangan heran banyak

programer stress dan banyak mahasiswa mentok di tengah jalan karena tergiur berbagai fitur framework yang ditawarkan.

Selain itu, alangkah baiknya anda memahami Ekosistem Ionic secara menyeluruh, minimal tahu teknologi dibalik ionic agar mudah dalam mempelajari dan medebug Program. Anda bisa melihat Ekosistem Ionic dan Hubungannya dengan teknologi lain dari diagram dibawah ini.



Jika anda tidak ingin memahami semuanya, saya sarankan fokuslah ke teknologi AngularJS karena jika diibaratkan HTML5,CSS3,JS adalah badannya maka angularJS adalah rohnya. Dengan bahasa lain, Framework Ionic (UI component) adalah Viewnya dan AngularJS adalah Controllernya.

Ionic Platform

Selain Ionic Framework yang free dan opensource, Ionic juga memperkenalkan ionic Platform. Ionic Platform memberikan layanan tambahan bagi para developer ionic. Ionic Platfrom menambahkan fitur Create, deploy, update, Analytics, Marketing

(Ionic Market) dan Push notification kedalam aplikasi yang dibangun dengan ionic. Saat ini, ionic Platform masih dalam tahap Alpha testing dan anda gratis mencobanya.

Ionic makes building, deploying, and scaling your app easy.

Focus on what matters. Ionic takes care of the infrastructure. Get up and running instantly with prototyping, packaging, analytics, push notifications, and live updating tools.



Untuk Platform ini memang masih alha dan belum ready, namun jika sambutan dari Developer positif, maka kedepannya ionic adalah platform paling menjanjikan untuk membuat aplikasi mobile yang terasa Native untuk Platform Android dan iOS.

3.2.4 Reactive

React Native adalah sebuah framework javascript yang dikembangkan oleh facebook dan memungkinkan kita membuat aplikasi mobile android atau ios menggunakan teknologi web. Seperti yang kita ketahui sekarang ini juga sudah banyak framework javascript yang bisa kita gunakan untuk membuat aplikasi android atau ios, akan tetapi untuk React Native ini berbeda dengan framework - framework javascript yang lainnya. Dengan React Native ini kita tidak membuat aplikasi Hybrid yang dimana aplikasi ini berjalan di javascript runtime akan tetapi kita akan membuat Real Aplikasi

dimana untuk android akan di compile di java dan untuk IOS akan di compile di Objective-C.

Fitur React Native

Didalam React Native ini terdapat beberapa komponen yang membuatnya bisa kita gunakan untuk membuat aplikasi mobile cross-platform. Berikut ini adalah beberapa fitur dari React Native ini :

ReactJS

Sebelum react native di keluarkan oleh facebook yang pertama di kenalkan adalah ReactJS yaitu framework javascript yang bisa digunakan untuk membuat aplikasi web dengan java script. Jika kamu sudah pernah menggunakan ReactJS maka kamu akan mudah untuk memahami React Native dan bagi kamu yang belum pernah mencoba ReactJS juga tidak masalah karna saya sendiri juga belum pernah coba ReactJS sebelumnya.

Native

Di Android dan IOS memiliki arsitektur yang berbeda, nah disini Native ini yang berfungsi untuk menghandle komponen dari masing-masing platform. Di native ini nanti kita akan mempelajari beberapa komponen untuk keperluan kita belajar membuat aplikasi dengan React Native.

Platform

Untuk saat ini platform yang sudah terintegrasi adalah Android dan IOS. Mungkin kedepanya bisa lebih dari dua platform ini yang bisa terintegrasi dengan react native ini kita tunggu saja.

Kelebihan React Native

Sebelum kita lanjut lebih dalam untuk coding kita perlu tau dulu sedikit apa sih kelebihan React Native dan kenapa kita harus mencoba membuat aplikasi dengan React Native ini. Nah berikut ini adalah kelebihan kita untuk menggunakan React Native.

JavaSript

Seperti yang sudah kita ketahui beberapa waktu yang lalu di survey stackoverflow JavaScript adalah salah satu bahasa pemrogramman yang paling banyak digunakan oleh developer di dunia. Dengan begitu artinya kita sebagai developer akan sangat mudah untuk mempelajari JavaScript ini karna komunitas yang sudah besar dan juga kita mempunyai banyak tempat untuk bertanya kalau kita mengalami kebingungan. Dan bagi kamu yang sudah berpengalaman dengan web development tentunya sudah akrab dengan javascript ini dan kamu sudah punya dasar untuk mulai membuat aplikasi dengan react native ini.

Code Sharing

Sebagian codebase yang kamu tulis itu bisa kamu gunakan untuk berbeda platform dimana disni bisa digunakan untuk android dan ios. Dengan begitu untuk hal - hal yang mendasar jika kita memiliki fungsi - fungsi umum yang tidak bersentuhan langsung dengan komponen yang berbeda platform ini kita bisa kita gunakan codebase tersebut. Sebagai contoh semisal kita punya sebuah fungsi untuk mengkonfersi currency, maka dengan begitu kita bisa gunakan untuk android ataupun di ios.

Community

React Native ini cukup diminati oleh para developer di dunia berikut ini adalah screenshoot dari repository react native.

Kelemahan React Native

React Native sendiri juga memiliki kelemahan berikut ini adalah beberapa keleamahan dari react native

Components

Untuk mengembangkan aplikasi mobile dengan react native kita harus banyak memahami Components dari masing - masing platform. Dikarenakan android dan ios ini memiliki arsitektur yang berbeda dengan begitu Components Native dari masing - masing platform masih terbatas. Untuk itu jika kita ingin menggunakan

komponen yang belum ada maka kita harus membuatnya sendiri. DImana dengan begini masih belum bisa di lakukan oleh pemula. Tetapi jangan khawatir sampai saat ini kontributor untuk react native ini sudah mencapai 1,231 orang, artinya dengan begitu perkembangan dari react native hari demi hari akan lebih baik. Atau kalau kamu punya pengalaman yang cukup untuk javascript dan mobile development maka silahkan berkontribusi untuk react native ini.

JavaScript

Nah java script ini juga menjadi salah satu kelemahan dari react native, dikarenakan java script ini berkembang begitu cepatnya yang mengakibatkan beberapa hal itu tidak bisa di lakukan dengan perubahan yang begitu cepat. Untuk itu kita harus pintar - pintar beradaptasi dengan JavaScript ini.

Framework

Framework React Native juga menjadi kelemahan karena perubahan yang begitu cepatnya yang mengakibatkan ketika kamu mengupdate react native ke versi yang baru untuk versi yang lama beberapa tidak bisa kamu gunakan. So dengan begitu kamu harus pandai - pandai melakukan management codebase kamu agar tidak berpengaruh banyak ketika ada update dari framework react native ini.

Platform

Nah buat kamu yang ingin mengembangkan react native untuk ios maka kamu perlu punya macbook so memang karena ekosistem dari ios ini semua harus menggunakan produk dari Apple. Untuk android juga kamu butuh java dan gradle dimana dua hal ini yang menjadi issue dikalangan developer karena kebutuhan devices yang minimal dengan memory 4gb untuk bisa bekerja maksimal. Jadi memang salah satu tantangan jika kamu memutuskan untuk terjun di pengembangan aplikasi mobile kamu harus memiliki devices yang memadahi.

Perfomance

Untuk perfomance juga masih menjadi perdebatan dikalangan developer ada yang mengatakan bagus dan ada yang mengatakan kurang bagus juga. Akan tetapi bagi saya untuk react native ini sendiri permofance nya cukup lebih baik dibanding kan dengan kita menggunakan framework javascript yang lain karena pada dasarnya dengan menggunakan react native kita membuat apliaksi mobile dengan perfomance native, akan tetapi di beberapa kondisi react native belum sebaik kita pure menggunakan native.

Persiapan

Nah sekarang kita lupakan tentang kelebihan dan kekurangan dari react native ini dan mari kita mulai untuk belajar bagaimana kita membuat aplikasi mobile dengan react native. Pada tutorial ini kita

akan belajar membuat aplikasi ios dan berikut ini hal - hal yang perlu kita siapkan untuk memulai membuat aplikasi IOS dengan menggunakan react native.

MacBook dan OSX

Pastikan kamu sudah pakai macbook dan osx karena disini kita akan latihan membuat aplikasi untuk IOS, untuk saat ini saya belum pernah coba untuk menggunakan hackintos. Akan tetapi buat kamu yang belum punya macbook mungkin bisa coba untuk menginstal laptop atau pc kamu sistem operasi hackintos. dan jangan lupa untuk menginstall Xcode karna nantinya kita perlu ios emulator untuk run aplikasi yang kita buat.

NPM dan NodeJS

Dikarenakan kita menggunakan bahasa pemrogramman javascript maka kita perlu nodejs dan npm untuk kebutuhan installasi dari react native ini.

Persiapan Installasi

Sekarang mari kita lanjutkan dengan melakukan proses installasi untuk beberapa kebutuhan mengembangkan aplikais dengan reac native ini.

Installasi Homebrew

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Installasi Watchman

```
brew install watchman
```

Installasi React Native

```
npm install -g react-native-cli
```

Jangan lupa pastikan kamu juga sudah menginstall **XCode**.

Buat Aplikasi Pertamamu

Sekarang mari kita buat aplikasi IOS pertama kamu dengan mengikuti langkah - langkah berikut ini.

Buat Project Aplikasi

```
react-native init IOSReactTutorial
```

Jalankan React Native Compiler

Yang pertama kita masuk ke direktori dari project yang telah kita buat.

```
cd IOSReactTutorial
```

Kemudian jalankan react native packeger, dalam pengembangan jangan matikan perintah ini tetap biarkan jalan selama kita dalam proses development.

```
react-native start
```

Jalan Aplikasi Di Emulator

Sekarang kita jalankan aplikasi yang kita buat di emulator.

```
react-native run-ios
```

Jika kamu berhasil maka akan muncul tampilan seperti berikut ini.



CODEPOLITAN

Mari Buat Hello Word

Secara default berikut ini adalah beberapa boilerplate code yang di generate secara default oleh react native.

CODEPOLITAN

Bisa kita lihat pada gambar di atas secara default react native akan mengenerate folder android dan ios dimana disana lah hasil dari compiler untuk aplikasi yang di jadikan native oleh React Native ini. Untuk di ios kita akan menggunakan file index.ios.js sebagai tempat kita menaruh code - code yang kita buat dengan react native ini. Sekarang mari kita buat aplikasi hello world dengan react native dengan membuat code seperti berikut ini.

3.3 *Software* untuk Membuat Aplikasi Dekstop

3.3.1 *Electron*

Electron adalah sebuah *framework* yang memudahkan *developer* untuk membuat aplikasi *desktop* lintas-perangkat (*cross-platform*) seperti pada Linux, Windows dan MacOS dengan menggunakan teknologi web (JavaScript).

Electron pada dasarnya menggunakan [Node.js](#) dan [Chromium](#). Iya benar, sebenarnya aplikasi kita akan jalan di atas sebuah *browser*. Jadi ide dasarnya adalah mengabungkan

kemudahan membuat *website* dengan kekuatan *native desktop app* yang dapat berinteraksi secara langsung dengan sistem operasi.

Electron sekarang telah dipakai untuk membuat beragam aplikasi *desktop*, seperti **Atom**, **Visual Code Studio**, **Slack** hingga **Wordpress Desktop**. Untuk memulai menggunakan Electron, kita harus meng-*install* Node.js serta **npm** (*node package manager*) untuk mengatur *dependency*-nya.

Pada tutorial kali ini Saya menggunakan **TeaLinuxOS**, sistem operasi (*distro*) turunan Ubuntu yang dirancang spesial untuk dunia pemrograman (bukan promosi ya :D).

Catatan : Kita juga bisa menggunakan sistem operasi lain seperti Windows dan MacOS untuk bermain dengan Electron.

Buka Terminal kita (CTRL + ALT + T)

Untuk menginstall Node.js, kita membutuhkan **curl** :

```
sudo apt-get install curl
```

Ambil paket *setup* Node.js dengan perintah berikut ini :

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```

Catatan : Kita akan menginstall Node.js versi 6 sekian-sekian.

Setelah paket terunduh, pasang dengan perintah berikut :

```
sudo apt-get install -y nodejs
```

Sekarang Node.js sekaligus npm telah terpasang pada mesin kita.

Untuk memastikan Node.js dan npm terpasang dengan benar, kita dapat melakukan perintah berikut :

```
node -v
```

```
npm -v
```

Setelah selesai menginstall Node.js

Buat suatu directory / folder untuk proyek kita

```
mkdir codepolitan-desktop
```

Masuk kedalam foldernya

```
cd codepolitan-desktop
```

Pada tutorial kali ini kita akan menggunakan dua cara, yaitu

- **Cara Gampang**, karena kita akan menggunakan tools CLI (Command Line Interface) yang mudah, cepat dan tanpa *coding*.
- **Cara Seru**, karena kita akan memasang kerangka kerja proyek Electron, lalu menulis *code*, menjalakannya sampai memaketnya

menjadi *executable* sehingga memahami bagaimana Eletron berkerja.

Cara Gampang

Cara gampang / mudah ini kita akan menggunakan *tools* bernama **Nativefier** yang akan membuat website apapun seperti **CodePolitan** atau **HACKTIV8** atau yang lainnya menjadi *desktop app* dengan mudah dan cepat. Nativefier pada dasarnya membungkus web tersebut ke dalam Electron dengan sedikit perintah pada *console terminal*.

Untuk memasang Nativefier, cukup ketikan perintah

```
npm install nativefier -g
```

Keterangan : npm akan menginstall nativefier secara ***globally*** dengan paramater "-g" sehingga dapat digunakan dimana saja *directory* kita berada (tidak hanya pada *directory* proyek kita).

Untuk menjalakanya sangat sederhana, cukup ketikan Nativefier diikuti URL websitenya, seperti ini :

```
nativefier "http://codepolitan.com"
```

Keterangan : Nativefier akan secara otomatis membaca nama dari website tersebut, namun kita juga dapat menginisialisasi nama

aplikasi kita dengan menambahkan *parameter* --name, menjadi seperti ini :

```
nativefier --name "Codepolitan.com - Media Edukasi dan Informasi Pemrograman Indonesia" "http://codepolitan.com"
```

Tunggu sejenak proses berjalan, setelah selesai Nativefier akan membuat sebuah *directory* yang berisi *source* dan juga *executable* aplikasi kita, cukup klik dua kali untuk menjalakanya. Tadaa, aplikasi web rasa *desktop* kita sudah jadi.

Cara Seru

Langkah awal sebelum mulai *coding*, kita siapkan dulu berkas-berkas dengan mengunduh [boilerplate](#) dari Electron, jadi telah ada *template* atau kerangka dasar yang sudah siap gunakan untuk belajar membuat aplikasi pada Electron, jadi tidak perlu repot-repot mulai dari nol. Berikut *boilerplate* Eletron detailnya : [Electron-Quick-Start](#)

Clone dengan Git menggunakan perintah :

- * Bagi kamu yang belum pernah menggunakan Git, kamu bisa kepoin di [Memulai Dasar Git](#)
- ** Guna memfokuskan tutorial ini untuk mengenal Electron kamu dapat dengan mudah memasang Git dengan perintah :

```
sudo apt-get install git
```

*** Untuk pengguna Windows dan MacOS, dapat mengunduh *installer* Git di laman [unduh git](#)

Mari kita *clone* proyek tersebut :

```
git clone https://github.com/electron/electron-quick-start
```

atau unduh **Electron-Quick-Start** di halaman GitHub-nya, jika kita tidak ingin menggunakan Git.

Masuk kedalam foldernya

```
cd eletron-quick-start
```

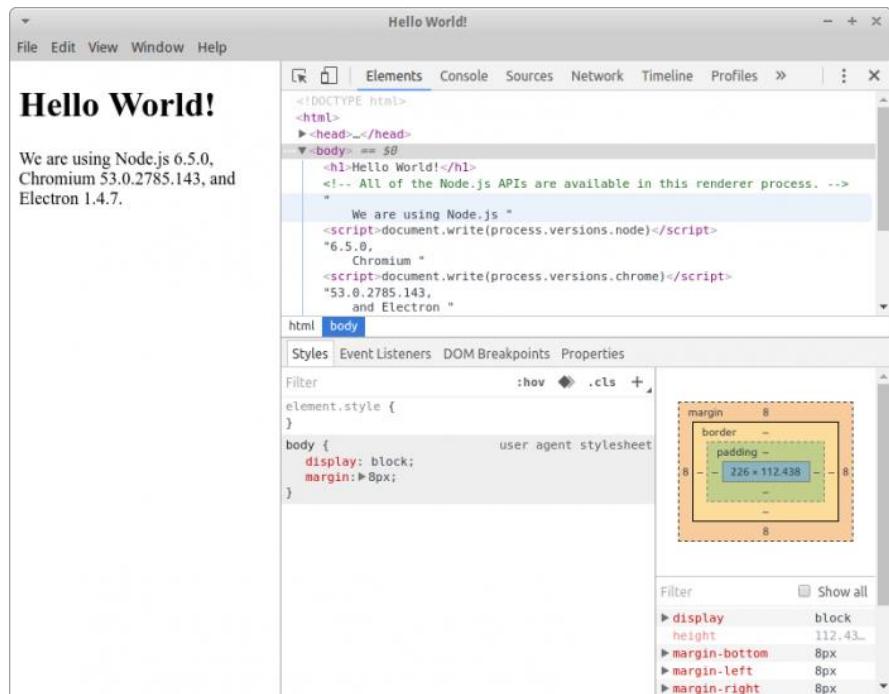
Install *dependency*-nya dengan perintah :

```
npm install
```

Untuk memastikan proses instalasi berhasil dengan baik, mari kita uji dengan jalankan proyek ini dengan perintah :

```
npm start
```

Jika semua berhasil, maka akan tampil seperti ini :



Kita tidak harus memahami semua *folder* dan fungsi setiap file pada pengenalan saat ini, fokus pada *file* main.js yang menjadi *file* utama dan dijalankan diawal ketika proyek dijalankan.

Pada file main.js telah diberi komentar untuk menerangkan fungsi masing-masing baris kode. Saya telah mentranslasikan komentar tersebut kedalam bahasa Indonesia supaya lebih mudah dipahami.

Pada code main.js memanggil file index.html yang akan dimuat saat semua proses telah selesai dan siap (*on ready*) digunakan.

Ketika membuka file index.html, kita akan menjumpai *markup* HTML pada umumnya, meliputi HTML, CSS dan JS

Coba ubah-ubah isi file tersebut kemudian jalankan ulang proyek kita untuk melihat bagaimana dia berkerja :D

Okay, balik ke pembuatan aplikasi kita.

Ingat kita akan membuat aplikasi yang sebenarnya adalah aplikasi berbasis web menjadi *desktop*, Electron telah menyediakan *method* < *webview* >, bahasa gampangnya, kita akan meng-*embed* halaman web kedalam aplikasi kita. Berikut coding-an setelah di otak-atik, Saya sudah menambahkan keterangan pada komentar di sisipan baris *code*-nya.

Jalankan kembali proyek kita dengan

```
npm start
```

Sebenarnya proyek kita sudah bisa digunakan sampai disini, tapi bagaimana cara mendistribusikan aplikasi kita ke pengguna ? Kita perlu membungkus atau memaketnya untuk menjadi *executable*.

Untuk itu kita butuh **Electron-packager** , tambahkan pada proyek kita dengan perintah :

```
npm install electron-packager --save-dev
```

Atau secara *globally*

```
sudo npm install electron-packager -g
```

Setelah terpasang, cukup jalankan perintah dengan format seperti ini :

```
electron-packager --platform= --arch= [optional flags...]
```

Keterangan :

1. electron-packager adalah nama perintah untuk menjalakanya
2. <sourcedir> adalah lokasi sumber proyek kita, cukup isi dengan " ./ " ketika kita menjalakan perintah pada *folder* proyek kita.
3. <appname> adalah nama aplikasi yang hendak kita paket, gunakan tanda petik "" untuk membungkus namanya.
4. --platform adalah sistem operasi target proyek kita
 - linux : untuk OS berbasis Linux seperti Ubuntu, Debian dan turunannya
 - darwin : untuk MacOS X
 - win32 : untuk OS Windows
5. --arch atau singkatan dari *architecture* atau arsitektur komputer target
 - ia32 : untuk 32 bit
 - x64 : untuk 64 bit
6. Opsi-opsi lain seperti

- --all : untuk memaket kesemua sistem operasi sekaligus semua arsitekturnya, karena perintah ini sama saja dengan --platform=all --arch=all
- Penjelasan opsi lebih lengkap kita dapat lihat disini usage.txt
Contoh, Saya menggunakan platform Linux dan berarsitektur 64bit
`electron-packager ./ "codepolitanseru" --platform=linux --arch=x64`

Setelah itu akan terbentuk folder baru dengan nama yang sesuai kita perintahkan. Sampai disini aplikasi kita telah siap digunakan. Namun, agar lebih manis dan *user-friendly*, kita dapat membuat menjadi *wizard-installer* sekaligus dapat mempunyai fitur *auto-update* jika aplikasi kita mengalami perkembangan dengan menggunakan **electron-builder**.

Untuk membuat aplikasi *desktop* yang sebenarnya, tentu masih banyak yang harus kita oprek, seperti menambahkan *icon*, memasang *splash-screen*, berinteraksi dengan sistem bawaan sistem operasi, seperti *File-manager*, *Printer*, sistem notifikasi, dll. Kita juga dapat menggunakan *framework SPA (Single Page Application)*, seperti **ReactJS**, **AngularJS**, **VueJS**, dll.

3.3.2 Intelij Idea

IntelliJ IDEA merupakan IDE (Integrated Development Environment), yaitu alat pengembang terpadu dari JetBrains. Di dalamnya terdapat fasilitas untuk programmer mengembangkan program / aplikasi.

IntelliJ merupakan IDE yang dapat dijalankan pada berbagai platform, seperti Windows, OS X, dan Linux. IntelliJ dipilih karena kesesuaian dengan pengembangan berbasis mobile dikemudian hari. Bunder IntelliJ telah mencakup JRE, jadi tidak perlu instalasi Java untuk menjalankan aplikasi yang dibuat pada IntelliJ, namun bundel ini tidak termasuk JDK.

Cara menginstall IntelliJ IDEA

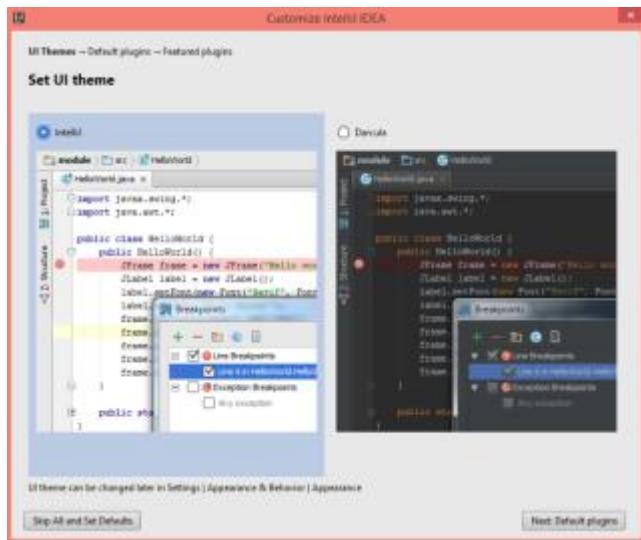
Seperti post sebelumnya, sudah di cantumkan link untuk mendownload software IntelliJ. Namun, akan saya post lagi. Silahkan download di link berikut <https://www.jetbrains.com/idea/download/index.html#section-windows>.

Minimum hardware requirement bagi instalasi IntelliJ adalah:

- RAM 1 GB, direkomendasikan 2 GB
- Hard disk space 300 MB + setidaknya 1 GB untuk cache
- Screen resolution: 1024×768

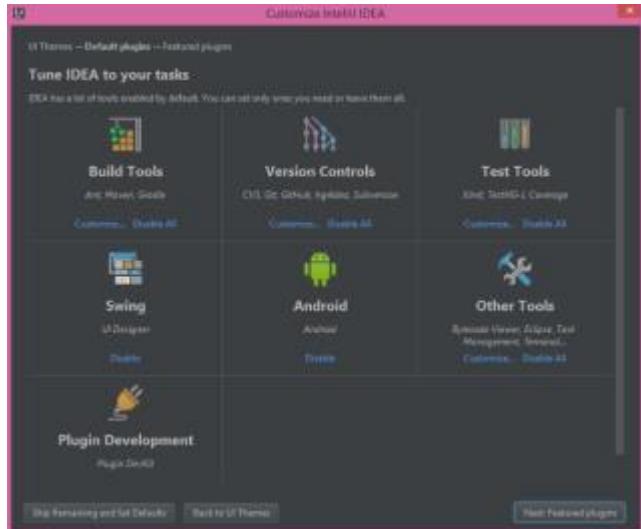
Langkah instalasi IntelliJ adalah sebagai berikut:

1. Klik installer. File idealC atau idealU-*.exe
2. Klik next, lalu continue.
3. Ikuti langkah instalasi sampai selesai.
 1. Pilih untuk membuat icon IntelliJ pada desktop.
 2. Pilih tampilan user interface (IntelliJ untuk background cerah dan Darcula untuk background gelap).



Pilihan user Interface IntelliJ

1. Pilih Plugin (enable all).



Pilihan Plug-in IntelliJ

1. Untuk menjalankan IntelliJ, klik icon IntelliJ pada desktop, atau bisa dicari pada kotak search Windows.

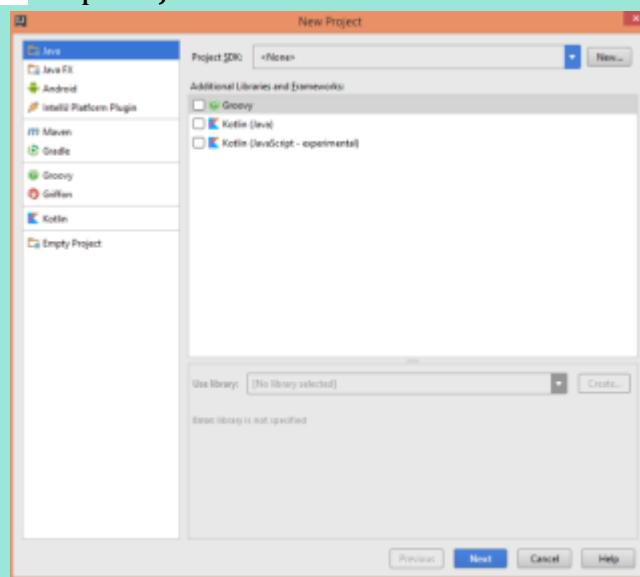
Membuat Project baru di IntelliJ IDEA

Tampilan pada saat kali pertama membuka IntelliJ



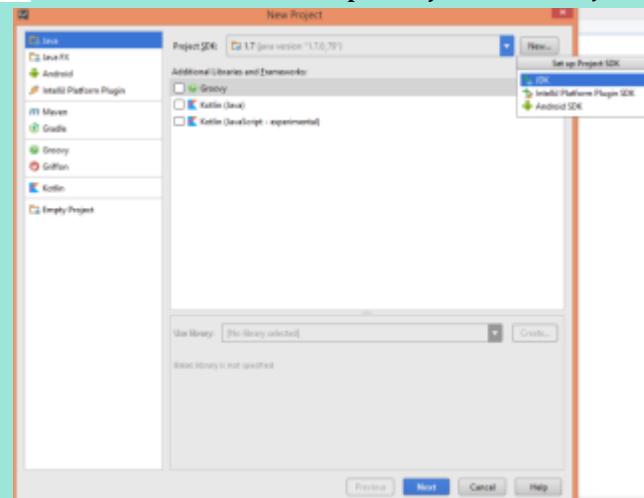
Tampilan Awal IntelliJ

1. Pilih create New Project.
1. Tampilan pada saat memulai project baru > pilih Java.



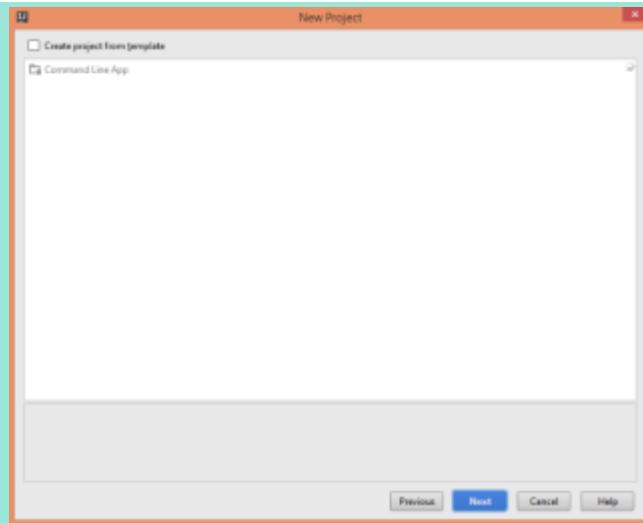
Tampilan projek baru

1. Pilih New > Set Up Project SDK > JDK.



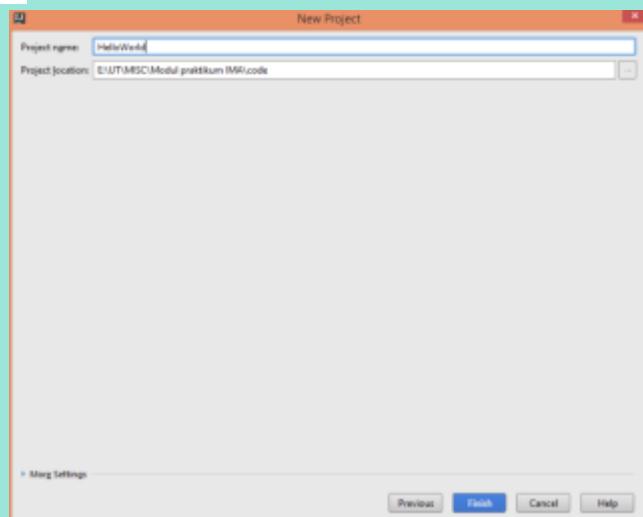
Setting JDK

1. Masukkan folder bin dari Java JDK yang sudah diinstall sebelum instalasi IntelliJ (IntelliJ tidak men-support JDK).
2. Klik Next (karena yang akan dibuat hanya project sederhana).
1. Klik Next pada bagian “Create Project from Template”.



Create Project Form Template

1. Tentukan nama dan lokasi project, klik finish.

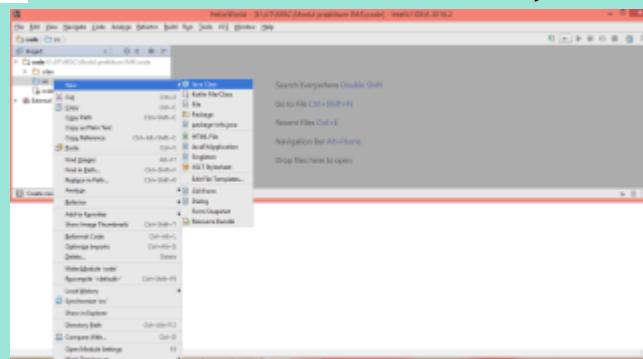


Setting Nama dan lokasi penyimpanan project

Hello World!

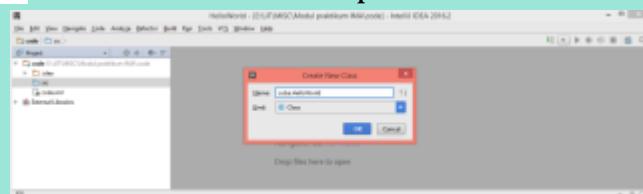
Hello World merupakan program sederhana pertama yang akan dibuat. Pada Java, program dibuat di dalam Class yang terdapat pada package. Untuk membuat program pertama, lakukan langkah berikut:

1. Pada project yang dibuat, terdapat folder src. Klik kanan folder ini > New > Java Class.



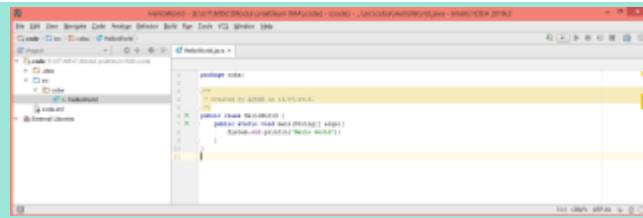
Pembuatan Class Baru

1. Beri nama Class ini, misal coba.HelloWorld. Coba merupakan nama package tempat Class HelloWorld ditempatkan.



Pemberian Nama Class

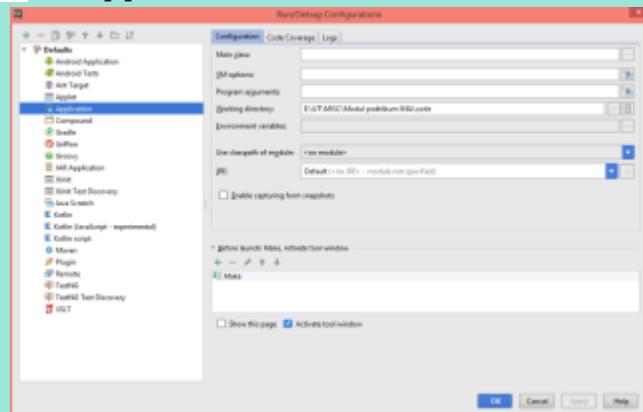
1. Ketikkan code berikut:



Penulisan Code pada jendela kerja

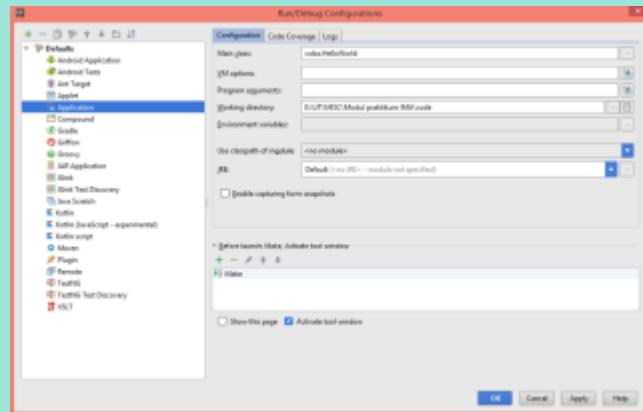
```
package coba;
public class HelloWorld {
    public static void main (String[]args){
        System.out.println("Hello World");
    }
}
```

1. Sebelum menjalankan program, terlebih dahulu harus dilakukan konfigurasi pada project yang dibuat. Kemenu Run > Edit Configuration.... Kemudian pada tampilan Run/Debug Configuration klik pada default > Application.



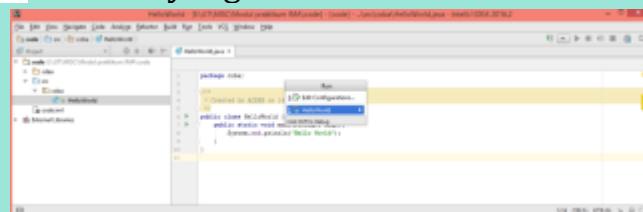
Konfigurasi Project

- Pilih main class dari project yang dibuat (klik tombol di sebelah kanan kotak Main Class).



Pemilihan Main Class

- Jalankan program dengan meng-klik kanan kelas yang berisi main method.



Running Program

- Hasil akhir akan terlihat pada console di bagian bawah jendela Intellij.



Hasil Program

3.3.3 Code Lobster PHP

CodeLobster dirancang terutama untuk menangani scripting, khususnya, PHP dan JavaScript pada sistem Windows.

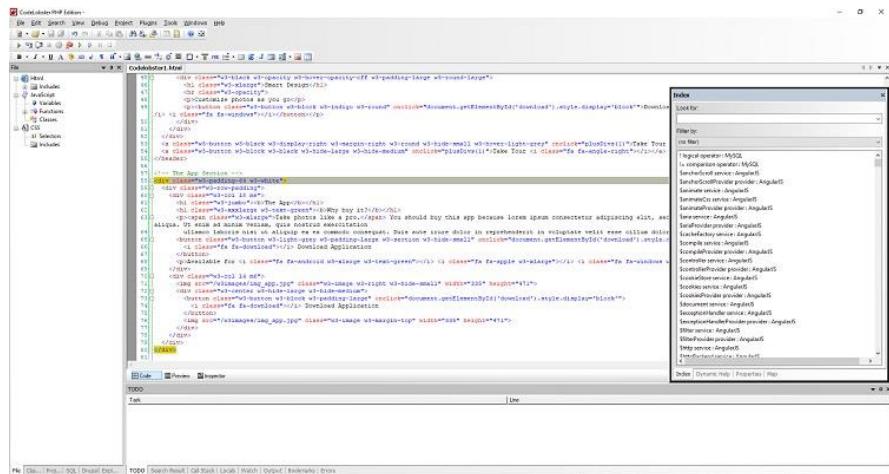
Pada intinya adalah platform kode, yang kemudian tim pengembangan membangun dinding pendukung fungsi yang solid. Di sini kita akan membahas beberapa fungsi paling dasar dan bagaimana hal ini dapat membantu.

IDE utama sekali dimuat cukup standar dan terdiri dari berbagai panel yang berhubungan dengan set informasi yang berbeda. Ini mendukung kanvas pengkodean utama yang berada di tengah. Anda dapat menyulap ini sesuai keinginan Anda dan tentu saja, Anda dapat menambahkan lebih banyak atau menyingkirkan beberapa juga.

Dengan begitu banyak fungsi tersedia, kami hanya akan memberikan daftar singkat dari mereka yang tampaknya paling berguna.

Kode Warna

Mengingat bahwa ada kesempatan di mana jenis skrip dicampur dalam pengembangan web, CodeLobster menyoroti berbagai jenis kode dengan warna mereka sendiri. Sementara ini mungkin tampak seperti tidak ada, ketika Anda sedang mengerjakan sepotong kode yang beberapa ratus baris itu sangat berguna.



CodeLobster menawarkan kode warna untuk kemudahan membaca

Bantuan

Meskipun kami yakin banyak yang merasa bahwa 'bantuan' dalam program jarang membantu, sistem bantuan CodeLobster digerakkan oleh konteks.

Ini berarti bahwa bantuan mudah tersedia untuk hal-hal yang sangat spesifik seperti fungsi, tag, atau atribut. Ini seperti memiliki kamus pemrograman inbuild dalam sistem dan dapat terbukti tak ternilai bagi mereka yang baru memasuki dunia pemrograman.

Autocomplete

Meskipun umum di ponsel cerdas dan mungkin pencarian Google, ini adalah yang pertama kami lihat di IDE. Setelah struktur kode Anda dikenali oleh CodeLobster, itu menurunkan daftar metode yang dapat Anda pilih.

Pelengkapan otomatis fungsi datang dalam daftar drop-down yang mudah digunakan

Debugger

Berguna baik untuk veteran berpengalaman maupun pemula, debugger PHP memungkinkan eksekusi skrip secara bertahap. Saat skrip berjalan Anda akan dapat melihat nilai yang tepat yang sedang dihitung dan diwariskan.

Koneksi basis data

Hampir tidak pernah ada kode yang tidak menggunakan database saat ini dan CodeLobster telah membangun koneksi untuk ini. Manajer SQL dapat melakukan hampir semua hal yang Anda perlukan dengan basis data, bahkan mengeksekusi kueri.

Pengerahan jarak jauh

Setelah kode Anda diuji dan siap, Anda dapat menggunakan dukungan FTP inbuilt untuk memindahkan kode Anda ke server web. Kemudian, perubahan lebih lanjut dapat dilakukan langsung pada server tersebut jika diperlukan.

Barang bagus lainnya

Ini termasuk penyorotan pasangan, kemungkinan pemilihan blok, runtuh, tooltips, navigasi pada deskripsi fungsi dan termasuk file pada pemotongan kunci CTRL, tampilan struktur file dan proyek, pratinjau di browser, book-mark, dan semua lainnya kemungkinan standar untuk bekerja dengan kode.

CodeLobster Mendukung di WordPress, Joomla dan Node.js

Dengan berbagai pilihan yang dihadapi pengembang web saat ini, kami akan fokus pada dua Sistem Manajemen Konten (CMS) yang sangat populer yang didukung oleh CodeLobster, yaitu WordPress dan Joomla. Yang juga menjadi fokus adalah dukungan untuk node.js yang merupakan runtime JavaScript yang dibangun di atas mesin JavaScript V8 Chrome.

1. WordPress

Bersinonim dengan istilah blogger, WordPress memungkinkan pengguna untuk fokus lebih sedikit pada kode daripada pada bagaimana mereka ingin menyajikan situs mereka. Ini gratis dan menurut W3Techs, digunakan oleh lebih dari 34% dari semua situs web.

Di sinilah sedikit menarik. Daripada sebuah plugin sederhana yang memberikan beberapa fungsi WordPress dalam IDE, CodeLobster praktis telah membangun sebuah seluruh lingkungan WordPress ke dalam plugin mereka.

Dari awal, Anda dapat menginstal versi lokal WordPress termasuk bahkan database yang akan Anda gunakan, semua menggunakan Wizard. Isi saja yang kosong.

Saat mengkodekan situs Anda, Anda akan mendapat manfaat dari fitur pelengkapan otomatis yang menawarkan penyelesaian fungsi pop-up. Tentu saja, ada juga bantuan sintaks bawaan yang sama sehingga Anda tidak perlu menjelajahi web untuk meminta bantuan.

Jika Anda tidak membangun dari awal, CodeLobster memiliki Editor Tema WordPress yang memungkinkan Anda memvisualisasikan situs Anda saat Anda mengedit kode. Ini seperti

sistem WYSIWYG yang WordPress sendiri tawarkan sebagai pratinjau.

2. Joomla

Di samping WordPress, Joomla adalah CMS populer lainnya yang memungkinkan pengguna untuk fokus pada presentasi daripada pengkodean. Tentu ini berarti bahwa itu juga merupakan hal besar bagi pengembang dan juga termasuk dalam Daftar lengkap plug-in CodeLobster.

Seperti halnya WordPress, plug-in Joomla menawarkan kepada para pengembang seluruh tipe IDE Joomla-spesifik yang memungkinkan pembuatan situs lokal instan dengan dukungan basis data.

Di mana WordPress hadir dengan widget, Joomla memiliki modul-modulnya yang IDE memungkinkan pembuatan dengan mudah. Panduan bantuan mendorong semua ini, jadi bahkan untuk pengembang ada sedikit pengkodean yang diperlukan kecuali untuk persyaratan desain khusus.

3. Node.js

Sebagai runtime JavaScript, node.js telah menjadi tidak ternilai bagi pengembang web saat ini. Ini sangat disukai karena ringan dan efisien. Mengingat seberapa baik itu diterima, ekosistem node.js adalah salah satu perpustakaan open source terbesar yang tersedia saat ini.

CodeLobster berfungsi dengan baik dengan node.js dan menawarkan autocomplete untuk kelas, pustaka, dan metode. Ini juga dilengkapi dengan bantuan dinamis yang berguna yang sangat spesifik konteksnya.

Harga dan Kompetisi

Banyak IDE premium saat ini datang tepat seperti itu; Harga premium.

Ambil sebagai contoh Microsoft Visual Studio yang dibangun untuk (tentu saja) C / C ++, VB.net, C # dan F#. Harga untuk rentang

tersebut mulai dari \$ 49 per bulan, hingga biaya tahunan dari langganan tahunan \$ 2,999.

Ada pilihan gratis yang tersedia seperti netbeans, tetapi tidak ada yang mendekati CodeLobster dalam hal jumlah fitur yang ditawarkan.

Baca juga - Berapa yang harus dibayarkan untuk hosting web

CodeLobster diberi harga berdasarkan model freemium.

Ini berarti aplikasi dasar itu gratis, tetapi lebih banyak fungsi dapat diakses untuk versi berbayar. Itu datang dalam versi gratis, LITE dan PRO. Versi gratis dilengkapi dengan editor, inspektor dan debugger.

Untuk sedikit naik ke LITE di \$ 39.95, Anda akan mendapatkan akses ke dukungan FTP / SFTP, manajer SQL, dukungan node.js, serta beberapa fungsi lainnya. Versi PRO adalah real deal dan disitulah Anda akan mendapatkan paket plug-in yang luar biasa.

Untuk \$ 99.95, CodeLobster berubah menjadi setiap IDE impian pengembang dengan semua yang disertakan. Setiap plugin tunggal yang mendukung berbagai kerangka kerja dan sistem ada di sana dan daftarnya komprehensif:

- Kaku
- Tulang punggung
- CakePHP
- CodeIgniter
- Drupal
- jQuery
- Joomla

- LARAVEL
- Meteor
- Phalcon
- Smarty
- Symfony + Twig
- WordPress
- Yii

Semua harga untuk upgrade ke CodeLobster adalah pembayaran satu kali, itu tidak didasarkan pada model berlangganan.

Kesimpulan

PROS

- Hadir dalam format portabel
- Kerangka luas dan dukungan CMS
- Peningkatan konstan

KONTRA

- Pengguna pertama kali dapat kewalahan

CodeLobster PHP Edition sebenarnya adalah IDE yang sangat kuat dan serbaguna. Meskipun mungkin sedikit terlalu kuat bagi beberapa orang, kami merasa bahwa menawarkan lebih banyak dan memberi pengguna opsi untuk hanya menggunakan apa yang Anda butuhkan jauh lebih baik daripada membuat mereka membayar setiap inci dari jalan.

Menurut Stanislav Ustimenko, Project Manager di CodeLobster, keuntungan terbesar yang ditawarkan platform mereka adalah dukungan khusus untuk banyak kerangka kerja populer dan CMS - dan kami setuju.

Kami telah melacak kemajuan CodeLobster selama beberapa waktu dan menyadari bahwa ini adalah salah satu IDE yang selalu mendapatkan pembaruan baru. Tidak hanya itu, tetapi pembaruan yang tampaknya mempertimbangkan umpan balik pengguna nyata seperti penyertaan fitur drag and drop file.

Orang-orang di CodeLobster juga telah membenarkan perasaan ini karena mereka telah memberi tahu kami bahwa kami akan melihat versi multi-platform yang keluar dalam beberapa bulan ke depan.

Dalam kata-kata penggemar dan pengembang Ruslan Kuliev, "Saya suka Codelobster PHP Edition karena ini adalah PHP gratis yang hebat, Editor HTML dan CSS. Ini memiliki semua fungsi yang dibutuhkan untuk pekerjaan saya - tips-alat, menyoroti dan

pelengkapan otomatis lanjutan. Ia bahkan memiliki autocomplete untuk SQL ”.

Fitur-Fitur Dari CodeLobster PHP Edition

Beberapa fitur utama yang ditawarkan oleh CodeLobster PHP Edition diantaranya:

Color Coding

Ada beberapa tipe script yang dikombinasikan pada pengembangan web, CodeLobster akan memberikan highlight dari tipe kode yang berbeda dengan warnanya masing-masing.

Help

Sistem Help yang ditawarkan oleh CodeLobster tersedia untuk berbagai hal spesifik seperti function, tags atau attribute. Fitur ini sangat bermanfaat bagi mereka yang sedang belajar dalam dunia pemrograman.

Autocomplete

Fitur ini pastinya sudah sering kamu gunakan pada saat menggunakan smartphone atau ketika melakukan pencarian di Google Search, dan fitur autocomplete ini juga ada di CodeLobster. Setelah tool IDE ini mengenali struktur kode yang kamu tulis, kamu akan disuguh pilihan (drop down list) dan kamu tinggal memilih kode yang kamu inginkan tanpa perlu lagi mengetiknya.

Debugger

Sangat berguna baik untuk programmer veteran atau yang masih belajar, PHP debugger memungkinkan untuk mengeksekusi script

secara bertahap. Ketika script dijalankan, kamu bisa melihat nilai pasti yang dihitung.

Koneksi Database

Hampir tidak kode saat ini yang yang tidak menggunakan database, kabar baiknya CodeLobster memiliki fitur koneksi ke database. SQL manager dapat melakukukan semua yang kamu inginkan dengan database, bahkan menjalankan query.

Remote Deployment

Setelah kode kamu telah di uji coba dan siap, kamu bisa menggunakan fitur FTP untuk memindahkan kode yang kamu buat ke web server. Dan, jika ada perubahan langsung dapat dilakukan langsung dari server.

Fitur Lainnya

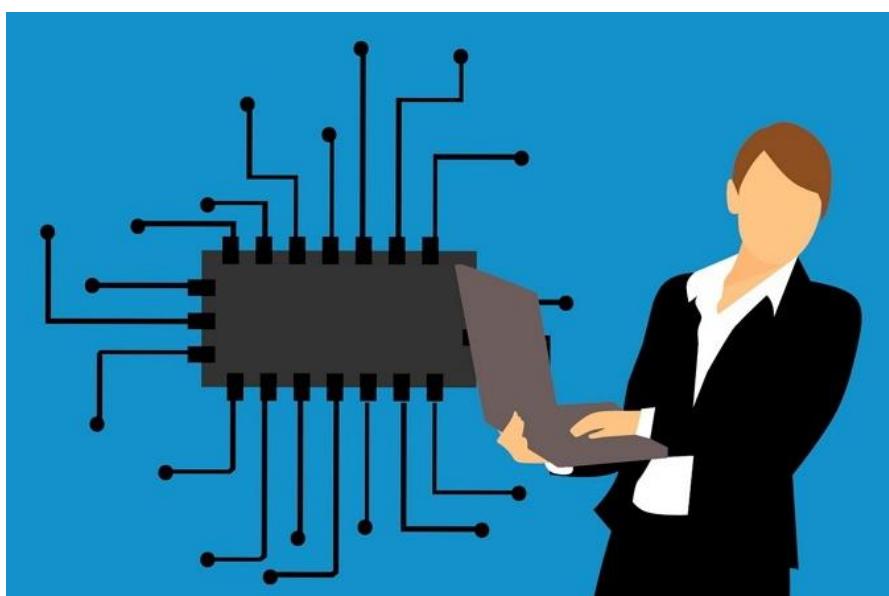
Fitur lainnya termasuk pair highlighting, block selection, collapsing, tooltips, navigation, melihat struktur file dan proyek, melihat preview code di web browser, dan bookmark.

BAB IV

FRAME WORK

4.1 Pengertian Framework

Secara sederhana, framework disebut sebagai kerangka kerja. Para developer menggunakan framework untuk memudahkan mereka dalam membuat dan mengembangkan aplikasi atau software. Pada framework itu sendiri berisi kumpulan fungsi-fungsi dasar atau perintah yang biasa digunakan dalam mengembangkan suatu software, dengan harapan agar software yang dibangun menjadi lebih cepat dan terstruktur.



Selain itu, ada juga yang mengatakan bahwa framework adalah komponen-komponen yang sudah siap dipakai oleh developer untuk menangani berbagai permasalahan dalam pemrograman, seperti pemanggilan variabel, file, koneksi ke database dan sebagainya.

Dengan begitu, developer akan lebih fokus dan mampu menyelesaikan software menjadi lebih cepat dan efektif.

Framework adalah sebuah software untuk memudahkan para programer untuk membuat sebuah aplikasi web yang di dalamnya ada berbagai fungsi diantaranya plugin, dan konsep untuk membentuk suatu sistem tertentu agar tersusun dan terstruktur dengan rapih.

4.2 Fungsi Framework

Berdasarkan pengertian framework diatas, sudah tampak bahwa framework memiliki fungsi utama untuk membantu dan memudahkan para developer dalam menyelesaikan suatu proyek pengembangan software atau aplikasi. Selain itu, ada beberapa fungsi framework yang bisa Anda ketahui dibawah ini.

1. Menghemat Waktu Pengembangan

Penggunaan framework dalam pengembangan suatu software akan mengurangi beban kerja developer, sehingga tidak ada waktu yang terbuang untuk memikirkan fungsi-fungsi umum yang akan digunakan.

Selain itu, developer akan lebih fokus pada alur cerita pada aplikasi seperti yang dibutuhkan oleh pengguna. Dengan begitu, waktu pengembangan software akan berjalan lebih cepat dan dapat

diserahkan kepada pengguna sesuai dengan waktu yang telah disepakati bersama.

2. Pemrograman menjadi Lebih Terstruktur

Ketika developer menghadapi suatu proyek pengembangan software yang besar, maka akan terdapat banyak program yang ditulis didalamnya. Terkadang, semakin banyaknya program akan menjadikan proses *debugging* semakin lambat. Selain itu, untuk mengecek kode program yang menjadi penyebab terjadinya *error* akan semakin sulit karena program tidak terstruktur dengan baik.



Oleh karena itu, penggunaan framework dapat menjadikan pemrograman menjadi lebih terstruktur sehingga Anda dapat dengan mudah menemukan bagian-bagian dari kode yang perlu diperbaiki. Apalagi, ada framework yang menerapkan konsep MVC (Model View Control) yang memudahkan developer untuk memisahkan dan

menyusun program berdasarkan bagiannya, yaitu Model, View ataupun Control.

3. Pengulangan Kode

Seperti yang sudah disinggung pada poin pertama bahwa pembuatan software membutuhkan waktu yang lama, apalagi Anda berperan sebagai programmer tunggal. Maka Anda dapat menggunakan framework untuk mengurangi beban tersebut.

Sebab, framework sudah menanamkan berbagai fungsi-fungsi umum yang bisa Anda gunakan tanpa harus mengulangi pembuatan kode dari awal. Pastinya, akan memudahkan Anda juga untuk menggunakan kembali di proyek-proyek berikutnya.

4. Meningkatkan Keamanan

Keamanan menjadi suatu hal yang sangat vital dalam pengembangan software atau aplikasi. Apalagi, software yang memuat data pengguna yang privasinya harus dilindungi. Disini, framework terus memperbarui versinya yaitu menawarkan fitur yang handal dalam menangani berbagai jenis ancaman yang menyerang sistem keamanan

4.3 Jenis-jenis Framework

Pada umumnya, terdapat dua jenis framework yaitu Desktop Framework dan Web Framework. Kedua jenis framework tersebut memiliki kegunaannya masing-masing. Desktop framework merupakan framework yang digunakan untuk membangun aplikasi berbasis desktop, sedangkan web framework merupakan framework yang digunakan untuk membangun aplikasi berbasis web. Berikut

ini akan dijelaskan mengenai contoh-contoh dari kedua jenis framework.



4.3.1 Dekstop Framework

1. WPF

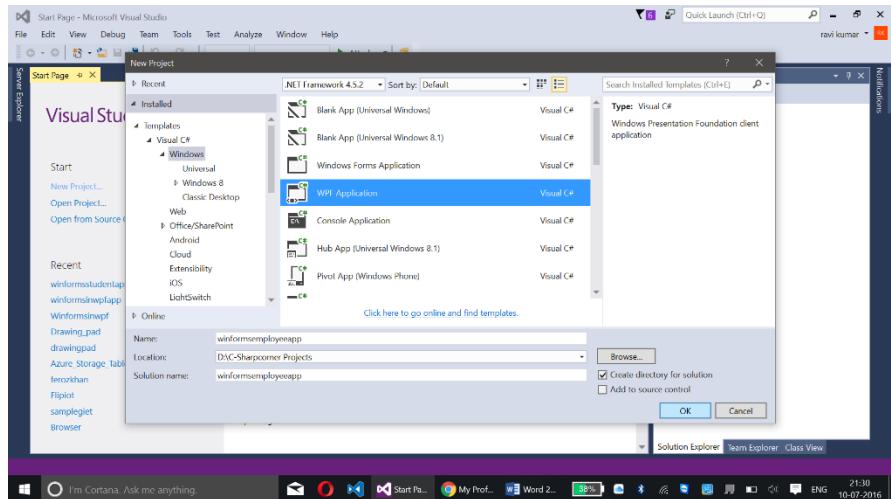


Foto: Microsoft

Ini adalah salah satu *framework* paling populer untuk pengembangan aplikasi Windows asli. *Windows Presentation Foundation* (WPF) adalah kerangka kerja dalam .NET *framework* yang terutama digunakan untuk mengembangkan grafik aplikasi desktop. Kamu bisa menggunakan untuk membuat antarmuka pengguna untuk *software*. *Library* runtime WPF biasanya tertanam dalam sistem operasi Windows. Karakteristik utama WPF adalah kemampuannya untuk menyatukan berbagai elemen antarmuka pengguna. Elemen-elemen ini termasuk grafik vektor, dokumen adaptif, objek media yang dirender sebelumnya, dan rendering 2D dan 3D.

2. Universal Windows Platform (UWP)



Foto: Technotification

UWP adalah *framework* penting lainnya untuk pengembangan aplikasi desktop. Ini juga merupakan *framework* yang sangat dikreditkan untuk popularitas platform .NET. Ini karena

memungkinkan pengembang untuk membuat aplikasi desktop lintas platform. Pada dasarnya, UWP memungkinkan pengembang untuk membuat aplikasi yang dapat berjalan di banyak platform milik Microsoft. Ini berarti perangkat lunak-mu akan bisa berjalan di beberapa perangkat. Ini dimungkinkan oleh algoritma khusus untuk pengembangan aplikasi Windows. *Framework* UWP sangat skalabel. Jika kamu telah membangun aplikasi untuk desktop, *framework* ini akan memungkinkan untuk mengukurnya untuk perangkat seluler. Aplikasi ini akan dapat berjalan dengan lancar di ponsel, tablet, dan bahkan di Xbox. Ini adalah kerangka kerja yang telah meningkatkan fungsionalitas Visual Studio.

3. Cocoa

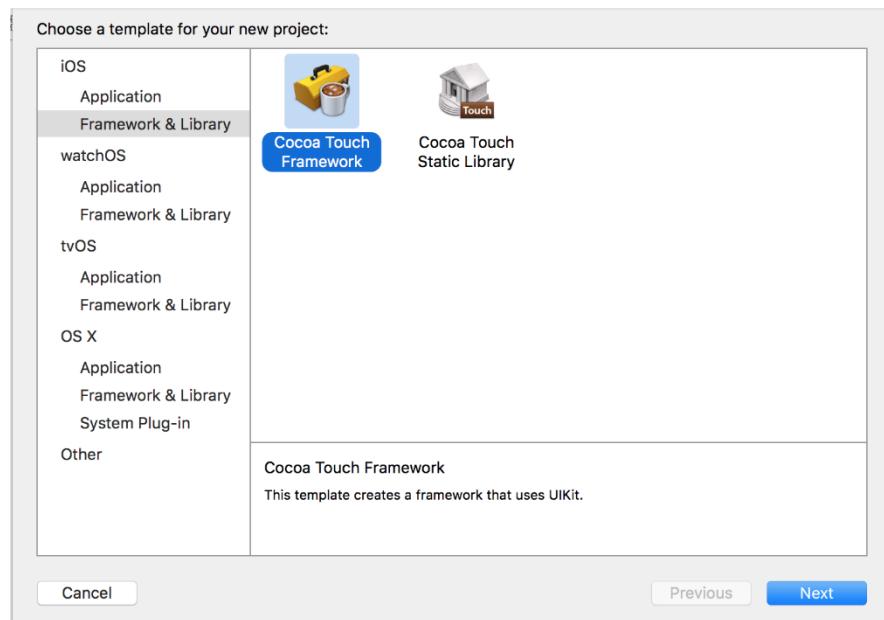


Foto: Medium

Cocoa adalah *framework* asli untuk pengembangan MacO asli. Ini adalah *framework* berorientasi objek untuk membangun antarmuka pengguna untuk MacOS, iOS, dan tvOS. Cocoa tidak hanya menambah fungsionalitas UI tetapi juga membuat antarmuka lebih menghibur. Dengan *framework* Cocoa, kamu bisa menambahkan fitur animasi, elemen kontrol grafis dan fitur pengenalan gerakan ke dalam aplikasi desktop. Semua alat pengembangan untuk Cocoa disediakan oleh Apple. Ketika datang untuk menulis kode, *framework* dapat digunakan dengan bahasa Python, Perl, dan Ruby. Untuk menggunakan bahasa-bahasa ini, kamu perlu jembatan seperti PyObjC, PasCocoa, dan RubyCocoa. *Framework* Cocoa akan membantu-mu mematuhi panduan ketat Apple.

4. Electron Js

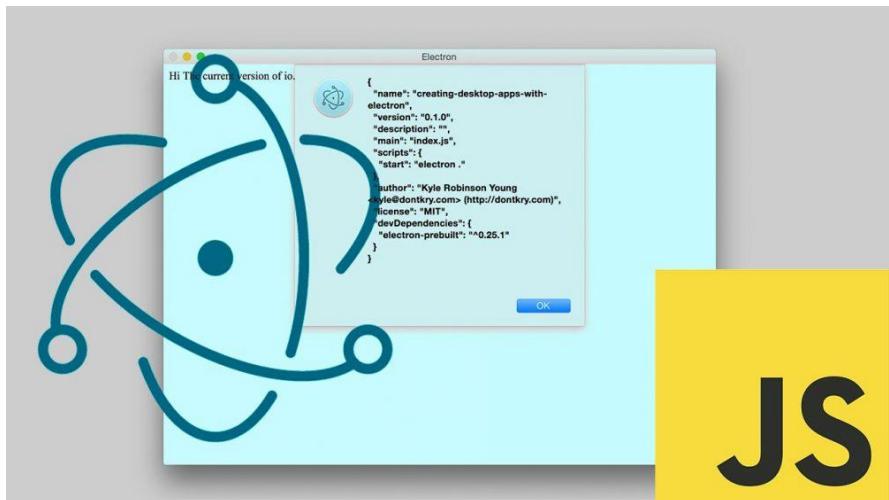


Foto: YouTube

Electron Js adalah *framework* pengembangan lintas platform yang dikembangkan oleh GitHub. *Framework* menggunakan Node.js dan

pengembang dapat menggunakannya untuk membangun aplikasi desktop lintas platform. Dari sisi tampilan, ada cukup banyak perusahaan besar yang menggunakan platform ini untuk mengembangkan aplikasi mereka. Mereka termasuk Facebook, Microsoft, dan Stack. Sebagai pengembang web, kamu bisa menggunakan Electron untuk membangun aplikasi desktop. *Framework* ini memungkinkan pengembang untuk fokus pada fungsionalitas inti dari perangkat lunak. Ini karena *framework* menangani bagian tersulit dari proses pengembangan perangkat lunak.

5. Swing

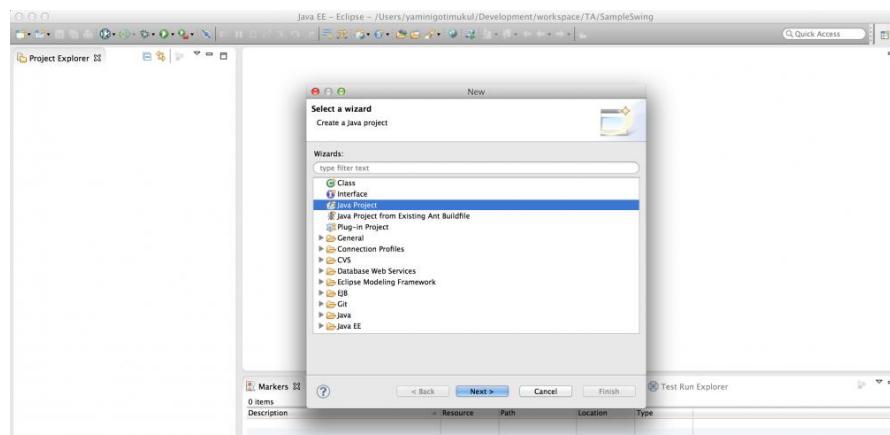


Foto: Cs.UCLA.Edu

Swing adalah *framework* berbasis Java yang dapat kamu gunakan untuk mengembangkan aplikasi desktop. Menjadi *framework* lintas platform, aplikasi yang dibangun oleh Swing dapat berjalan di platform apa pun. Fungsi utama Swing adalah untuk membangun GUI yang lebih baik

untuk aplikasi desktop. Ia mampu meniru desain, tampilan, dan gaya banyak aplikasi desktop. Kerangka kerja ini dilengkapi dengan berbagai komponen UI seperti tombol, panel, tabel panel gulir, dan kotak centang.

4.3.2 Web Framework

A. *Django*

Django merupakan suatu framework yang berbasis Python dengan menggunakan pola MTV, yaitu Model, Template dan View. Django menawarkan berbagai fitur untuk mengembangkan aplikasi web yang berkualitas, terutama bagian keamanan yang digunakan untuk mencegah terjadinya eksekusi program dilapisan Template. Web framework ini sudah banyak digunakan oleh perusahaan besar seperti Disqus, Pinterest, Instagram dan Quora.



Apa itu Django?

Django adalah sebuah *framework full-stack* untuk membuat aplikasi web dengan bahasa pemrograman Python. Framework akan membantu kita membuat web lebih cepat, dibandingkan menulis kode dari nol.¹

Full-stack artinya, django meliputi sisi *front-end* dan juga *back-end*. *Front-end* adalah sisi depan yang akan dilihat oleh pengguna, sedangkan *back-end* adalah sisi belakang yang berhubungan dengan database dan logika bisnis.

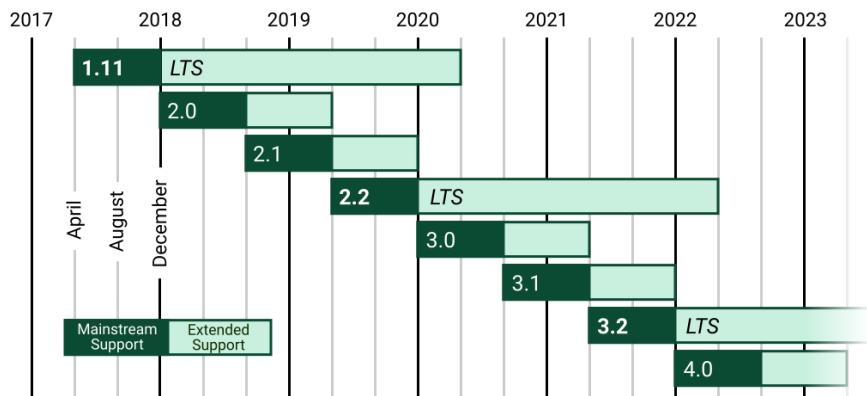
Sejarah dan Perkembangan Django

Django awalnya dikembangkan pada tahun 2003 dan 2005 oleh beberapa web developer yang bertugas membuat dan merawat web portal (*newspaper website*).

Setelah membuat beberapa website, orang-orang tersebut mulai membuat ulang kode-kode yang pernah mereka tulis dengan menerapkan beberapa *design pattern*. Lalu disebarluaskan sebagai project open source dengan nama “Django” pada bulan july 2005.

Django kemudian semakin berkembang, lalu dirilis versi 1.0 pada bulan september 2008. Sekarang (2018) Django sudah mencapai versi 2.0.

Berikut ini roadmap jadwal rilis Django:



Versi LTS akan dirilis setiap 2 tahun dan memiliki masa dukungan yang lebih panjang.

Peralatan yang Harus dipersiapkan untuk Belajar Django
Peralatan yang kita butuhkan untuk membuat project Django adalah:

1. Virtualenv;
2. pip.

Virtualenv adalah sebuah tool yang berfungsi untuk membuat lingkungan virtual yang terisolasi.

Kenapa kita harus menggunakan virtualenv?

Karena agar tidak bentrok dengan project yang lain.

Pembahasan tentang virtualenv, bisa kamu baca di: Mengenal Virtualenv: Apa Saja yang Harus Kamu Ketahui?

Lalu, pip adalah sebuah tool yang akan kita gunakan untuk manajemen paket python. Termasuk juga menginstall Django.

Install virtualenv dan pip:

```
apt install python-virtualenv python-pip python3-virtualenv  
python3-pip
```

Instalasi Django di Lingkungan OS

Instalasi Django dapat dilakukan melalui pip dengan perintah seperti ini:

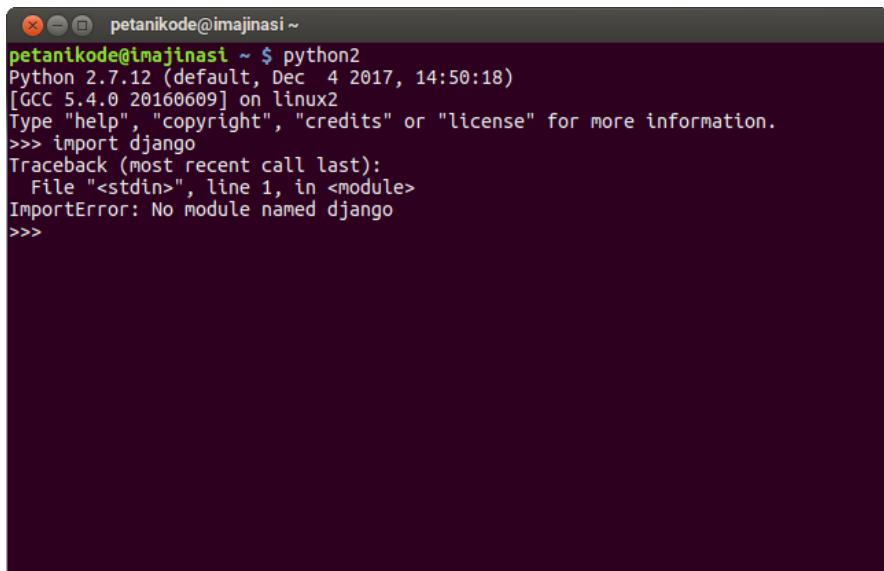
```
pip install Django==3.0.2
```

Maka Django yang akan terinstall adalah Django versi 3.0.2.

Untuk menguji django yang sudah diinstal, masuk ke *shell python*. Kemudian ketik import django dan django.VERSION:

```
>>> import django  
>>> django.VERSION  
(2, 0, 2, 'final', 0)  
>>>
```

Jika terjadi error seperti ini:



```
petanikode@imajinasi ~
petanikode@imajinasi ~ $ python2
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named django
>>>
```

Artinya Django belum terinstal dengan benar.

Coba instal ulang!

Instalasi Django di Virtualenv

Virtualenv adalah lingkungan virtual yang terisolasi dari lingkungan luar (OS).

Artinya: kita bebas menginstal apapun di dalam lingkungan virtual, tanpa harus mengganggu aplikasi lain di lingkungan luar.

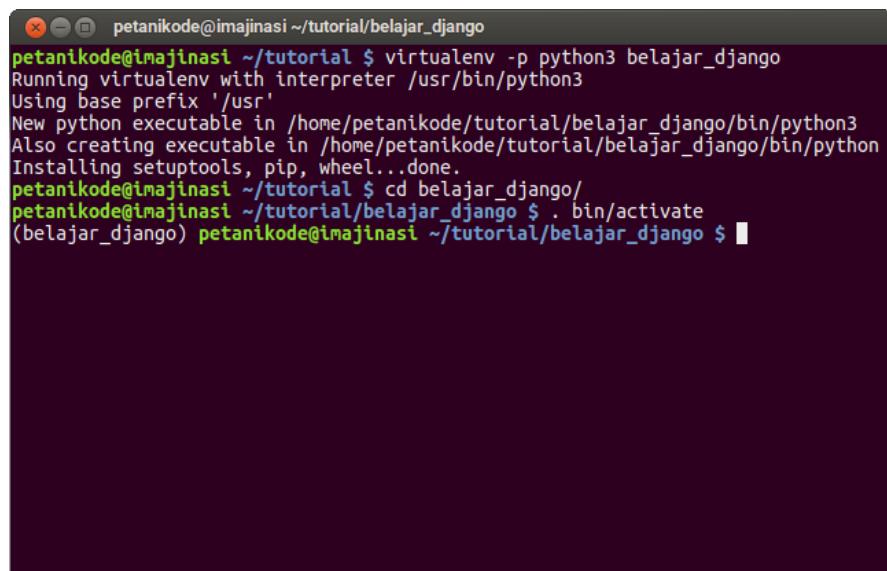
Baiklah, karena kita akan menggunakan Python 3 pada virtualenv, maka kita harus memberikan argumen -p python3 saat membuat virtualenv:

```
virtualenv -p python3 belajar_django
```

Maka sekarang kita punya direktori baru bernama belajar_django. Silahkan masuk ke direktori tersebut, kemudian aktifkan virtual enviroment-nya.

```
cd belajar_django  
. bin/activate
```

Kalau berhasil, tampilannya akan seperti ini:

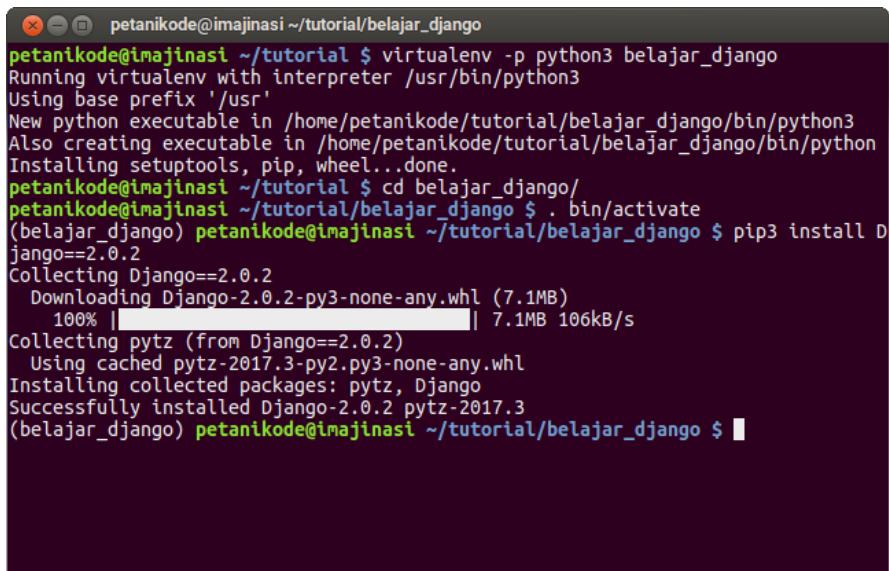


```
petanikode@imajinasi ~/tutorial/belajar_django  
petanikode@imajinasi ~/tutorial $ virtualenv -p python3 belajar_django  
Running virtualenv with interpreter /usr/bin/python3  
Using base prefix '/usr'  
New python executable in /home/petanikode/tutorial/belajar_django/bin/python3  
Also creating executable in /home/petanikode/tutorial/belajar_django/bin/python  
Installing setuptools, pip, wheel...done.  
petanikode@imajinasi ~/tutorial $ cd belajar_django/  
petanikode@imajinasi ~/tutorial/belajar_django $ . bin/activate  
(belajar_django) petanikode@imajinasi ~/tutorial/belajar_django $
```

Setelah itu, install Django dengan perintah ini:

```
pip3 install Django==3.0.2
```

Perintah pip3 adalah pip untuk Python 3.



```
petanikode@imajinasi ~/tutorial/belajar_django
petanikode@imajinasi ~/tutorial $ virtualenv -p python3 belajar_django
Running virtualenv with interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/petanikode/tutorial/belajar_django/bin/python3
Also creating executable in /home/petanikode/tutorial/belajar_django/bin/python
Installing setuptools, pip, wheel...done.
petanikode@imajinasi ~/tutorial $ cd belajar_django/
petanikode@imajinasi ~/tutorial/belajar_django $ . bin/activate
(belajar_django) petanikode@imajinasi ~/tutorial/belajar_django $ pip3 install Django==2.0.2
Collecting Django==2.0.2
  Downloading Django-2.0.2-py3-none-any.whl (7.1MB)
    100% |██████████| 7.1MB 106kB/s
Collecting pytz (from Django==2.0.2)
  Using cached pytz-2017.3-py2.py3-none-any.whl
Installing collected packages: pytz, Django
Successfully installed Django-2.0.2 pytz-2017.3
(belajar_django) petanikode@imajinasi ~/tutorial/belajar_django $
```

Membuat Project Django

Untuk membuat project aplikasi web dengan Django, kita bisa ketik perintah berikut.

```
django-admin startproject mysite
```

Keterangan:

- startproject adalah perintah untuk membuat project;
- mysite adalah nama direktori project-nya.

Setelah itu, kita akan mendapatkan direktori baru bernama mysite dengan isi sperti ini:

```
petanikode@imajinasi ~/tutorial/belajar_django
(belajar_django) petanikode@imajinasi ~/tutorial/belajar_django $ django-admin startproject mysite
(belajar_django) petanikode@imajinasi ~/tutorial/belajar_django $ tree mysite/
mysite/
└── manage.py
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py

1 directory, 5 files
(belajar_django) petanikode@imajinasi ~/tutorial/belajar_django $
```

Mengenal Stuktur Direktori Project Django

Berikut ini struktur direktori yang dibuatkan Django:

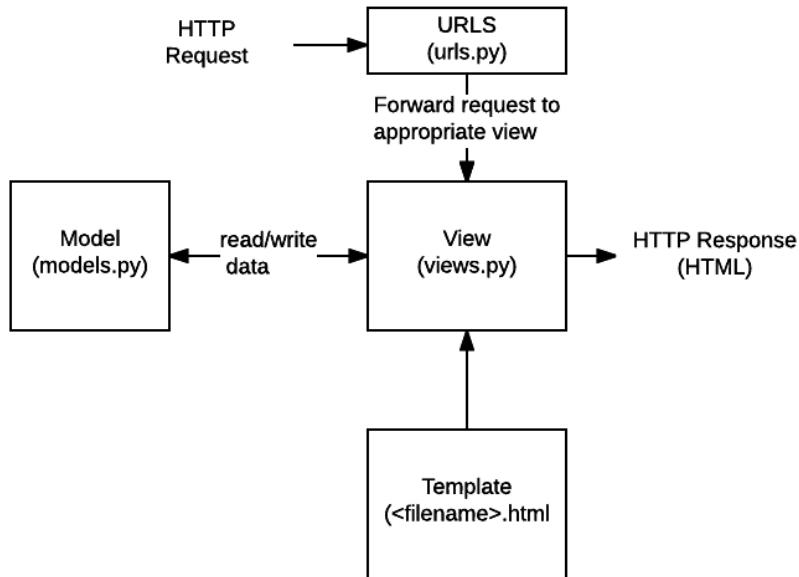
```
mysite/
└── manage.py
└── mysite
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

Ada beberapa file dan direktori yang harus kita ketahui:

- Direktori mysite/ adalah root direktori yang berisi seluruh file dari project. Nama direktori ini bisa diganti dengan apa saja, karena tidak akan jadi masalah bagi Django.

- File manage.py adalah program untuk mengelola project Django. Kita akan sering mengeksekusi manage.py saat ingin melakukan sesuatu terhadap project, misalnya: menjalankan server, melakukan migrasi, menciptakan sesuatu, dll.
- File mysite/__init__.py adalah sebuah file kosong yang menyatakan direktori ini adalah sebuah paket (*python package*).
- File mysite/settings.py adalah tempat kita mengkonfigurasi project.
- File mysite/urls.py adalah tempat kita mendeklarasikan URL.
- File mysite/wsgi.py adalah entri point untuk WSGI-compatible

Bagaimana Konsep Kerja Django?



Setiap *request* dari *client* akan diproses pertama kali oleh *urls.py*, karena di sini berisi definisi alamat URL (*route*) dan fungsi yang akan dieksekusi di setiap rute.²

Berikutnya, fungsi yang ada di *views.py* akan melakukan pemrosesan seperti:

- Tulis data atau ambil data dari *Model*
- Racik tampilan data dengan template *HTML*
- Kirim *HTTP Response* ke *client*

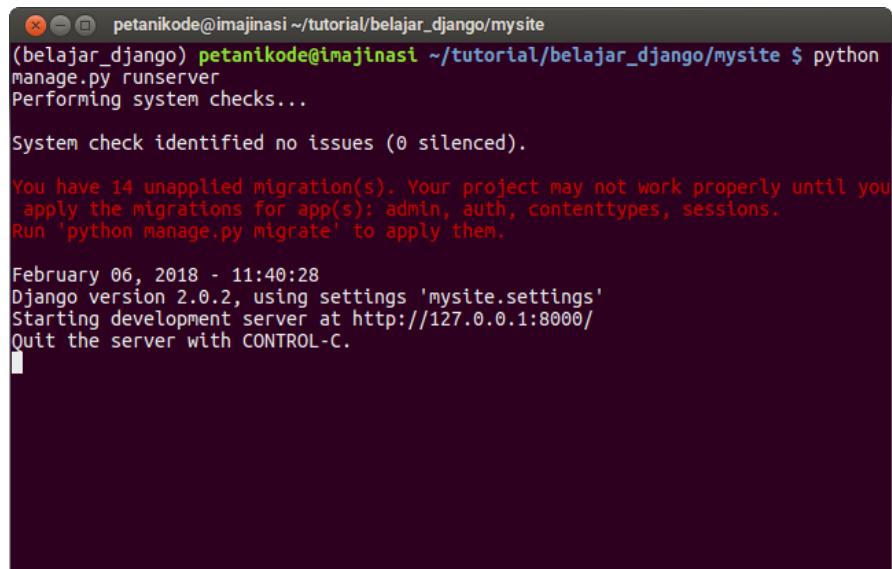
Menjalankan Server Django

Django memiliki server yang digunakan untuk *development*.

Cara menjalankannya, ketik perintah ini:

```
python manage.py runserver
```

Maka, server akan segera dijalankan...



```
petanikode@imajinasi ~/tutorial/belajar_django/mysite
(belajar_django) petanikode@imajinasi ~/tutorial/belajar_django/mysite $ python
manage.py runserver
Performing system checks...

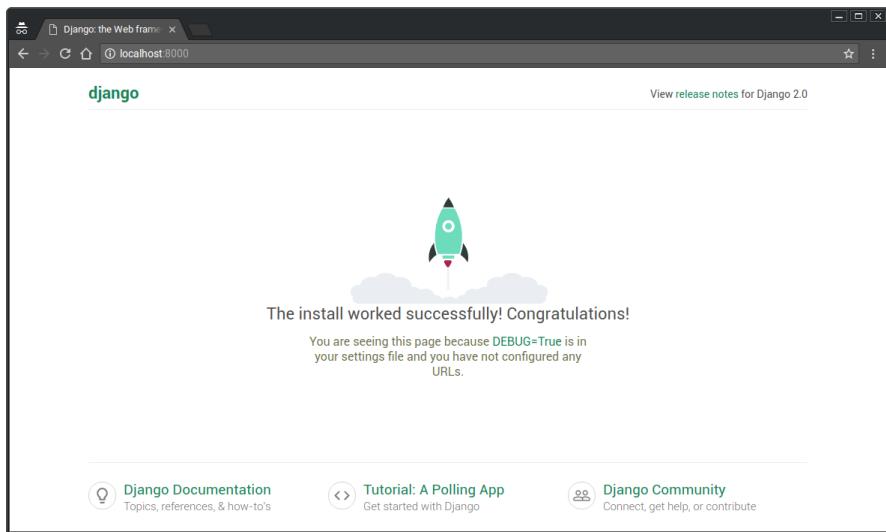
System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you
apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

February 06, 2018 - 11:40:28
Django version 2.0.2, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Abaikan teks yang berwarna merah, karena kita belum melakukan apapun di dalam project ini.

Sekarang coba buka <http://localhost:8000>, maka hasilnya:



B. Angular JS

Angular JS merupakan framework JavaScript yang sengaja dirilis oleh Google dalam mengembangkan aplikasi web. Bahkan, angular dapat membangun halaman web di sisi *client* dengan kinerja yang tinggi. Framework ini memiliki konsep MVC (Model, View, Controller). Web framework ini sudah banyak digunakan oleh perusahaan besar seperti Youtube on PS3, Weather dan Netflix.

C. Laravel

Laravel merupakan framework MVC yang dibuat oleh Taylor Otwell pada 2011 dengan menggunakan PHP. Framework ini adalah pengembangan website berbasis MVP yang dirancang untuk meningkatkan kualitas perangkat lunak, yaitu dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, meningkatkan pengalaman bekerja dengan aplikasi yang menyediakan sintaks ekspresif, jelas dan menghemat waktu. Web framework ini sudah

banyak digunakan oleh perusahaan seperti Deltanet Travel dan Neighbourhood Lender.

Love beautiful code? We do too.

The PHP Framework For Web Artisans



Bericara soal framework PHP, salah satu nama yang sudah familiar di kalangan para developer adalah Laravel. Framework ini dikenal karena memiliki syntax yang mudah dipahami dan digunakan.

Dengan Laravel, website dan aplikasi dapat dikembangkan dalam waktu singkat. Bahkan Anda tidak perlu memikirkan pekerjaan-pekerjaan dasar karena Laravel telah menyediakan fungsi, seperti otentikasi user, manajemen sesi, dan *caching*. Singkatnya, Laravel menawarkan semua fungsionalitas yang Anda butuhkan untuk membuat serta mengembangkan aplikasi PHP modern.

Ada beberapa alasan teknis mengapa kami menempatkan Laravel di urutan pertama. Dari segi performa, framework ini memiliki core yang dapat diandalkan serta dikembangkan dengan menggunakan berbagai add-ons.

Laravel, di sisi lain, terintegrasi dengan *library* dan platform pihak ketiga, seperti Amazon Web Services (AWS), sehingga Anda dapat membuat aplikasi yang mudah diskalakan. Untuk task yang membutuhkan waktu penyelesaian yang lama (biasanya lebih dari 30 detik), Anda bisa mengurutkannya agar dapat dijalankan secara *asynchronously* di *background*. Performa pun dapat ditingkatkan dengan lebih baik.

Keunggulan Utama:

- Memiliki syntax yang mudah dipahami dan dipelajari.
- Fungsionalitas intinya dapat dikembangkan dengan menggunakan add-ons.
- Menyediakan fungsi bawaan untuk mengelola *routing*, manajemen user, *caching*, dan lain sebagainya.
- Terintegrasi dengan *library* dan platform pihak ketiga, seperti AWS.
- Menjalankan task secara *asynchronously* di *background* untuk meningkatkan performa.

Laravel memiliki komunitas yang sangat aktif. Adanya komunitas ini membantu setiap usernya dalam mencari bantuan atau bahkan tutorial. Karena kemudahannya, Laravel merupakan pilihan terbaik bagi yang masih baru di dunia pengembangan web dan PHP framework.

D. Rails

Rails adalah web framework Model-View-Controller yang ditulis oleh David Heinemeier Hansson. Selain itu, Rails mencakup semua yang diperlukan untuk membuat aplikasi web berbasis database menggunakan Ruby. Web framework ini sudah banyak digunakan oleh perusahaan seperti AirBnB, GitHub, UrbanDictionary, GroupOn dan Shopify.

E. Spring

Spring adalah web framework Model-View-Controller yang dikembangkan oleh Pivotal Software. Framework ini dirancang untuk menciptakan aplikasi web berbasis JVM yang sederhana, portabel, cepat dan fleksibel. Web framework ini sudah banyak digunakan oleh perusahaan seperti Mascus dan Allocine.

F. CodeIgniter

CodeIgniter merupakan framework yang memiliki konsep MVC (Model, View, Controller) untuk membangun sebuah website yang dinamis menggunakan PHP. Dengan menggunakan codeIgniter, developer akan semakin cepat dalam membangun aplikasi web, walaupun memulainya dari awal. Web framework ini sudah banyak digunakan oleh perusahaan seperti Bufferapp dan The Mail and Guardian.

CodeIgniter Rocks

CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications.

Star 16,504 Fork 7,505 Follow 22.5K followers

CodeIgniter merupakan framework PHP yang menggunakan arsitektur Model View Controller (MVC). Dalam kalimat yang lebih sederhana, CodeIgniter menggunakan komponen yang berbeda-beda untuk mengelola task development yang spesifik. Metode yang ditawarkan CodeIgniter lebih disukai oleh para developer karena mereka bisa membuat serta mengembangkan aplikasi web yang mudah diskalakan tanpa menyita banyak ruang.

Kemudahan dalam pembelajaran dan penggunaan membuat CodeIgniter menjadi salah satu framework PHP terbaik bagi pemula. Kelebihan lainnya adalah informasi bantuan atau tutorial yang lengkap. Anda dapat memahami hal-hal dasar terkait penggunaan CodeIgniter dengan lebih mudah. Performa CodeIgniter pun dapat diandalkan, terlebih jika Anda ingin membuat aplikasi yang ‘ringan’ yang dijalankan di server sederhana.

Keunggulan Utama:

- Framework ringan dengan performa yang dapat diandalkan.
- Mudah dipahami dan digunakan berkat adanya informasi bantuan dan tutorial.

- Membuat aplikasi yang mudah diskalakan dengan menggunakan arsitektur framework berbasis MVC.

Di samping kelebihan, juga ada kekurangan. Sebagai contoh, tidak ada jadwal yang tetap dan teratur mengenai update CodeIgniter. Jadi, PHP framework ini tidak kami sarankan untuk digunakan bilamana aplikasi yang Anda kembangkan memiliki standar keamanan tinggi.

4.4 G
4.5 G
4.6 K

BAB V

TAHAPAN PERANCANGAN APLIKASI

5.1 Mencari Ide Aplikasi

Dalam tahapan ini, kita harus menemukan ide untuk membuat sebuah aplikasi. Proses dari tahapan ini dimulai dari mencari permasalahan yang ada dan apa solusi dari masalah tersebut. Dari permasalahan tersebut kita bisa mengetahui tujuan dari pembuatan aplikasi dan fungsinya.

5.2 Melakukan Analisis

Pada tahapan analisis dapat dilakukan dengan dokumentasi atau wawancara. Dari dokumentasi atau wawancara tersebut terkumpulah data yang baru baik dalam bentuk dokumen maupun rekaman. Tahapan ini juga bertujuan mengetahui proses bisnis yang berlangsung.

Setelah melakukan analisis dokumentasi atau wawancara, kita dapat menentukan masalah apa saja. Setelah permasalahan ditemukan, langkah selanjutnya adalah menentukan tujuan yang bisa didapatkan, mengetahui sistem yang akan dibangun, serta menganalisis kebutuhan data.

Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem. Tahap ini merupakan tahap yang kritis dan sangat penting, karena kesalahan dalam tahap ini menyebabkan kesalahan pada tahap selanjutnya. Misalnya anda dihadapkan pada suatu sistem untuk menentukan seberapa jauh sistem tersebut telah mencapai

sasarannya. Jika sistem mempunyai beberapa kelemahan, anda harus dapat menemukannya. Tugas ini yang disebut sebagai analisis sistem.

Tugas utama dari menganalisis sistem meliputi :

- ❑ Menentukan lingkup sistem
- ❑ Mengumpulkan fakta
- ❑ Menganalisis fakta
- ❑ Mengkomunikasikan temuan-temuan tersebut melalui laporan analisis sistem

Fakta merupakan bagian dari informasi yang menunjukkan realita, situasi dan relasi yang menjamin analisis dan pemodelan.

LANGKAH-LANGKAH DI DALAM ANALISIS SISTEM

Langkah-langkah dasar yang harus dilakukan oleh analis sistem adalah :

1. Identify, mengidentifikasi masalah
2. Understand, memahami kerja sistem yang ada
3. Analyze, menganalisis sistem Analisis dan Perancangan Sistem

Minggu Ke 2-3 Halaman 2

4. Report, membuat laporan hasil analisis

Untuk masing-masing langkah ini, beberapa tugas perlu dilakukan oleh analis sistem. Supaya memudahkan untuk melakukan koordinasi dan pengawasan, koordinator tim analis dapat membuat suatu kertas kerja

yang memuat tugas-tugas yang harus dikerjakan untuk masing-masing langkah analisis sistem ini.

3. MENGIDENTIFIKASI MASALAH Mengidentifikasi (mengenal) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai. Oleh karena itu langkah pertama yang harus dilakukan oleh analis sistem adalah mengidentifikasi terlebih dahulu masalahmasalah yang terjadi. Tugas yang harus dilakukan analis sistem adalah :

- ❑ Mengidentifikasi penyebab masalah
- ❑ Mengidentifikasi titik keputusan
- ❑ Mengidentifikasi personil-personil kunci

3.1. Mengidentifikasi Penyebab Masalah Analis sistem harus mempunyai pengetahuan yang cukup tentang aplikasi yang sedang dianalisisnya. Untuk aplikasi bisnis, analis sistem perlu mempunyai pengetahuan tentang sistem bisnis yang diterapkan di organisasi, sehingga dapat mengidentifikasi penyebab terjadinya masalah ini. Tugas mengidentifikasi penyebab masalah dimulai dengan mengkaji ulang terlebih dahulu subyek permasalahan yang telah diutarakan oleh manajemen atau yang telah ditemukan oleh analis sistem di tahap perencanaan sistem. Analisis dan Perancangan Sistem Minggu Ke 2-3 Halaman 3

3.2. Mengidentifikasi Titik Keputusan Setelah penyebab terjadinya masalah dapat diidentifikasi, selanjutnya juga harus

mengidentifikasi titik keputusan penyebab masalah tersebut. Titik keputusan menunjukkan suatu kondisi yang menyebabkan sesuatu terjadi. Analis sistem bila telah dapat mengidentifikasi terlebih dahulu titik-titik keputusan penyebab masalah, maka dapat memulai penelitiannya di titik-titik keputusan tersebut. Sebagai dasar identifikasi titik-titik keputusan ini, dapat digunakan dokumen paperwork flow atau form flowchart bila dokumentasi ini dimiliki oleh perusahaan.

3.3. Mengidentifikasi Personil-personil Kunci Setelah titik-titik keputusan penyebab masalah dapat diidentifikasi beserta lokasi terjadinya, maka selanjutnya yang perlu diidentifikasi adalah personil-personil kunci baik yang langsung maupun yang tidak langsung dapat menyebabkan terjadinya masalah tersebut. Identifikasi personil-personil kunci ini dapat dilakukan dengan mengacu pada bagan alir dokumen perusahaan serta dokumen deskripsi kerja (job description).

4. MEMAHAMI KERJA SISTEM

Langkah ini dapat dilakukan dengan mempelajari secara terinci bagaimana sistem yang ada beroperasi. Diperlukan data yang dapat diperoleh dengan cara melakukan penelitian. Bila di tahap perencanaan sudah pernah diadakan penelitian, sifatnya masih penelitian pendahuluan (preliminary survey). Sedangkan pada tahap analisis sistem, penelitiannya bersifat penelitian terinci (detailed survey).

Analisis sistem perlu mempelajari apa dan bagaimana operasi dari sistem yang ada sebelum mencoba untuk menganalisis permasalahan, kelemahan dan kebutuhan pemakai sistem untuk dapat memberikan rekomendasi pemecahannya. Sejumlah data perlu dikumpulkan, dengan

menggunakan teknik pengumpulan data yang ada, yaitu wawancara, oberservasi, daftar pertanyaan dan pengambilan sampel.

Tugas yang perlu dilakukan di langkah ini adalah :

❑ Menentukan jenis penelitian Analisis dan Perancangan Sistem
Minggu Ke 2-3 Halaman 4

❑ Merencanakan jadual penelitian o Mengatur jadual wawancara
o Mengatur jadual observasi o Mengatur jadual pengambilan sampel

❑ Membuat penugasan penelitian ❑ Membuat agenda wawancara

❑ Mengumpulkan hasil penelitian

4.1. Menentukan Jenis Penelitian Jenis penelitian perlu ditentukan untuk masing-masing titik keputusan yang akan diteliti. Jenis penelitian tergantung dari jenis data yang diperoleh, dapat berupa data tentang operasi sistem, data tentang perlengkapan sistem, pengendalian sistem, atau I/O yang digunakan oleh sistem.

4.2. Merencanakan Jadual Penelitian Supaya penelitian dapat dilakukan secara efisien dan efektif, maka jadual penelitian harus direncanakan terlebih dahulu yang meliputi : o Dimana penelitian akan dilakukan o Apa dan siapa yang akan diteliti o Siapa yang akan meneliti o Kapan penelitian dilakukan Dari rencana jadual ini, berikutnya ditentukan ke dalam jenis penelitiannya masingmasing.

4.3. Membuat Penugasan Penelitian Setelah rencana jadual penelitian dibuat, maka tugas dilanjutkan dengan menentukan tugas dari masing-masing anggota tim analis sistem, yang ditentukan oleh koordinator analis sistem melalui surat penugasan dengan menyertakan

lampiran kegiatan penelitian yang harus dilakukan. Analisis dan Perancangan Sistem Minggu Ke 2-3 Halaman 5

4.4. Membuat Agenda Wawancara Sebelum wawancara dilakukan, waktu dan materi wawancara perlu didiskusikan. Rencana ini dapat ditulis di agenda wawancara dan dibawa selama wawancara berlangsung. Tujuannya adalah supaya wawancara dapat diselesaikan tepat pada waktunya dan tidak ada materi yang terlewatkan.

4.5. Mengumpulkan Hasil Penelitian Fakta atau data yang diperoleh dari hasil penelitian harus dikumpulkan sebagai suatu dokumentasi sistem lama, yaitu : 1. Waktu untuk melakukan suatu kegiatan 2. Kesalahan melakukan kegiatan di sistem yang lama 3. Pengambilan sampel 4. Formulir dan laporan yang dihasilkan oleh sistem lama 5. Elemen-elemen data 6. Teknologi yang digunakan di sistem lama 7. Kebutuhan informasi pemakai sistem / manajemen

5. MENGANALISIS HASIL PENELITIAN

Langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan.

5.1. Menganalisis Kelemahan Sistem

Penelitian dilakukan untuk menjawab pertanyaan :

% apa yang dikerjakan ?

% bagaimana mengerjakannya ?

% siapa yang mengerjakan ?

% dimana dikerjakan ? Menganalisis kelemahan sistem sebaliknya dilakukan untuk menjawab pertanyaan :

% mengapa dikerjakan ?

% perlukah dikerjakan ?

% apakah telah dikerjakan dengan baik ?

Sasaran yang diinginkan oleh sistem yang baru ditentukan oleh kriteria penilaian sebagai berikut : relevance, capacity, efficiency, timeliness, accessibility, flexibility, accuracy, reliability, security, economy, simplicity.

Berdasarkan pertanyaan dan kriteria ini, selanjutnya analis sistem akan dapat melakukan analis dari hasil penelitian dengan baik untuk menemukan kelemahan dan permasalahan dari sistem yang ada.

6. MEMBUAT LAPORAN HASIL ANALISIS Laporan hasil analisis diserahkan ke Panitia Pengarah (Steering Committee) yang nantinya akan diteruskan ke manajemen. Pihak manajemen bersama-sama dengan panitia pengarah dan pemakai sistem akan mempelajari temuan-temuan dan analis yang telah dilakukan oleh analis sistem yang disajikan dalam laporan ini. Tujuan utama dari penyerahan laporan ini kepada manajemen adalah :

□ Analisis telah selesai dilakukan

□ Meluruskan kesalah-pengertian mengenai apa yang telah ditemukan dan dianalisis oleh analis sistem tetapi tidak sesuai menurut manajemen

□ Meminta pendapat dan saran dari pihak manajemen

□ Meminta persetujuan kepada pihak manajemen untuk melakukan tindakan selanjutnya (dapat berupa meneruskan ke tahap

disain sistem atau menghentikan proyek bila dipandang tidak layak lagi) Semua hasil yang didapat dari penelitian perlu dilampirkan pada laporan hasil analisis ini, sehingga manajemen dan user dapat memeriksa kembali kebenaran data yang telah diperoleh.

5.3 Membuat Perancangan

Perancangan adalah penggambaran atau pembuatan sketsa dari beberapa komponen yang terpisah ke dalam satu kesatuan dan berfungsi. Perancangan sistem dapat dirancang dalam bentuk aliran data atau proses bisnis.

Berikut ini adalah beberapa pengertian perancangan sistem menurut para ahli.

1. Verzello / John Reuter III

Perancangan adalah tahap setelah analisis dari siklus pengembang sistem : Pendefinisan dari kebutuhankebutuhan fungsional dan persiapan untuk rancang bangun implementasi : “ Mengembarkan bagaimana suatu sistem dibentuk”

2. John Buch & Gary Grudnitski

Perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

3. George M. Scott

Perancangan adalah menentukan bagaimana sistem akan menyelesaikan apa yang mesti diselesaikan ; tahap ini menyangkut mengkonfigurasikan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem.

Pada saat membuat sistem membuat sebuah sistem yang akan digunakan pada suatu perusahaan, setiap pengembang aplikasi diharuskan membuat sebuah rancangan dari sistem yang ingin dibuat. Rancangan ini bertujuan untuk memberi gambaran umum dari sistem yang akan berjalan nantinya kepada setiap stakeholder.

Terdapat pula teori-teori mengenai pengertian perancangan sistem menurut para ahli diantaranya adalah :

- a) Menurut Satzinger, Jackson dan Burd (2012 : 5)

perancangan sistem adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Hal itu bertujuan untuk menghasilkan produk perangkat lunak yang sesuai dengan kebutuhan user.

- b) Menurut Kenneth dan Jane (2006 : G12)

perancangan sistem adalah kegiatan merancang dan menentukan cara mengolah sistem informasi dari hasil analisa sistem sehingga sistem tersebut sesuai dengan requirement

c) Menurut O'Brien dan Marakas (2009 : 639)

perancangan sistem adalah sebuah kegiatan merancang dan menentukan cara mengolah sistem informasi dari hasil analisa sistem sehingga dapat memenuhi kebutuhan dari pengguna termasuk diantaranya perancangan user interface, data dan aktivitas proses. Dari beberapa teori-teori diatas dapat disimpulkan bahwa perancangan sistem adalah proses perancangan untuk merancang suatu sistem baru atau memperbaiki suatu sistem yang telah ada sehingga sistem tersebut menjadi lebih baik dan biasanya proses ini terdiri dari proses merancang input, output dan file.

Tujuan Perancangan Sistem

Adapun tujuan yang hendak dicapai dari tahap perancangan system mempunyai maksud atau tujuan utama, yaitu sebagai berikut:

1. Untuk memenuhi kebutuhan pemakaian sistem (*user*)
2. Untuk memberikan gambaran yang jelas dan menghasilkan rancangan bangun yang lengkap kepada pemograman komputer dan ahli-ahli teknik lainnya yang terlibat dalam pengembangan atau pembuatan sistem.

Use Case Diagram

Use case diagram adalah spesifikasi dari serangkaian tindakan yang dilakukan oleh sistem. Hubungan antara aktor dan kasus penggunaan, antara kasus penggunaan (Sertakan dan Perpanjang) dan antara actor diwakili oleh hubungan [5].

Pada gambar 3.1 berikut contoh simbol-simbol use case diagram.

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan [3]. Pada gambar 3.8 berikut contoh simbol-simbol Class Diagram

Daftar Simbol

NO	GAMBAR	NAMA	KETERANGAN
1	—	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2	◇	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5	<----->	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6	----->	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7	—	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek[3]. Berikut ini adalah gambar 3.9 simbol-simbol Sequence Diagram

a. An Actor



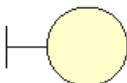
Menggambarkan orang yang sedang berinteraksi dengan sistem

b. Entity Class



Menggambarkan hubungan kegiatan yang akan dilakukan

c. Boundary Class



Menggambarkan sebuah penggambaran dari form

d. Control Class



Menggambarkan penghubung antara boundary dengan tabel

e. A focus Of Control & A life line



Menggambarkan tempat mulai dan berakhirnya sebuah message

f. A message



Menggambarkan Pengiriman Pesan

Collaboration Diagram ini bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (message)[6]. Berikut ini adalah gambar 2.4 simbol-simbol pada Collaboration Diagram

Simbol Collaboration Diagram

Simbol	Keterangan
	Actor
	Object instance : Obyek yang dibuat, melakukan tindakan, dan / atau dimusnahkan selama lifeline
	Interaksi link : Merupakan indikasi bahwa obyek kejadian dan berkolaborasi aktor dan pertukaran pesan.

Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis[3]. Diagram ini bersifat dinamis. Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek[6]. Gambar ?? adalah simbol-simbol pada Activity Diagram berikut.

Simbol	Keterangan
	Titik awal atau permulaan
	Titik akhir atau akhir dari aktivitas
	Aktivitas yang dilakukan oleh aktor
	Decision atau pilihan untuk mengambil keputusan
	Arah tanda panah alur Proses

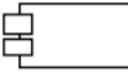
Statechart Diagram

Diagram ini bersifat dinamis. Diagram ini memperlihatkan state – state pada sistem, memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem – sistem yang reaktif[6]. Berikut ini gambar 3.12 merupakan simbol-simbol dari Statechart Diagram.

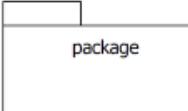
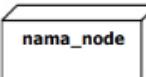
NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

Component Diagram

Diagram ini bersifat statis. Diagram ini memperlihatkan organisasi serta kebergantungan pada komponen - komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelaskelas, antarmuka – antarmuka serta kolaborasi – kolaborasi[6]. Pada gambar 3.3 ini adalah contoh simbol - simbol component diagram.

no	Simbol	Nama simbol	deskripsi
1		Package	Package merupakan simbol bungkus dari satu atau lebih komponen
2		Komponen	Komponen sistem
3		Dependency / kebergantungan	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4		Interface / antar muka	Sama dengan konsep interface pada pemrograman berorientasi objek, yaitu sebagai antar muka komponen agar tidak mengakses komponen langsung
5		Link	Relasi antar komponen

Deployment Diagram Deployment Diagram digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur sistem[3]. Diagram ini bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (saat run time). Dengan ini memuat simpul – simpul (node) beserta komponen – komponen yang ada di dalamnya. Deployment diagram berhubungan erat dengan diagram komponen dimana deployment diagram memuat satu atau lebih komponen – komponen. Diagram ini sangat berguna saat aplikasi berlaku sebagai aplikasi yang dijalankan pada banyak mesin (distributed computing)[6]. Pada gambar 3.6 ini adalah contoh simbol - simbol Deployment Diagram.

Simbol	Deskripsi
Package 	package merupakan sebuah bungkus dari satu atau lebih node
Node 	biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>). jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
Kebergantungan / dependency 	Kebergantungan antar node, arah panah mengarah pada node yang dipakai
Link 	relasi antar node

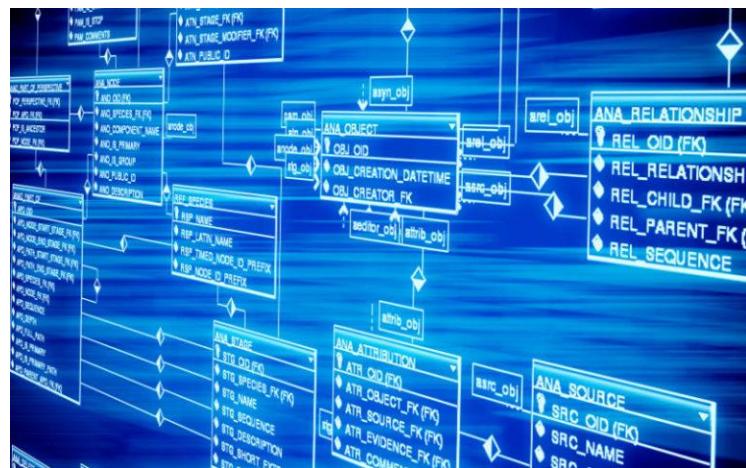
Object Diagram

Diagram objek bersifat statis. Diagram ini memperlihatkan objek-objek serta relasi antar objek. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas[6].

Simbol	Deskripsi
Objek - <div style="border: 1px solid black; padding: 5px; width: fit-content;"> nama_objek : nama_kelas atribut = nilai </div>	objek dari kelas yang berjalan saat sistem dijalankan
Link	relasi antar objek

5.4 Membuat Data Base

Merupakan tahap pembuatan database sesuai dengan kebutuhan aplikasi yang akan dibuat dengan menggunakan aplikasi tertentu seperti Microsoft Access, Visdata, Foxpro dan lain sebagainya. Tahap ini bersifat optional, artinya tidak semua perancangan program melewati tahapan ini, tahap ini hanya dikerjakan bila aplikasi yang akan dibuat memerlukan hubungan dengan database.



Tujuan Perancangan *Database*

- Untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan *user* secara khusus dan aplikasi – aplikasinya.
 - Memudahkan pengertian struktur informasi.
 - Mendukung kebutuhan-kebutuhan pemrosesan dan beberapa obyek penampilan (*response time, processing time, dan storeage space*).

Proses Perancangan *Database*

Proses perancangan *database* terdiri dari 6 tahap:

- Tahap 1, Pengumpulan data dan analisis
- Tahap 2, Perancangan database secara konseptual
- Tahap 3, Pemilihan DBMS
- Tahap 4, Perancangan *database* secara logika (*data model mapping*)
- Tahap 5, Perancangan *database* secara fisik
- Tahap 6, Implementasi Sistem *database*

Secara khusus proses perancangan berisi 2 aktifitas paralel:

1. Aktifitas yang melibatkan perancangan dari isi data dan struktur *database*,
2. Aktifitas mengenai perancangan pemrosesan *database* dan aplikasi-aplikasi perangkat lunak.

Di lain pihak, kita biasanya menentukan perancangan aplikasi-aplikasi *database* dengan mengarah kepada konstruksi skema *database* yang telah ditentukan selama aktifitas yang pertama.

6 tahapan diatas tadi tidak harus diproses berurutan. Pada tahap ke 1 merupakan kumpulan informasi yang berhubungan dengan penggunaan database. Tahap 6 merupakan implementasi *database*-nya.

Tahap 1 dan 6 kadang-kadang bukan merupakan bagian dari perancangan *database*. Sedangkan yang merupakan inti dari proses perancangan *database* adalah pada tahap 2, 4, 5.

- Tahap 1 – Pengumpulan data dan analisa
Merupakan suatu tahap dimana kita melakukan proses indentifikasi dan analisa kebutuhan-kebutuhan data dan ini

disebut pengumpulan data dan analisa. Untuk menentukan kebutuhan-kebutuhan suatu sistem *database*, kita harus mengenal terlebih dahulu bagian-bagian lain dari sistem informasi yang akan berinteraksi dengan sistem *database*, termasuk para *user* yang ada dan para *user* yang baru beserta aplikasi-aplikasinya. Kebutuhan-kebutuhan dari para *user* dan aplikasi-aplikasi inilah yang kemudian dikumpulkan dan dianalisa. Berikut ini adalah aktifitas-aktifitas pengumpulan data dan analisa:

1. Menentukan kelompok pemakai dan bidang-bidang aplikasinya
 2. Peninjauan dokumentasi yang ada
 3. Analisa lingkungan operasi dan pemrosesan data
 4. Daftar pertanyaan dan wawancara
- Tahap 2, Perancangan *database* secara konseptual Pada tahap ini akan dihasilkan *conceptual schema* untuk *database* yang tergantung pada sebuah DBMS yang spesifik. Sering menggunakan sebuah *high-level data model* seperti ER/EER modelselama tahap ini. Dalam *conceptual schema*, kita harus merinci aplikasi-aplikasi *database* yang diketahui dan transaksi-transaksi yang mungkin.Tahap perancangan *database* secara konseptual mempunyai 2 aktifitas pararel:
 1. Perancangan skema konseptual Menguji kebutuhan-kebutuhan data dari suatu *database* yang merupakan hasil dari tahap 1 dan

menghasilkan sebuah *conceptual database schema* pada DBMS-*independent* model data tingkat tinggi seperti EER (*Enhanced Entity Relationship*) model.Untuk menghasilkan skema tersebut dapat dihasilkan dengan penggabungan bermacam-macam kebutuhan user dan secara langsung membuat skema database atau dengan merancang skema-skema yang terpisah dari kebutuhan tiap-tiap user dan kemudian menggabungkan skema-skema tersebut. Model data yang digunakan pada perancangan skema konseptual adalah DBMS-*independent* dan langkah selanjutnya adalah memilih DBMS untuk melakukan rancangan tersebut.

2. Perancangan transaksi

Menguji aplikasi-aplikasi *databasedimana* kebutuhannya telah dianalisa pada fase 1, dan menghasilkan perincian transaksi-transaksi ini.Kegunaan tahap ini yang diproses secara paralel bersama tahap perancangan skema konseptual adalah untuk merancang karakteristik dari transaksi-transaksi *database* yang telah diketahui pada suatu DBMS-*independent*. Transaksi-transaksi ini akan digunakan untuk memproses dan memanipulasi *database* suatu saat dimana *database* tersebut dilaksanakan.

- Tahap 3, Pemilihan DBMS
Pemilihan *databaseditentukan* oleh beberapa faktor diantaranya faktor teknik, ekonomi, dan politik organisasi.Contoh faktor teknik:

Keberadaan DBMS dalam menjalankan tugasnya seperti jenis-jenis DBMS (*relational*, *network*, *hierarchical*, dan lain-lain), struktur penyimpanan, dan jalur akses yang mendukung DBMS, pemakai, dan lain-lain. Faktor-faktor ekonomi dan organisasi yang mempengaruhi satu sama lain dalam pemilihan DBMS :

1. Struktur data
Jika data yang disimpan dalam *database* mengikuti struktur hirarki, maka suatu jenis hirarki dari DBMS harus dipikirkan.
 2. Personal yang telah terbiasa dengan suatu sistem
Jika staf *programmer* dalam suatu organisasi sudah terbiasa dengan suatu DBMS, maka hal ini dapat mengurangi biaya latihan dan waktu belajar.
 3. Tersedianya layanan penjual
Keberadaan fasilitas pelayanan penjual sangat dibutuhkan untuk membantu memecahkan beberapa masalah sistem.
- Tahap 4, Perancangan *database* secara logika (*data model mapping*)
Tahap selanjutnya adalah membuat sebuah skema konseptual dan skema eksternal pada model data dari DBMS yang terpilih. Tahap ini dilakukan oleh pemetaan skema konseptual dan skema eksternal yang dihasilkan pada tahap 2. Pada tahap ini, skema konseptual ditransformasikan dari model data tingkat tinggi yang digunakan pada tahap 2 ke dalam model data dari model data dari DBMS yang dipilih

pada tahap 3.Pemetaan tersebut dapat diproses dalam 2 tingkat:

1. Pemetaan *system-independent*

Pemetaan ke dalam model data DBMS dengan tidak mempertimbangkan karakteristik atau hal-hal yang khusus yang berlaku pada implementasi DBMS dari model data tersebut.

2. Penyesuaian skema ke DBMS yang spesifik

Mengatur skema yang dihasilkan pada langkah 1 untuk disesuaikan pada implementasi yang khusus di masa yang akan datang dari suatu model data yang digunakan pada DBMS yang dipilih.Hasil dari tahap ini memakai perintah-perintah DDL (*Data Definition Language*) dalam bahasa DBMS yang dipilih yang menentukan tingkat skema konseptual dan eksternal dari sistem *database*. Tetapi 10 dalam beberapa hal, perintah-perintah DDL memasukkan parameter-parameter rancangan fisik sehingga DDL yang lengkap harus menunggu sampai tahap perancangan *databases* secara fisik telah lengkap.Tahap ini dapat dimulai setelah pemilihan sebuah implementasi model data sambil menunggu DBMS yang spesifik yang akan dipilih. Contoh: jika memutuskan untuk menggunakan beberapa *relational* DBMS tetapi belum memutuskan suatu relasi yang utama. Rancangan dari skema eksternal untuk aplikasi-aplikasi yang spesifik seringkali sudah selesai selama proses ini.

- Tahap 5, Perancangan *database* secara fisik
Perancangan *database* secara fisik merupakan proses pemilihan struktur-struktur penyimpanan dan jalur-jalur akses pada *file-file database* untuk mencapai penampilan yang terbaik pada bermacam-macam aplikasi. Selama fase ini, dirancang spesifikasi-spesifikasi untuk *database* yang disimpan yang berhubungan dengan struktur-struktur penyimpanan fisik, penempatan record dan jalur akses. Berhubungan dengan *internal schema* (pada istilah 3 level arsitektur DBMS). Beberapa petunjuk dalam pemilihan perancangan *database* secara fisik :

1. *Response time*

Waktu yang telah berlalu dari suatu transaksi *database* yang diajukan untuk menjalankan suatu tanggapan. Pengaruh utama pada *response time* adalah di bawah pengawasan DBMS yaitu : waktu akses *database* untuk data item yang ditunjuk oleh suatu transaksi. *Response time* juga dipengaruhi oleh beberapa faktor yang tidak berada di bawah pengawasan DBMS, seperti penjadwalan sistem operasi atau penundaan komunikasi.

2. *Space utility*

Jumlah ruang penyimpanan yang digunakan oleh *file-file database* dan struktur-struktur jalur akses.

3. *Transaction throughput*

Rata-rata jumlah transaksi yang dapat diproses per menit oleh sistem *database*, dan merupakan parameter kritis

dari sistem transaksi (misal : digunakan pada pemesanan tempat di pesawat, bank, dll). Hasil dari fase ini adalah penentuan awal dari struktur penyimpanan dan jalur akses untuk *file-file database*.

- Tahap 6, Implementasi Sistem *database* Setelah perancangan secara logika dan secara fisik lengkap, kita dapat melaksanakan sistem *database*. Perintah-perintah dalam DDL dan SDL(*Storage Definition Language*) dari DBMS yang dipilih, dihimpun dan digunakan untuk membuat skema *database* dan *file-file database* (yang kosong). Sekarang *database* tersebut dimuat (disatukan) dengan datanya. Jika data harus dirubah dari sistem komputer sebelumnya, perubahan-perubahan yang rutin mungkin diperlukan untuk format ulang datanya yang kemudian dimasukkan ke *database* yang baru. Transaksi-transaksi *database* sekarang harus dilaksanakan oleh para programmer aplikasi. Spesifikasi secara konseptual diuji dan dihubungkan dengan kode program dengan perintah-perintah dari *embedded DML* yang telah ditulis dan diuji. Suatu saat transaksi-transaksi tersebut telah siap dan data telah dimasukkan ke dalam *database*, maka tahap perancangan dan implementasi telah selesai, dan kemudian tahap operasional dari sistem *database* dimulai.

5.5 Membuat Kode Program

Merancang sebuah program tentunya tidak mudah, perlu adanya beberapa tahap pembuatan program yang baik dan benar. Program disusun dengan beberapa algoritma yang di implementasikan ke dalam bahasa pemrograman. Sedangkan orang yang profesiya membuat program disebut dengan programmer. Programmer dekat dengan system analyst, system analyst adalah profesi diatas seorang programmer , biasanya system analyst adalah senior programmer yang sudah banyak memakan asam garam dunia programming.

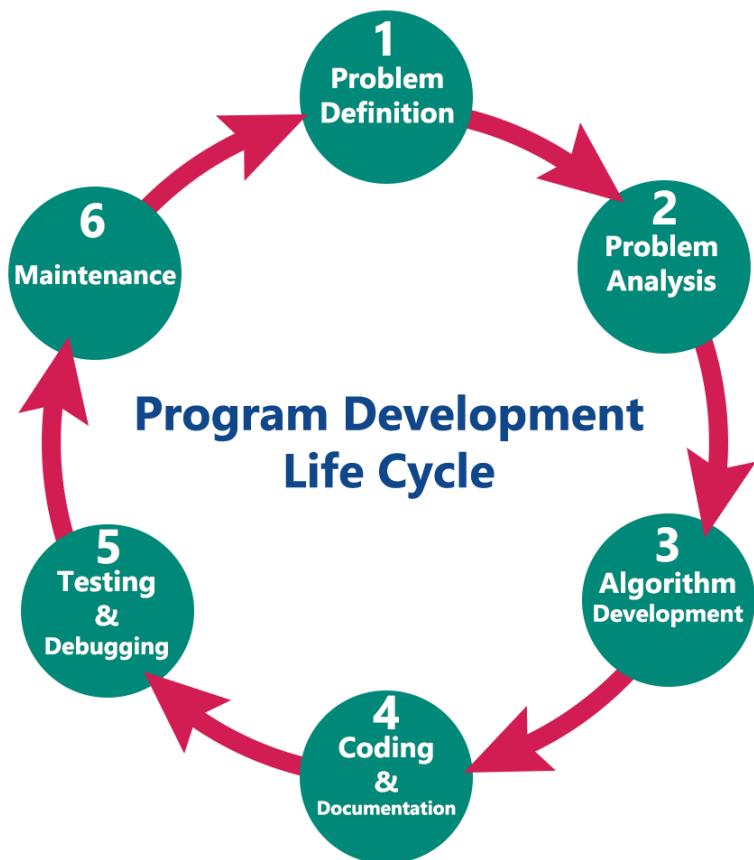
Program memang sudah menjadi ***kebutuhan pokok*** bagi masyarakat IT. karena segala sesuatu yang dilakukan di dalam IT pastilah memerlukan program. Program yang paling sederhana sekalipun setidaknya memiliki 3 bagian:

1. Input – Masukan data.
2. Proses – pemrosesan input.
3. Output – keluaran program, kebutuhan yang kita harapkan.

Dalam membuat program, pemrograman adalah pokok dari proses pembuatan program itu sendiri namun pemrograman bergantung dari pemahaman persoalan, analisis sistem, perencanaan-perencanaan dalam mendesain program itu sendiri.

Selain pemrograman hal yang utama harus dilakukan adalah merencanakan langkah-langkah yang harus diambil dalam menyelesaikan masalah. Karena dengan mengetahui masalah dan langkah-langkah penyelesaikan berarti kita sudah menyelesaikan

program tersebut sebanyak 50% dari total pekerjaan, selanjutnya adalah teknis pembuatan itu sendiri yang di kenal dengan pemrograman/koding. Sebaliknya jika kita tidak bisa mengetahui masalah dan belum bisa membuat perencanaan berarti kita sudah merencanakan kegalan itu sendiri.



Tahap-tahap pembuatan program secara teori dasar :

1. Defining the problem (*definisi masalah*) and solusi

Masalah disini yaitu masalah yang akan di analisis dan selanjutnya di pecahkan melalui program atau aplikasi melihat dari sundut pandang kebutuhan. Mendefinisikan sebuah masalah sangat penting bagi seorang programmer dalam menentukan langkah awal. Mendefinisikan masalah digunakan untuk mengetahui apa masalah yang ingin di selesaikan. Jika tidak mengetahui secara benar dan detail masalah apa yang akan diselesaikan, seorang programmer tidak bisa menentukan langkah selanjutnya dengan benar. Banyak juga programmer yang menganggap remeh tahapan ini, dengan terburu-buru langsung membuat program sehingga hasil program tersebut tidak sesuai dengan masalah awal dan hal itu akan merugikan programmer.

Masalah yang sudah didefinisikan dengan benar, setelah itu programmer harus memikirkan berbagai solusi yang dapat digunakan. Berbagai solusi tersebut harus dipilih salah satu yang paling efektif untuk diterapkan. Jika masalah yang harus diselesaikan adalah masalah besar, maka programmer bisa mengerucutkan masalah tersebut ke lingkup yang lebih kecil lagi sehingga terbentuk beberapa modul masalah agar mempermudah dalam penyelesaiannya.

2. Planning and Design System (*perencanaan dan desain system*)

Pada tahapan ini sangat diperlukan, karena tanpa tahapan planning pekerjaan yang dilakukan akan sia-sia tentunya dan memakan waktu yang lama. dengan planning system akan jelas serta tampilan

aplikasi yang dibuat juga terarah. biasanya perancangan system dilakukan oleh system analyst yang selanjutnya di kerjakan oleh seorang programmer. Design System ini biasanya menggunakan Flowchart System atau symbol alur program dan sebagainya.

3. Implementasi

Pada tahap ini programmer tentunya mengimplementasikan rancangan ke dalam bentuk struktur coding atau sering di sebut **code a programmer language**. Bahasa program tentu banyak tergantung pemakaian.

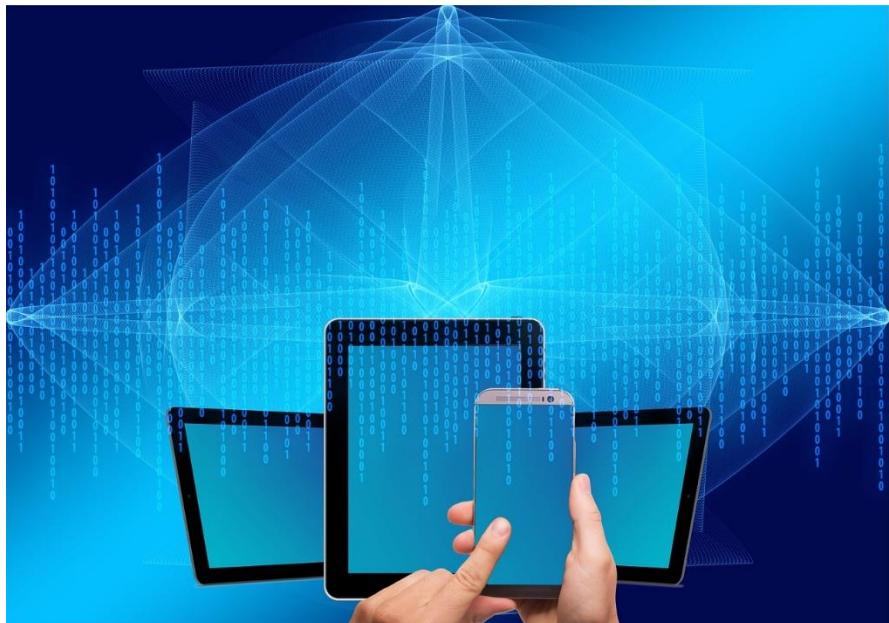
4. Documentation (*dokumentasi*)

Setelah tahap implementasi dilakukan dan program sudah mencakup 80 % selanjutnya buatlah dokumentasi atau komentar-komentar pada cuplikan program, tujuannya adalah jika terjadi debug atau problem kita dapat mengoreksinya dengan mudah dan sebaliknya orang dapat mengoreksi program yang kita buat.

Menulis	Dokumentasi
Saat membuat sebuah program, tidak sedikit programmer yang mengabaikan langkah dalam mendokumentasikan langkah-langkah yang telah dilakukan. Padahal mendokumentasikan saat membuat program sangatlah penting. Dengan dokumentasi tersebut, seorang programmer dapat melihat lagi dokumentasi tersebut kapan saja sehingga adakemungkinan untuk lebih mengembangkan program yang telah ada tersebut menjadi lebih baik dan baik lagi. Sebuah	

dokumentasi juga dapat membantu orang lain untuk mengetahui bahasa pemrograman ataupun algoritma yang digunakan pada suatu masalah yang ingin diketahuinya.

5. Testing



Berikut ini 8 Jenis Testing Wajib Seorang Software Tester / Software QA:

1. Performance Testing

8 Jenis Testing Wajib Seorang Software Tester / Software QA
(binish-siddiqui.com)

Performance test adalah integration dan usability test yang menentukan apakah system or subsystem dapat memenuhi kriteria

kinerja berbasis waktu seperti response time atau throughput. Response time menentukan batas waktu maksimum yang diijinkan dari respon software untuk query dan update. Throughput menentukan jumlah minimum query dan transaksi yang harus diproses per menit atau per jam.

2. System Testing

8 Jenis Testing Wajib Seorang Software Tester / Software QA (Biblioteca Digital do IPG)

System test adalah integration test dari behavior seluruh sistem atau independent subsystem. System testing biasanya dilakukan pertama kali oleh pengembang atau personil pengujian untuk memastikan bahwa keseluruhan sistem tidak berfungsi dan bahwa sistem telah memenuhi persyaratan pengguna (user requirement). System testing biasanya dilakukan di akhir setiap iterasi untuk mengidentifikasi isu -- isu penting, seperti masalah performance yang perlu ditangani di iterasi berikutnya. Biasanya test ini harus dilakukan sesering mungkin.

3. Unit Testing

8 Jenis Testing Wajib Seorang Software Tester / Software QA

Unit testing adalah proses metode pengujian individual, class, atau komponen sebelum mereka terintegrasi dengan perangkat lunak

lainnya. Tujuan dari unit testing adalah untuk mengidentifikasi dan memperbaiki kesalahan sebanyak mungkin sebelum modul -- modul digabungkan menjadi unit perangkat lunak yang lebih besar, seperti program, class dan subsistem. Kesalahan menjadi lebih sulit dan mahal untuk ditemukan dan diperbaiki ketika banyak unit telah digabungkan.

Unit testing memerlukan implementasi dari driver dan/atau *stub*. *Stub* adalah class atau method dummy yang dapat dipanggil namun biasanya tidak melakukan apapun kecuali mengembalikan tipe yang diperlukan. Modul driver adalah program yang menjalankan method atau fungsi dari class yang akan ditest.

Berikut adalah langkah yang harus dilakukan :

- Menentukan nilai dari parameter input
- Memanggil unit yang dites, melewatkannya dengan parameter input
- Menerima parameter kembalian dari unit yang dites dan mencetaknya, menampilkannya, atau mengetes hasilnya terhadap hasil yang diharapkan.

4. Integration Testing

8 Jenis Testing Wajib Seorang Software Tester / Software QA

Integration test adalah mengevaluasi behavior dari kelompok method atau class. Tujuan dari integration test adalah untuk mengidentifikasi kesalahan yang tidak dapat dideteksi oleh unit

testing. Kesalahan tersebut mungkin disebabkan oleh beberapa masalah, diantaranya :

- *Interface incompatibility*, misalnya sebuah method melewatkkan parameter dengan tipe data yang salah ke method lainnya
- *Parameter values*, misalnya sebuah method mengembalikan nilai yang tidak terduga seperti nomor negatif untuk harga
- *Run-time exceptions*, misalnya method menyebabkan kesalahan seperti out of memory atau file already in use karena ada konflik kebutuhan sumber daya
- *Unexpected state interactions*, misalnya state dari dua atau lebih objek yang berinteraksi menyebabkan kesalahan yang kompleks seperti ketika method class Order menjalankan satu kesalahan dari semua kemungkinan state objek Customer.

Beberapa masalah di atas merupakan kesalahan paling umum yang sering ditemui dalam integration testing, tetapi sebenarnya masih banyak masalah lainnya yang dapat menjadi penyebab kesalahan (error).

5. Usability Testing

8 Jenis Testing Wajib Seorang Software Tester / Software QA

Usability test adalah test untuk menentukan apakah method, class, subsistem, atau sistem telah memenuhi persyaratan pengguna. Oleh karena banyaknya tipe persyaratan sistem baik yang fungsional

maupun non-fungsional, maka banyak tipe dari usability test yang harus dilakukan di waktu yang berbeda. Umumnya usability test mengevaluasi persyaratan fungsional dan kualitas dari user interface. User berinteraksi dengan sistem untuk menentukan apakah fungsi telah seperti yang diharapkan dan apakah user interface membuat sistem dapat mudah digunakan. Pengujian ini sering dilakukan untuk mendapatkan feedback yang cepat dalam meningkatkan interface dan mengoreksi kesalahan dalam komponen perangkat lunak.

6. Smoke Testing

8 Jenis Testing Wajib Seorang Software Tester / Software QA

Smoke testing adalah system test yang biasanya dilakukan setiap hari atau beberapa kali per minggu. Build and smoke test sangat penting karena menyediakan feedback yang cepat dalam masalah yang signifikan.

7. Stress Testing

8 Jenis Testing Wajib Seorang Software Tester / Software QA

Stress Testing adalah pengujian yang biasanya dilakukan dalam membuat sebuah website, dimana stress testing dilakukan untuk mengetahui sekuat apa server website kita menampung visitor dalam

website tersebut, dengan cara melakukan hit dummy ke website menggunakan tools.

8. User Acceptance Test (UAT)

8 Jenis Testing Wajib Seorang Software Tester / Software QA

User acceptance test digunakan untuk menentukan apakah sistem yang dikembangkan telah memenuhi kebutuhan pengguna. Dalam beberapa proyek, acceptance testing dilakukan pada putaran terakhir proses pengujian yaitu sebelum sistem diserahkan kepada user.

Acceptance Testing biasanya dilakukan setelah rangkaian testing seperti *Unit Testing*, *Integration Testing*, dan *System Testing* selesai dan menggunakan metode *Black Box Testing*, dengan menggunakan dokumen test case untuk di presentasikan di akhir ke user / client.

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak

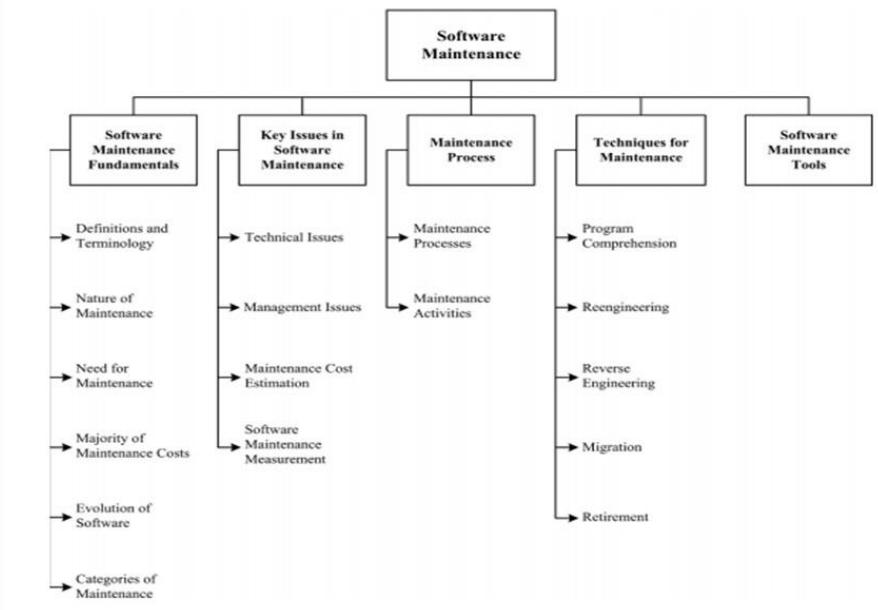
6. Maintenance

Program tanpa maintenance tentu tidak update dalam masalah bug error report yang dikeluhkan oleh pemakaian. Agar aplikasi dapat dikembangkan kembali beberapa tahun kemudian atau open source serta ada juga close source yang tidak bisa dikembangkan oleh

pihak lain biasanya ditentukan oleh lisensi sebuah program hanya untuk pemakaian saja.

Hargailah profesi seorang programmer, karena menciptakan sebuah program itu tidak mudah seperti memasak air. Tanpa programmer dunia IT tidak berkembang pesat seperti saat ini. Programmer juga butuh hidup dan sosialisasi. karena menjadi programmer itu tidak mudah.

Software Maintenance adalah bagian dari Software Development Life Cycle (SDLC). Software maintenance dilakukan diakhir pembuatan software. Bagian ini kadang terabaikan, padahal maintenance adalah hal penting yang harus ada dalam sebuah proyek pembuatan software. Pada Software Engineering Book of Knowledge (SWEBOK) dijelaskan mengenai software maintenance secara rinci, bahkan pada swebok sudah di break down kedalam beberapa sub area. Adapun Breakdown dari topik Software Maintenance ditunjukkan pada Gambar dibawah.



1. Software Maintenance Fundamental

Software Maintenance fundamental membahas mengenai definisi dari software maintenance dan pentingnya adanya maintenance

1.1. Definisi

Software maintenance adalah aktivitas perubahan atau modifikasi perangkat lunak setelah perangkat lunak digunakan (*after the delivery of software*).

1.2. Mengapa Butuh Maintenance

- Memperbaiki kesalahan
- Meningkatkan desain
- Peningkatan implementasi
- Ada interaksi antarmuka dengan perangkat lain (contoh : mengakses API)

- Ada perubahan perangkat keras, sehingga perangkat lunak perlu menyesuaikan
- Migrasi perangkat lunak à perangkat lunak sebelumnya diubah

1.3. Evolusi dari Software

Software maintenance pertama kali dikenalkan dan diterapkan pada akhir tahun 1960. Pada software maintenance ada sebuah paradigma bahwa “*Software yang besar itu tidak pernah selesai dan akan selalu berlanjut untuk berkembang, Software tumbuh semakin kompleks kecuali jika ada beberapa tindakan yang dilakukan untuk mengurangi kompleksitas*“

1.5. Kategori dari Perawatan

Terdapat 4 kategori dalam Software Maintenance, yaitu :

A. Corrective Maintenance

Modifikasi reaktif (atau perbaikan) perangkat lunak dilakukan **setelah adanya koreksi/masalah yang ditemukan**. Digunakan untuk menjaga operasional produk perangkat lunak

B. Adaptive Maintenance

Modifikasi produk perangkat lunak yang dilakukan agar perangkat lunak dapat tetap **digunakan dengan lingkungan yang berubah**. Contoh : perubahan sistem operasi sehingga perangkat lunak harus menyesuaikan.

C. Perfective Maintenance

Modifikasi perangkat lunak untuk **menyediakan perangkat tambahan untuk pengguna**, **peningkatan dokumentasi program**, dan **pengkodean ulang** untuk meningkatkan kinerja perangkat lunak atau atribut perangkat lunak lainnya. Misalnya : Penambahan Fitur

D. Preventive Maintenance

Modifikasi perangkat lunak untuk mendeteksi dan **memperbaiki kesalahan laten** dalam perangkat lunak sebelum mereka menjadi kesalahan operasional.

Pengelompokan Kategori Software Maintenance

	Correction	Enhancement
Proaktif	Preventive Maintenance	Perfective Maintenance
Reaktif	Correction Maintenance	Adaptive Maintenance

- **Correction** : hanya memperbaiki
- **Enhancement** : meningkatkan kinerja, meningkatkan kemampuan (upgrade)

2. Key Issues in Software Maintenance

Tedapat 3 Issue terkait software maintenance, yaitu perawatan secara teknis (technical issues), perawatan manajerial (management issue), Maintenance Cost Estimation, dan Maintenance Measurement.

2.1. Technical Issues

1. Limited Understanding

Software engineer baru harus dengan cepat paham terhadap

program/aplikasi yang dibuat oleh developer lain. Sehingga ketika terjadi error atau penambahan fitur dapat langsung memperbaiki.

2. Testing

Melakukan pengujian pada software untuk menemukan bug. Hal tersebut merupakan salah satu kegiatan yang dilakukan pada perawatan software.

3. Impact Analysis

Menganalisa pengaruh yang ditimbulkan ketika ada penambahan fitur. Jangan sampai saat terjadi penambahan fitur, malah membuat program error atau fungsi lain jadi tidak berfungsi.

4. Maintainability

Melakukan perawatan kepada software untuk menjaga agar tidak ada kerusakan yang fatal, serta meng-update fitur nya agar tidak kalah dengan software-software terbaru

2.2. Management Issues

1. Alignment with Organizational Objectives

Sebuah organisasi atau perusahaan yang berkecimpung dibidang IT. Selayaknya memberikan jobdesc kepada pegawainya sesuai dengan role bisnis. Kebutuhan maintener disesuaikan dengan kebutuhan, karena biasanya pekerjaan dilakukan per proyek. Deadline penggerjaan proyek yang terbatas biasanya membuat perusahaan meniadakan maintener.

2. Staffing

Menjadi seorang maintener (orang yang bekerja untuk merawat software) adalah pekerjaan yang kadang dipandang sebelah mata, karena kalah bergengsi dengan developer. Sehingga jika terdapat staff yang dapat bekerja sebagai seorang maintener, sebaiknya

diberikan fasilitas, penghasilan yang baik agar tidak berhenti.

3. Process

Proses dalam melakukan perawatan itu tidak sedikit dan cenderung kompleks. Software life cycle process adalah sebuah set aktifitas, metode, latihan, dan transformasi yang orang gunakan untuk mengembangkan dan merawat software dan produk-produk yang terkait. Maintenance juga membutuhkan beberapa aktivitas yang tidak ditemukan di software development (lihat section 3.2 di swebok)

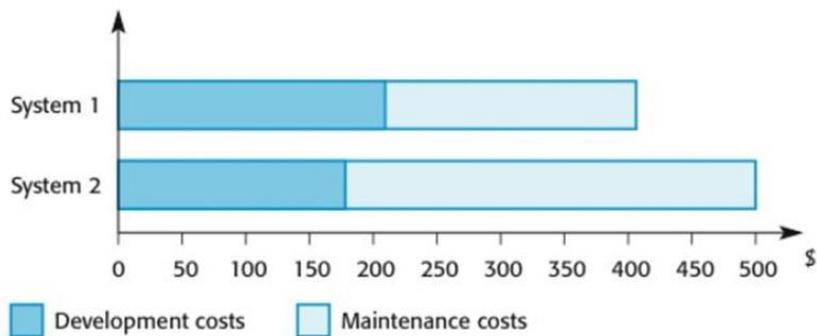
4. Outsourcing

Outsourcing adalah pegawai disebuah perusahaan yang memiliki sedikit kontrol terhadap perusahaan. Bukan merupakan pegawai tetap perusahaan dan biasanya melalui pelantara pihak ketiga. Contoh pekerjaan outsourcing misalnya sebagai penerjemah dari software yang dikerjakan.

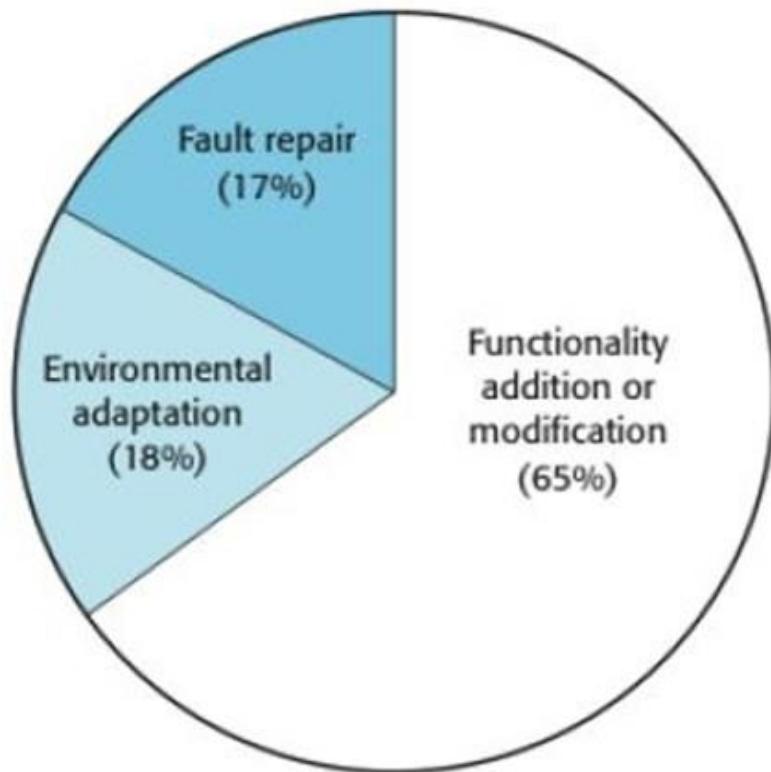
2.3. Maintenance Cost Estimation

Cost/Biaya [1]

Biasa perawatan Software, biasanya lebih besar dari biaya development. Software yang relatif “tua” mungkin membutuhkan biaya yang tinggi (karena bahasa pemrogramran “lama”, compilers, dll)



2.4. Software Maintenance Measurement Maintenance Effort Distribution



Faktor yang Mempengaruhi Maintenance Cost

1. Stabilitas Tim

Biaya pemeliharaan dapat ditekan jika staf yang sama terlibat dalam

jangka waktu yang lama

2. Tanggung Jawab Kontraktual

Adakah kontrak yang tertulis dalam perjanjian dengan pengembang software? Insentif pengembangan software melakukan maintenance tertera atau tidak, jika tidak maka lebih murah.

3. Ketrampilan Staff

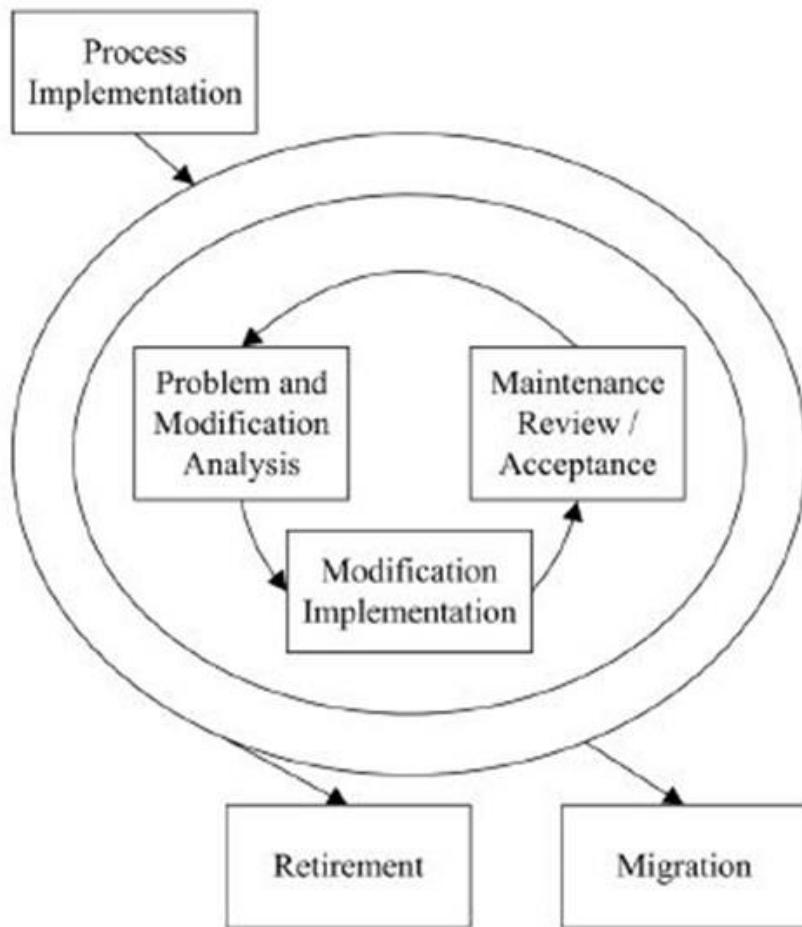
Semakin berpengalaman dan memiliki domain pengetahuan luas maka dapat menekan biaya maintenance

4. Usia dan Struktur Program

Seiring usia program, strukturnya semakin menurun dan menjadi lebih sulit untuk dipahami dan diubah

3. Maintenance Process

Merupakan proses yang diperlukan untuk melakukan perawatan software. Proses dalam perawatan software sangat banyak sekali, mulai dari menentukan kebutuhan, modifikasi, dan review kesesuaian. Berikut adalah langkah-langkah dari proses perawatan software.



Berdasarkan buku *Software Engineering 10th Edition* karangan Ian Sommerville perawatan software dimulai dengan menentukan permasalahan dan analisa dari modifikasi. Setelah menganalisa masalah dan menentukan modifikasi yang perlu diubah, langkah selanjutnya adalah melakukan implementasi dari modifikasi tersebut. Implementasi modifikasi diperlukan koordinasi antara analyst dan developer. Langkah terakhir setelah dilakukan implementasi adalah melakukan review dan acceptance test terhadap hasil implementasi.

Proses perawatan software dilakukan secara terus menerus mengikuti kebutuhan. Secara umum cara perawatan dilakukan dengan mengimplementasikan fitur (coding), melukan migrasi dari fitur baru atau penambahan fitur. Kemudian apabila software tersebut sudah tidak dapat mendukung kebutuhan pengguna, misal sebagai akibat bahasa pemrograman yang sudah tidak mendukung fungsi dari program.

Adapun model dari proses perawatan software lainnya adalah sebagai berikut.

- Quick Fix
- Spiral
- Osborne's
- Iterative Enhancement
- Reuse Oriented

Beberapa Aktifitas dalam Maintenance

- **Pemahaman Program**

Kegiatan yang diperlukan untuk mendapatkan pengetahuan umum tentang perangkat lunak, bagaimana cara kerjanya dan bagaimana unit di dalamnya berinteraksi

- **Transisi**

Penyampaian yang terkontrol dan terkoordinasi dengan baik segala hal yang berkaitan dengan perangkat lunak, dari pengembang ke pihak maintainer

- **Penerimaan/ Penolakan permintaan maintenance**

Permintaan maintenance dapat ditolak oleh maintainer dan dialihkan ke pengembang

- **Analisis Dampak**

Melakukan teknik analisis dampak dari perubahan di dalam perangkat lunak

- **Maintenance Service-Level Agreements (SLAs) and maintenance licenses and contracts**

Perjanjian kontrak

4. Techniques of Maintenance

4.1. Program Comprehension

- Programmer menghabiskan banyak waktu untuk membaca dan memahami program yang akan diubah
- Dokumentasi yang jelas dan ringkas dapat membantu dalam pemahaman program

4.2. Reverse Engineering

- Menganalisis perangkat lunak untuk mengidentifikasi komponen perangkat lunak dan hubungan di dalamnya untuk **membuat representasi perangkat lunak dalam bentuk lain atau pada tingkat abstraksi yang lebih tinggi**
- Reverse engineering bersifat pasif. **Tidak mengubah perangkat lunak ataupun menghasilkan perangkat lunak baru.**
- Contoh reverse engineering : re-dokumentasi

4.3. Reengineering [1]

- Reengineering didefinisikan sebagai pemeriksaan dan perubahan perangkat lunak dengan **menyusun kembali atau menuliskan kembali** sebagian atau seluruh bagian dari perangkat lunak, **tanpa mengubah fungsionalitasnya**

- Biasanya digunakan bukan untuk meningkatkan kinerja perangkat lunak, **namun mengganti perangkat lunak yang “menua”**.
- Sistem mungkin distrukturisasi dan didokumentasikan ulang

Reengineering [2] : Keuntungan

1. Mengurangi risiko

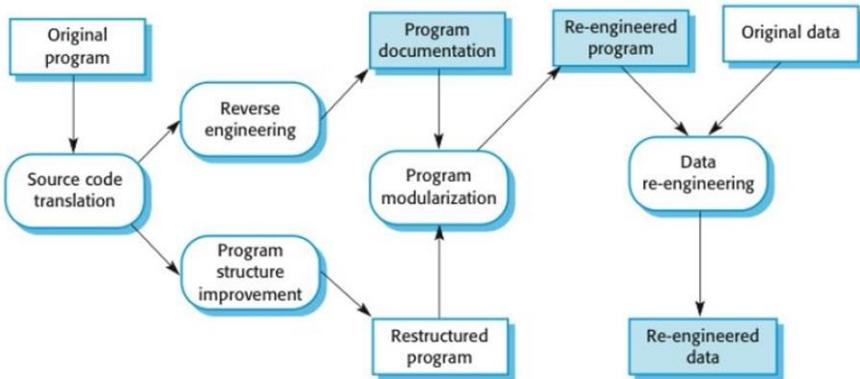
Ada risiko tinggi dalam pengembangan perangkat lunak baru.

Mungkin ada masalah pengembangan, masalah staf dan masalah spesifikasi.

2. Biaya yang dikurangi

Biaya reengineering seringkali jauh lebih rendah daripada biaya pengembangan perangkat lunak baru

Reengineering [3]



4.4. Migration [1]

- Dilakukan jika terjadi perubahan environment yang menjalankan perangkat lunak
- Maintainer perlu mengetahui aktivitas dalam proses migrasi : **migration requirements, migration tools, conversion of product and data, execution, verification and supports**

Migration [2] : Additional Activities

1. Notification of intent

Pernyataan berisi alasan lingkungan lama tidak lagi didukung, diikuti dengan deskripsi lingkungan yang baru dan tanggal ketersediaannya

2. Parallel Operations

Menyediakan lingkungan lama dan baru agar migrasi berjalan mulus

3. Notification of Completion

Saat migrasi selesai, informasi harus disampaikan ke seluruh pihak yang terkait

4. Post operation review

Penilaian setelah parallel operations dan dampak perubahan menjadi lingkungan baru

5. Data Archival

Menyimpan data dari perangkat lunak lama

4.5. Retirement

- Setelah perangkat lunak mencapai akhir kegunaan, maka harus dipensiunkan.
- Analisis harus dilakukan untuk membantu dalam membuat keputusan retirement dari sebuah software.
- Analisis ini harus dimasukkan dalam dokumen retirement planning, yang mencakup persyaratan retirement, dampak, penggantian, jadwal, dan effort.

5. Software Maintenance Tools

1. Program Slicers

Dapat memilih program yang terkena pengaruh perubahan

2. Static Analyzers

Memberikan general view/ pandangan umum dan ringkasan dari isi program

3. Dynamic Analyzers

Membantu melacak jalur eksekusi dari program

4. Data Flow Analyzers

Membantu melacak aliran data

5. Cross References

Menghasilkan indeks komponen program

6. Dependency Analyzers

membantu menganalisis dan memahami hubungan timbal balik antar komponen suatu program.

5.6 k

BAB VI	H
BAB VII	J
BAB VIII	G
BAB IX	GF
BAB X	G

1. Android studio
<https://medium.com/@tulisanndeso/pengertian-android-studio-dan-berbagai-macam-bagiannya-a21db3160a6e>
<https://developer.android.com/studio/intro?hl=id>
2. Eclipse
<https://wi01.files.wordpress.com/2011/02/pengenalan-eclipse.pdf>
<https://text-id.123dok.com/document/nq767wdny-sejarah-eclipse-arsitektur-eclipse-perkembangan-eclipse.html>
3. Framework ionic <https://edusoftcenter.com/mengenal-apitu-ionic-framework/>
<http://www.candra.web.id/mengenal-ionic-framework/>
4. Electron <https://www.codepolitan.com/membuat-website-apapun-menjadi-aplikasi-desktop-dengan-electron-dalam-15-menit-5830b4e408fcc-2140>
5. Intelij idea <https://litikgo.wordpress.com/2016/11/27/mengenal-ide-intellij-idea/>
6. Code lobster
<https://www.webhostingsecretrevealed.net/id/blog/web-tools/codelobster-php-edition-more-than-your-average-ide/>
<http://phphostingindonesia.com/codelobster-php-edition-lebih-dari-sekedar-text-editor/>
7. Framework
<https://www.nesabamedia.com/pengertian-framework/>
<https://idcloudhost.com/mengenal-apa-itu-framework-beserta-jenisnya/>
8. Desktop framework
<https://teknologi.id/insight/5-framework-terbaik-untuk-pengembangan-aplikasi-desktop/>
9. Ci laravel
<https://www.hostinger.co.id/tutorial/framework-php/>
10. Django
<https://www.petanikode.com/django-untuk-pemula/>
11. perancangan
<http://eprints.umg.ac.id/2054/3/14.%20Bab%20II.pdf>

12. Analisis
<file:///C:/Users/User/Downloads/Analisis%20Sistem.pdf>
13. Kode program
<https://novo-tani.wordpress.com/2013/08/01/langkah-langkah-membuat-program-secara-teori-dasar/>
14. Database
<https://muhammadghazali.wordpress.com/tag/6-tahap-perancangan-database/>
15. Testing
<https://www.kompasiana.com/toghr9896/5b67f796caf7db26250adff6/8-jenis-testing-wajib-seorang-software-tester-sqa?page=all>