



**VSGA** Vocational School  
Graduate Academy

# Modul Pelatihan **Junior Mobile Programmer**

Vocational School Graduate Academy  
Digital Talent Scholarship  
Tahun 2022

## KATA PENGANTAR

Era Digitalisasi pada Industri 4.0 di Indonesia pada saat ini dihadapkan pada tantangan hadirnya permintaan talenta digital dalam mendukung perkembangan ekosistem industri teknologi. Tantangan tersebut perlu dihadapi salah satunya melalui inisiasi dan kegiatan inovasi oleh berbagai pihak demi meningkatkan kapabilitas talenta digital Indonesia.

Dari unsur *triple helix*, upaya ini dapat diwujudkan melalui kerja sama antara pemerintah, instansi pendidikan dan pelatihan, serta dunia industri dengan mempersiapkan angkatan kerja muda menjadi talenta digital nasional. Oleh karena itu, Kementerian Komunikasi dan Informatika melalui Badan Penelitian dan Pengembangan Sumber Daya Manusia sejak tahun 2018, menginisiasi Program Beasiswa Pelatihan Digital bernama *Digital Talent Scholarship* (DTS) yang telah berhasil dianugerahkan kepada lebih dari 100.000 penerima pelatihan bidang teknologi informasi dan komunikasi. Program *Digital Talent Scholarship* ini ditujukan untuk memberikan pelatihan dan sertifikasi berbagai tema terkait teknologi informasi dan komunikasi, serta diharapkan menjadi bagian untuk memenuhi kebutuhan talenta digital Indonesia.

Program DTS 2022 secara garis besar dibagi menjadi tujuh akademi, yaitu Vocational School Graduate Academy (VSGA), Fresh Graduate Academy (FGA), Professional Academy (PROA), Digital Entrepreneurship Academy (DEA), Talent Scouting Academy (TSA), Government Transformational Academy (GTA), dan Thematic Academy (TA). VSGA merupakan program pelatihan berbasis kompetensi kerja nasional bagi lulusan pendidikan vokasi SMK/ sederajat dan diploma bidang *Science, Technology, Engineering, Mathematics* (STEM) yang belum mendapatkan pekerjaan atau sedang tidak bekerja. Tujuan Program VSGA adalah menyiapkan talenta digital dengan standar kompetensi sesuai Standar Kompetensi Kerja Nasional Indonesia (SKKNI). Oleh karena itu, penyusunan modul pelatihan untuk Program VSGA disusun dengan berbasis pada kompetensi (*Competency Based Training*). Kami berpesan agar modul pelatihan berbasis kompetensi yang telah disusun ini dapat menjadi referensi bagi peserta dan pengajar agar pelatihan berjalan efektif dan efisien.

Selamat mengikuti Pelatihan *Digital Talent Scholarship*, mari persiapkan diri kita menjadi talenta digital Indonesia yang kompeten.

Jakarta, Mei 2022  
Kepala Badan Penelitian dan Pengembangan Sumber Daya Manusia  
Kementerian Komunikasi dan Informatika Republik Indonesia

**Dr. Hary Budiarto, M.Kom**

## Pendahuluan

Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membuat aplikasi berbasis mobile Android.

### A. Tujuan Umum

Setelah mempelajari modul ini peserta latih diharapkan mampu membuat aplikasi berbasis mobile sederhana.

### B. Tujuan Khusus

Adapun tujuan mempelajari unit kompetensi melalui buku informasi Pelatihan Mobile Programming ini guna memfasilitasi peserta latih sehingga pada akhir pelatihan diharapkan memiliki kemampuan dalam melakukan instalasi alat bantu pemrograman mobile dan dapat menggunakan alat bantu pemrograman mobile dengan benar sesuai dengan kebutuhan.

## Latar belakang

Unit kompetensi ini dinilai berdasarkan tingkat kemampuan dalam merancang mobile programming. Adapun penilaian dilakukan dengan menggabungkan serangkaian metode untuk menilai kemampuan dan penerapan pengetahuan pendukung penting. Penilaian dilakukan dengan mengacu kepada Kriteria Unjuk Kerja (KUK) dan dilaksanakan di Tempat Uji Kompetensi (TUK), ruang simulasi atau workshop dengan cara:

- 1.1 Lisan
- 1.2 Wawancara
- 1.3 Tes tertulis
- 1.4 Demonstrasi
- 1.5 Metode lain yang relevan.

## Deskripsi Pelatihan

Materi ini berisi mengenai instalasi alat bantu pemrograman mobile, review alat bantu yang telah diinstalasi untuk digunakan dalam pengembangan aplikasi mobile.

## Tujuan Pembelajaran

### a. Tujuan Umum

Setelah mempelajari modul ini peserta latih diharapkan mampu membuat aplikasi berbasis mobile sederhana.

**b. Tujuan Khusus**

Adapun tujuan mempelajari unit kompetensi melalui buku informasi Pelatihan Junior Mobile Programmer ini guna memfasilitasi peserta latih sehingga pada akhir pelatihan diharapkan memiliki kemampuan sebagai berikut:

1. Peserta mengetahui langkah demi langkah instalasi alat bantu pengembangan aplikasi mobile
2. Peserta memahami komponen-komponen alat bantu hardware, software dan platform aplikasi mobile
3. Peserta dapat melakukan instalasi alat bantu pemrograman mobile dengan benar
4. Peserta mampu menjalankan alat bantu pemrograman mobile dengan benar
5. Peserta mampu menggunakan alat bantu pemrograman mobile dengan benar sesuai dengan kebutuhan

**Kompetensi Dasar**

Mampu menggunakan dan menjalankan alat bantu pemrograman mobile dengan benar sesuai dengan kebutuhan.

**Indikator Hasil Belajar**

1. Peserta mengetahui langkah demi langkah instalasi alat bantu pengembangan aplikasi mobile
2. Peserta memahami komponen-komponen alat bantu hardware, software dan platform aplikasi mobile
3. Peserta dapat melakukan instalasi alat bantu pemrograman mobile dengan benar
4. Peserta mampu menjalankan alat bantu pemrograman mobile dengan benar
5. Peserta mampu menggunakan alat bantu pemrograman mobile dengan benar sesuai dengan kebutuhan

## INFORMASI PELATIHAN

Akademi	Vocational School Graduate Academy
Mitra Pelatihan	Perguruan Tinggi
Tema Pelatihan	<b>Junior Mobile Programmer</b>
Sertifikasi	<i>Sertifikasi Kompetensi Junior Mobile Programming dari BNSP</i>
Persyaratan Sarana Peserta/spesifikasi device Tools/media ajar yang akan digunakan	<p>Laptop dengan spesifikasi:</p> <ol style="list-style-type: none"> <li>1. RAM minimal 4 GB (disarankan 8 GB)</li> <li>2. Laptop dengan 32/64-bit processor</li> <li>3. Laptop dengan Operating System Windows 7,8,10, Linux, atau MAC OSX</li> <li>4. Laptop dengan konektivitas WiFi dan memiliki webcam</li> </ol> <p>Akses Internet Dedicated 126 kbps per peserta per perangkat</p>
Aplikasi yang akan di gunakan selama pelatihan	<ol style="list-style-type: none"> <li>1. JDK (Java Development Kit)</li> <li>2. Android Studio</li> <li>3. Pencil</li> <li>4. Sublime Text</li> <li>5. XAMPP</li> </ol>
Tim Penyusun	<ul style="list-style-type: none"> <li>• Alif Akbar Fitrawan, S.Pd, M. Kom (Politeknik Negeri Banyuwangi);</li> <li>• Anwar, S.Si, MCs. (Politeknik Negeri Lhokseumawe);</li> <li>• Eddo Fajar Nugroho (BPPTIK Cikarang);</li> <li>• Eddy Tungadi, S.T., M.T. (Politeknik Negeri Ujung Pandang);</li> <li>• Fitri Wibowo (Politeknik Negeri Pontianak);</li> <li>• Ghifari Munawar (Politeknik Negeri Bandung);</li> </ul>

	<ul style="list-style-type: none"> <li>• Hetty Meileni, S.Kom., M.T. (Politeknik Negeri Sriwijaya) ;</li> <li>• I Wayan Candra Winetra, S.Kom., M.Kom (Politeknik Negeri Bali) ;</li> <li>• Irkham Huda (Vokasi UGM) ;</li> <li>• Josseano Amakora Koli Parera, S.Kom., M.T. (Politeknik Negeri Ambon) ;</li> <li>• I Komang Sugiarta, S.Kom., MMSI (Universitas Gunadarma) ;</li> <li>• Lucia Sri Istiyowati, M.Kom (Institut Perbanas) ;</li> <li>• Maksy Sendiang, ST, MIT (Politeknik Negeri Manado) ;</li> <li>• Medi Noviana (Universitas Gunadarma) ;</li> <li>• Muhammad Nashrullah (Politeknik Negeri Batam) ;</li> <li>• Nat. I Made Wiryana, S.Si., S.Kom., M.Sc. (Universitas Gunadarma) ;</li> <li>• Rika Idmayanti, ST, M.Kom (Politeknik Negeri Padang) ;</li> <li>• Rizky Yuniar Hakkun (Politeknik Elektronik Negeri Surabaya) ;</li> <li>• Robinson A.Wadu, ST, MT (Politeknik Negeri Kupang) ;</li> <li>• Roslina. M.IT (Politeknik Negeri Medan) ;</li> <li>• Sukamto, SKom., MT. (Politeknik Negeri Semarang) ;</li> <li>• Syamsi Dwi Cahya, M.Kom. (Politeknik Negeri Jakarta) ;</li> <li>• Syamsul Arifin, S.Kom, M.Cs (Politeknik Negeri Jember) ;</li> <li>• Usmanudin (Universitas Gunadarma) ;</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>Wandy Alifha Saputra (Politeknik Negeri Banjarmasin) ;</li> </ul>
--	--

## INFORMASI PEMBELAJARAN

Unit Kompetensi	Materi pembelajaran	Kegiatan pembelajaran	Durasi Pelatihan	Rasio Praktek : Teori	Sumber pembelajaran
Kebutuhan dalam membuat aplikasi berbasis mobile Android	Modul dan Slide dalam melakukan instalasi alat bantu pemrograman mobile android	Daring/Online	Live Class 2 JP LMS 4 JP @ 45 Menit	60:40	

Materi Pokok
<p>Instalasi alat bantu pemrograman mobile, review alat bantu yang telah diinstalasi untuk digunakan dalam pengembangan aplkasi mobile</p>

Sub Materi Pokok
<p>Langkah-langkah instalasi alat bantu pengembangan aplikasi mobile</p> <p>Memahami komponen-komponen alat bantu hardware, software dan platform aplikasi mobile</p> <p>Melakukan instalasi alat bantu pemrograman mobile dengan benar</p> <p>Menjalankan alat bantu pemrograman mobile dengan benar</p> <p>Menggunakan alat bantu pemrograman mobile dengan benar sesuai dengan kebutuhan</p>

## PENDAHULUAN

### Pengenalan Bahasa java

#### MATERI:

- A. Pendahuluan
- B. Pengenalan Bahasa Java
- C. Environment dan Tools

#### A. Pendahuluan

##### a. Sejarah Singkat Java

Pada tahun 1991, sekelompok insinyur SUN dipimpin oleh Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat konsumen seperti cable TV Box. Dikarenakan perangkat tersebut tidak memiliki banyak memori, bahasa harus berukuran kecil dan mengandung kode yang liat. Juga karena manufaktur-manufaktur berbeda memilih processor yang berbeda pula, maka bahasa harus bebas dari manufaktur manapun. Proyek diberi nama kode Green.

Kebutuhan untuk fleksibilitas, kecil, liat dan kode yang netral terhadap platform mengantar tim mempelajari implementasi Pascal yang pernah dicoba. Niklaus Wirth, pencipta bahasa Pascal telah merancang bahasa portabel yang menghasilkan *intermediate code* untuk mesin hipotesis. Mesin ini sering disebut dengan mesin maya (virtual machine). Kode ini kemudian dapat digunakan di sembarang mesin yang memiliki interpreter. Proyek Green menggunakan mesin maya untuk mengatasi isu utama tentang netral terhadap arsitektur mesin.

Karena orang-orang di proyek Green berbasis C++ dan bukan Pascal maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek dan bukan prosedural. Mulanya bahasa yang diciptakan diberi nama **Oak** oleh James Gosling yang mendapat inspirasi dari sebuah pohon yang berada pada seberang kantornya, namun dikarenakan nama Oak sendiri merupakan nama bahasa pemrograman yang telah ada sebelumnya, kemudian SUN menggantinya dengan **JAVA**. Nama JAVA sendiri terinspirasi pada saat mereka sedang menikmati secangkir kopi di sebuah kedai kopi yang kemudian dengan tidak sengaja salah satu dari mereka menyebutkan kata JAVA yang mengandung arti asal bijih kopi. Akhirnya mereka sepakat untuk memberikan nama bahasa pemrograman tersebut dengan nama Java. Produk pertama proyek Green adalah Star 7 (\*7), sebuah kendali jarak jauh yang sangat cerdas. Dikarenakan pasar masih belum tertarik pada produk konsumen cerdas maka proyek Green harus menemukan pasar lain dari teknologi yang diciptakan. Pada saat yang sama, implementasi WWW dan



Internet sedang mengalami perkembangan pesat. Di lain pihak, anggota dari proyek Green juga menyadari bahwa Java dapat digunakan pada pemrograman internet, sehingga penerapan selanjutnya mengarah menjadi teknologi yang berperan di web.

Bahasa/Alat pengembangan	Arsitektur Program			
	Modul Web Server	Scripting Web Server	Modul Web Browser	Scripting Web Browser
Java	servlet	JSP	Applet	Javascript
C++	CGI exe		ActiveX*	
Perl	CGI script			
Phyton	CGI script			
PHP		PHP script		
Visual Basic		ASP*	ActiveX*	VB Script*

\*) Hanya di landasan Windows, tidak bisa di Linux.

Gambar 1. Arsitektur Program (J.E.N.I - Pengenalan Pemrograman 1)

Java telah mengakomodasi hampir seluruh fitur penting bahasa-bahasa pemrograman yang ada semenjak perkembangan komputasi modern manusia:

- Dari SIMULA, bahasa pada tahun 65-an, bahasa yang paling mempengaruhi Java sekaligus C++. Dari bahasa ini diadopsi bentukan-bentukan dasar dari pemrograman berorientasi objek.
- Dari LISP bahasa tahun 55-an. Diadopsi fasilitas garbage collection, serta kemampuan untuk meniru generic list processing, meski fasilitas ini jarang yang memanfaatkannya.
- Dari Algol bahasa pada tahun 60-an, diambil struktur kendali yang dimilikinya.
- Dari C++, diadopsi sintaks, sebagian semantiks dan exception handling.
- Dari bahasa Ada, diambil strongly type, dan exception handling.
- Dari Objective C, diambil fasilitas interface.
- Dari bahasa SmallTalk, diambil pendekatan single-root class hiérarchie, dimana object adalah satu kesatuan hirarki pewarisan.
- Dari bahasa Eiffel, fasilitas assertion yang mulai diterapkan di sebagian JDK 1.4

#### b. JAVA Dikenal Sebagai Perangkat Teknologi, yaitu :

- Sebuah Bahasa Pemrograman

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, desktop, web dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Java adalah

bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat open source.

- Sebuah Development Environment

Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak tools : compiler, interpreter, penyusun dokumentasi, paket kelas dan sebagainya.

- Sebuah Aplikasi

Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki Java Runtime Environment (JRE).

- Sebuah Deployment Environment

Terdapat dua komponen utama dari Deployment Environment. Yang pertama adalah JRE, yang terdapat pada paket J2SDK, mengandung kelas-kelas untuk semua paket teknologi Java yang meliputi kelas dasar dari Java, komponen GUI dan sebagainya. Komponen yang lain terdapat pada Web Browser. Hampir seluruh Web Browser komersial menyediakan interpreter dan runtime environment dari teknologi.

### c. Karakteristik Bahasa JAVA

Berdasarkan white paper resmi dari SUN, Java memiliki karakteristik berikut :

1. **Sederhana:** Bahasa pemrograman Java menggunakan sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan multiple inheritance. Java juga menggunakan automatic memory allocation dan memory garbage collection.
2. **Berorientasi objek (Object Oriented):** Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.
3. **Dapat didistribusi dengan mudah:** Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya libraries networking yang terintegrasi pada Java.
4. **Interpreter:** Program Java dijalankan menggunakan interpreter yaitu Java Virtual Machine (JVM). Hal ini menyebabkan source code Java yang telah dikompilasi menjadi Java bytecodes dapat dijalankan pada platform yang berbeda-beda.

5. **Robust:** Java mempunyai reliabilitas yang tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai runtime-Exception handling untuk membantu mengatasi error pada pemrograman.
6. **Aman:** Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.
7. **Architecture Neutra:** Program Java merupakan platform independent. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform yang berbeda dengan Java Virtual Machine.
8. **Portabel:** Source code maupun program Java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.
9. **Performance:** Performance pada Java sering dikatakan kurang tinggi. Namun performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan Just In Time Compilers (JIT).
10. **Multithreaded:** Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.
11. **Dinamis:** Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu class dengan menambahkan properties ataupun method dapat dilakukan tanpa mengganggu program yang menggunakan class tersebut.

#### d. Fitur Dan Bahasa JAVA

##### 1. Java Virtual Machine (JVM)

JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi hardware dan platform dimana kompilasi kode Java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari platform manapun karena proses kompilasi diselesaikan oleh JVM.

Aplikasi program Java diciptakan dengan file teks berekstensi *.java*. Program ini dikompilasi menghasilkan satu berkas bytecode berekstensi *.class*. Bytecode adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara bytecode berjalan pada java interpreter yang tersedia di semua platform sistem komputer dan sistem operasi.

##### 2. Garbage Collection

Banyak bahasa pemrograman lain yang mengijinkan seorang programmer mengalokasikan memori pada saat dijalankan. Namun, setelah menggunakan alokasi memori tersebut, harus terdapat cara untuk menempatkan kembali blok memori tersebut supaya program lain dapat menggunakannya. Dalam C, C++ dan bahasa lainnya, adalah programmer

yang mutlak bertanggung jawab akan hal ini. Hal ini dapat menyulitkan bilamana programmer tersebut lupa untuk mengembalikan blok memori sehingga menyebabkan situasi yang dikenal dengan nama memory leaks.

Program Java melakukan garbage collection yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh programmer dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat pada bahasa yang memungkinkan alokasi dinamis.

### 3. Code Security

Code Security terimplementasi pada Java melalui penggunaan Java Runtime Environment (JRE). Java menggunakan model pengamanan 3 lapis untuk melindungi sistem dari untrusted Java Code.

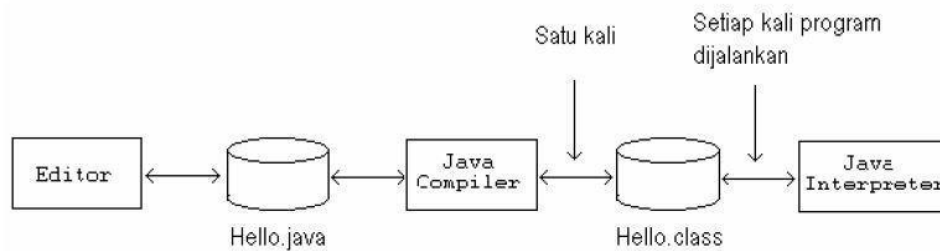
1. Pertama, *class-loader* menangani pemuatan kelas Java ke runtime interpreter. Proses ini menyediakan pengamanan dengan memisahkan kelas-kelas yang berasal dari local disk dengan kelas-kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas-kelas yang berasal dari local disk yang dimuat terlebih dahulu.
2. Kedua, *bytecode verifier* membaca bytecode sebelum dijalankan dan menjamin bytecode memenuhi aturan-aturan dasar bahasa Java.
3. Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, port jaringan, proses eksternal dan sistem windowing.

Setelah seluruh proses tersebut selesai dijalankan, barulah kode program dieksekusi.

Java juga menyediakan beragam teknik pengamanan lain, yaitu:

1. Bahasa dirancang untuk mempersulit eksekusi kode perusak. Peniadaan pointer merupakan langkah besar pengamanan. Java tidak mengenal operasi pointer. Di tangan programmer handal, operasi pointer merupakan hal yang luar biasa untuk optimasi dan pembuatan program yang efisien serta mengagumkan. Namun mode ini dapat menjadi petaka di hadapan programmer jahat. Pointer merupakan sarana luar biasa untuk pengaksesan tak diotorisasi. Dengan peniadaan operasi pointer, Java dapat menjadi bahasa yang lebih aman.
2. Java memiliki beberapa pengamanan terhadap applet. Untuk mencegah program bertindak mengganggu media penyimpanan, maka applet tidak diperbolehkan melakukan open, read ataupun write terhadap berkas secara sembarangan. Karena Java applet dapat membuka jendela browser yang baru, maka jendela mempunyai logo Java dan teks identifikasi terhadap jendela yang dibuka. Hal ini mencegah jendela pop-up menipu sebagai permintaan keterangan username dan password.

## e. Fase Pemrograman JAVA



**Gambar 2. Fase Pemrograman Java**

Langkah pertama dalam pembuatan sebuah program berbasis Java adalah menuliskan kode program pada text editor. Contoh text editor yang dapat digunakan antara lain : notepad, vi, emacs dan lain sebagainya. Kode program yang dibuat kemudian tersimpan dalam sebuah berkas berekstensi *.java*.

Setelah membuat dan menyimpan kode program, kompilasi file yang berisi kode program tersebut dengan menggunakan Java Compiler. Hasil dari kompilasi berupa berkas bytecode dengan ekstensi *.class*.

Berkas yang mengandung bytecode tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan platform yang digunakan.

<i>Proses</i>	<i>Tool</i>	<i>Hasil</i>
Menulis kode program	Text editor	Berkas berekstensi <i>.java</i>
Kompilasi program	Java Compiler	Berkas berekstensi <i>.class</i> (Java Bytecodes)
Menjalankan program	Java Interpreter	Program Output

**Gambar 3. Ringkasan Fase Pemrograman Java**

## B. Ruang Lingkup Pemrograman Java

### 1. Integrated Development Environment (IDE)

IDE adalah sebuah peralatan dalam pemrograman yang diintegrasikan ke dalam aplikasi software yang mendukung pembangunan GUI, teks atau penulisan kode, compiler dan debugger. Tutorial ini menggunakan Ubuntu 15.04 (Vivid Vervet) sebagai sistem operasinya. Sebelum menggunakan tutorial ini, pastikan komputer telah terinstal Java dan NetBeans dalam sistem yang digunakan.

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems yang saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi

sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal.

Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam *p-code* (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

## 2. Java Virtual Machine

Java Virtual Machine adalah merupakan jantung dari Java Platform. JVM ini adalah pihak yang bertanggung jawab untuk mengeksekusi program Java menjadi bahasa mesin untuk diproses oleh prosesor. JVM mampu menerjemahkan code-code Java ke hampir semua platform. JVM ini-lah yang membuat Java "*write once, run everywhere*" alias multi-platform.



Gambar 4. Skema Java Runtime Environment

## 3. Java Development Kit

Java Development Kit (JDK) adalah lingkungan pemrograman untuk menulis program aplikasi dan applet java, JDK terdiri dari lingkungan eksekusi program yang berada diatas Operating System, sebagaimana dibutuhkan oleh para programmer untuk meng-compile, membenahi bug(s) yang ada, dan menjalankan tambahan dari program intinya (applets) yang ditulis dengan menggunakan Bahasa pemrograman Java.

#### 4. Java Runtime Environment

Java Runtime Environment (JRE), beberapa praktisi cukup menyebutkannya dengan Java Runtime, merupakan bagian dari JDK, dan merupakan kumpulan dari “tools” programming, class-class inti, dan banyak file-file pendukung aplikasi. JRE menyediakan kebutuhan minimum untuk mengeksekusi aplikasi-aplikasi Java. salah satu bagian dari JRE adalah JVM (Java Virtual Machine).

#### C. Panduan Instalasi Java Development Kit

Sebelum menjalankan Eclipse, hal pertama yang harus dilakukan adalah menginstall Java SE Development Kit (JDK). Langkah langkah untuk menginstall JDK adalah sebagai berikut:

1. Download Java Development Kit (JDK) di <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Selanjutnya akan diarahkan ke halaman Download Oracle lalu pilih “Java Platform (JDK) 8u20” seperti gambar berikut:



Gambar 5. Halaman Java SE Downloads

2. Kemudian beri tanda centang pada Accept License Agreement yang untuk menerima dan menyetujui ketentuan dari Oracle dan Java. Kemudian download JDK sesuai dengan Sistem Operasi yang digunakan.



Gambar 6. Pemilihan Installer dan Agreement

- Setelah proses download selesai, buka dan install file JDK tersebut. Berikut adalah tampilan persiapan sebelum instalasi berlangsung:



Gambar 7. Persiapan Instalasi

- Kemudian akan muncul halaman setup seperti ini, tungguilah sampai proses inisialisasi siap barulah klik next.



Gambar 8. Welcome Screen Installer JDK



5. Pilih komponen yang akan diinstal komponen-komponen tersebut yaitu Development tools, Source code, dan Public JRE.



Gambar 9. Pemilihan Instalasi Komponen

6. Berikut in adalah gambar proses ekstrak sedang berlangsung, tunggu beberapa menit.



Gambar 10. Proses Ekstraksi File Instalasi JDK

7. Pilih lokasi folder tujuan kemana JDK akan diinstall. Pada bagian ini bisa diubah atau membiarkannya secara default.



Gambar 11. Menentukan Lokasi Installasi

8. Proses instalasi sedang berlangsung, tungguilah beberapa menit.



Gambar 12. Proses Installasi JDK

9. Setelah proses instalasi selesai maka tampilannya adalah seperti gambar berikut ini. Gambar di bawah ini menandakan bahwa proses instalasi JDK/JRE telah sukses terinstall dan dialog tersebut dapat ditutup dengan menekan tombol close. Atau jika ingin mempelajari lebih lanjut silahkan tekan Next Steps.



Gambar 13. Konfirmasi Instalasi JDK Selesai

## PENGENALAN ANDROID STUDIO

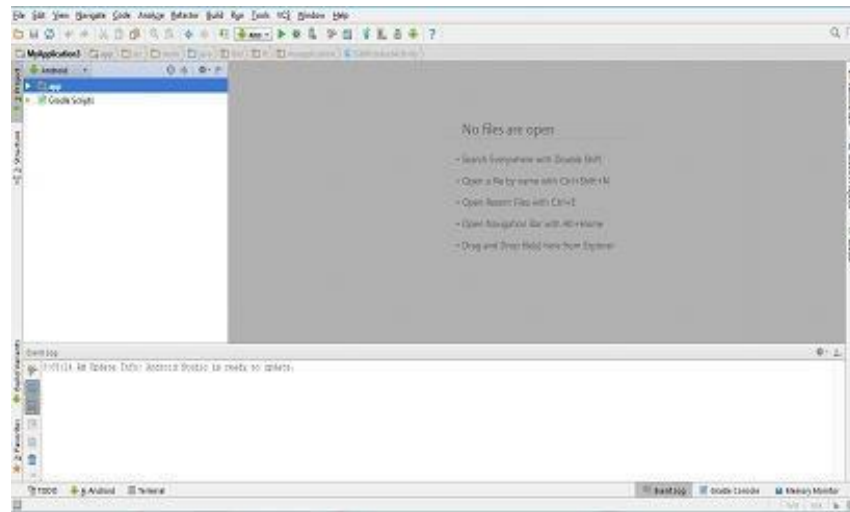
Android Studio adalah sebuah IDE untuk Android Development diperkenalkan google pada acara I/O 2013. Terdapat perbedaan antara Android Studio dengan Eclipse, Android Studio menggunakan Gradle untuk manajemen project. Gradle Merupakan Build Automation Tool, ini yang membedakan gradle dari Ant atau Maven yang memakai XML. Dengan gradle, programmer tidak perlu menambahkan kode di file manifest secara manual ketika menambah activity baru.



Gambar 14. Android Studio

Android Studio merupakan pengembangan dari eclipse, dikembangkan menjadi lebih kompleks dan professional yang telah tersedia didalamnya Android Studio IDE, Android SDK tools Android 5.0 (Lollipop) Platform, Android 5.0 emulator system image with Google APIs, terlebih Android studio

ini sudah memiliki dukungan untuk Android Lollipop 5.0 yang membuat para developer lebih luas dalam mengimplementasikan berbagai project aplikasi mereka pada berbagai interface semua jenis smartphone atau tablet Android.



Gambar 15. Interface Android Studio

## 1. Environment & Tools Android Studio

Android Studio memungkinkan untuk melihat perubahan visual apapun yang dilakukan pada aplikasi secara langsung. Perbedaannya dapat dilihat jika dipasang pada beberapa perangkat Android berbeda, termasuk konfigurasi dan resolusinya secara bersamaan.

Fitur lain di Android Studio adalah alat-alat baru untuk mengepak dan memberi label kode. Program ini juga memakai sistem *drag and drop* untuk memindahkan komponen melalui antar muka pengguna. Selain itu, lingkungan baru ini juga mendukung Google Cloud Messaging. Sebuah fitur untuk mengirim data dari server ke perangkat Android Anda melalui cloud. Program ini juga membantu untuk merelokasi aplikasi, dengan memberi gambaran visual untuk tetap memprogram sambil mengontrol alur dari aplikasi.

Mirip dengan Eclipse dengan ADT Plugin, Android Studio menyediakan alat pengembang terintegrasi untuk pengembangan dan debugging. Kelebihan Android Studio yang membuatnya unggul yaitu:

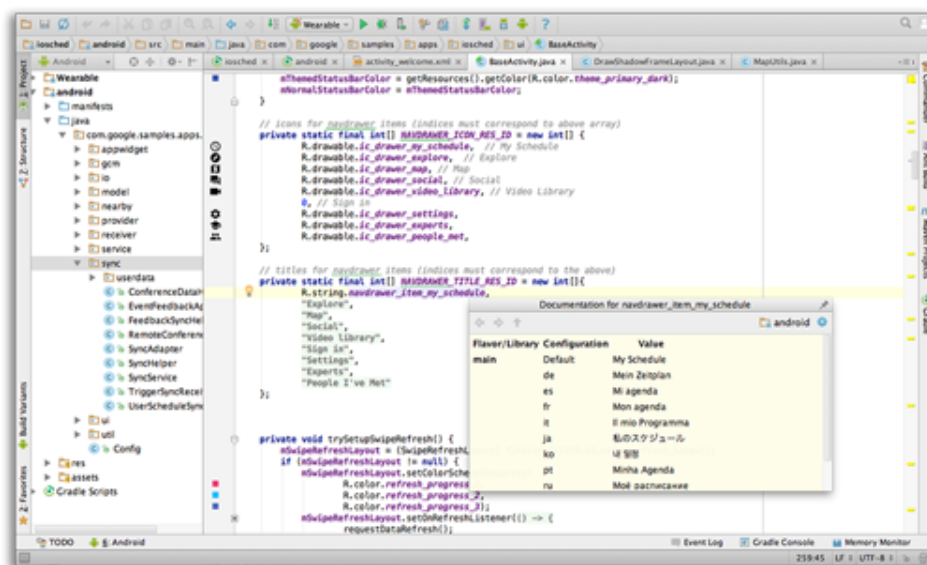
- Berbasis Gradle.
- Android-spesifik refactoring dan perbaikan yang cepat.
- Alat Lint untuk menangkap kinerja, kegunaan, versi kompatibilitas dan masalah lainnya.
- ProGuard dan app-signature.
- Wizard untuk design dan membuat komponen-komponen.

- Built-in dukungan untuk Google Cloud platform, sehingga mudah untuk mengintegrasikan Google Cloud Messaging dan App Engine sebagai komponen server-side.

Fleksibilitas dari Sistem Pengembangan Android Studio memungkinkan untuk mencapai keseluruhan hal ini tanpa harus memodifikasi file inti dari sebuah project.

#### a. Code Editor Cerdas

Inti dari Android Studio adalah editor kode cerdas mampu code completion canggih, refactoring, dan analisis kode. Editor kode yang kuat membantu pengguna menjadi pengembang aplikasi Android yang lebih produktif.



Gambar 16. Code Editor Android Studio

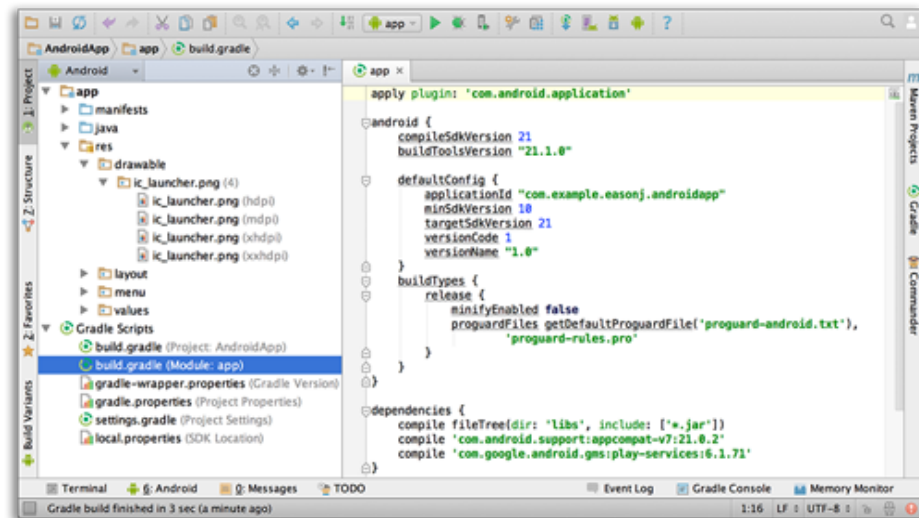
#### b. Kode template dan integrasi GitHub

Kelebihan yang paling diunggulkan oleh Android Studio yaitu Gradle Android Studio build system, dimana Gradle merupakan toolkit yang digunakan untuk membangun, test, menjalankan, dan membuat paket dari aplikasi yang akan dibuat. Build system tersebut tidak tergantung (independent) dari Android Studio, sehingga dapat dipanggil di dalam Android Studio maupun menggunakan command line.

Setelah semua kode ditulis untuk sebuah aplikasi, fitur Build System tersebut dapat digunakan untuk:

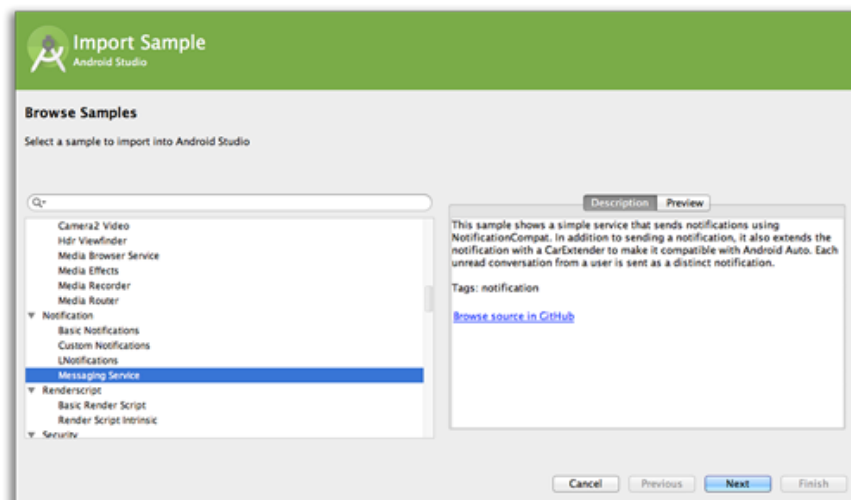
- Kustomisasi, konfigurasi, dan mengextends proses build.
- Membuat beberapa APK untuk aplikasi android anda dengan fitur yang berbeda menggunakan project yang sama.

- Menggunakan ulang kode dan resources



Gambar 17. Gradle Android Studio

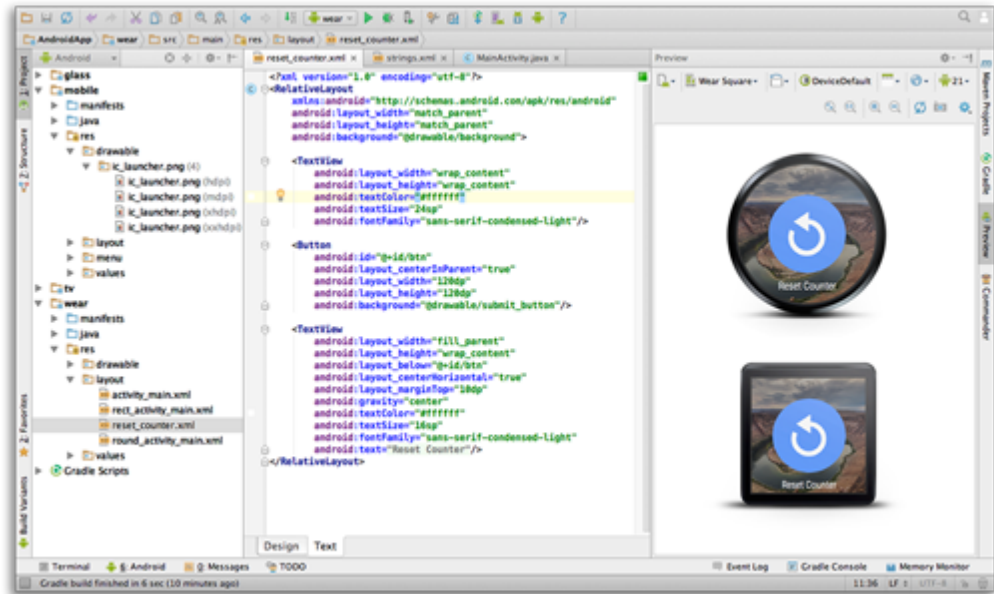
Android Studio juga terintegrasi dengan GitHub, sehingga pengguna dapat mengimpor contoh kode dari Google.



Gambar 18. Integrasi Android Studio dengan GitHub

### c. Pengembangan Aplikasi multi-layar

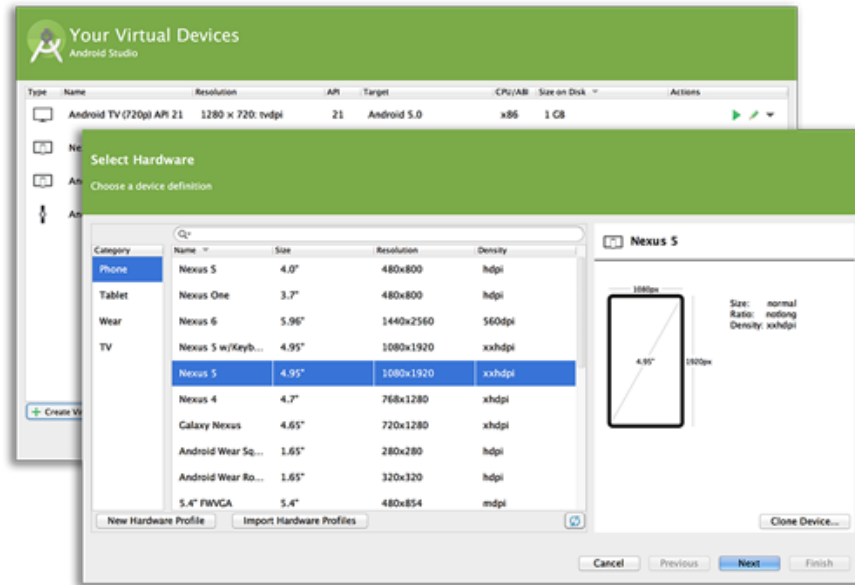
Android Studio menawarkan kemudahan untuk membangun aplikasi untuk ponsel Android, tablet, Android Wear, Android TV, Android Auto dan Google Glass. Dengan Android View Project dan modul dukungan di Android Studio, lebih mudah untuk mengelola proyek aplikasi dan sumber daya.



Gambar 19. Pengembangan Aplikasi Multi-Layar

#### d. Perangkat Virtual untuk segala bentuk dan ukuran

Android Studio dikonfigurasi dengan emulator gambar yang optimal. Virtual Device Manager yang telah diperbarui dan menjadi lebih efisien memberikan profil perangkat yang telah ditetapkan untuk perangkat Android umum. Android dengan Gradle memungkinkan membuat beberapa APK untuk aplikasi Android dengan fitur yang berbeda dalam satu project yang sama ataupun mengelola aplikasi dependensi dengan Maven.



Gambar 20. Integrasi Android Studio dengan GitHub

## 2. Instalasi Android Studio

### a. Download Android Studio

Sebagian besar pekerjaan yang terlibat dalam mengembangkan aplikasi untuk Android akan dilakukan dengan menggunakan lingkungan Android Studio. Android Studio dapat di unduh dari halaman web berikut: <http://developer.android.com/sdk/index.html>

Pada halaman tersebut, klik pada tombol Download Android Studio for Windows untuk pengguna Windows. Pada layar berikutnya, akan muncul jendela yang berisi syarat dan kondisi untuk memulai download

### b. Install Android Studio

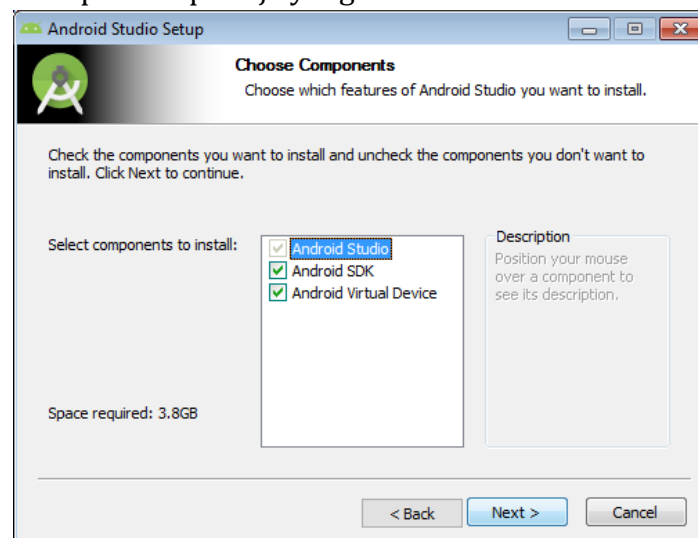
1. Setelah file selesai diunduh, cari file instalasi executable Android Studio dan klik dua kali untuk memulai proses instalasi. Klik tombol Yes pada dialog User Account Control dan klik tombol 'Next'.
2. Klik tombol 'Next' ketika layar berikutnya.





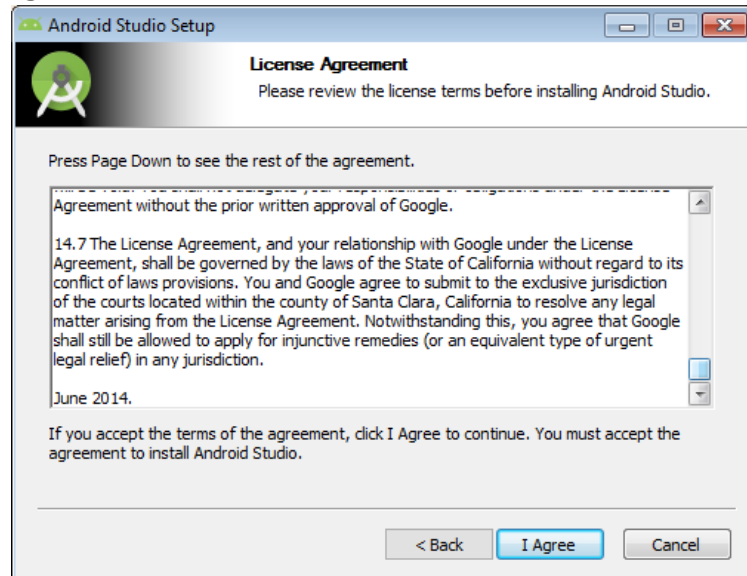
Gambar 21. Android Studio Setup

3. Pilih komponen apa saja yang akan diinstall. Klik 'Next'.



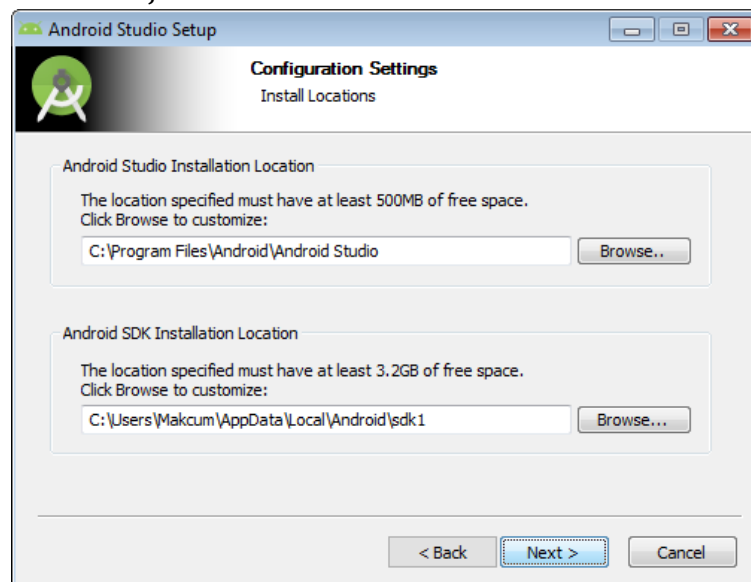
Gambar 22. Choose Components

4. Layar berikutnya adalah perjanjian lisensi. Klik pada tombol "I Agree".



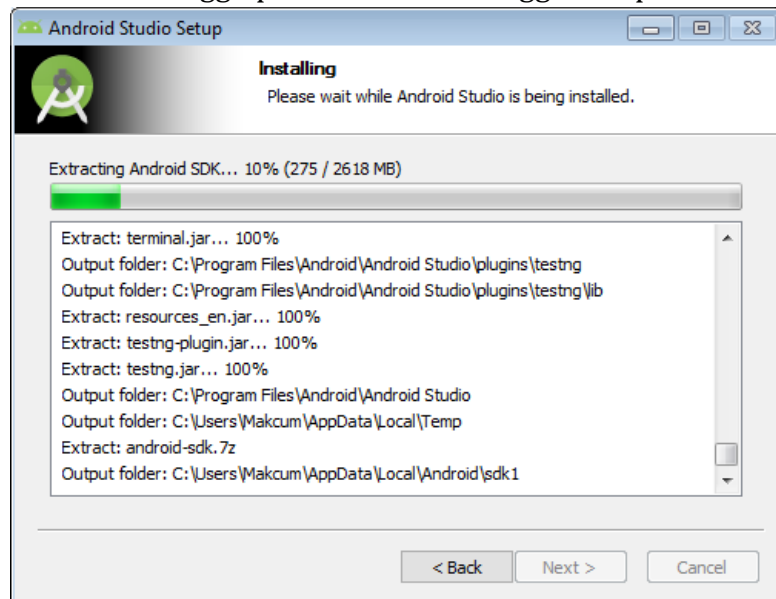
Gambar 23. Android Studio License Agreement

5. Layar berikutnya adalah pengaturan konfigurasi di mana semua jalur instalasi akan ditampilkan. Klik tombol 'Next' untuk melanjutkan.



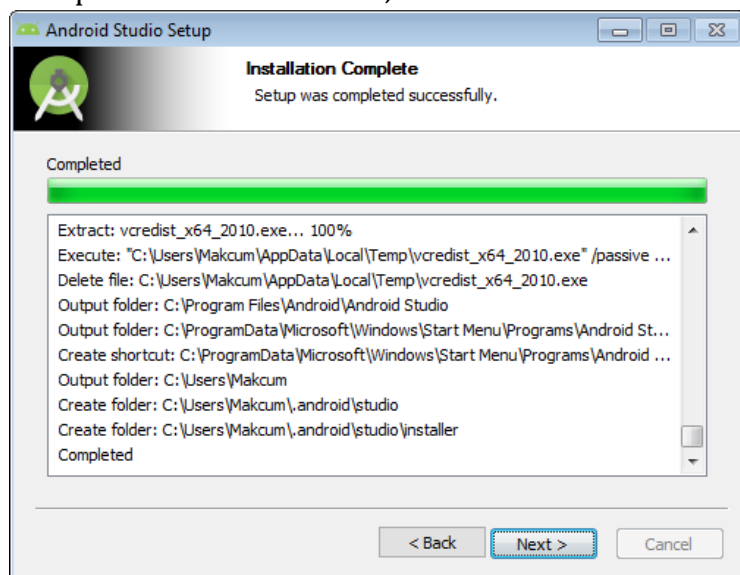
Gambar 24. Location Configuration Settings

6. Layar berikutnya adalah untuk konfirmasi untuk membuat sebuah shourcuts pada start menu, klik pada tombol 'Install' dan tunggu proses instalasi hingga Completed.



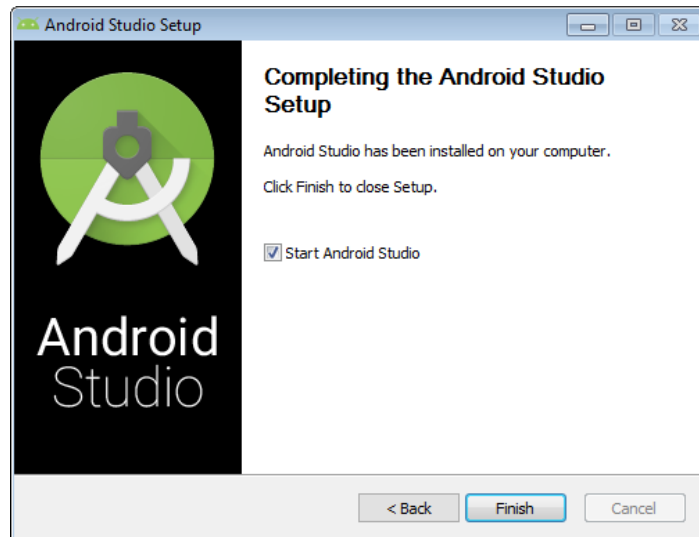
Gambar 25. Installing Android Studio

7. Setelah proses instalasi selesai, klik tombol 'Next'.



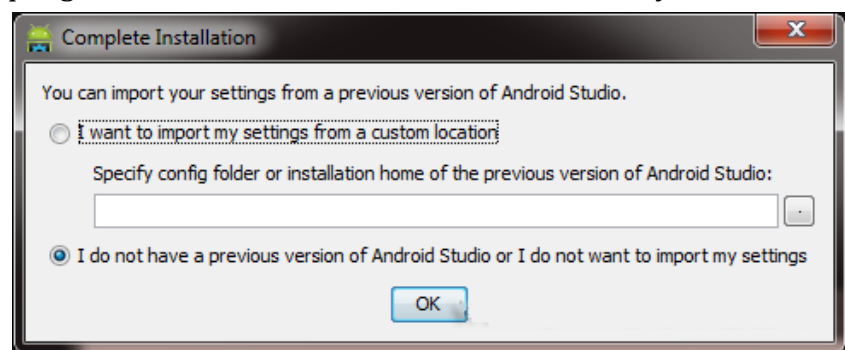
Gambar 26. Installation Complete

8. Centang Start Android Studio untuk membuka Android Studio sesaat setelah proses instalasi selesai dan Klik tombol Finish.



Gambar 27. Completing Android Studio Setup

9. Pertama kali Android Studio diluncurkan setelah diinstal, dialog akan muncul menyediakan opsi untuk mengimpor pengaturan dari versi Android Studio sebelumnya.



Gambar 28. Importing previous Version

10. Pilih "I do not have a previous version of Android Studio or I do not want to import my settings" dan klik tombol OK untuk melanjutkan.
11. Kemudian akan muncul jendela konfirmasi untuk mendownload dan menginstall komponen Android SDK Tools. Pastikan komputer terhubung dengan internet. Tunggulah proses download dan instalasi Android SDK tools hingga selesai.
12. Setelah proses download dan install componen telah selesai. Klik tombol Finish.

Tugas Dan Proyek Pelatihan
1. Kuis 1 Mobile Programming
Link Referensi Modul Pertama
1. Video Pembelajaran 2. E-book 3. Link Youtube/Website rujukan
Link Pertanyaan Modul Pertama
<a href="https://app.sli.do/">https://app.sli.do/</a> (bisa menggunakan aplikasi ini)
Bahan Tayang
Bisa berupa Link/ Screen Capture Slide pelatihan
Link room Pelatihan dan Jadwal live sesi bersama instruktur
Zoom, Blue Jeans, Meets
Penilaian
Komposisi penilaian Kuis 1 Mobile Programming: Nilai 10 (Range 0 -10)
Target Penyelesaian Modul Pertama
1hari/sampai 6JP

## REFERENSI

1. Dhingra, M. et al, Wireless Network Security Threats and Their Solutions: a short

- study, International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN), ISSN No. 2248-9738 (Print), Vol-2, Iss-1,2, 2012
2. Efraim Turban, D. K.-P. (2012). Electronic commerce (Seventh Edition ed.). Pearson.
  3. Gottapu Sasibhushana Rao, 2013. Mobile Cellular Communication (First Edition ). Pearson.
  4. [Guowang Miao](#), Jens Zander, Ki Won Sung, and Ben Slimane, Fundamentals of Mobile Data Networks, Cambridge University Press, [ISBN 1107143217](#), 2016.
  5. Herbert Schildt, *Java2 : A beginner's Guide*, Second Edition, McGraw-Hill/Osborne
  6. Larsson, J. and Waller, Ida. Security in wireless networks: Vulnerabilities and countermeasures, Department of Software Engineering and Computer Science Blekinge Institute of Technology, Springer, 2003
  7. Matthew Mathias, Swift Programming, 2nd edition, Big Nerd Ranch
  8. <https://developer.android.com/reference/android/os/Environment.html>
  9. <https://developer.android.com/reference/java/io/File.html>
  10. <https://developer.android.com/guide/topics/data/data-storage?hl=id>
  11. <https://developer.android.com/training/data-storage/files#InternalVsExternalStorage>
  12. Owens, Michael (2006). The Definitive Guide to SQLite. <https://www.sqlite.org/chronology.html>
  13. <https://developer.android.com>
  14. <https://www.quora.com/How-does-the-algorithm-of-Google-Maps-work>
  15. <https://tips-droid.blogspot.com/2013/07/mengenai-android-sensor.html>
  16. <https://mainthebest.com/smartphones/sensor-smartphone-android/#accelerometer>
  17. PROFESSIONAL Android™ Sensor Programming Greg Milette Adam Strou
  18. Irwanto, A. (2013, September 7). area teknik. Retrieved Oktober 25, 2017, from <http://area-teknik.blogspot.co.id/2013/09/perbedaan-teknologi-1g-2g-25g-3g-35g-4g.html>

19. Wulan. (2015, November 2015). Srydari. Retrieved Oktober 25, 2017, from <http://srydari.blogspot.co.id/2015/11/perkembangan-teknologi-dari-1g-sampai-5g.html>
20. “ ”, Java™ Programming Language, Oracle America
21. <https://developer.apple.com/library/archive/referencelibrary/GettingStarted/DevelopiOSAppsSwift/index.html/>
22. <https://developer.android.com/topic/libraries/architecture>
23. <https://www.oracle.com/technetwork/java/javase/overview/index.html>
24. Android Cook Book, McGraw-Hill/Osborne, 2013



KOMINFO

Badan Penelitian dan Pengembangan SDM  
Kementerian Komunikasi dan Informatika