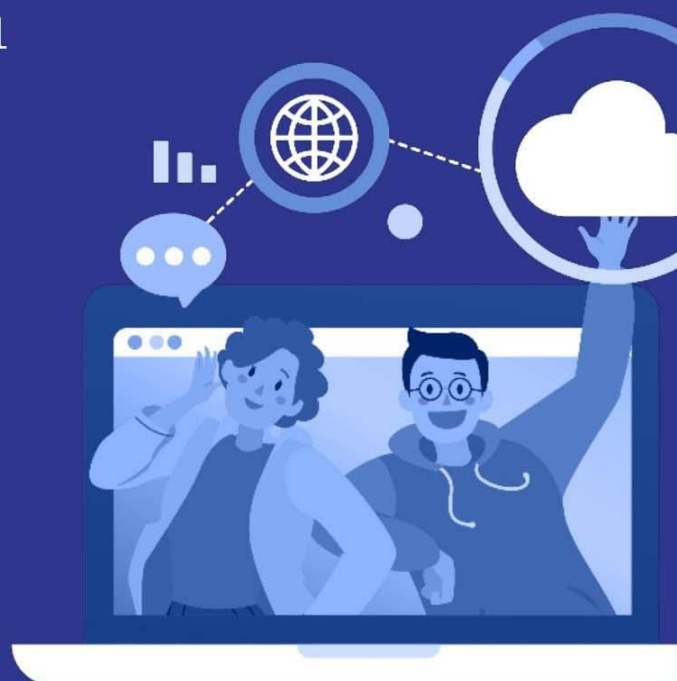


# Modul Pelatihan

## Vocational School Graduate Academy

### Tema

Digital Talent Scholarship 2021



## KATA PENGANTAR

Era Digitalisasi pada Industri 4.0 di Indonesia saat ini dihadapkan pada tantangan hadirnya permintaan dan penawaran talenta digital dalam mendukung perkembangan ekosistem industri teknologi. Tantangan tersebut perlu dihadapi salah satunya melalui kegiatan inovasi dan inisiasi dari berbagai pihak dalam memajukan talenta digital Indonesia, baik dari pemerintah maupun mitra kerja pemerintah yang dapat menyiapkan angkatan kerja muda sebagai talenta digital Indonesia. Kementerian Komunikasi dan Informatika melalui Badan Penelitian dan Pengembangan Sumber Daya Manusia sejak tahun 2018-2019, telah menginisiasi Program Digital Talent Scholarship yang telah berhasil dianugerahkan kepada 26.000 penerima pelatihan di bidang teknologi informasi dan komunikasi. Program Digital Talent Scholarship ini ditujukan untuk memberikan pelatihan dan sertifikasi tema-tema bidang teknologi informasi dan komunikasi, diharapkan menjadi bagian untuk memenuhi permintaan dan penawaran talenta digital Indonesia.

Tahun ini, Program Digital Talent Scholarship menargetkan pelatihan peningkatan kompetensi bagi 60.000 peserta yang bertujuan untuk meningkatkan keterampilan dan daya saing SDM bidang teknologi informasi dan komunikasi sebagai bagian dari program pembangunan prioritas nasional. Program pelatihan DTS 2021 ditujukan untuk meningkatkan keterampilan, keahlian angkatan kerja muda Indonesia, masyarakat umum dan aparatur sipil negara di bidang teknologi informasi dan komunikasi sehingga dapat meningkatkan produktivitas dan daya saing bangsa di era Industri 4.0.

Program DTS 2021 secara garis besar dibagi menjadi Tujuh akademi, yaitu: Fresh Graduate Academy (FGA), Program pelatihan berbasis kompetensi bersama GlobalTech yang ditujukan kepada peserta pelatihan bagi lulusan S1 bidang TIK dan MIPA, dan terbuka bagi penyandang disabilitas; Vocational School Graduate Academy (VSGA), Program pelatihan berbasis kompetensi nasional yang ditujukan kepada peserta pelatihan bagi lulusan SMK dan Pendidikan Vokasi bidang TI, Telekomunikasi, Desain, dan Multimedia; Coding Teacher Academy (CTA), Program pelatihan merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan kepada peserta pelatihan bagi Guru setingkat SMA/SMK/MA/SMP/SD di bidang pemrograman. Online Academy (OA), Program pelatihan OA merupakan program pelatihan Online di bidang Teknologi Informasi yang ditujukan kepada peserta pelatihan bagi Masyarakat umum, ASN, mahasiswa, dan pelaku industri; Thematic Academy (TA), Program pelatihan TA merupakan program pelatihan multisektor bagi pengembangan sumberdaya manusia yang ditujukan kepada peserta pelatihan dari jenjang dan multidisiplin yang berbeda; Regional Development Academy (RDA), Program pelatihan RDA merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan untuk meningkatkan kompetensi ASN di Kawasan Prioritas Pariwisata dan 122 Kabupaten Prioritas Pembangunan. Digital Entrepreneurship Academy (DEA), Program pelatihan DEA merupakan program pelatihan pengembangan sumberdaya manusia yang ditujukan kepada talenta digital di bidang Usaha Mikro, Kecil, dan Menengah (UMKM).

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

Jakarta, 2021

Kepala Badan Penelitian dan Pengembangan Sumber Daya Manusia  
Kementerian Komunikasi dan Informatika Republik Indonesia  
**Dr. Ir. Basuki Yusuf Iskandar, MA**

## 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 menunjukkan platform operating system dan bahasa pemrograman di dalam perangkat lunak (Dasar Platform mobile).

## 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 penjelasan mengenai cara menunjukkan *Platform Operating Sistem* serta Bahasa Pemrograman di dalam Perangkat Lunak (Dasar Java untuk Mobile Programming).

## Tujuan Pembelajaran

Setelah pertemuan selesai peserta pelatihan mampu :

- a. Peserta mampu menunjukkan arsitektur dasar sistem operasi berbasis mobile sesuai dengan perangkat keras yang digunakan.
- b. Peserta mampu menunjukkan platform sistem operasi berbasis mobile sesuai dengan perangkat keras yang terkait.
- c. Peserta mampu menunjukkan security pada platform sistem operasi berbasis mobile sesuai dengan perangkat keras yang digunakan
- d. Peserta mampu merancang kebutuhan user berdasarkan spesifikasinya

## Kompetensi Dasar

Cara menunjukkan *Platform Operating* Sistem serta Bahasa Pemrograman di dalam Perangkat Lunak (Dasar Java untuk Mobile Programming)

## Indikator Hasil Belajar

- a. Peserta mampu menunjukkan arsitektur dasar sistem operasi berbasis mobile sesuai dengan perangkat keras yang digunakan.
- b. Peserta mampu menunjukkan platform sistem operasi berbasis mobile sesuai dengan perangkat keras yang terkait.
- c. Peserta mampu menunjukkan security pada platform sistem operasi berbasis mobile sesuai dengan perangkat keras yang digunakan
- d. Peserta mampu merancang kebutuhan user berdasarkan spesifikasinya

## INFORMASI PELATIHAN

Akademi	Vocational School Graduate Academy
Mitra Pelatihan	Perguruan Tinggi
Tema Pelatihan	<b>Mobile Programmimg</b>
Sertifikasi	<i>Sertifikasi Kompetensi Junior Mobile Programming dari BNSP</i>
Persyaratan Sarana Peserta/spesifikasi device Tools/media ajar yang akan digunakan	Laptop dengan spesifikasi: 1. RAM minimal 4 GB (disarankan 8 GB) 2. Laptop dengan 32/64-bit processor 3. Laptop dengan Operating System Windows 7,8,10, Linux, atau MAC OSX 4. Laptop dengan konektivitas WiFi dan memiliki webcam  Akses Internet Dedicated 126 kbps per peserta per perangkat
Aplikasi yang akan di gunakan selamat pelatihan	1. JDK (Java Development Kit) 2. Android Studio 3. Pencil 4. Sublime Text 5. XAMPP
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><li>● Hetty Meileni, S.Kom., M.T. (Politeknik Negeri Sriwijaya) ;</li><li>● I Wayan Candra Winetra, S.Kom., M.Kom</li></ul>

	<p>(Politeknik Negeri Bali) ;</p> <ul style="list-style-type: none"><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>● Sukanto, 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><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 Sistem Operasi, Bahasa Pemrograman, dan Dasar pemrograman	Daring/Online	Live Class 2 JP LMS 4 JP @ 45 Menit	60:40	

### Materi Pokok

Menunjukkan *Platform Operating* Sistem serta Bahasa Pemrograman di dalam Perangkat Lunak (Dasar Java untuk Mobile Programming).

### Sub Materi Pokok

- 1.1 Menunjukkan jenis platform sistem operasi berbasis mobile
- 1.2 Menentukan platform sistem operasi yang sesuai kebutuhan user
- 1.3 Menjelaskan bahasa pemrograman berbasis mobile

## MENUNJUKKAN PLATFORM OPERATING SYSTEM DAN BAHASA PEMROGRAMAN DI DALAM PERANGKAT LUNAK

**A. Pengetahuan yang diperlukan dalam menunjukkan platform operating system dan bahasa pemrograman di dalam perangkat lunak**

**Menunjukkan jenis platform sistem operasi berbasis mobile**

**a. Karakteristik Mobile Device**

Mobile devices have similar characteristics.



- 1) Portable, secara garis besar dapat diartikan sebagai Praktis, User friendly, mudah dibawa kemana mana dan bersifat mandiri
- 2) Small size, Ukuran dan berat yang bisa dibawa dengan satu tangan dan dapat dimanipulasi dengan tangan lainnya
- 3) Wi-Fi or Wireless Communication, memiliki koneksi tanpa kabel dengan frekuensi tinggi standart IEEE
- 4) Hampir secara keseluruhan mobile device saat ini memiliki interface touchscreen
- 5) Memory Terbatas, mungkin argument ini cocok digunakan untuk ponsel ponsel kelahiran 2015 kebawah, dimana ponsel tersebut memang memiliki RAM dan ROM yang terbatas, tetapi untuk Ponsel saat ini mungkin sudah mendekati layaknya laptop, atau bahkan sudah ada yang melebihi kelas laptop. Contoh misalnya smartphone kita memiliki ram 4 GB dan ROM 128 GB, untuk sekelas Smartphone ini merupakan suatu memory yang lebih dari cukup.
- 6) Adanya virtual assistant, seperti Siri, Cortana atau Google Assistant, bahkan dulu kita sering mendengar nama Jarvis di film ironman 2 sebagai Virtual asisten, kalau sekarang virtual asisten ironman di avenger end game bernama friday

**b. Arsitektur dasar sistem operasi**

Sistem Operasi mobile adalah software utama yang melakukan manajemen dan kontrol terhadap hardware pada mobile device secara langsung.



## 1) Jenis-jenis Sistem Operasi

### a) Windows

Hampir 90% pengguna Sistem Operasi menggunakan Windows. Diperkenalkan pada 1985 sudah Sistem operasi ini sudah mature dan telah memiliki bagian-bagian software yang lengkap.

Kelebihan

- Kompatibilitas: Hampir setiap aplikasi, driver atau game akan berfungsi di Windows.
- Dukungan teknis: Memiliki begitu banyak pengguna, mudah menemukan seseorang (baik online atau offline) yang dapat membantu dengan Windows.
- Banyak fungsi: Ketika mengenal Windows dengan baik, maka akan menemukan bahwa ada begitu banyak fungsi yang dapat dilakukan hampir semua hal dengan mudah.

Kelemahan:

- Virus: perlu membeli program antivirus, meskipun ada yang gratis.
- Lambat: Windows, terutama Vista dan 7, membutuhkan banyak sumber daya komputer (memori, prosesor, ruang disk), dan dengan demikian, berjalan lebih lambat.

## **b) Macintosh**

Sistem Operasi Macintosh dari Apple lebih tua dari Windows merupakan sistem operasi berbasis grafis pertama yang sukses, yang dirilis satu tahun sebelum Microsoft.

Kelebihan:

- Virus: Apple Mac hampir tidak memiliki virus.
- Reliabilitas: Mac hanya berjalan di komputer Apple, dan dengan demikian kurang rentan terhadap kerusakan perangkat keras dan perangkat lunak.
- Tampilan: Mac terlihat lebih baik daripada Windows.

Kekurangan:

- Mahal: biaya Mac lebih dari Windows.
- Hanya tersedia di komputer Apple: Jika Anda sudah memiliki komputer, Anda tidak dapat menginstal MAC di atasnya kecuali Apple.
- Kompatibilitas: Hanya beberapa program yang akan berjalan di Mac, dan hampir tidak ada game.

## **c) Linux**

Linux adalah jawaban GNU untuk Mac dan Windows. Linux gratis, pengguna dapat mengunduh, memodifikasi, dan mendistribusikan ulang tanpa biaya. Linux telah ditulis pada tahun 1991, dan dioptimalkan untuk penggunaan modern (baik, lebih dari Windows dan Mac).

Kelebihan:

- Harga: Linux adalah gratis (F-R-E-E). Pengguna dapat mengunduh, menginstal, menggunakannya, memodifikasinya.
- Ragam: Linux bukan sistem operasi penuh. Itu hanya sebuah kernel. Untuk menggunakan kernel, perangkat lunak tambahan harus dipaketkan dengan Linux. Tersedia beberapa ratus bundel ini (disebut “distribusi” atau hanya “distro”). Yang paling populer termasuk Ubuntu, Mint dan Fedora. Hal yang baik adalah, dengan begitu banyak rasa Linux yang berbeda, selalu ada satu yang sesuai dengan kebutuhan pengguna

- Virus: Meskipun lebih rentan terhadap virus daripada Mac (karena open source), Linux masih memiliki sangat sedikit virus.

Kekurangan:

- Rumit: Meskipun beberapa distro cukup mudah digunakan, kebanyakan dari mereka akan membutuhkan banyak pengetahuan komputer untuk membuatnya bekerja.
- Kompatibilitas: Seperti Mac, yang hanya mewakili beberapa persen pangsa pasar, Linux tidak memiliki banyak program dan game seperti Windows.
- Vendor: Pengguna tidak akan menemukan banyak vendor yang menjual komputer Linux. Biasanya, Anda akhirnya harus membeli komputer Windows, memformat ulang hard drive, dan menginstal Linux sendiri.

#### **d) Android**

Android dikembangkan oleh Google Inc. Android adalah platform Open Source, yang diturunkan dari Linux yang didukung oleh Google, bersama dengan pengembang perangkat keras dan perangkat lunak utama (seperti Intel, HTC, ARM, dan eBay, untuk beberapa nama), yang membentuk Open Aliansi Handset. OS ini, meskipun sangat baru, sudah memiliki sekte berikut di antara programmer yang ingin mengembangkan aplikasi untuk, Open Source, back end yang fleksibel. Android menjanjikan untuk memberi pengembang akses ke setiap aspek operasi ponsel.

Kelebihan: Multitasking, Kemudahan akses ke ribuan aplikasi, beragam pilihan telepon.

Kekurangan: Diperlukan koneksi Internet berkelanjutan, Iklan, Rentang aplikasi masih dapat diperluas.

#### **e) iOS**

System operasi seluler yang dikembangkan oleh Apple Inc. dan didistribusikan secara eksklusif untuk perangkat keras Apple.

Iphone menggunakan sistem operasi yang disebut iPhone OS, yang berasal dari Mac OS X.

Kelebihan: Integrasi Twitter Langsung, Advance Voice Recognition, Facetime untuk melakukan panggilan Video.

Kekurangan: Tidak ada dukungan flash, Tergantung pada perangkat keras Apple, Proses Persetujuan Aplikasi sebagian besar adalah kotak hitam untuk pengembang, facetime eksklusif untuk perangkat yang didukung iOS.

	Android	Linux	MACOS	Windows
Computer Architectures supported	ARMv7, ARMv8-A, x86, ARMv5TE	x86, x86-64, PowerPC, SPARC, Alpha, others	68k, PowerPC	x86, x86-64
Gui default is on	✓	⚡ depends on distribution	✓	✓
Target system type	Mobile devices, handheld devices	Desktop/Server Depends on distribution	Workstation, personal computer	Workstation, personal computer, media center, Tablet PC, embedded
File systems supported	NTFS, exFAT, FAT32	ext2, ext3, ext4, ReiserFS, FAT, ISO 9660, UDF, NFS, and others	HFS+, HFS, MFS (Mac OS 8.0 and before), AFP, ISO 9660, FAT (System 7 and later), UDF	NTFS, FAT exFAT ISO 9660, UDF; 3rd-party drivers support ext2, ext3, ReiserFS, [t 10] HFS+, FATX, and HFS (with third party driver)
User friendly for lay users	✓ very very user friendly	⚡ depends on distribution	✓	✓ very user friendly
Integrated firewall	Android Firewall	chroot, capability-based security, [s 5] seccomp, SELinux	Application Firewall	Windows Firewall
Encrypted file systems	⚡	✓	✓	✓
Security threats	yes	negligible	negligible	Huge
Shell terminal	Terminal Emulator	Bash shell powerful shell with alot of features	BASH	CMD
Kernel type	Linux kernel	Monolithic with modules	Monolithic with modules	Hybrid

Gambar 5

## Perbandingan Sistem Operasi

(<http://socialcompare.com/en/comparison/operating-system-1g1rzjcn>)

### c. Arsitektur Dasar Sistem Operasi berbasis mobile

#### 1) Java Micro Edition

##### Configuration

menggambarkan fitur minimal runtime dasar dari suatu sistem J2ME, merupakan core library, virtual machine, fitur keamanan dan jaringan.

- CDC (Connected Device Configuration)
- CLDC (Connected Limited Device Configuration)

##### Profile

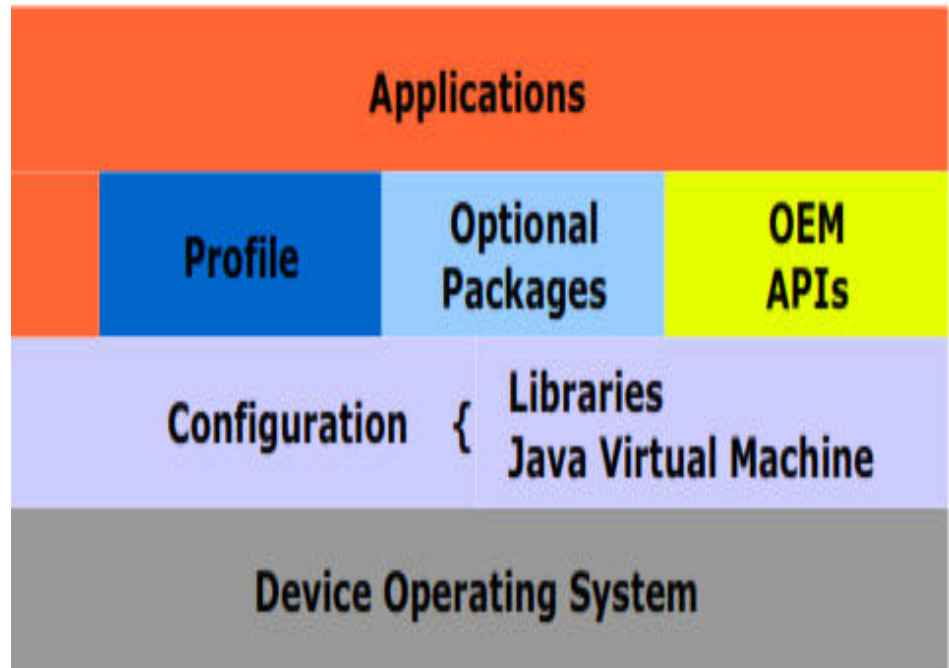
memberikan library tambahan untuk suatu kelas tertentu pada sebuah perangkat, seperti menyediakan user interface (UI) API, persistence, messaging library

- MIDP 1.0 : Aplikasi berbasis text
- MIDP 2.0 : Aplikasi berbasis Multimedia

## MIDlet

Aplikasi Java Micro Edition menggunakan teknologi MIDP

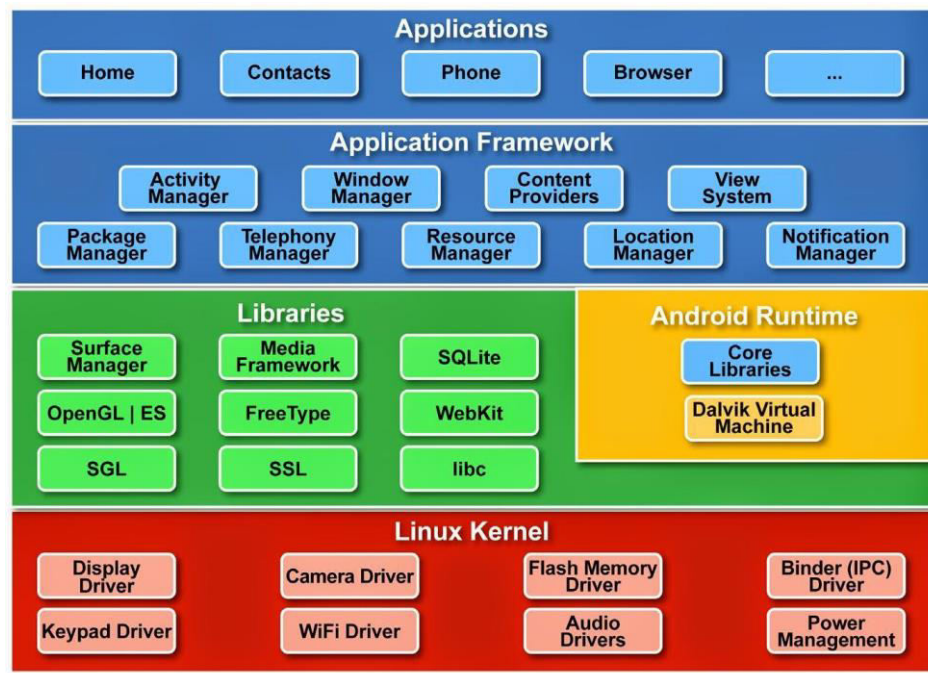
- `public void startApp() {}`
- `public void pauseApp() {}`
- `public void destroyApp(boolean unconditional) {}`



## 2) Android

Sistem operasi Android terdiri dari komponen perangkat lunak yang diatur di stack. Komponen yang berbeda dari sistem operasi android adalah:

- Kernel Linux
- Perpustakaan (libraries)
- Android Run time
- Kerangka Aplikasi (application framework)
- Aplikasi



Gambar 6

Arsitektur sistem operasi Android

#### a) Layer Applications dan Widget

Layer ini merupakan layer yang berhubungan dengan aplikasi-aplikasi inti yang berjalan pada Android OS. Seperti klien email, program SMS, kalender, browser, peta, kontak, dan lain-lain. Semua aplikasi ini dibuat dengan menggunakan bahasa Java.

layer pertama pada OS Android, biasa dinamakan layer **Applications dan Widget**. Layer ini merupakan layer yang berhubungan dengan aplikasi-aplikasi inti yang berjalan pada Android OS. Seperti klien email, program SMS, kalender, browser, peta, kontak, dan lain-lain. Semua aplikasi ini dibuat dengan menggunakan bahasa Java. Apabila kita membuat aplikasi, maka aplikasi itu ada di layer ini.

#### b) Layer Applications Framework

**Applications Framework** merupakan layer dimana para pembuat aplikasi menggunakan komponen-komponen yang ada di sini untuk membuat aplikasi mereka.

#### c) Libraries

**Libraries** merupakan layer tempat fitur-fitur android berada. Pada umumnya *libraries* diakses untuk menjalankan aplikasi. Beberapa *library* yang terdapat pada android diantaranya adalah *libraries* Media untuk memutar media video atau audio, *libraries* untuk menjalankan tampilan, *libraries* Graphic, *libraries* SQLite untuk dukungan database, dan masih banyak *library* lainnya.

#### **d) Android RunTime**

Android RunTime terletak pada level yang sama dengan *libraries*, dibagi menjadi dua bagian yaitu:

- Core Libraries : berfungsi untuk menerjemahkan bahasa Java/C
- Dalvik Virtual Machine : sebuah mesin virtual berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi pada Android secara efisien.

#### **e) Linux Kernel**

**Linux Kernel** merupakan layer tempat keberadaan inti dari operating system android. Kernel yang digunakan adalah kernel Linux versi 2.6, dan versi 3.x pada Android versi 4.0 ke atas. Kernel ini berbasis monolithic

Layer ini berisi file-file system yang mengatur system processing, memory, resource, drivers, dan sistem android lainnya. Inilah yang membuat file sistem pada Android mirip dengan file sistem pada sistem operasi berbasis Linux.

### **Fitur Perangkat Keras Android**

#### **Touchscreen**

Perangkat Android memiliki fitur layar sentuh (touchscreen) yang memberikan beberapa kemungkinan bagi pengguna untuk berinteraksi dengan aplikasi dengan menggunakan jari. Pengguna dapat melakukan swipe, flip, drag, dan pinch untuk zoom. Android juga mendukung multitouch yang berarti keseluruhan layar dapat disentuh dengan satu atau lebih jari pada saat yang bersamaan.

#### **GPS**



Sistem operasi Android mendukung GPS yang memungkinkan developer untuk mengakses lokasi pengguna. Contoh aplikasi yang memanfaatkan GPS adalah Aplikasi Peta (Map) yang menunjukkan lokasi pengguna dan memberikan petunjuk untuk menuju suatu lokasi.

#### **Accelerometer**

Android mendukung Accelerometer, yaitu perangkat yang digunakan untuk mengukur percepatan. Accelerometer dapat memberitahukan apabila suatu Perangkat Android bergerak, atau terguncang, atau berbalik arah posisinya.

#### **SD Card**

Android memiliki fitur yang memungkinkan pengguna atau aplikasi untuk mengakses (menyimpan atau membuka) file pada SD Card. SD Card merupakan media penyimpanan medium yang digunakan Perangkat Android dan beberapa perangkat mobile lain non Android sebagai media penyimpanan.

### **3) IOS**

**IOS** (*Iphone Operating System*) adalah sistem operasi perangkat keras dari Apple. Awalnya dibuat hanya untuk Iphone. iOS kemudian berkembang hingga bisa mendukung perangkat Apple yang lain seperti IPod, touch,iPad dan Apple TV. Apple tidak melisensikan iOS untuk digunakan di perangkat keras lain. Hal ini berbeda dengan Android yang bisa ditemukan di berbagai merk perangkat genggam.

iPhone OS memiliki empat lapisan abstraksi: lapisan OS Inti, lapisan Layanan Inti, lapisan Media, dan lapisan Cocoa Touch. Sistem operasi ini menggunakan sekitar 500megabyte penyimpanan perangkat.

#### **Lapisan Cocoa Touch**

Lapisan teratas dari layer OS iPhone dan berisi kerangka kerja yang paling sering digunakan oleh pengembang aplikasi iPhone. Cocoa Touch

terutama ditulis dalam Objective-C, dan ini didasarkan pada standar Mac OS X Cocoa API.

### **Lapisan Media**

Lapisan kedua dari bagian atas layer, menyediakan OS iPhone dengan audio, video, animasi dan kemampuan grafis. Seperti halnya lapisan lain dari layer OS iPhone, lapisan Media terdiri dari sejumlah kerangka yang dapat digunakan saat mengembangkan aplikasi iPhone.

### **Lapisan Layanan Inti**

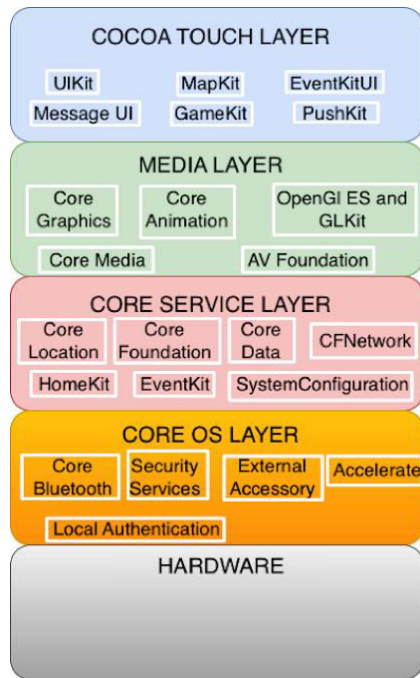
Lapisan ketiga iPhone menyediakan banyak fondasi di mana lapisan di atas dibangun.

### **Lapisan OS Inti**

Lapisan OS Inti adalah lapisan bawah dari layer OS iPhone dan langsung berada di atas perangkat keras perangkat. Lapisan ini menyediakan berbagai layanan termasuk jaringan tingkat rendah, akses ke aksesori eksternal dan layanan sistem operasi dasar yang biasa seperti manajemen memori, penanganan sistem file dan untaian.

### **Perangkat Keras iPhone**

Perangkat keras dikelola oleh OS iPhone dan menyediakan teknologi yang diperlukan untuk mengimplementasikan aplikasi asli di telepon. OS dikirimkan dengan beberapa aplikasi sistem seperti Mail, Safari, Phone, yang memberikan layanan standar kepada pengguna.



Gambar 7

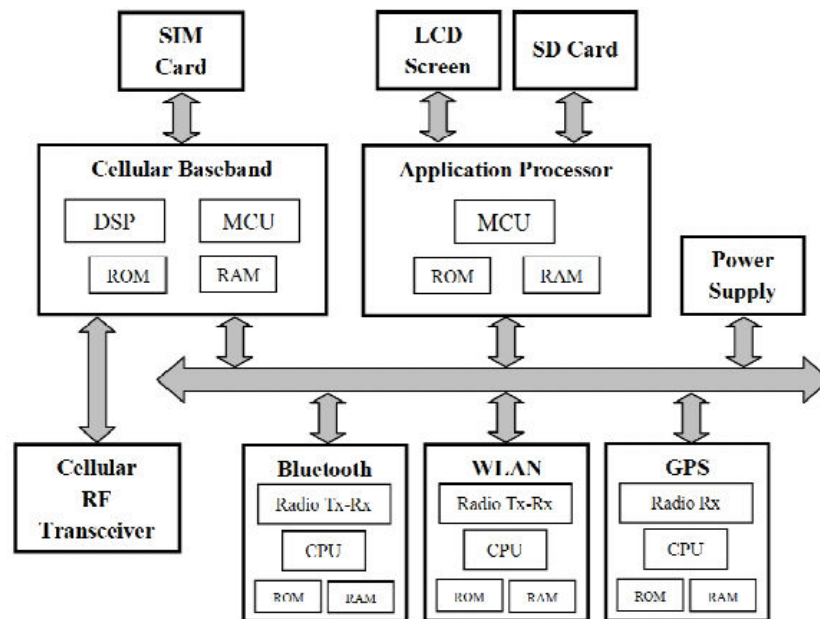
Arsitektur sistem operasi IOS

<https://codeingwithios.blogspot.com/2017/09/ios-layered-architecture.html>

#### d. Perangkat keras Mobile

Setiap smartphone modern saat ini menggunakan Arsitektur Sistem pada Chip (System on chip/SoC) dengan 3 komponen utama berikut:

- 1) Prosesor aplikasi yang mengeksekusi perangkat lunak aplikasi pengguna dengan instruksi dari middleware dan sistem operasi (OS)
- 2) Sebuah prosesor baseband (atau modem) dengan komponen OS-nya sendiri yang melakukan transmisi radio baseband dan penerimaan audio, video, dan data
- 3) Berbagai perangkat periferal untuk antarmuka pengguna



Gambar 8  
Arisitektur Perangkat mobile

### Security (Enkripsi)

Enkripsi merupakan cara untuk menjaga kerahasiaan data dengan mengubah data yang dapat dibaca (*plaintext*) menjadi tidak dapat dibaca (*chipertext*) menggunakan algoritma dan kunci tertentu. Enkripsi ini merupakan cara agar data yang ada pada *smartphone* kita terjaga kerahasiaannya. Beberapa sistem operasi memiliki fitur enkripsi sebagian yang mana maksudnya yaitu *user* dapat memilih data mana yang ingin dijaga kerahasiaannya.

 <p>Versi 3.0 dan selanjutnya memiliki fitur enkripsi <i>file system</i> (enkripsi menggunakan algoritma AES128, CBC, SHA256)</p>	 <p>iOS4 dan selanjutnya memiliki fitur enkripsi dengan algoritma AES256 untuk mengenkripsi semua data yang tersimpan di memori</p>	 <p>Memiliki beberapa algoritma enkripsi seperti AES, SHA1, SHA256, HMACSHA1, HMACSHA256</p>	 <p>Enkripsi menggunakan kunci enkripsi simetrik saat transfer data antara <i>Blackberry device</i> dan <i>Blackberry enterprise serve</i></p>
--	--	--	---

## **Menentukan platform sistem operasi yang sesuai kebutuhan user**

### **a. Cara merancang kebutuhan user berdasarkan spesifikasinya**

Agar proyek perangkat lunak berhasil, kebutuhan (*requirement*) harus dikumpulkan dan dianalisis secara profesional. Dokumen Persyaratan Pengguna (*User Requirement Document*) atau Spesifikasi kebutuhan pengguna (*User Requirement Specification/URS*) adalah dokumen yang biasanya digunakan dalam rekayasa perangkat lunak yang menentukan harapan pengguna terhadap apa yang dapat dilakukan oleh perangkat lunak. URS merupakan dokumen kunci di seluruh siklus hidup pengembangan sistem yang diperlukan untuk bisnis (perlindungan investasi) dan alasan regulasi (mendefinisikan tujuan yang dimaksudkan).

Persyaratan harus spesifik dan sesuai untuk sistem yang diinginkan.

Prinsip SMART umumnya harus diikuti ketika menulis persyaratan

- *Specific* (Spesifik)
- *Measurable* (Terukur)
- *Achievable* (Dapat diraih)
- *Realistic* (Realistis)
- *Testable* (Dapat diuji)

Panduan dalam menyusun URS:

- Setiap pernyataan persyaratan harus direferensikan secara unik.
- URS harus konsisten dan pernyataan persyaratan tidak boleh diduplikasi atau bertentangan.
- Fokusnya harus pada apa yang diperlukan, bukan bagaimana hal itu harus dicapai.
- Setiap persyaratan harus dapat diuji. Ini memungkinkan pengujian dirancang segera setelah URS selesai.
- URS harus ditulis sedemikian rupa sehingga pelanggan dan vendor dapat memahami dokumen.
- Persyaratan harus diprioritaskan sebagai wajib atau diinginkan.
- URS harus dimodifikasi tetapi perubahan harus di bawah prosedur kontrol formal.

Dalam urutan kronologis, tujuh topik mendasar yang harus dieksplorasi sebagai bagian dari proses pengumpulan persyaratan adalah:

1. Output yang harus dihasilkan oleh aplikasi
2. Input yang akan masuk ke aplikasi perangkat lunak
3. File-file logis yang harus dikelola oleh aplikasi
4. Entitas dan hubungan yang akan ada di file logis dari aplikasi
5. Jenis pertanyaan yang dapat digunakan dengan aplikasi
6. Antarmuka antara aplikasi dan sistem lain
7. Algoritma utama yang harus ada dalam aplikasi

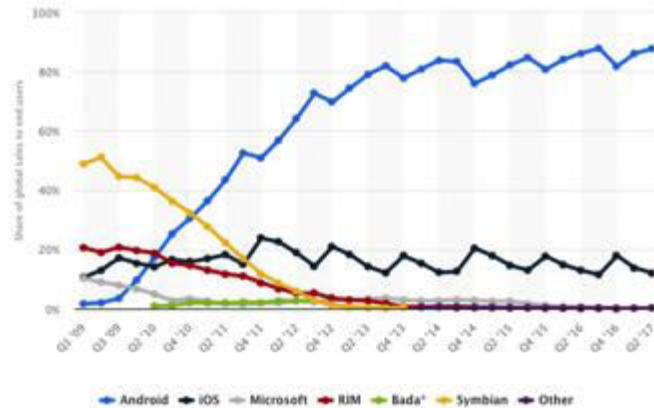
Beberapa hal yang juga perlu dieksplorasi dan diputuskan selama fase persyaratan. Beberapa di antaranya adalah persyaratan nonfungsional, dan beberapa persyaratan bisnis diperlukan untuk menentukan apakah pendanaan harus disediakan untuk aplikasi. Topik tambahan ini antara lain:

- 1) Ukuran aplikasi dalam poin fungsi dan kode sumber
- 2) Jadwal aplikasi mulai dari persyaratan hingga pengiriman
- 3) Staf tim pengembangan, termasuk spesialis utama
- 4) Biaya aplikasi berdasarkan aktivitas dan juga dalam hal biaya per poin fungsi
- 5) Nilai bisnis dari aplikasi dan laba atas investasi (ROI)
- 6) Nilai non finansial, seperti keunggulan kompetitif dan loyalitas pelanggan
- 7) Risiko utama yang dihadapi aplikasi, yaitu, penghentian, penundaan, overruns, dan sebagainya
- 8) Fitur aplikasi yang kompetitif oleh pesaing bisnis
- 9) Metode pengiriman, seperti SOA, SaaS, disk, unduhan, dan sebagainya
- 10) Rantai suplai aplikasi, atau aplikasi terkait di hulu atau hilir
- 11) Persyaratan warisan yang berasal dari aplikasi yang lebih lama digantikan

- 12) Hukum dan peraturan yang memengaruhi aplikasi (yaitu, undang-undang perpajakan, privasi, dll.)
- 13) Tingkat kualitas dalam hal cacat, keandalan, dan kemudahan penggunaan kriteria
- 14) Fitur penanganan kesalahan dalam kasus kesalahan pengguna atau pemadaman listrik, dan sebagainya
- 15) Ketentuan garansi aplikasi dan tanggapan atas klaim garansi
- 16) Platform perangkat keras tempat aplikasi akan beroperasi
- 17) Platform perangkat lunak (s), seperti sistem operasi dan basis data
- 18) Kriteria nasionalisasi, atau jumlah versi bahasa asing
- 19) Kriteria keamanan untuk aplikasi dan basis data pendampingnya
- 20) Kriteria kinerja, jika ada, untuk aplikasi
- 21) Persyaratan pelatihan atau bentuk materi tutorial yang mungkin diperlukan
- 22) Prosedur penginstalan untuk memulai dan menginisialisasi aplikasi
- 23) Kriteria penggunaan kembali untuk aplikasi dalam hal kedua bahan yang digunakan kembali ke dalam aplikasi dan juga apakah fitur aplikasi dapat ditujukan untuk digunakan kembali berikutnya oleh aplikasi hilir
- 24) Kasus penggunaan atau tugas utama yang diharapkan dapat dilakukan oleh pengguna melalui aplikasi
- 25) Aliran kontrol atau urutan informasi yang bergerak melalui aplikasi
- 26) Kemungkinan persyaratan masa depan untuk rilis selanjutnya
- 27) Tingkat bahaya dari setiap persyaratan yang mungkin berpotensi “beracun”
- 28) Harapan hidup aplikasi dalam hal kehidupan layanan sekali digunakan
- 29) Proyeksi total biaya kepemilikan (TCO) dari aplikasi
- 30) Frekuensi rilis untuk fitur dan perbaikan baru (setiap tahun, setiap bulan, dll.)

**b. Cara menentukan platform sistem operasi untuk mengembangkan aplikasi mobile**

Hal yang perlu dilakukan saat membuat aplikasi mobile adalah memilih sistem operasi. Banyak platform yang ada saat ini seperti pada gambar 9, dalam buku informasi ini dibahas pasar platform seluler saat ini yaitu iOS dan Android.



Gambar 9

Pasar global Sistem Operasi mobile 2009 – 2017

Ketika memilih platform perlu mempertimbangkan hal-hal berikut:

- a. **Audiens target.** Lakukan penelitian dan buat profil pelanggan. Perhatikan kriteria geografis karena mungkin yang paling penting bagi Anda. Platform sangat bervariasi menurut wilayah.



Gambar 10

Peta pengguna Andorid dan iOS di dunia

- b. **Perilaku pengguna (*User behavior*)**

Perilaku pengguna berbeda-beda tergantung pada OS dan juga negara. Jika mengembangkan aplikasi gratis tanpa monetisasi apa pun, maka mungkin tidak penting untuk mengetahui di mana orang cenderung



membeli aplikasi dan melakukan pembelian dalam aplikasi. Pengguna Apple lebih cenderung menghabiskan uang untuk aplikasi. Pengguna Android cenderung menggunakan aplikasi gratis dan tidak mau membeli apa pun dari Google Play Store atau melakukan pembelian dalam aplikasi.

**c. fitur (*Features*)**

Pertimbangkan fitur yang diperlukan serta pengalaman pengguna dan bandingkan dengan kemampuan platform tertentu. Jika ingin mengembangkan aplikasi khusus dengan tampilan dan nuansa asli, pertimbangkan solusi seperti iOS, Android, WP. Aplikasi asli selalu berkinerja terbaik.

Jika mengikuti pendekatan “run-once-run-everywhere” lebih lanjut, pertimbangkan pengembangan lintas platform. Namun perlu disadari bahwa tidak semua fitur, yang dapat dengan mudah diimplementasikan, didukung oleh solusi lintas platform.

**d. Biaya pengembangan (*The cost of development*)**

Harga pengembangan aplikasi bergantung pada fitur dan desain aplikasi, bukan pada bahasa pemrograman atau sistem operasi.

Untuk mendapatkan perkiraan biaya pembangunan yang tepat, Anda dapat melipatgandakan tarif per jam tim Anda dengan perkiraan waktu pengembangan. Tidak ada cara lain untuk menentukan harga tanpa membuat perhitungan untuk proyek tertentu.

**e. Perangkat yang akan didukung (*Devices you need to support*)**

Perangkat yang didukung Android sejumlah 160 item dan terus berusaha untuk meningkatkan jumlah ini, Apple memiliki strategi yang berbeda. iOS 9 mendukung 15 perangkat, di antaranya terdapat berbagai versi iPad, iPhone dan iPod. Melihat persaingan dua pemegang pasar utama ini, jelas bahwa itu merupakan tantangan untuk menyesuaikan aplikasi untuk setiap gadget. Windows Phone 8.1 berfungsi pada 100 perangkat, sementara Blackberry hanya didukung oleh 10. Windows phone dan blackberry sudah semakin menurun, beberapa aplikasi paling populer seperti WhatsApp menolak untuk

mendukungnya. Untuk mengembangkan aplikasi lintas platform, perlu menyesuaikan dengan semua perangkat itu secara bersamaan.

**f. Biaya tambahan terkait dengan pemrograman (*Additional costs related to programming*)**

Salah satu hal yang perlu dipertimbangkan adalah biaya tambahan yang terkait dengan pengembangan aplikasi. Untuk mengirimkan aplikasi ke toko aplikasi, Anda harus membeli akun pengembang. Google Play mengenakan biaya \$ 25 untuk layanan ini, Apple membenakan biaya \$ 100 untuk akun pengembang dan \$ 299 untuk Program Perusahaan.

**g. Moderasi (*Moderation*)**

Aplikasi iOS selalu sangat dimoderasi. Apple memiliki aturan tertentu tentang bagaimana aplikasi harus berperilaku agar dapat dikirimkan ke pasar mereka. Jika tidak sesuai dengan mereka, kemungkinan besar akan ditolak. Anda dapat dengan mudah menemukan Pedoman Peninjauan App Store di situs web resmi mereka. Misalnya, mereka tidak akan menerima aplikasi yang mogok atau berisi spanduk ID kosong. Apple membutuhkan waktu hingga 10 hari untuk memeriksa sebuah proyek dan menerima atau menolak pengirimannya.

Lain halnya dengan Google Play yang juga memiliki pedoman tertentu, tetapi pasar ini terlibat dalam rumor buruk tentang banyak aplikasi pembajakan. Ini dapat dijelaskan oleh sistem moderasi yang lemah. Selain itu, pengguna dapat dengan mudah mengunduh file .apk dari aplikasi apa pun dan menginstalnya ke perangkatnya.

**h. Harapan pengguna (*User expectations*)**

Tergantung pada platform dan pembuatnya, pengguna memiliki harapan tertentu dari aplikasi di perangkat mereka. Jika ingin membuat pengguna merasa puas, pertimbangkan harapan mereka. Misalnya, Apple terkenal dengan kualitas dan inovasi kelas atas. Pengikut merek sangat setia dan berkomitmen terhadap layanan perusahaan dan berharap untuk mengalami desain trendi dan kegunaan ekstrem dari aplikasi yang mereka gunakan. Mereka bersedia menjadi yang pertama dalam hal inovasi dan siap membayar

lebih untuk mendapatkannya. Pengguna aplikasi Android tidak suka membelanjakan uang untuk aplikasi atau pembelian dalam aplikasi. Mereka lebih suka desain bergaya material dan gratisan.

#### **i. Technical specifications**

Saat merancang aplikasi Android, desainer umumnya mengikuti Pedoman Desain Material (*Material Design Guidelines*) dari Google, yang menentukan ukuran tabel, jarak ikon, rekomendasi palet warna, dan banyak hal lainnya. Untuk iOS, desainer dapat merujuk pada Pedoman Antarmuka Manusia (*Human Interface Guidelines*) dari Apple. Panduan ini tidak mendikte detail, melainkan menawarkan rekomendasi untuk mencapai desain intuitif.

Tiga perbedaan utama antara Pedoman Desain Material dan Pedoman Antarmuka Manusia:

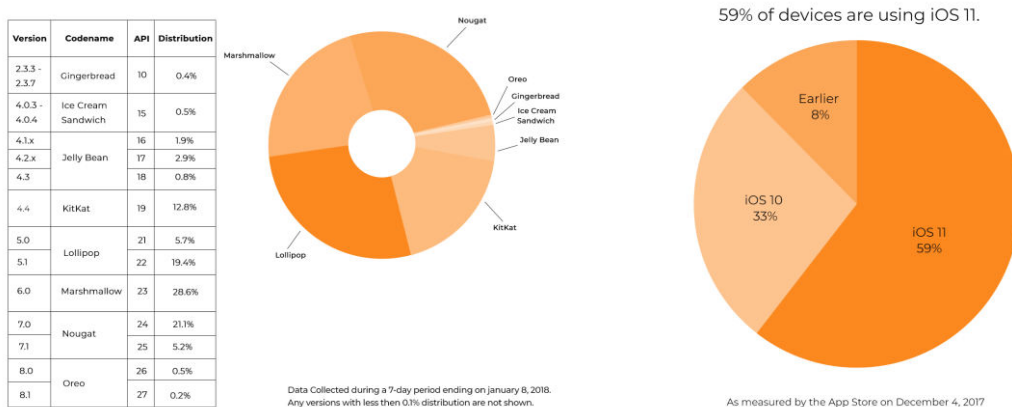
- 1) Kedalaman dan persepsi. Google menganjurkan desain material, yang menyerupai lembaran kertas yang ditumpuk di atas satu sama lain. Apple lebih memilih desain yang memiliki kedalaman. Di iOS, garis tipis dan margin kabur membuat elemen tampak mengapung di ruang mereka sendiri.
- 2) Animasi. Google ingin animasi untuk meningkatkan pengalaman pengguna dan memeriahkan desain; lampu dan getaran menciptakan pengalaman pengguna yang lebih praktis. Apple melihat animasi sebagai sesuatu dengan utilitas murni. Di iOS, animasi harus sederhana dan diskrit.
- 3) Navigasi. Google yang menggunakan navigasi agak santai. Itu bisa di mana saja; hanya membuatnya jelas. Kebebasan ini dapat meningkatkan kreativitas tetapi juga dapat mengarah pada desain yang tidak begitu ramah pengguna. Apple membatasi opsi navigasi, membuat perancang berpikir dengan hati-hati tentang fungsi apa yang harus disertakan. Tetapi pembatasan ini biasanya mengarah pada desain yang nyaman dan intuitif.

#### **j. Fragmentasi perangkat dan platform**

Android adalah platform terbuka. Pengembang dapat menggunakan kode sumber Android untuk menyesuaikan sistem operasi Android.

Lingkungan terbuka merupakan kelebihan namun dapat membuat pengembangan mobile menjadi bermasalah karena:

- 1) Produsen *smartphone* cenderung mengubah Android agar sesuai dengan perangkat mereka, yang memaksa pengembang untuk menyesuaikan aplikasi agar mereka bekerja pada *smartphone* dan tablet tertentu;
- 2) Beberapa perangkat memiliki komponen yang lebih murah (seperti prosesor dan kamera), yang dapat menyebabkan bug atau memperlambat aplikasi, membuat aplikasi kurang menarik bagi sebagian pengguna.



Gambar 11  
Platform OS Android dan iOS

### Menjelaskan bahasa pemrograman berbasis mobile

Pemrograman Mobile adalah pemrograman yang ditujukan untuk pembuatan aplikasi diperangkat mobile. Diterapkan di mobile/Handphone atau Laptop.

Contoh pemrograman mobile :

- Android menggunakan beberapa Bahasa pemrograman yaitu C++, Java, Kotlin dll
- IOS menggunakan bebrapa Bahasa pemrograman yaitu C++, Objective-C, Swift
- Windows Phone, bahasa yang digunakan adalah bahasa pemograman C# dan juga Visual Basic. Dan untuk mendesign dan membuat

animasinya kita bisa menggunakan Silverlight yang ada di Microsoft Expression Blend.

- Blacberry menggunakan bebrapa Bahasa pemrograman yaitu C++, Java.

Beberapa faktor utama yang mempengaruhi pemilihan bahasa pemrograman tertentu adalah:

- Keseluruhan strategi pengembangan aplikasi seluler *native*, *cross platform*, *hybrid* dan sebagainya
- Skill set dari tim pengembangan
- Sasaran pengguna aplikasi dan dukungan platform yang dipilih
- Kebutuhan khusus aplikasi.
- Kategori aplikasi : Utility, games, B2B atau B2C dan lainnya.

#### **a. menentukan jenis bahasa pemrograman pemrograman berbasis mobile**

##### **1) Bahasa pemrograman**

Bahasa pemrograman adalah kosakata dan seperangkat aturan gramatikal (sintaksis) untuk menginstruksikan komputer untuk melakukan tugas-tugas tertentu dan digunakan untuk membuat program komputer.

Program harus diubah menjadi bahasa mesin sehingga komputer dapat memahaminya. Ada dua cara untuk melakukan ini yaitu kompilasi (*Compile*) program menafsirkan (Interpret).

Bahasa Pemrograman mana yang sesuai dengan Strategi Pengembangan Aplikasi Mobile? Jika kita ingin mendvelop aplikasi IOS maka pelajari Bahasa pemrograman yang digunakan IOS.

Jika kita ingin mendvelop aplikasi IOS maka pelajari Bahasa pemrograman yang digunakan IOS

Sama halnya jika kita ingin mendvelop aplikasi Android maka kita pelajari Bahasa pemrograman yang support untuk android

Tetapi jangan coba coba untuk mendvelop aplikasi Blackberry, sebab smatphone tersebut sudah jarang dipasarkan,

Ketika kita ingin menjadi pengembang aplikasi, lebih baik untuk kedepan kita harus menguasai dasar dan konsep dari mobile programming

Jika dilihat secara keseluruhan, Bahasa programming C,C++ terdapat di semua platform mobile application, begitu pula Bahasa pemrograman Java

Hanya saja standar pemrograman antar platform memiliki beberapa perbedaan, misalnya android dengan apple.

#### **b. membandingkan perbedaan bahasa pemrograman berbasis mobile**

Perangkat seluler adalah platform yang tumbuh paling cepat di industri TI. Karena semakin banyak orang yang memilih Apps untuk memperluas bisnis mereka di tingkat global, industri pengembangan aplikasi seluler mendapatkan popularitas yang sangat besar. Tujuan utama dari pembuat perangkat lunak adalah untuk mengembangkan aplikasi yang berjalan pada smartphone, tablet dan *smartwatches* yang baru.

Pemilihan bahasa pemrograman yang tepat untuk digunakan untuk pengembangan aplikasi Mobile menjadi perhatian utama bagi para pengembang aplikasi. Pemilihan bahasa pemrograman tidak hanya tergantung pada bahasa tetapi juga tergantung pada sistem operasi yang berbeda yang mendukung bahasa-bahasa pemrograman tersebut. Tantangannya adalah memilih bahasa pemrograman yang tepat. Sistem Operasi Mobile dan bahasa pemrograman:

- Java Micro Edition (sebelumnya J2ME)
- NET Compact Framework : C++, C#, VB.NET untuk Windows Mobile
- Qualcomm's BREW : C atau C++
- Symbian : C++
- BlackBerry : Java
- Android : Java

- iPhone : Objective-C, Swift

### 1) Swift

Swift, dirilis oleh Apple Inc. Pada Juni 2014 adalah bahasa pemrograman utama yang digunakan untuk mengembangkan aplikasi iOS dan OSX. Swift adalah salah satu bahasa yang cepat berkembang. Permintaan untuk pengembang Swift telah meningkat secara signifikan dan menyebabkan mereka menjadi pengembang yang paling banyak dipekerjakan. Swift berfungsi di Linux dan hanya tersedia untuk pengembangan iOS

Fitur-fitur utama:

- Open source.
- Mudah jika sudah memahami Objective-C.
- masa depan dari pengembangan iOS
- Perawatan mudah
- Swift membutuhkan lebih sedikit pengkodean (*coding*) dibandingkan dengan bahasa lain

### 2) Python

Python adalah bahasa tingkat tinggi yang kuat yang mudah dipelajari dan bagus dalam hal keterbacaan (*readability*). Semua jenis aplikasi seluler dapat dibuat dengan Python. Python adalah salah satu bahasa yang paling mudah dipelajari untuk membuat aplikasi Android dan Desktop.

Situs terkenal seperti Dropbox, BitTorrent, Instagram, dan Quora dibuat dengan Python.

Fitur-fitur utama:

- Berorientasi objek dan bahasa interaktif
- Mudah dipelajari dan mudah dibaca
- Berjalan di Windows, Mac, Unix, dan Linux
- Scalable
- Mendukung aplikasi GUI

### 3) Java

It is most suitable language because it runs on all the platforms including the all the versions of Android. Apart from Android apps, developers can create server apps, web apps, games, websites, big data technology, etc.

Java adalah bahasa pemrograman pengembangan aplikasi seluler yang paling banyak digunakan. Jadi seseorang memiliki pengetahuan tentang Java, niscaya akan dapat membuat aplikasi Android dari semua jenis. Java adalah bahasa yang paling cocok (*suitable*) karena dapat berjalan di semua platform termasuk semua versi Android. Selain aplikasi Android, pengembang dapat membuat aplikasi server, aplikasi web, game, situs web, teknologi big data, dll.

Fitur-fitur utama:

- Bahasa berorientasi objek.
- Mendukung integrasi API.
- Berjalan di semua platform
- Mudah dipelajari dan dibaca.
- Java menyediakan *rich documentation*.
- Banyak pakar tersedia di komunitas Android.
- Powerful IDEs
- IDE membuat coding mudah dan bebas dari kesalahan.

#### **4) PHP**

Hypertext Preprocessor (PHP) adalah open source bahasa scripting server-side (*server-side scripting open source language*). Sengaja dikembangkan untuk membuat situs web, tetapi sekarang digunakan untuk pengembangan tujuan umum. Ini membantu dalam membuat situs web dinamis, aplikasi web, dan semua jenis aplikasi seluler. Banyak Sistem Manajemen Konten menampilkan PHP sebagai bahasa pemrograman mereka.

Dengan PHP, seseorang dapat membuat GUI, aplikasi Facebook, PDFs, WordPress dan plugin Joomla, situs web e-commerce dan banyak lagi.

Fitur-fitur utama:

- Open-source
- Uses Object-oriented and procedure oriented approaches
- Independent of platform
- Easy-to-learn



- Integration with database is easy
- Have numerous uses
- Server compatibility

## 5) JavaScript

JavaScript adalah bahasa pemrograman tingkat tinggi yang diinterpretasi. Mendukung pemrograman berorientasi objek dan fungsional. JavaScript dijalankan oleh browser yang digunakan untuk mengembangkan dan mengontrol halaman web. Aplikasi seluler dapat dibuat hanya dengan JavaScript bila digunakan dengan CSS, HTML, dan AJAX untuk membuat aplikasi seluler. Beberapa *framework* membuat aplikasi JavaScript profesional seperti jQuery Mobile, angular js, Ionic, dll.

Fitur-fitur utama:

- Salah satu bahasa yang paling mudah.
- Cepat dan efisien.
- Kode satu kali dan kemudian dijalankan di semua platform
- Dieksekusi pada sisi-klien.
- Menghemat bandwidth dengan tidak menggunakan server.
- Digunakan membuat animasi dan dinamis pada situs web yang membosankan.

## 6) Kotlin

Kotlin terhitung baru dalam dunia bahasa pemrograman untuk pengembangan aplikasi seluler. Dianggap sebagai era baru dalam pengembangan aplikasi Android, merupakan masa depan. Google mendukung Kotlin sebagai bahasa pemrograman pengembangan aplikasi Android "first-class". Basis Kotlin adalah Java Virtual Machine (JVM), diharapkan menjadi hal besar dalam pemrograman untuk Android. Karena berbasis JVM maka Kotlin dapat dijalankan pada hampir semua Platform OS.

Fitur-fitur utama:

- Open-source
- Membutuhkan lebih sedikit pengkodean

- Mudah dipelajari dan beradaptasi
- Aman digunakan karena keamanannya yang bebas gangguan
- Tanpa biaya adopsi
- Menyediakan fungsi ekstensi
- Java Interoperability penuh
- Hasil Aplikasi dapat dijalankan pada Platform Android.

### **Kelebihan Kotlin**

Kelebihan dari Kotlin dibandingkan dengan bahasa pemrograman lainnya (dalam hal ini Java) dapat disimpulkan ke dalam poin-poin berikut :

**Concise** : Kotlin mampu mengurangi *boilerplate of code* atau tingkat kerumitan dari kode yang biasa kita tulis, ketika menggunakan bahasa Java

**Safe** : Kotlin mampu menjamin bahwa setiap *syntax* yang kita tulis secara proses kompilasi dapat mencegah kemungkinan terjadinya *error*, misalnya mampu mencegah terjadinya **NullPointerException** ketika kita *coding* menggunakan bahasa Java

**Versatile** : Kotlin sejatinya sama seperti Java, karena memang Kotlin itu sendiri diturunkan dari bahasa induknya, yaitu Java. Sehingga Kotlin juga dapat dipakai dalam pengembangan aplikasi di *Web* maupun *Mobile*

**Interoperable** : Kotlin tidak sama seperti bahasa Java turunan lainnya (misal; Scala ataupun Clojure) yang tidak dapat dijalankan bersamaan dengan kode yang kita tulis menggunakan Java. Kotlin mampu membaca kode lama atau library yang kita gunakan atau kita tulis dengan bahasa Java dan begitupun sebaliknya

Sebelum memilih salah satu bahasa pengembangan aplikasi seluler yang ada, pertimbangkan persyaratan klien dan platform yang ditargetkan. Juga, pilih bahasa pemrograman dengan cara membangun kemampuan pengembang yang membuat aplikasi seluler berkualitas tanpa kompromi.

**Beberapa faktor utama yang mempengaruhi pemilihan bahasa pemrograman tertentu adalah:**

- Keseluruhan strategi pengembangan aplikasi seluler *native, cross platform, hybrid* dan sebagainya
- Skill set dari tim pengembangan
- Sasaran pengguna aplikasi dan dukungan platform yang dipilih
- Kebutuhan khusus aplikasi.

Kategori aplikasi : Utility, games, B2B atau B2C dan lainnya.

### **Komponen Bantu Pengembangan:**

#### **Hardware:**

Komputer development, mobile device

#### **Software /Tool Pengembangan:**

- Windows Phone : Windows Phone SDK, Visual Studio Express Expression Blend (hanya di windows)
- iOS : xCode , iOS SDK
- Android : Java JDK, Android SDK, Eclipse IDE, Android ADT, Android Studio IDE
- Java Micro Edition : Java JDK, Java Micro Edition SDK, Netbeans IDE, Eclipse
- Blackberry : Eclipse, Blackberry Plugin, Blackberry Simulator

#### **Pemilihan Platform :**

Android, iOS, Blackberry, Windows Phone, Symbian

## **1. Cara menghasilkan alur program untuk pembuatan aplikasi berbasis mobile**

### **a. Alur Program**

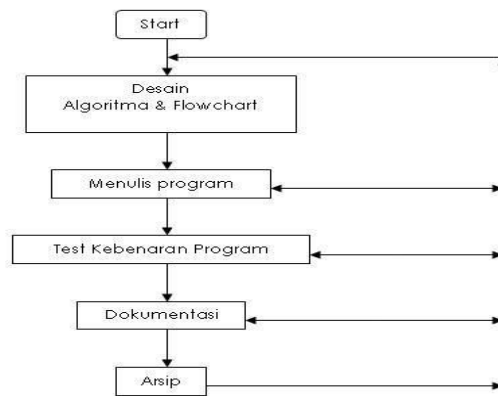
Alur program adalah suatu bagan dengan simbol tertentu yang menggambarkan urutan proses secara detail dan hubungan antara suatu proses dengan proses lainnya dalam suatu program.

Dalam membuat diagram alur tidak ada rumus dasar yang mutlak karena pada dasarnya adalah gambaran hasil pemikiran (logika) manusia dalam menganalisa masalah.

b. Cara Pemecahan masalah

Proses pemecahan masalah dengan algoritma tertentu hingga menjadi program dapat dibagi dalam sembilan tahap (Thomspson Susabda Ngoen, 2008:6) :

- 1) Mendefinisikan Masalah. Masalah yang ingin dipecahkan harus jelas lingkupnya.
- 2) Membuat Model. Model yang dimaksud adalah model (bentuk) matematis yang dapat digunakan untuk memecahkan masalah, misalnya apakah harus dilakukan pengurutan terhadap data, apakah menggunakan perhitungan kombinatorik dan sebagainya.
- 3) Merancang algoritma (flowchart/pseudocode). Yang perlu dipertimbangkan dalam merancang algoritma adalah masukannya bagaimana rincian prosesnya, apa keluarannya.
- 4) Menulis program. Merubah algoritma menjadi program (source code) sesuai dengan bahasa pemrograman tertentu.
- 5) Mengubah source code menjadi executable code melalui proses compiling.
- 6) Memeriksa hasil compiling, jika salah maka kembali ke tahap empat.
- 7) Menjalankan program (run) untuk diuji kebenarannya dengan menggunakan berbagai data
- 8) Memperbaiki kesalahan  
Apabila hasilnya tidak sesuai dengan keluaran yang diharapkan, kesalahan mungkin terjadi saat konversi rancangan algoritma menjadi program, atau salah rancang algoritma, atau salah menentukan model, atau salah mendefinisikan masalah. Ulangi langkah yang sesuai.
- 9) Membuat dokumentasi



Gambar 18

Tahapan pembuatan algoritma hingga program

## 2. Cara menentukan konsep variabel dan konstanta dalam salah satu bahasa pemrograman

### a. Dasar Java Android

Java adalah bahasa pemrograman yang terkenal, dibuat pada tahun 1995.

Dimiliki oleh Oracle. Java dapat digunakan untuk :

- Aplikasi Mobile (spesial Android apps)
- Aplikasi Desktop
- Aplikasi Web
- Web Server dan Application server
- Game
- Koneksi Basisdata, dll

Beberapa alasan menggunakan Java?

- Java berfungsi pada berbagai platform (Windows, Mac, Linux, Raspberry Pi, dll.)
- Salah satu bahasa pemrograman paling populer di dunia
- Mudah dipelajari dan mudah digunakan
- Open-source dan gratis
- Aman, cepat dan kuat
- Memiliki dukungan komunitas yang sangat besar (puluhan juta pengembang)

### b. Sintaks Java

kode berikut untuk mencetak "Hello World" ke layar:

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

#### c. Java Comment

Komentar dapat digunakan untuk menjelaskan kode Java, dan membuatnya lebih mudah dibaca. Juga dapat digunakan untuk mencegah eksekusi ketika menguji kode alternatif. Komentar baris tunggal dimulai dengan dua garis miring (//). Teks apa pun antara // dan akhir baris diabaikan oleh Java (tidak akan dieksekusi).

#### d. Java Multi-line Comments

Komentar banyak baris dimulai dengan /\* dan diakhiri dengan \*/.

Teks apa pun antara /\* dan \*/ akan diabaikan oleh Java.

```
// This is a comment  
/* The code below will print the words Hello World  
to the screen, and it is amazing */  
System.out.println("Hello World");
```

**Public class myclass** merupakan pembuatan class dengan nama Myclass yang bersifat Publik(bias diakses siapa saja)

**“Public static void main(string[] args)”** pembuatan void main yang akan dieksekusi dengan tipe data string

**{system.out.println(“hello world”);** merupakan isi dari void main, yaitu perintah untuk system agar mencetak kalimat “Hello World”

#### d. Konsep Variabel

Variabel adalah wadah untuk menyimpan nilai data. Ada berbagai jenis variabel, misalnya:

- String - menyimpan teks, seperti "Halo". Nilai string ditandai oleh tanda kutip ganda

- int - Integer (bilangan bulat), tanpa desimal, seperti 123 atau -123
- float - menyimpan angka floating point, dengan desimal, seperti 19,99 atau -19,99
- char - menyimpan karakter tunggal, seperti 'a' atau 'B'. Nilai-nilai Char ditandai oleh tanda kutip tunggal
- boolean - menyimpan nilai dengan dua status: benar atau salah

### **Mendeklarasikan variabel**

Untuk membuat variabel, Anda harus menentukan tipe dan memberinya nilai:

Sintaksis: *type variable = value;*

Untuk membuat variabel yang harus menyimpan teks, misalnya memuat variabel bernama nama tipe String dan berikan nilai "John":

```
String name = "John";
System.out.println(name);
```

Contoh Lain :

```
int myNum = 5;
float myFloatNum = 5.99f;
char myLetter = 'D';
boolean myBool = true;
String myText = "Hello"
```

Untuk menampilkan variabel sering digunakan metode `println ()` dan untuk menggabungkan teks dan variabel, gunakan karakter `+`:

```
String name = "John";
System.out.println("Hello " + name);
```

### **Java Identifiers**

Semua variabel Java harus diidentifikasi dengan nama unik. Nama-nama unik ini disebut **identifiers**. **Identifiers** dapat berupa nama pendek (seperti x dan y) atau nama yang lebih deskriptif (usia, jumlah, total Volume).

Aturan umum untuk membuat nama untuk variabel (unique identifiers) adalah:

- Nama dapat berisi huruf, angka, garis bawah, dan tanda dolar
- Nama harus dimulai dengan huruf
- Nama juga dapat dimulai dengan \$ dan \_ (tetapi tidak dalam tutorial ini)
- Case-sensitive ("myVar" dan "myvar" adalah variabel yang berbeda)
- Nama harus dimulai dengan huruf kecil dan tidak boleh mengandung spasi
- Kata-kata yang dicadangkan (seperti kata kunci Java, seperti int atau String) tidak dapat digunakan sebagai nama

## Tipe Data

Jenis Tipe data dibagi menjadi dua kelompok:

- 1) Tipe data primitif, seperti byte, short, int, long, float, double, boolean dan char

Tipe data primitif menentukan ukuran dan jenis nilai variabel, tidak memiliki metode tambahan.

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers from 3.4e-038 to 3.4e+038. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers from 1.7e-308 to 1.7e+038. Sufficient for storing 15 decimal digits
boolean	1 byte	Stores true or false values
char	2 bytes	Stores a single character/letter

- 2) Tipe data non-primitif - seperti String, Array, dan Class

Tipe data non-primitif disebut tipe referensi karena merujuk pada objek.

Perbedaan utama antara tipe data primitif dan non-primitif adalah:

- Tipe primitif sudah ditentukan sebelumnya (sudah ditentukan) di Java. Tipe non-primitif dibuat oleh programmer dan tidak didefinisikan oleh Java (kecuali untuk String).
- Tipe non-primitif dapat digunakan untuk memanggil metode untuk melakukan operasi tertentu, sedangkan tipe primitif tidak bisa.



- Tipe primitif selalu memiliki nilai, sedangkan tipe non-primitif bisa menjadi nol.
- Tipe primitif dimulai dengan huruf kecil, sedangkan tipe non-primitif dimulai dengan huruf besar.
- Ukuran tipe primitif tergantung pada tipe data, sedangkan tipe non-primitif memiliki semua ukuran yang sama.
- Contoh tipe non-primitif adalah String, Array, Class, Interface, dll.

### **Keyword final**

Pengubah non-akses yang digunakan untuk kelas, atribut, dan metode, yang membuatnya tidak dapat diubah (tidak mungkin diwariskan atau ditimpa)

Final pada variable

```
class Lingkaran {
    final double PI = 3.14;
}
```

Final Pada Method

```
class Kampus {
    final void tampil() {
        System.out.println("I LOVE JAVA") }
}
```

Final pada class

```
class Kampus {
    final void tampil() {
        System.out.println("I LOVE JAVA") }
}
```

### **Operator Java**

Operator Aritmatika digunakan untuk melakukan operasi matematika umum.

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value from another	$x / y$
%	Modulus	Returns the division remainder	$x \% y$
++	Increment	Increases the value of a variable by 1	$++x$
--	Decrement	Decreases the value of a variable by 1	$--x$

### 3. Cara menentukan konsep struktur kondisi dan perulangan dalam salah satu bahasa pemrograman berbasis mobile

#### a. Struktur Kondisi

Kondisi Java dan Pernyataan if

Java mendukung kondisi logis yang biasa dari matematika:

- Kurang dari:  $a < b$
- Kurang dari atau sama dengan:  $a \leq b$
- Lebih besar dari:  $a > b$
- Lebih besar atau sama dengan:  $a \geq b$
- Sama dengan  $a == b$
- Tidak sama dengan:  $a != B$

Anda dapat menggunakan kondisi ini untuk melakukan tindakan berbeda untuk keputusan yang berbeda. Java memiliki pernyataan kondisional berikut:

- ✓ Gunakan if untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan benar
- ✓ Gunakan else untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sama salah
- ✓ Gunakan else if menentukan kondisi baru untuk diuji, jika kondisi pertama salah
- ✓ Gunakan switch menentukan banyak blok kode alternatif yang akan dieksekusi

#### Pernyataan if

```
int x = 20;
int y = 18;
if (x > y) {
    System.out.println("x is greater than y");
}
```

### **Pernyataan else**

```
int time = 20;
if (time < 18) {
    System.out.println("Good day.");
} else {
    System.out.println("Good evening.");
}
```

### **Pernyataan else if**

```
int time = 22;
if (time < 10) {
    System.out.println("Good morning.");
} else if (time < 20) {
    System.out.println("Good day.");
} else {
    System.out.println("Good evening.");
}
```

### **Pernyataan Switch Java**

Gunakan pernyataan switch untuk memilih salah satu dari banyak blok kode yang akan dieksekusi.

- ✓ Ekspresi switch dievaluasi sekali.
- ✓ Nilai ekspresi dibandingkan dengan nilai setiap kasus.
- ✓ Jika ada kecocokan, blok kode terkait dijalankan.
- ✓ break dan kata kunci default adalah opsional.

### **Kata kunci break**

- ✓ Ketika Java mencapai kata kunci break, maka keluar dari blok switch.
- ✓ Ini akan menghentikan pelaksanaan lebih banyak kode dan pengujian kasus di dalam blok.
- ✓ Ketika kecocokan ditemukan, dan pekerjaan selesai, saatnya break(dihentikan). Tidak perlu pengujian lebih lanjut.
- ✓ Break dapat menghemat banyak waktu eksekusi karena "mengabaikan" eksekusi semua kode lainnya di blok switch.

### **Kata Kunci default**

Kata kunci default untuk dijalankan jika tidak ada kecocokan .

Contoh :

```
int day = 4;
switch (day) {
    case 6:
        System.out.println("Today is Saturday");
        break;
    case 7:
        System.out.println("Today is Sunday");
        break;
    default:
        System.out.println("Looking forward to the Weekend");
}
```

### **b. Struktur Perulangan**

#### **Perulangan while**

Perulangan while dikerjakan selama kondisi yang ditentukan benar.

```
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

#### **Perulangan Do/While**

Do / while loop adalah varian dari perulangan while. Perulangan ini akan mengeksekusi blok kode sekali, sebelum memeriksa apakah kondisinya benar, akan mengulang loop selama kondisinya benar.

```
int i = 0;
do {
    System.out.println(i);
    i++;
}
while (i < 5);
```

### Perulangan for

```
for (int i = 0; i <= 10; i = i + 2) {
    System.out.println(i);
}
```

- Pernyataan 1 dieksekusi (satu kali) sebelum eksekusi blok kode. Menetapkan variabel sebelum perulangan dimulai (int i = 0).
- Pernyataan 2 mendefinisikan kondisi untuk mengeksekusi blok kode. Mendefinisikan kondisi untuk menjalankan perulangan ( harus kurang dari sama dengan 10). Jika kondisinya benar, perulangan akan memulai lagi, jika itu salah, perulangan akan berakhir.
- Pernyataan 3 dieksekusi (setiap kali) setelah blok kode dieksekusi. Meningkatkan nilai (i ++) setiap kali blok kode dalam perulangan telah dieksekusi.

### Perulangan for-Each

Perulangan **for-each**, digunakan secara eksklusif untuk mengulang elemen-elemen dalam array

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
    System.out.println(i);
}
```

Java break pada perulangan for

```

for (int i = 0; i < 10; i++) {
    if (i == 4) {
        break;
    }
    System.out.println(i);
}

```

#### 4. Cara menjelaskan konsep layout dan object dalam salah satu pemrograman berbasis mobile

##### a) Konsep Layout

Layout mendefinisikan struktur visual untuk antarmuka pengguna, seperti UI sebuah aktivitas atau widget aplikasi. Anda dapat mendeklarasikan layout dengan dua cara:

- **Deklarasikan elemen UI dalam XML.**

Android menyediakan sebuah kosakata XML sederhana yang sesuai dengan kelas dan subkelas View, seperti halnya untuk widget dan layout. Keuntungan mendeklarasikan UI dalam XML

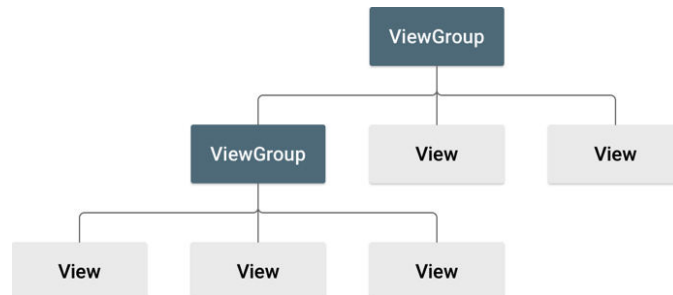
- ✓ Memungkinkan Anda memisahkan penampilan aplikasi dari kode yang mengontrol perilakunya dengan lebih baik.
- ✓ Keterangan UI Anda bersifat eksternal bagi kode aplikasi Anda, yang berarti bahwa Anda bisa memodifikasi atau menyesuaikannya tanpa harus memodifikasi dan mengompilasi ulang kode sumber.
- ✓ Misalnya, Anda bisa membuat layout XML untuk berbagai orientasi layar, berbagai ukuran layar perangkat, dan berbagai bahasa

- **Buat instance elemen layout saat runtime.**

Dalam suatu aplikasi bisa membuat objek View dan ViewGroup (dan memanipulasi propertinya) lewat program.

Layout (Tata letak) menentukan struktur untuk antarmuka pengguna (*user interface*) di aplikasi, seperti dalam suatu aktivitas. Semua elemen dalam tata letak dibangun menggunakan object hierarki View dan ViewGroup. View biasanya menampilkan sesuatu yang dapat dilihat dan berinteraksi

dengan pengguna. Sedangkan ViewGroup adalah wadah yang tidak terlihat yang mendefinisikan struktur tata letak untuk View dan objek ViewGroup lainnya, seperti yang ditunjukkan pada gambar 20.



Gambar 20

Ilustrasi hierarki tampilan, yang mendefinisikan tata letak UI

### Layout-XML

File --> New --> Project...

Choose Android --> Android Application Project

Edit res/layout/activity\_main.xml, and replace everything with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

</LinearLayout>
```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>

```

### Compile aplikasi

- 1) Masing-masing file layout XML akan dikompilasi dalam sebuah sumber daya [View](#).
- 2) Harus memuat sumber daya layout dari kode aplikasi, dalam implementasi callback [Activity.onCreate\(\)](#).
- 3) Lakukan dengan memanggil [setContentView\(\)](#), dengan meneruskan acuan ke sumber daya layout berupa:  
`R.layout.layout_file_name`.

Misalnya, jika XML layout Anda disimpan sebagai `main_layout.xml`, Anda akan memuatnya untuk Activity seperti ini:

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
}

```

### Atribut

Setiap objek View dan ViewGroup mendukung variasi atribut XML-nya sendiri. Sebagian atribut bersifat spesifik untuk objek View (misalnya, TextView mendukung atribut **textSize**). Sebagian atribut bersifat umum untuk semua objek View, karena diwarisi dari kelas Root View (seperti atribut **id**).



Atribut lain dianggap sebagai "parameter layout" yaitu atribut yang menjelaskan orientasi layout tertentu dari objek View, seperti yang didefinisikan oleh objek ViewGroup induk dari objek itu.

## ID

Objek View apa saja dapat memiliki ID integer yang dikaitkan dengannya, untuk mengidentifikasi secara unik View dalam pohon.

Bila aplikasi dikompilasi, ID ini akan diacu sebagai integer, namun ID biasanya ditetapkan dalam file XML layout sebagai string, dalam atribut id. Ini atribut XML yang umum untuk semua objek View (yang didefinisikan oleh kelas View) dan Anda akan sering sekali menggunakannya

`android:id="@+id/my_button"`

## Parameter Layout

Atribut layout XML bernama *layout\_something* mendefinisikan parameter layout View yang cocok untuk ViewGroup tempatnya berada.

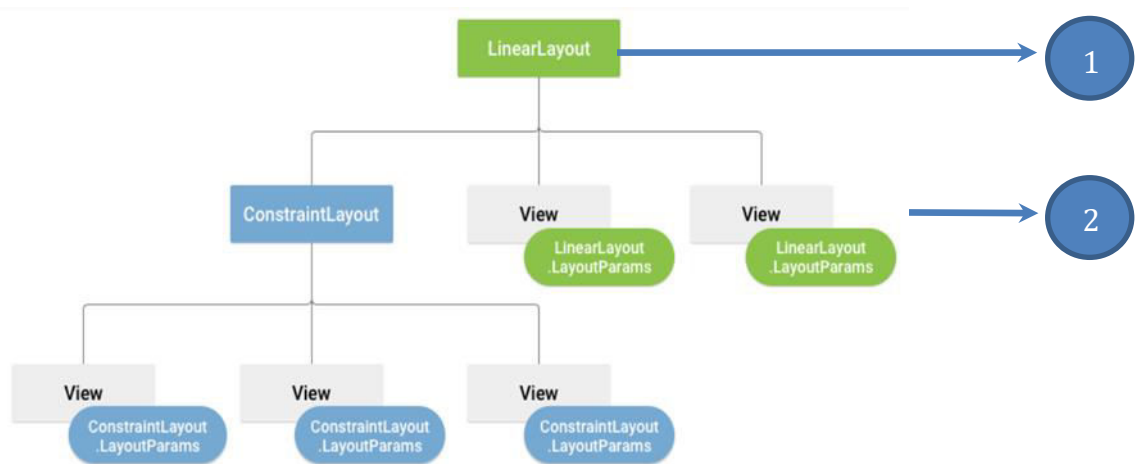
Semua grup tampilan berisi lebar dan tinggi (*layout\_width* dan *layout\_height*), dan masing-masing tampilan harus mendefinisikannya. Banyak *LayoutParams* yang juga menyertakan margin dan border opsional.

Salah satu konstanta ini untuk mengatur lebar atau tinggi:

- ✓ ***wrap\_content*** memberi tahu tampilan agar menyesuaikan sendiri ukurannya dengan dimensi yang dibutuhkan oleh materinya.
- ✓ ***match\_parent*** memberi tahu tampilan agar menjadi sebesar yang akan diperbolehkan oleh kelompok tampilan induknya.

Grup tampilan *root*.

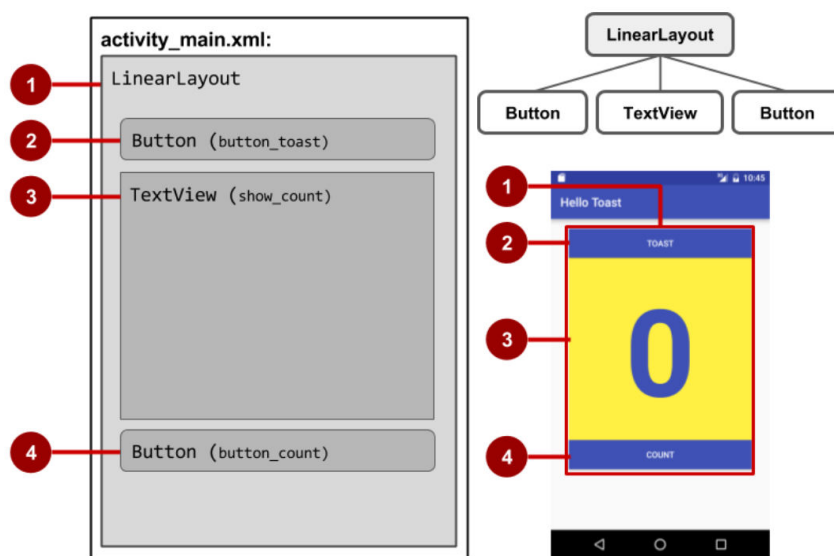
Rangkaian tampilan anak dan grup tampilan pertama yang induknya



Ada sejumlah layout yang disediakan antara lain

### 1) Linear Layout

LinearLayout adalah tampilan grup yang menyelaraskan semua anak dalam satu arah, secara vertikal atau horizontal. Menetapkan arah layout dengan atribut android:orientation.

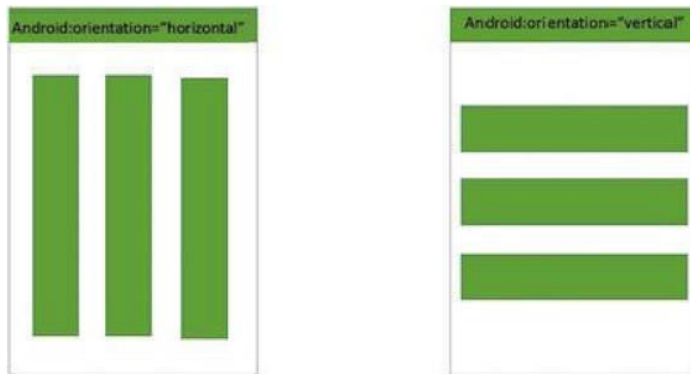


(1) Layout akar LinearLayout, yang berisi semua tampilan anak, disetel ke orientasi vertikal.

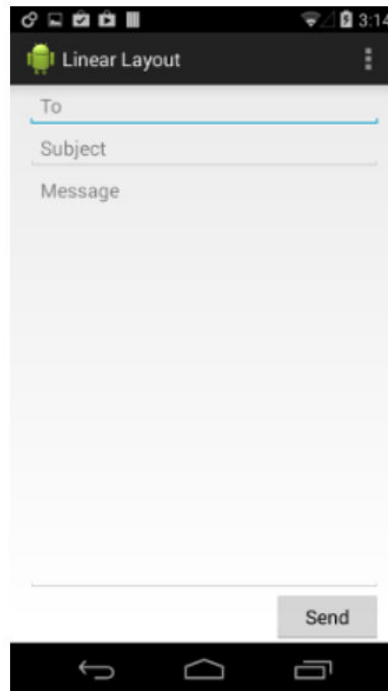
(2) Button (button\_toast) tampilan anak. Sebagai tampilan anak pertama, muncul di bagian atas di layout linear.

(3) TextView (show\_count) tampilan anak. Sebagai tampilan anak kedua, muncul di bawah tampilan anak pertama di layout linear.

(4) Button (button\_count) tampilan anak. Sebagai tampilan anak ketiga, muncul di bawah tampilan anak kedua di layout linear.

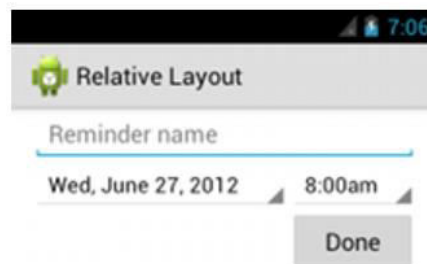


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```



## 2) Relative Layout

RelativeLayout adalah grup tampilan yang menampilkan tampilan anak dalam posisi relatif terhadap tampilan dalam grup tampilan. Dengan kata lain, posisi tampilan anak bisa dijelaskan dalam hubungan satu sama lain atau dengan grup tampilan induk.



```

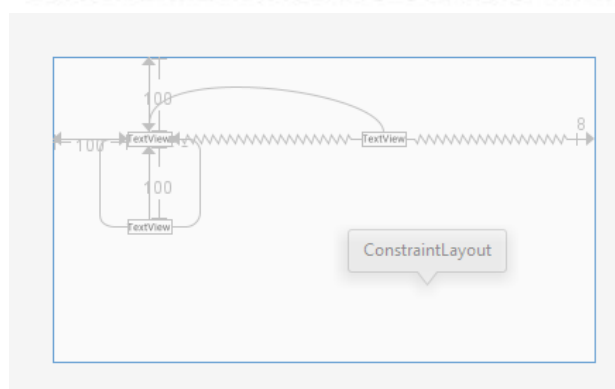
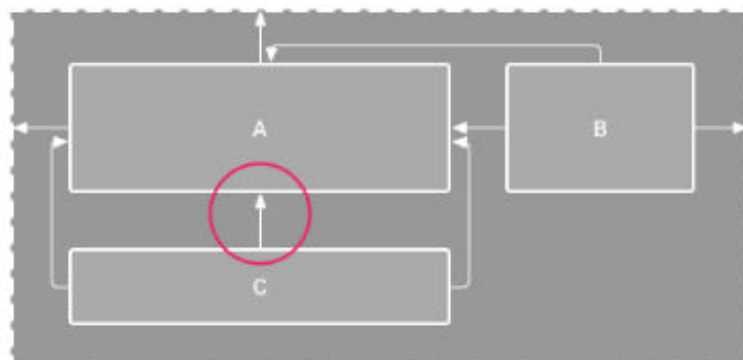
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>

```

### 3) Constraint Layout

Bangun tampilan yang responsive. Mirip seperti layout relatif, dimana tampilan bergantung dengan view lain atau parent layoutnya.

Didukung oleh android studio, gunakan drag and drop daripada mengedit kode xml.



```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/
android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="100dp"
android:layout_marginLeft="100dp"
android:layout_marginTop="100dp"
android:text="TextView"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="100dp"

```

```

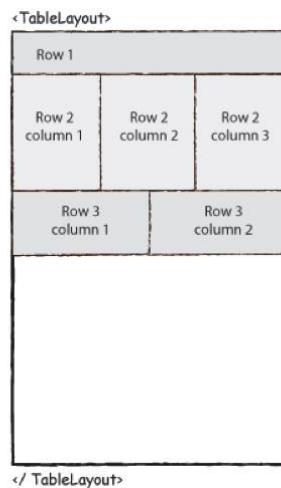
        android:text="TextView"
        app:layout_constraintEnd_toEndOf="@+id/
textView1"    app:layout_constraintStart_toStartOf="@
+id/textView1"
        app:layout_constraintTop_toBottomOf="@+id/textView1" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.503"
    app:layout_constraintStart_toEndOf="@+id/
textView1"
    app:layout_constraintTop_toTopOf="@+id/textView1" />
</android.support.constraint.ConstraintLayout>

```

#### 4) Table Layout

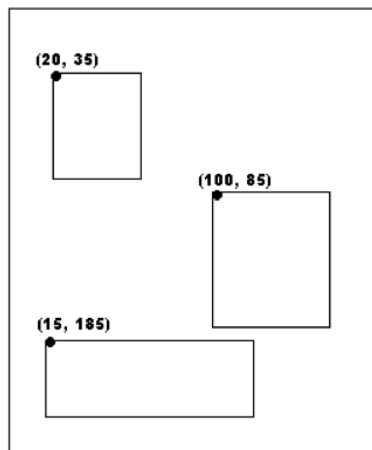
TableLayout adalah tampilan yang mengelompokkan tampilan dalam baris dan kolom.



#### 5) Absolute Layout

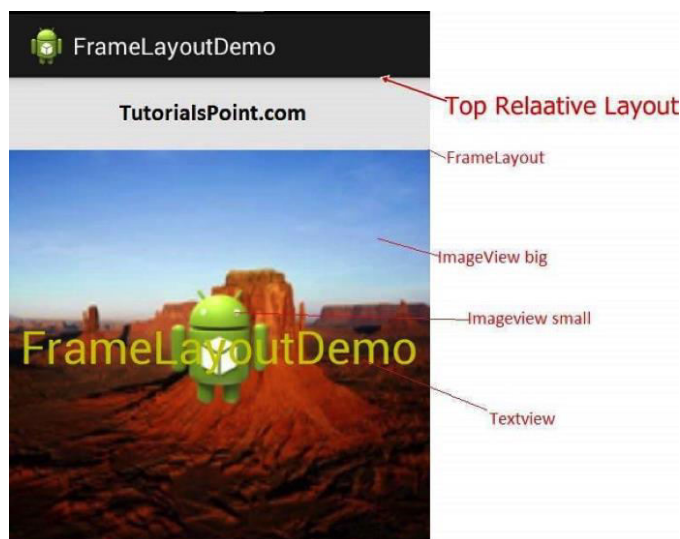
AbsoluteLayout memungkinkan untuk menentukan lokasi pasti anak-anaknya.

### Absolute Layout



### 6) Frame Layout

FrameLayout adalah placeholder di layar yang dapat digunakan untuk menampilkan satu tampilan.



### b) Konsep Object

Pemrograman berorientasi objek (OOP) memperkenalkan konsep objek. Objek OOP adalah batas enkapsulasi yang menyediakan beberapa jenis perilaku publik. Konsep objek tidak terbatas pada bahasa tertentu. Banyak objek yang diwakili dalam bahasa berbasis kelas seperti Java / C # / Ruby / Python, serta bahasa berbasis prototipe seperti JavaScript. Objek-objek ini adalah konstruksi bahasa, bukan konstruk konseptual.

Objek dalam OOP adalah konstruksi yang melakukan sesuatu – ia berperilaku dan bertindak.

Struktur data Orang memiliki atribut nama depan, nama belakang, dan nomor telepon. Objek Seseorang berjalan, berlari, melompat, dan berbicara. Objek Seseorang melakukan banyak hal.

Contoh objek OOP object:

```
public interface PhoneNumberValidator {  
    boolean validate(String phoneNumber);  
}
```

PhoneNumberValidator memvalidasi bahwa String yang diberikan mewakili nomor telepon yang diformat dengan benar. Tidak ada indikasi keadaan internal di dalam validator.

## **5. Cara membangun aplikasi mobile sederhana dengan bahasa pemrograman mobile**

### **B. Ketrampilan yang diperlukan dalam menunjukkan platform operating system dan bahasa pemrograman di dalam perangkat lunak**

1. Membaca, meneliti, menggali dan mengolah data informasi
2. Berkomunikasi dengan para pihak

### **C. Sikap yang diperlukan dalam menunjukkan platform operating system dan bahasa pemrograman di dalam perangkat lunak**

1. Ketelitian
2. Kesabaran
3. Sopan



## 1. Kuis 2 Mobile Programming

### Link Referensi Modul Pertama

1. Video Pembelajaran
2. E-book
3. Link Youtube/Website rujukan

### Link Pertanyaan Modul Pertama

<https://app.sli.do/> (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 2 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/mengenal-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 <sup>TM</sup> 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



DIGITAL  
TALENT  
SCHOLARSHIP



[digitalent.kominfo.go.id](https://digitalent.kominfo.go.id)



DTS\_kominfo



digitalent.kominfo



digitalent.kominfo



Digital Talent Scholarship



KOMINFO

Badan Penelitian dan Pengembangan Sumber Daya Manusia  
Kementerian Komunikasi dan Informatika  
Republik Indonesia