

PEMROGRAMAN ANDROID TEORI & APLIKASI



Yeka Hendriyani, S.Kom., M.Kom
Karmila Suryani, S.Kom., M.Kom

**Sanksi Pelanggaran Pasal 72:
Undang Undang Nomor 19 Tahun 2002**

Tentang Hak Cipta:

- a. Barangsiapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 Ayat (1) atau Pasal 49 Ayat (1) dan Ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp. 1.000.000,00 (Satu juta rupiah), atau pidana penjara paling lama 7 (tujuh tahun dengan atau denda paling banyak Rp. 5.000.000.000,00 (Lima miliar rupiah).
- b. Barangsiapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta atau hak terkait sebagai dimaksud pada Ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp. 500.000.000,00 (Lima ratus juta rupiah).

PEMROGRAMAN ANDROID

Teori dan Aplikasi

Penulis:
Yeka Hendriyani, S.Kom., M.Kom
Karmila Suryani, S.Kom., M.Kom



CV. Penerbit Qiara Media

PEMROGRAMAN ANDROID

Teori dan Aplikasi

CV. PENERBIT QIARA MEDIA

258 hlm: 14,5 x 21 cm

Copyright @2020 Yeka Hendriyani

ISBN: 978-623-7925-03-3

Penerbit IKAPI No. 237/JTI/2019

Penulis:

Yeka Hendriyani, S.Kom., M.Kom

Karmila Suryani, S.Kom., M.Kom

Editor:

Nur Fahmi Hariyanto

Desainer Sampul dan Layout:

Nur Fahmi Hariyanto

Gambar diperoleh dari www.google.com

Cetakan Pertama, 2020

Diterbitkan oleh:

CV. Penerbit Qiara Media - Pasuruan, Jawa Timur

Email: qiaramediapartner@gmail.com

Web: qiaramedia.wordpress.com

Blog: qiaramediapartner.blogspot.com

Instagram: [qiara_media](https://www.instagram.com/qiara_media)

Hak cipta dilindungi oleh undang-undang. Dilarang mengutip dan/atau memperbanyak sebagian atau seluruh isi buku tanpa izin tertulis penerbit.

Dicetak Oleh CV. Penerbit Qiara Media

Isi diluar tanggung Jawab Percetakan

KATA PENGANTAR

Puji dan syukur alhamdulillah kami panjatkan ke hadirat Allah SWT, sholawat dan salam semoga dilimpahkan kepada junjungan kita nabi Muhammad saw, penyusunan buku tentang teori dan aplikasi pemrograman android dapat diselesaikan tepat pada waktunya.

Buku yang berjudul "Pemrograman Android Teori dan Aplikasi", kami selesaikan sesuai dengan rencana, dengan harapan semoga buku ini dapat bermanfaat bagi semua pembaca, terutama bagi para generasi muda siswa, mahasiswa yang sedang mengembangkan potensi keahlian dibidang teknologi informasi.

Penulisan buku ini digunakan sebagai buku pembelajaran bagi masyarakat atau pelajar Indonesia yang sedang mendalami ilmu teknologi informasi. Sehingga akan terjadi kesesuaian antara yang dibelajarkan di sekolah maupun Perguruan Tinggi, Kami menyadari bahwa dalam penulisan buku ini masih jauh dari sempurna, oleh karena itu kami mohon saran dan kritikan demi sempurnanya buku ini yang akan kami masukan dalam terbitan berikutnya.

Apapun adanya, semoga buku ini bermanfaat bagi pembinaan dan pengembangan ilmu teknologi informasi berbasis android, sehingga mampu bersaing di era milenial ini; dan semoga hasil karya ini jadi amal shaleh mendapat balasan yang berlipat ganda dari Allah SWT, hanya kepada Allah lah kami beribadah dan hanya kepada Allah lah kami berserah diri.

Wasalaamu'alaikum wr.wb.

Penulis,

Yeka Hendriyani

DAFTAR ISI

KATA PENGANTAR	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
BAB I PENDAHULUAN	1
A. Pengantar Android	2
B. Fitur Pada Android.....	14
C. Jenis-Jenis Versi Android	15
D. Manfaat Aplikasi Android	30
E. Arsitektur Platform Android	32
F. Tantangan Development Aplikasi Android.....	40
BAB II BAHASA PEMROGRAMAN DAN APLIKASI ANDROID	45
A. Bahasa Pemrograman Android	46
B. Struktur Dasar Bahasa Pemrograman.....	101
C. User Interface dan User Experience Pada Aplikasi Android.....	106
D. Penggunaan Android Studio	117
BAB III DESAIN DAN PERANCANGAN APLIKASI ANDROID	139
A. Perancangan Proses Dalam Pembuatan Aplikasi.....	140
B. Desain Wireframe Aplikasi Android	171
C. Aktivitas dan Maksud pada Android Studio.....	174
BAB IV BANGUN APLIKASI ANDROID	217
A. Membuat Aplikasi Android Pertama	218
B. Perancangan User Interface dan User Experience Pada Aplikasi.....	227
C. Menguji dan Mendebug Aplikasi.....	232

BAB V PENUTUP.....	239
A. Kesimpulan.....	240
B. Kritik dan Saran	241
DAFTAR REFERENSI.....	243
BIODATA PENULIS.....	248

DAFTAR GAMBAR

Gambar 1.1 Tampilan Android.....	11
Gambar 1.2 Arsitektur Android.....	39
Gambar 1.3 Contoh Aplikasi Multilayer.....	40
Gambar 2.1 Ilustrasi Pemrograman Komputer.....	48
Gambar 2.2 Logo Sistem Operasi Linux.....	49
Gambar 2.3 Contoh Pemrograman Aplikasi Menggunakan Bahasa Pemrograman Python.....	50
Gambar 2.4 Contoh Aplikasi Website.....	52
Gambar 2.5 Contoh Pembuatan Game 3D.....	53
Gambar 2.6 Grafik Bahasa Pemrograman Terpopuler.....	60
Gambar 2.7 Proses Pemrograman JSP.....	74
Gambar 2.8 Contoh Pemrograman ASP.....	76
Gambar 2.9 Contoh Pemrograman ColdFusion.....	81
Gambar 2.10 Contoh Pemrograman C.....	83
Gambar 2.11 Logo Python.....	88
Gambar 2.12 Standart Komite Pemrograman C++.....	93
Gambar 2.13 Cara Kerja Java.....	96
Gambar 2.14 Platform Java.....	96
Gambar 2.15 Contoh Struktur Flowchart Pengulangan.....	103
Gambar 2.16 Contoh Struktur Flowchart Percabangan.....	105
Gambar 3.1 Use Case Sistem Aplikasi.....	142
Gambar 3.2 Activity Diagram Pencarian Dokumen Musik.....	146
Gambar 3.3 Sequence Diagram Aplikasi Pencarian Dokumen Musik.....	148
Gambar 3.4 Struktur Menu Infomusik.....	150
Gambar 3.5 Rancangan Tampilan Menu Home.....	151
Gambar 3.6 Rancangan Tampilan Menu Start.....	153
Gambar 3.7 Rancangan Tampilan Menu Hasil.....	154
Gambar 3.8 Wireframe Aplikasi Android.....	173
Gambar 3.9 Contoh Aktivitas Aplikasi Pemesanan.....	176
Gambar 4.1 Tampilan Android Studio saat Membuat Project Baru.....	221
Gambar 4.2 Contoh Perancangan User Interface.....	231
Gambar 4.3 Contoh Perancangan User Experience.....	232

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

A. Pengantar Android

1. Pengertian Aplikasi

Menurut Nazrudin Safaat H (2012 : 9) Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (application suite). Contohnya adalah Microsoft Office dan Open Office.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya.

Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi.

Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

Aplikasi menurut Dhanta dikutip dari Sanjaya (2015) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel. Aplikasi berasal dari kata application yang artinya penerapan lamaran penggunaan. Menurut Jogiyanto dikutip oleh Ramzi (2013) aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan,

dan pekerjaan itu sendiri. Menurut Kadir (2008:3) program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.

Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi software spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

Dari kedua pengertian di atas dapat disimpulkan bahwa aplikasi adalah sekumpulan perintah atau kode yang disusun secara sistematis untuk menjalankan suatu perintah yang diberikan oleh manusia melalui komponen atau hardware komputer yang digunakan oleh manusia dalam

menjalankan program aplikasi, dengan demikian bisa membantu manusia untuk memberikan solusi dari apa yang diinginkan.

2. Pengertian Android

Android adalah sebuah sistem operasi dan platform pemrograman yang dikembangkan oleh Perusahaan Google untuk ponsel pintar dan perangkat seluler lainnya (seperti, tablet). Android bisa berfungsi di berbagai macam perangkat dari banyak produsen yang berbeda. Android telah dilengkapi dengan kit development perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membangun aplikasi bagi pengguna Android. Secara keseluruhan, Android menyatakan ekosistem untuk aplikasi seluler.

Android menurut Nazaruddin (2012: 1) merupakan sistem operasi yang digunakan untuk telepon seluler berbasis Linux. Android juga menyediakan platform terbuka bagi pengembang aplikasi untuk menciptakan dan membangun aplikasi lain untuk bisa digunakan bermacam perangkat bergerak. Android umum digunakan di smartphone dan juga

tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS.

Menurut Hermawan (2011: 1), Android merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, iPhone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

3. Sejarah Perkembangan Android

Android pada mulanya berada di bawah naungan Android, Inc. Sebuah perusahaan yang didirikan pada tahun 2003 silam oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Markas besarnya berlokasi di Palo Alto, California. Lucunya, Android pada awalnya tidak dibuat untuk ponsel, melainkan untuk kamera digital.

Android, Inc. didirikan di Palo Alto, California, pada bulan Oktober 2003 oleh Andy Rubin (pendiri Danger), Rich Miner (pendiri Wildfire Communications, Inc.) (dalam Markoff, John. 2007), Nick Sears (dalam Kirsner, Scott, 2007), (mantan VP T-Mobile) (dalam Vogelstein, Fred, 2011), dan Chris White (kepala desain dan pengembangan antarmuka WebTV) untuk mengembangkan "perangkat seluler pintar yang lebih sadar akan lokasi dan preferensi penggunanya".

Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital (dalam Elgin, Ben, 2005). Tujuan itu bergeser ketika perusahaan menyadari peluang mereka lebih besar untuk berkembang jika menasar perangkat mobile. Kemudian Android diplot untuk menyaingi Symbian dan Windows Mobile yang kala itu menjadi pemain utama. Tapi saat mulai dikerjakan, karena suatu alasan, Android Inc menjalani misi itu secara tersembunyi. Android pada masa itu dialihkan bagi pasar telepon pintar untuk menyaingi Symbian dan Windows Mobile (iPhone Apple belum dirilis pada saat itu). Meskipun para pengembang Android adalah pakar-pakar teknologi yang berpengalaman,

Android Inc. dioperasikan secara diam-diam, hanya diungkapkan bahwa para pengembang sedang menciptakan sebuah perangkat lunak yang diperuntukkan bagi telepon seluler. Masih pada tahun yang sama, Rubin kehabisan uang. Steve Perlman, seorang teman dekat Rubin meminjamkan \$10.000 tunai dan menolak tawaran saham di perusahaan (Vance, Ashlee, 2011).

Google mengakuisisi Android Inc. pada tanggal 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google. Pendiri Android Inc. seperti Rubin, Miner dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google. Setelah itu, tidak banyak yang diketahui tentang perkembangan Android Inc., tetapi banyak anggapan yang menyatakan bahwa Google berencana untuk memasuki pasar telepon seluler dengan tindakannya ini. Di Google, tim yang dipimpin oleh Rubin mulai mengembangkan platform perangkat seluler dengan menggunakan kernel Linux. Google memasarkan platform tersebut kepada produsen perangkat seluler dan operator nirkabel, dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bisa diperbarui. Google telah memilih

beberapa mitra perusahaan perangkat lunak dan perangkat keras, serta mengisyaratkan kepada operator seluler bahwa kerja sama ini terbuka bagi siapapun yang ingin berpartisipasi.

HTC Dream, ponsel Android pertama. Spekulasi tentang niat Google untuk memasuki pasar komunikasi seluler terus berkembang hingga bulan Desember 2006. Menurut McKay, Martha (2006) mengemukakan bahwa BBC dan Wall Street Journal melaporkan bahwa Google sedang bekerja keras untuk menyertakan aplikasi dan mesin pencari di perangkat seluler. Berbagai media cetak dan media daring mengabarkan bahwa Google sedang mengembangkan perangkat seluler dengan merek Google. Beberapa di antaranya berspekulasi bahwa Google telah menentukan spesifikasi teknisnya, termasuk produsen telepon seluler dan operator jaringan. Pada bulan Desember 2007, InformationWeek melaporkan bahwa Google telah mengajukan beberapa aplikasi paten di bidang telepon seluler (dalam Claburn, Thomas, 2007 dan Pearce, James Quintana, 2007).

Pada tanggal 5 November 2007, Open Handset Alliance (OHA) didirikan. OHA adalah konsorsium dari

perusahaan-perusahaan teknologi seperti Google, produsen perangkat seluler seperti HTC, Sony dan Samsung, operator nirkabel seperti Sprint Nextel dan T-Mobile, serta produsen chipset seperti Qualcomm dan Texas Instruments. OHA sendiri bertujuan untuk mengembangkan standar terbuka bagi perangkat seluler. Saat itu, Android diresmikan sebagai produk pertamanya; sebuah platform perangkat seluler yang menggunakan kernel Linux versi 2.6 (dalam Open Handset Alliance, 2007). Telepon seluler komersial pertama yang menggunakan sistem operasi Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008.

Pada tahun 2010, Google merilis seri Nexus; perangkat telepon pintar dan tablet dengan sistem operasi Android yang diproduksi oleh mitra produsen telepon seluler seperti HTC, LG, dan Samsung. HTC bekerja sama dengan Google dalam merilis produk telepon pintar Nexus pertama, yakni Nexus One. Seri ini telah diperbarui dengan perangkat yang lebih baru, misalnya telepon pintar Nexus 4 dan tablet Nexus 10 yang diproduksi oleh LG dan Samsung. Pada 15 Oktober 2014, Google mengumumkan Nexus 6 dan Nexus 9 yang diproduksi oleh Motorola dan HTC. Pada 13 Maret 2013, Larry Page

mengumumkan dalam postingan blognya bahwa Andy Rubin telah pindah dari divisi Android untuk mengerjakan proyek-proyek baru di Google. Ia digantikan oleh Sundar Pichai, yang sebelumnya menjabat sebagai kepala divisi Google Chrome, yang mengembangkan Chrome OS.

Sejak tahun 2008, Android secara bertahap telah melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi, menambahkan fitur baru, dan memperbaiki bug yang terdapat pada versi sebelumnya. Setiap versi utama yang dirilis dinamakan secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau camilan bergula; misalnya, versi 1.5 bernama Cupcake, yang kemudian diikuti oleh versi 1.6 Donut. Versi terbaru adalah 9.0 Pie, yang dirilis pada 7 maret 2018.



Gambar 1.1 Tampilan Android

Aplikasi dikembangkan untuk berbagai alasan yaitu untuk kebutuhan bisnis, membangun layanan baru, membuat bisnis baru, dan menyediakan game serta jenis materi lainnya untuk pengguna. Developer memilih untuk mengembangkan bagi Android agar bisa menjangkau sebagian besar pengguna perangkat seluler.

Android Sebagai platform perangkat seluler paling populer di dunia, Android mendukung ratusan juta perangkat seluler di lebih dari 190 negara di seluruh dunia. Android memiliki basis pemasangan terbesar dari platform seluler apapun dan masih tumbuh dengan cepat. Setiap hari, jutaan pengguna lain menyalakan perangkat Android mereka untuk pertama kalinya dan mulai mencari aplikasi, game, dan materi digital lainnya.

Android menyediakan antarmuka pengguna (UI) layar sentuh untuk berinteraksi dengan aplikasi. Antarmuka pengguna Android sebagian besar berdasarkan pada manipulasi langsung, menggunakan isyarat sentuhan seperti menggesek, mengetuk, dan mencubit untuk memanipulasi objek di layar. Selain keyboard, ada keyboard virtual yang bisa disesuaikan untuk masukan teks. Android juga bisa

mendukung pengontrol game dan keyboard fisik berukuran penuh yang dihubungkan dengan Bluetooth atau USB.

Layar utama Android bisa berisi sejumlah laman ikon aplikasi, yang akan meluncurkan aplikasi terkait, dan widget, dengan menampilkan materi langsung yang diperbarui secara otomatis seperti cuaca, kotak masuk email pengguna, atau ticker berita. Android juga bisa memutar materi multimedia seperti musik, animasi, dan video. Gambar di atas menampilkan ikon aplikasi pada layar utama (kiri), music yang diputar (tengah), dan widget yang ditampilkan (kanan). Sepanjang bagian atas layar terdapat bilah status, yang menampilkan informasi tentang perangkat dan konektivitasnya. Layar utama Android bisa terdiri dari sejumlah laman, yang bisa digesek mundur dan maju oleh pengguna.

Android didesain untuk menyediakan respons cepat terhadap masukan pengguna. Selain antarmuka sentuh yang berubah-ubah, kemampuan getaran perangkat Android bisa menyediakan umpan balik sentuhan. Perangkat keras internal seperti akselerometer, giroskop, dan sensor kedekatan, digunakan oleh banyak aplikasi untuk merespons tindakan

pengguna tambahan. Sensor tersebut bisa mendeteksi rotasi layar dari potret ke lanskap untuk tampilan yang lebih lebar atau sensor bisa memungkinkan pengguna untuk menyetir kendaraan virtual dengan memutar perangkat seolah-olah setir mobil.

Platform Android, berdasarkan pada kernel Linux, terutama didesain untuk perangkat seluler layar sentuh seperti ponsel cerdas dan tablet. Karena perangkat Android biasanya bertenaga baterai, Android didesain untuk mengelola proses guna menjaga konsumsi daya tetap minimum, sehingga menyediakan penggunaan baterai lebih lama.

B. Fitur Pada Android

Android merupakan sistem operasi open source dan bebas untuk di-customize. Oleh karena itu, tidak ada konfigurasi standar untuk software dan hardware pada perangkat berbasis Android, namun, Android mendukung beberapa fitur dasar yaitu: (Lee, 2012, pp. 3)

- a. Storage – menggunakan SQLite, karena SQLite merupakan database yang ringan untuk penyimpanan data.

- b. Connectivity – mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, WiFi, LTE, WiMAX.
- c. Messaging – mendukung SMS dan MMS
- d. Web Browser - berdasarkan pada open source webkit, bersama dengan chrome's v8 Javascript engine
- e. Media Support - mendukung beberapa media seperti AAC, HE-AAC, MPEG-4 SP, AMR, AMR-WB, WAV, MP3, JPG, PNG
- f. Hardware Support - accelerometer sensor, kamera, digital kompas, proximity sensor, GPS
- g. Multi-touch - mendukung multi-touch screens
- h. Multi-tasking – mendukung aplikasi multi-tasking
- i. Flash support – Android 2.3 mendukung flash 10.1
- j. Tethering - mendukung untuk sharing internet.

C. Jenis-Jenis Versi Android

Android adalah salah satu Sistem Operasi (OS) smartphone yang paling populer di dunia. Dan juga telah merebut perhatian banyak produsen ponsel di dunia. Android telah resmi menjadi Sistem Operasi terbaru untuk perangkat mobile modern seperti smartphone dan tablet. Sejak awal dipublikasikan, Android telah mempunyai banyak versi.

Berbagai versi android dengan fitur-fitur keunggulannya masing-masing. Hal ini tentu disamping dengan pesaingnya yang sudah lama eksis, seperti Apple iOS, Windows, Blackberry (BB), Symbian dan sebagainya.

Uniknya, setiap versi Android yang baru diluncurkan pasti selalu disertai dengan nama lucu yang sama sekali tidak ada hubungannya dengan karakter robot hijau (Android) tersebut. Misalkan seperti, Cupcake (kue cangkir), Gingerbread (roti jahe), Donat, Honeycomb (sarang madu), Ice Cream Sandwich (es krim isi atau lapis), Jelly Bean (Kacang Jeli), Kitkat (coklat keping atau potong), Lollipop (permen lolipop), Marshmallow, Naugat, dan yang terbaru yakni Oreo. Bahkan, yang lebih uniknya, nama-nama versi android pasti berurutan menurut abjad dari A (Apple Pie), hingga sampai sekarang ini O (Oreo). Oleh sebab itu, mari kita mengenal lebih lanjut dari macam-macam versi Android di bawah ini.

1. Android 1.0 (Apple Pie)



Android versi pertama yaitu Apple Pie, yang dirilis pada 23 September 2008 dan hanya dilengkapi berbagai fitur seperti Play Store, kamera, Web Browser, Sinkronisasi antara G-mail, Contacts dan Google Agenda. Selain itu, diawal peluncurannya, Android juga sudah dilengkapi aplikasi Google Maps dan dukungan streaming Youtube.

2. Android 1.1 (Banana Bread)



Sistem Operasi android yang rilis selanjutnya yaitu Banana Bread, rilis pada bulan Februari 2009. Dan fiturnya yaitu tidak jauh berbeda dengan versi sebelumnya. HTC

merupakan salah satu smartphone Android pertama yang menggunakan versi ini.

3. Android 1.5 (Cupcake)



Dirilis pada awal bulan April 2009 dan juga tidak berbeda dengan versi Android sebelumnya. Hanya saja terdapat fitur tambahan seperti sudah Support Bluetooth A2DP, AVRCP, Soft-keyboard dengan prediksi text dan record atau watch videos.

4. Android 1.6 (Donut)



Android Donut dirilis pada 15 September 2009, dan terdapat fitur tambahan seperti Gesture Framework hingga

Turn-by-turn navigation. Kemudian, Android ini juga terlihat lebih sempurna pada saat itu. Dengan minimnya bug, ditambah lebih lengkapnya berbagai fitur yang disediakan oleh Google.

5. Android 2.0 (Eclair)



Android versi 2.0 ini bernama Eclair dan dirilis pada 26 Oktober 2009 silam. Selain terdapat bluetooth, versi ini juga mendapatkan fitur tambahan seperti multi-touch, Live Wallpaper dan juga flash kamera. Kemudian, beberapa fitur yang dapat anda nikmati dalam versi eclair adalah HTML, Digital zoom, Support Microsoft Exchange, dan Updated UI. Versi android tercepat IDwebhost.

6. Android 2.2.9 (Froyo)



Pada bulan Mei 2010 lalu, Perusahaan raksasa Google telah merilis Android versi terbaru Yakni adalah Android 2.2.9 (Froyo). Versi ini adalah salah satu sistem operasi Android yang juga telah disempurnakan, tujuannya tentu untuk meningkatkan kecepatan kinerja suatu sistem Android. Dan berikut ini adalah beberapa fitur dan perbaikan yang disediakan oleh Android Froyo:

- Peningkatan Speed
- Implementasi JIT
- USB Tethering
- Aplikasi instalasi untuk perluasan memori atau storage
- Support file upload pada aplikasi browser
- Animated GIFs

7. Android 2.3 (Gingerbread)



Pada bulan Desember 2010 lalu, Google merilis kembali Android versi terbarunya yaitu Gingerbread. Yang secara fitur sudah jelas sangat sempurna. Ditambah lagi, Android 2.3 ini juga telah diadopsi oleh salah satu perusahaan Smartphone paling populer, yaitu Samsung dengan menanamkan sistem operasi ini dalam smartphone seri Nexus-nya.

8. Android 3.0 – 3.2 6 (Honeycomb)



H, untuk Honeycomb adalah salah satu sistem operasi Android versi terbaru yang dirilis pada bulan Februari 2011 silam. Namun, versi ini lebih ditujukan untuk perangkat Tablet yang mana pada tahun itu sangat laris atau laku dipasaran. Beberapa fitur dan perbaikan pada Android Honeycomb, yaitu:

- Support Multi core
- Support Tablet lebih baik
- Updated 3D UI
- Layar Utama (homescreens) yang dapat diatur
- Melihat aplikasi yang barusan dibuka
- Menyempurnakan layout keyboard
- Transport protocol untuk Media atau Picture
- video chat Google Talk

- Google eBooks
- "Private browsing"
- System-wide Clipboard
- HTTP Live streaming
- Update 3.1
- Peningkatan UI
- Open Accessory API
- USB host API
- Support mouse, joysticks dan gamepad
- Widget Home screen yang bisa di atur size atau ukurannya
- Notifikasi MTP
- RTP API untuk audio
- Update 3.2
- Optimise pada berbagai tablets
- Mode kompatibilitas display (zoom for fixed sized apps)
- Sinkronisasi Media dari SD card
- versi android terbaik
- IDwebhost

- Update 3.2.1
- Update Android Market merupakan automatic updates yang lebih mudah
- Update Google Books
- Peningkatan kinerja Wi-Fi
- Perbaikan prediksi tulisan tangan dengan huruf Chinese
- Update 3.2.2
- Perbaikan kecil
- Update 3.2.4
- Update tambahan 'Pay as you go' bagi tablet
- Update 3.2.6
- Perbaikan kecil

9. Android 4.0 (Ice Cream Sandwich)



Puncak kesempurnaan Android yakni ketika pada versi ini, dimana Ice Cream Sandwich dirilis pada bulan Oktober 2011 silam. Dan operasi sistem ini mulai bekerja dengan baik di semua jenis smartphome apapun. Selain bertambahnya berbagai fitur yang menarik, Ice Cream Sandwich juga merupakan versi yang paling banyak disukai pada saat itu. Bahkan, Android Ice Cream Sandwich juga sudah dilengkapi dengan fitur ekstra multitasking serta notifikasi yang lebih banyak.

10. Android 4.1.2 (Jelly Bean)



Jelly Bean dirilis pada 9 Juli 2012 lewat konferensi I/O Google. Versi ini adalah salah satu versi Android yang kerap mendapatkan update fitur-fitur yang bermanfaat dan menarik, beberapa contohnya semacam memperbaiki rotasi

layar, seperti Support resolusi video 4K, Support penulisan huruf Hebrew dan Arabic dari kanan ke kiri, peningkatan kinerja, dan sistem keamanan serta masih banyak lainnya.

11. Android 4.4 (Kitkat)



Android versi inilah yang saat ini banyak dipakai oleh mayoritas masyarakat Indonesia. Kitkat dirilis pada tahun 2013 lalu. pada versi ini, Android banyak mendapatkan pembaharuan/update fitur. Seperti, terdapatnya fitur Screen recording, untuk merekam kegiatan yang terjadi pada layar smartphone, Peningkatan akses notifikasi, New Translucent system UI, System wide settings untuk closed captioning, dan Peningkatan kinerja serta lain sebagainya.

12. Android 5.0 (Lollipop)



Dirilis pada tahun 2014, Android Lollipop lebih banyak menawarkan fitur tambahan untuk menyempurnakan berbagai fitur yang sudah ada. Dan Nexus 6 merupakan salah satu ponsel yang pertama mencicipi Android Lollipop ini. Selain itu, Google juga lebih menyempurnakan pada kinerja dari Android Lollipop sendiri.

13. Android 6.0 (Marshmallow)



Android versi 6.0 dirilis pada tahun 2015 silam, yang banyak membawa pembaharuan. Salah satunya yaitu sudah support USB Type-C. Selain itu, Android Marshmallow ini juga terdapat fasilitas autentikasi sidik jari dan daya baterai yang lebih baik.

14. Android 7.0 (Nougat)



Android Nougat versi 7.0 dirilis pada bulan Agustus 2016 yang lebih meningkatkan pada kinerja versi sebelumnya. Selain itu, Android Nougat juga menambah banyak fitur-fitur baru yang diantaranya seperti sudah dapat multitasking, meningkatkan fitur Doze yang dahulu telah dirilis di versi sebelumnya. Inilah beberapa fitur terbaru yang terdapat pada versi Nougat:

- Support Multi window

- Dapat langsung membalas pesan dari menu notifikasi atau jendela.
- Tampilan panel notifikasi serta quick settings yang baru.
- Mode Doze yang lebih baik, (Doze Mode 2.0)
- Menu di antara system settings.

15. Android 8.0 (Oreo)



Android versi Oreo dirilis pada bulan Agustus 2017 lalu. Tentu saja Android Oreo merupakan versi final untuk sekarang ini. Beberapa fiturnya juga turut diluncurkan Google selaku pihak pengelola. Adapun fitur-fiturnya tersebut antara lain yaitu:

- Android O lebih berfokus pada kecepatan dan efisiensi
- Kecepatan Boot up 2X lebih cepat
- Mode Picture in picture lebih flexibel

- Aplikasi yang berjalan di latarbelakang atau background lebih diperketat untuk lebih menghemat battery
- Battery lebih tahan lama
- Emoji yang diperbaharui dan diperbanyak

D. Manfaat Aplikasi Android

Android memiliki beberapa manfaat dan fitur utama, diantaranya adalah sebagai berikut:

a. Dapat Berjalan dalam Banyak Perangkat

Android berjalan di banyak smartphone berbeda. Beberapa model terbaru yang lebih populer adalah HTC One, HTC Evo 3D, Motorola Photon 4G, LG Optimus, Samsung Galaxy S 4, Nexus 4 dari Google sendiri.

b. Dapat Menggunakan Fitur iPhone

Android mampu melakukan sebagian besar inovasi yang dibawa oleh iPhone, seperti penelusuran Web dan Email yang mudah dan menu layar sentuh. Namun, Android juga memiliki inovasi sendiri yang kini telah diadopsi oleh iPhone, seperti Multitasking.

c. Memiliki Fitur yang Canggih

Ketersediaan fitur yang lebih canggih tergantung pada model ponsel yang digunakan. Namun, hampir semua model akan memiliki kamera untuk mengambil gambar dan Video dan GPS untuk navigasi satelit.

d. Mempunyai App Store

Seperti iPhone, ponsel Android memiliki “App Store” mereka sendiri dalam bentuk Google Play (sebelumnya Android Market). Pada Juli 2013 lebih dari satu juta aplikasi tersedia, mulai dari pengolah kata, Game, Fotografi dan Video hingga Aplikasi Travelling, Berita dan Seni Lukisan, dan lain sebagainya (daftaranya sangat besar).

e. Android Adalah Open-Source

Android adalah sistem “Open Source” dengan demikian, jauh lebih mudah dimodifikasi dan dipersonalisasi daripada iOS iPhone. App Store-nya juga tidak terlalu membatasi. Namun, pendekatan bentuk bebas ini lebih berarti bahwa menggunakan ponsel Android adalah pengalaman yang sedikit lebih cepat dan ramah pengguna daripada menggunakan iPhone.

E. Arsitektur Platform Android

Android adalah tumpukan perangkat lunak berbasis Linux sumber terbuka yang dibuat untuk berbagai perangkat dan faktor bentuk. Menurut Lee (2012, pp. 4), Android terbagi menjadi lima bagian dari empat lapisan utama yaitu:

a. Linux Kernel

Linux kernel merupakan dasar dari Android. Lapisan ini semua low level device driver untuk berbagai komponen perangkat keras dari perangkat Android. Sebagai contoh, Android Runtime (ART) bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan pengelolaan memori tingkat rendah. Menggunakan kernel Linux memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan driver perangkat keras untuk kernel yang cukup dikenal.

b. Hardware Abstraction Layer (HAL)

Hardware Abstraction Layer (HAL) memberikan antarmuka standar yang mengungkap kemampuan perangkat keras perangkat ke kerangka kerja API Java yang lebih tinggi. HAL terdiri atas beberapa modul pustaka,

masing-masing menerapkan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Ketika API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul pustaka untuk komponen perangkat keras tersebut.

c. Libraries

Lapisan ini mengandung seluruh kode yang menyediakan fitur utama dari sistem operasi Android.

Sebagai contoh, SQLite library menyediakan dukungan database sehingga aplikasi dapat menggunakannya sebagai tempat penyimpanan.

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode bawaan yang memerlukan pustaka bawaan yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja API Java meningkatkan fungsi beberapa pustaka bawaan pada aplikasi. Misalnya, Anda dapat mengakses OpenGL ES melalui kerangka kerja API OpenGL Java Android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D pada aplikasi Anda.

Jika mengembangkan aplikasi yang memerlukan kode C atau C++, Anda dapat menggunakan Android NDK untuk mengakses beberapa pustaka platform bawaan langsung dari kode asal.

d. Android Runtime

Android Runtime terdapat pada lapisan yang sama dengan libraries, yang menyediakan kumpulan libraries utama yang dapat digunakan pengembang untuk membuat aplikasi pada Android menggunakan bahasa pemrograman Java. Android Runtime juga termasuk Dalvik virtual machine, yang memungkinkan aplikasi Android berjalan dengan proses tersendiri.

Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap Android Runtime (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX, format bytecode yang dirancang khusus untuk Android yang dioptimalkan untuk footprint memori minimal. Buat rantai aplikasi, misalnya Jack, mengumpulkan sumber Java ke bytecode DEX, yang dapat berjalan pada platform Android.

Beberapa fitur utama ART mencakup:

- Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT).
- Pengumpulan sampah (GC) yang dioptimalkan.
- Di Android 9 (API level 28) dan yang lebih tinggi, konversi file format Dalvik Executable (DEX) paket aplikasi menjadi kode mesin yang lebih ringkas.
- Dukungan debugging yang lebih baik, mencakup profiler penyampelan terpisah, pengecualian diagnostik mendetail dan laporan kerusakan dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

Sebelum ke Android versi 5.0 (API level 21), Dalvik adalah waktu proses Android. Jika aplikasi Anda berjalan baik pada ART, semestinya berfungsi baik juga pada Dalvik, tetapi mungkin tidak sebaliknya.

Android juga menyertakan serangkaian pustaka waktu proses inti yang menyediakan sebagian besar fungsi bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8, yang digunakan kerangka kerja API Java.

e. Application Framework

Memperlihatkan berbagai kemampuan dari sistem operasi Android untuk pengembang aplikasi Android sehingga mereka dapat menggunakannya dalam membuat aplikasi.

Keseluruhan rangkaian fitur pada Android OS tersedia untuk Anda melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang harus Anda buat aplikasi Android dengan menyederhanakan penggunaan ulang inti, komponen dan layanan sistem modular, yang mencakup berikut ini:

- Tampilan Sistem yang kaya dan luas dapat Anda gunakan untuk membuat UI aplikasi, termasuk daftar, kisi, kotak teks, tombol, dan bahkan browser web yang dapat disematkan.
- Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file layout.
- Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status

- Pengelola Aktivitas yang mengelola siklus hidup aplikasi dan memberikan back-stack navigasi yang umum.
- Penyedia Materi yang memungkinkan aplikasi mengakses data dari aplikasi lainnya, seperti aplikasi Kontak, atau untuk berbagi data milik sendiri.

Developer memiliki akses penuh ke API kerangka kerja yang digunakan oleh aplikasi sistem Android.

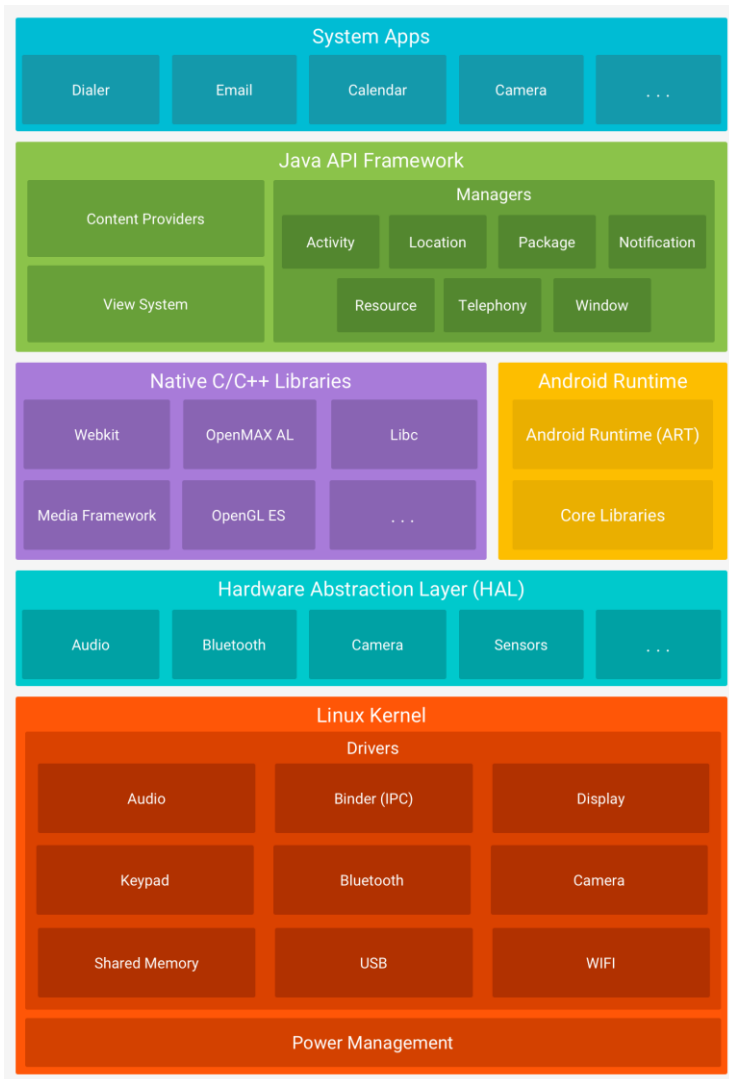
f. Applications

Applications merupakan lapisan paling atas, yang memungkinkan para pengembang untuk menemukan aplikasi bawaan yang terdapat pada perangkat Android. Contohnya saja adalah Phone, Contacts, Browser, dan lainnya. Selain itu juga aplikasi yang dapat di unduh dan diinstall dari Android Market. Setiap aplikasi yang dibuat terdapat pada lapisan ini.

Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi yang disertai dengan platform tidak memiliki status khusus pada aplikasi yang pengguna ingin

instal. Jadi, aplikasi pihak ketiga dapat menjadi browser web utama, pengolah pesan SMS atau bahkan keyboard utama (beberapa pengecualian berlaku, seperti aplikasi Settings sistem).

Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh developer dari aplikasi mereka sendiri. Misalnya, jika aplikasi Anda ingin mengirimkan pesan SMS, Anda tidak perlu membangun fungsi tersebut sendiri—Anda bisa juga menjalankan aplikasi SMS mana saja yang telah diinstal guna mengirimkan pesan kepada penerima yang Anda cantumkan.



Gambar 1.2. Arsitektur Android

F. Tantangan Development Aplikasi Android

Sewaktu platform Android menyediakan fungsionalitas kaya untuk development aplikasi, masih ada sejumlah tantangan yang perlu ditangani, seperti:

a. Membangun untuk dunia multilayer

Android berjalan pada miliaran perangkat genggam di seluruh dunia, dan mendukung beragam factor bentuk termasuk perangkat yang dapat dikenakan dan televisi. Perangkat bisa tersedia dalam ukuran dan bentuk berbeda yang memengaruhi desain layar untuk elemen UI di aplikasi.



Gambar 1.3. Contoh Aplikasi Multilayer

Dalam membangun untuk dunia multilayer, produsen perangkat dapat menambahkan elemen UI, gaya, dan warna khusus untuk membedakan produknya. Setiap produsen dapat menawarkan fitur yang berbeda dalam hal bentuk keyboard, ukuran layar, atau tombol kamera.

Aplikasi yang berjalan pada satu perangkat mungkin terlihat sedikit berbeda di perangkat lain. Tantangan bagi sebagian besar developer adalah untuk mendesain elemen UI yang bisa bekerja di semua perangkat. Selain itu, developer juga bertanggung jawab untuk menyediakan sumber daya aplikasi seperti ikon, logo, grafik lain, dan gaya teks untuk mempertahankan keseragaman penampilan diseluruh perangkat yang berbeda.

b. Memaksimalkan kinerja aplikasi yang tepat

Kinerja aplikasi, seberapa cepat aplikasi berjalan, seberapa mudah aplikasi menghubungkan ke jaringan, dan seberapa baik aplikasi mengelola baterai dan penggunaan memori, dipengaruhi oleh beberapa factor seperti daya tahan baterai, materi multimedia, dan akses Internet. Anda harus memperhatikan batasan tersebut dan menulis kode

sedemikian rupa sehingga penggunaan sumber daya diseimbangkan dan didistribusikan secara optimal. Misalnya, Anda harus menyeimbangkan layanan latar belakang dengan mengaktifkannya hanya jika perlu; hal ini akan menghemat daya tahan baterai perangkat pengguna.

c. Membuat kode dan pengguna tetap aman

Dalam pengembangan aplikasi perlu melakukan tindakan pencegahan untuk mengamankan kode dan pengalaman pengguna saat menggunakan aplikasi. Gunakan alat seperti ProGuard (disediakan di Android Studio), yang mendeteksi dan membuang kelas, bidang, metode, dan atribut yang tidak digunakan serta mengenkripsi semua kode dan sumber daya aplikasi sewaktu memaketkan aplikasi.

Untuk melindungi informasi penting milik pengguna seperti proses masuk dan sandi, maka harus mengamankan saluran komunikasi untuk melindungi data yang bergerak (di Internet) serta data yang tidak bergerak (di perangkat).

d. Tetap kompatibel dengan versi platform yang lebih lama

Pertimbangkan cara menambahkan fitur versi platform Android baru ke aplikasi, sambil memastikan bahwa aplikasi masih bisa berjalan pada perangkat dengan versi platform yang lebih lama. Hal ini tidak akan bermanfaat untuk hanya focus pada versi Android terbaru, karena tidak semua pengguna sudah meningkatkan versi atau dapat meningkatkan versi perangkatnya.

e. Memahami pasar dan pengguna.

Dalam membangun aplikasi android, target pengguna aplikasi yang dibangun harus sesuai dengan permintaan pengguna atau kebutuhan pasar agar aplikasi memiliki banyak pengguna dan bisa digunakan oleh banyak dari kalangan pengguna.

BAB II

BAHASA PEMROGRAMAN

DAN APLIKASI ANDROID

BAB II

BAHASA PEMROGRAMAN DAN APLIKASI ANDROID

A. Bahasa Pemrograman Android

1. Definisi Pemrograman

Program komputer atau sering kali disingkat sebagai program adalah serangkaian instruksi yang ditulis untuk melakukan suatu fungsi spesifik pada komputer. Komputer pada dasarnya membutuhkan keberadaan program agar bisa menjalankan fungsinya sebagai komputer, biasanya hal ini dilakukan dengan cara mengeksekusi serangkaian instruksi program tersebut pada prosesor.

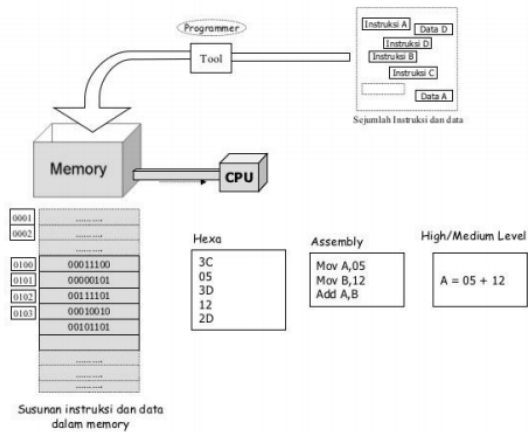
Sebuah program biasanya memiliki suatu bentuk model pengeksekusian tertentu agar dapat secara langsung dieksekusi oleh komputer. Program yang sama dalam format kode yang dapat dibaca oleh manusia disebut sebagai kode sumber, bentuk program yang

memungkinkan programmer menganalisis serta melakukan penelaahan algoritma yang digunakan pada program tersebut.

Kode sumber tersebut pada akhirnya dikompilasi oleh utilitas bahasa pemrograman tertentu sehingga membentuk sebuah program. Bentuk alternatif lain model pengekseskuan sebuah program adalah dengan menggunakan bantuan interpreter, kode sumber tersebut langsung dijalankan oleh utilitas interpreter suatu bahasa pemrograman yang digunakan.

Beberapa program komputer dapat dijalankan pada sebuah komputer pada saat bersamaan, kemampuan komputer untuk menjalankan beberapa program pada saat bersamaan disebut sebagai multitasking. Program komputer dapat dikategorikan menurut fungsinya; perangkat lunak sistem atau perangkat lunak aplikasi.

Ilustrasi Pemrograman Komputer



Gambar 2.1. Ilustrasi Pemrograman Komputer

Pemrograman komputer merupakan suatu proses iteratif penulisan dan penyuntingan kode sumber sehingga membentuk sebuah program. Penyuntingan kode sumber meliputi proses pengetesan, analisis, pembetulan kesalahan, pengoptimasian algoritma, normalisasi kode, dan kadang-kadang pengkoordinasian antara satu programmer dengan programmer lainnya jika sebuah program dikerjakan oleh beberapa orang dalam sebuah tim. Seorang praktisi yang memiliki keahlian untuk melakukan penulisan

kode dalam bahasa pemrograman disebut sebagai programmer komputer atau programmer, pengembang perangkat lunak, atau koder. Istilah rekayasa perangkat lunak (bahasa Inggris: Software engineering) seringkali digunakan karena proses penulisan program tersebut dipandang sebagai suatu disiplin ilmu perkerayaan.

2. Jenis-Jenis Pemrograman

a. Pemrograman Sistem Operasi

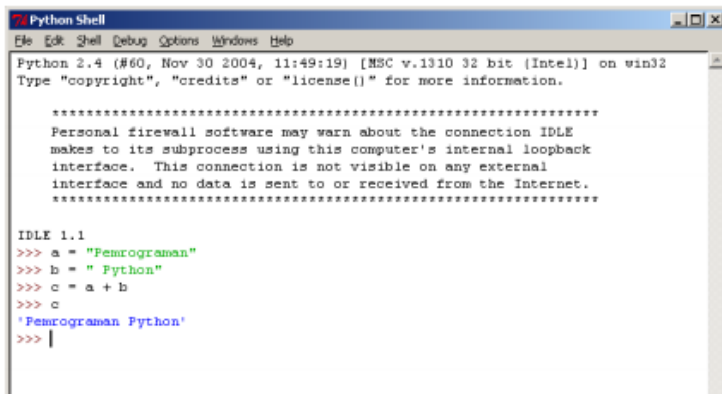
Salah satu programmer yang terkenal berhasil membangun sebuah system operasi adalah Linus Benedict Torvalds dikenal dengan system operasi linux.



Gambar 2.2. Logo Sistem Operasi Linux

Linux dibangun menggunakan bahasa pemrograman C. hal tersebut bisa diketahui karena linux bersifat open source (source code nya bisa dibaca siapapun). System operasi lain seperti halnya Windows tidak diketahui dengan pasti dibangun dengan bahasa apa meski beberapa sumber mengatakan windows dibangun dengan bahasa assembler.

b. Pemrograman Aplikasi



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.4 (#60, Nov 30 2004, 11:49:19) [MSC v.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.1
>>> a = "Pemrograman"
>>> b = " Python"
>>> c = a + b
>>> c
'Pemrograman Python'
>>> |
```

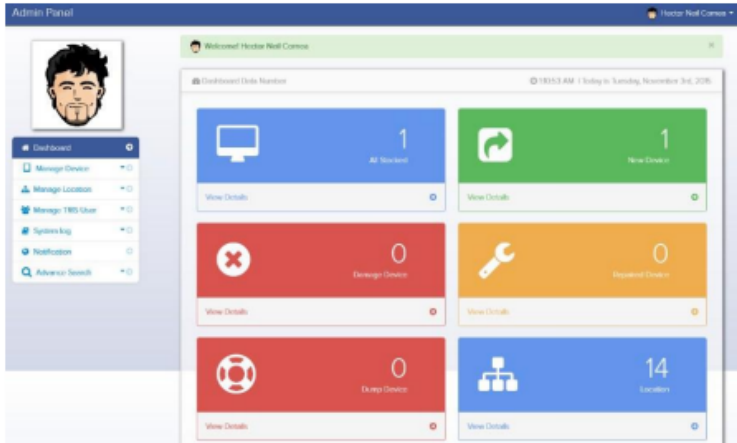
Gambar 2.3. Contoh Pemrograman Aplikasi Menggunakan Bahasa Pemrograman Python

Bahasa pemrograman yang dapat digunakan untuk membangun aplikasi misalnya java, visual basic,

Delphi, C/++/C#, VISUAL FOXPRO, dan Python. Aplikasi yang dihasilkanpun bermacam-macam mulai dari editor teks, image viewer, pemutar VCD, sampai aplikasi-aplikasi perkantoran pengolah gambar, pengolah foto, pengolah data dan lain-lain.

c. Pemrograman Web

Pemrograman web pada dasarnya digunakan untuk mendesain halaman situs web yang dinamis dan interaktif. Suatu halaman web dibangun dengan menggunakan bahasa HTML dan perlu di digaris bawahi bahwa HTML bukan bahasa pemrograman, HTML hanyalah melakukan markup (penandaan) pada suatu teks sehingga akan menghasilkan format tertentu apabila dibaca oleh browser. Itu sebabnya HTML hanya bersifat statis.



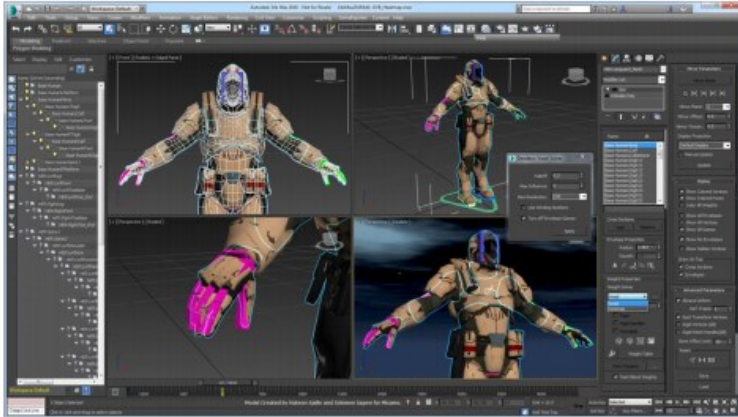
Gambar 2.4. Contoh Aplikasi Website

Pemrograman web didesain untuk dapat mengubah-ubah output HTML tersebut sehingga tidak lagi statis melainkan dinamis. Bahasa pemrograman yang digunakan untuk pemrograman web adalah ASP, PHP, JSP, VBScript, dan Java Script.

d. Pemrograman Game

Pemrograman game adalah pemrograman yang paling rumit. Sebelum dikembangkan, sebuah game harus memiliki konsep cerita yang jelas dan menarik. Kemudian pemrograman game harus menggabungkan

seluruh pustaka dan API (Application Programming Interface) yang ada.



Gambar 2.5. Contoh Pembuatan Game 3D

Karena alasan diatas bahasa pemrograman yang paling populer untuk pemrograman game adalah C++, java dan C. sifat orientasi objek dan compiler dari bahasa tersebut mendukung untuk pemrograman game. Namun bukan berarti bahasa pemrograman yang lain tidak mendapat tempat, jika game yang dibuat adalah game sederhana maka bahasa pemrograman seperti visual basic pun dapat digunakan untuk menciptakan game.

3. Bahasa Pemrograman

Bahasa pemrograman adalah notasi yang digunakan untuk menulis program (komputer). Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk pemerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Bahasa Pemrograman (programming language) adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat

menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

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

- 1) Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110
- 2) Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah.Ingggris Assembly), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode mnemonic), contohnya kode_mesin | MOV, SUB, CMP, JMP, JGE, JL, LOOP, dsb.
- 3) Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&, ||, dsb.

4) Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia, contohnya begin, end, if, for, while, and, or, dsb. Komputer dapat mengerti bahasa manusia itu diperlukan program compiler atau interpreter.

Sebagian besar bahasa pemrograman digolongkan sebagai Bahasa Tingkat Tinggi, hanya bahasa C yang digolongkan sebagai Bahasa Tingkat Menengah dan Assembly yang merupakan Bahasa Tingkat Rendah.

4. Fungsi Bahasa Pemrograman

Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program/aplikasi. Contohnya adalah program yang digunakan oleh kasir di mal-mal atau swalayan, penggunaan lampu lalu lintas di jalan raya, dll.

Bahasa Pemrograman yang kita kenal ada banyak sekali di belahan dunia, tentang ilmu komputer dan

teknologi dewasa ini. Perkembangannya mengikuti tingginya inovasi yang dilakukan dalam dunia teknologi. Contoh bahasa pemrograman yang kita kenal antara lain adalah untuk membuat aplikasi game, antivirus, web, dan teknologi lainnya.

Bahasa pemrograman komputer yang kita kenal antara lain adalah Java, Visual Basic, C++, C, Cobol, PHP, .Net, dan ratusan bahasa lainnya. Namun tentu saja kebutuhan bahasa ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya.

Secara umum bahasa pemrograman terbagi menjadi 4 kelompok, yaitu:

- Object Oriented Language (Visual dBase, Visual FoxPro, Delphi, Visual C)
- High Level Language (seperti Pascal dan Basic)
- Middle Level Language (seperti bahasa C), dan
- Low Level Language (seperti bahasa Assembly)

5. Tingkatan Bahasa Pemrograman

Menurut tingkat kesulitan dalam mempelajari dan implementasinya, bahasa pemrograman dibedakan menjadi beberapa jenis:

a. Bahasa Tingkat Tinggi

Bahasa pemrograman masuk tingkat ini karena bahasa tersebut mendekati bahasa manusia. Contohnya bahasa Basic, Visual Basic, Pascal, Java, PHP.

b. Bahasa Tingkat Menengah

Disebut tingkat menengah karena bisa masuk ke dalam bahasa tingkat tinggi maupun rendah. Contohnya bahasa C.

c. Bahasa Tingkat Rendah

Bahasa pemrograman masuk tingkat ini karena bahasanya masih jauh dari bahasa manusia. Contohnya bahasa Assembly.

6. Jenis-Jenis Bahasa Pemrograman

Bahasa pemrograman terdiri dari berbagai jenis seperti pemrograman untuk membuat aplikasi website,

desktop, game, ataupun sistem operasi. Berikut adalah jenis jenis bahasa pemrograman yang sering dipakai:

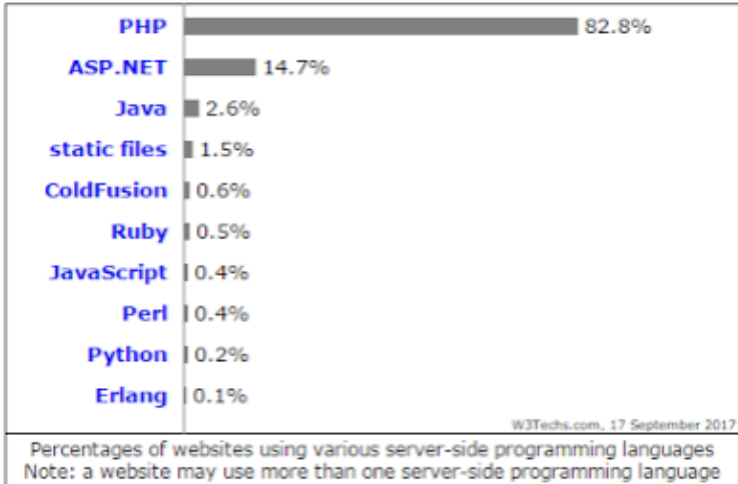
a. Bahasa Pemrograman PHP

Menurut Wikipedia.org, PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>.

PHP merupakan bahasa pemrograman server-side karena di proses di komputer server.

Awalnya, PHP adalah singkatan dari Personal Home Page. PHP digunakan untuk Membuat halaman web pribadi. Dalam beberapa tahun perkembangannya, PHP berubah menjadi bahasa pemrograman yang sangat luar biasa dan banyak digunakan oleh website-website besar seperti wikipedia, facebook, joomla, wordpress, dan lainlain. Saat ini PHP merupakan singkatan dari PHP: Hypertext Preprocessor. PHP

berbasis freeware dan open source, artinya kamu bisa mengubahnya sesuai kemauan kamu.



Gambar 2.6. Grafik Bahasa Pemrograman Terpopuler

Bahasa pemrograman PHP sangat sering dipakai orang karena kemudahan dalam mempelajari dan menerapkannya. Script PHP secara umum menggunakan bahasa Inggris, jadi programmer awam dapat mudah memahami bahasa tersebut. Dilansir dari W3Tech, PHP merupakan bahasa pemrograman server-side yang paling banyak dipakai.

Tabel 2.1 Versi-versi Pemrograman PHP

Versi	Tanggal rilis	Didukung sampai	Catatan
1.0	8 Juni 1995		Secara resmi disebut "Personal Home Page Tools (Alat PHP)". Ini adalah penggunaan pertama dari nama "PHP"
2.0	1 November 1997		Secara resmi disebut "PHP / FI 2.0". Ini adalah rilis pertama yang sebenarnya bisa dicirikan sebagai PHP, menjadi bahasa mandiri dengan banyak fitur yang bertahan sampai sekarang.

Versi	Tanggal rilis	Didukung sampai	Catatan
3.0	6 Juni 1998	20 Oktober 2000	Pengembangan berpindah dari satu orang ke beberapa pengembang. Zeev Suraski dan Andi Gutmans menulis ulang dasar untuk versi ini.
4.0	22 Mei 2000	23 Juni 2001	Ditambahkan lebih maju dua tahap parse / mengeksekusi tag-parsing sistem yang disebut mesin Zend.
4.1	10 Desember 2001	12 Maret 2002	Diperkenalkan "superglobals" (<code>\$_GET</code> , <code>\$_POST</code> , <code>\$_SESSION</code> , dll.)

Versi	Tanggal rilis	Didukung sampai	Catatan
4.2	22 April 2002	6 September 2002	Disabled register_globals secara default. Data yang diterima melalui jaringan tidak dimasukkan langsung ke ruang nama global lagi, menutup kemungkinan lubang keamanan pada aplikasi.
4.3	27 Desember 2002	31 Maret 2005	Diperkenalkan antarmuka baris perintah (CLI), untuk melengkapi CGI.
4.4	11 Juli 2005	7 Agustus 2008	Memperbaiki bug korupsi memori, yang mengharuskan

Versi	Tanggal rilis	Didukung sampai	Catatan
			kompatibilitas biner terpecahkan dengan ekstensi yang dikompilasi melawan PHP versi 4.3.x.
5.0	13 Juli 2004	5 September 2005	Zend Engine II dengan model objek baru.
5.1	24 November 2005	24 Agustus 2006	Peningkatan kinerja dengan diperkenalkannya variabel compiler dalam engine PHP yang direkayasa ulang. Ditambahkan Objek Data PHP (PDO) sebagai antarmuka yang

Versi	Tanggal rilis	Didukung sampai	Catatan
			konsisten untuk mengakses database.
5.2	2 November 2006	6 Januari 2011	Diaktifkan ekstensi filter secara default.
5.3	30 Juni 2009	14 Agustus 2014	Dukungan JSON asli, Dukungan namespace, akhir binding statis , label loncat (goto terbatas), penutupan , arsip PHP (phar) , koleksi sampah untuk referensi melingkar, dukungan Windows yang ditingkatkan, sqlite3, mysqlnd

Versi	Tanggal rilis	Didukung sampai	Catatan
			<p>sebagai pengganti libmysql sebagai library yang mendasari ekstensi yang bekerja dengan MySQL , fileinfo sebagai pengganti mime_magic untuk mendapatkan dukungan MIME yang lebih baik, ekstensi Internasionalisasi, dan penghentian perluasan ereg.</p>
5.4	1 Maret 2012	3 September 2015	Dukungan karakter, dukungan sintaks array pendek. Item yang

Versi	Tanggal rilis	Didukung sampai	Catatan
			<p>dihapus: register_globals , safe_mode, allow_call_time_pass _reference , session_register () , session_unregister () dan session_is_registered (). Built-in server web. [60] Beberapa perbaikan pada fitur, kinerja dan persyaratan memori yang ada.</p>
5.5	20 Juni 2013	21 Juli 2016	Dukungan untuk generator , finally blok untuk penanganan

Versi	Tanggal rilis	Didukung sampai	Catatan
			pengecualian, OpCache (berdasarkan Zend Optimizer +) yang dibundel dalam distribusi resmi.
5.6	28 Agustus 2014	31 Desember 2018	Ekspresi skalar konstan, fungsi variadik , pembatalan ar gumen, operator eksponen baru, ekstensi pernyataan penggunaan untuk fungsi dan konstanta, debugger phpdbg baru sebagai modul

Versi	Tanggal rilis	Didukung sampai	Catatan
			SAPI, dan perbaikan kecil lainnya.
6.x	Tidak dilepaskan	N / A	Versi terbengkalai PHP yang direncanakan untuk menyertakan dukungan Unicode asli.
7.0	3 Desember 2015	3 Desember 2018	Zend Engine 3 (peningkatan kinerja dan dukungan integer 64-bit pada Windows), sintaks variabel seragam, proses kompilasi berbasis AST , menambahkan Closure

Versi	Tanggal rilis	Didukung sampai	Catatan
			<p>:: call () , bitwise Pergeseran konsistensi antar platform, ?? (null coalesce) operator, urutan Unicode codepoint escape , deklarasi tipe kembali, tipe scalar (bilangan bulat, float, string dan boolean) deklarasi, <=> " perbandingan tiga arah" operator, delegasi generator , kelas anonim , API CSPRNG yang lebih sederhana dan lebih konsisten tersedia,</p>

Versi	Tanggal rilis	Didukung sampai	Catatan
			mengganti banyak kesalahan internal PHP yang ada dengan pengecualian yang lebih modern, dan sintaks singkatan untuk mengimpor beberapa item dari namespace.
7.1	1 Desember 2016	1 Desember 2019	void return type , class constant visibility modifiers, jenis nullable, tipe pseudo iterable, menangkap beberapa jenis pengecualian.

Versi	Tanggal rilis	Didukung sampai	Catatan
7.2	30 November 2017	30 November 2020	

b. Bahasa Pemrograman JSP

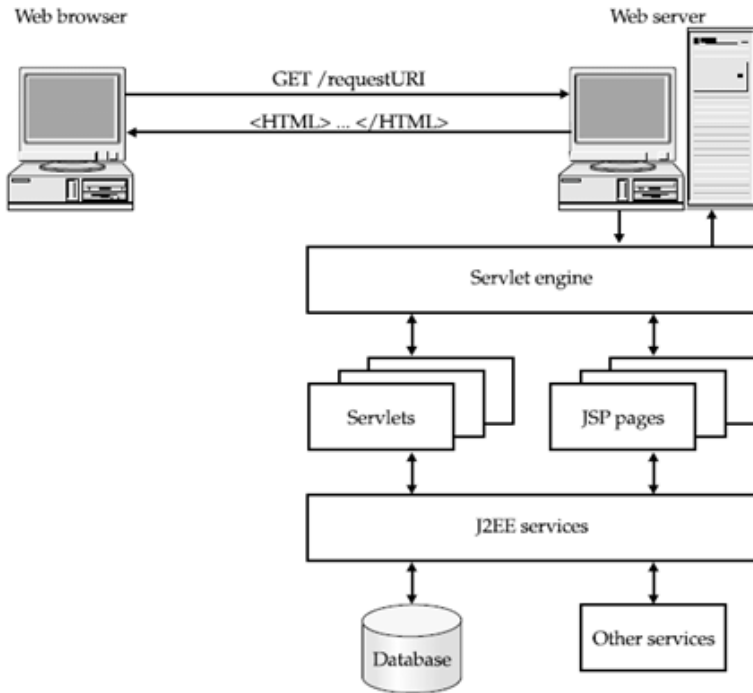
JSP adalah suatu teknologi web berbasis bahasa pemrograman Java dan berjalan di Platform Java, serta merupakan bagian teknologi J2EE (Java 2 Enterprise Edition). JSP sangat sesuai dan tangguh untuk menangani presentasi di web. Sedangkan J2EE merupakan platform Java untuk pengembangan sistem aplikasi enterprise dengan dukungan API (Application Programming Interface) yang lengkap dan portabilitas serta memberikan sarana untuk membuat suatu aplikasi yang memisahkan antara business logic (sistem), presentasi dan data.

JSP merupakan bagian dari J2EE dan khususnya merupakan komponen web dari aplikasi J2EE secara keseluruhan. JSP juga memerlukan JVM (Java Virtual

Machine) supaya dapat berjalan, yang berarti juga mengisyaratkan keharusan menginstal Java Virtual Machine di server, dimana JSP akan dijalankan. Selain JVM, JSP juga memerlukan server yang disebut dengan Web Container.

Teknologi JSP menyediakan cara yang lebih mudah dan cepat untuk membuat halaman-halaman web yang menampilkan isi secara dinamik. Teknologi JSP didesain untuk membuat lebih mudah dan cepat dalam membuat aplikasi berbasis web yang bekerja dengan berbagai macam web server, application server, browser dan development tool.

Java Server Pages (JSP) adalah bahasa scripting untuk web programming yang bersifat server side seperti halnya PHP dan ASP. Berbeda dengan servlet, JSP tidak perlu melakukan kompilasi terlebih dahulu sebelum dijalankan, tetapi server yang akan melakukan tugas tersebut.



Gambar 2.7. Proses Pemrograman JSP

JSP pada dasarnya juga halaman HTML dengan tambahan tag-tag JSP (scriptlet) `<% %>` yang berupa Java Code. File JSP mempunyai extension .jsp bukan .htm atau html. JSP engine (Java virtual mechine) melakukan proses parsing pada file.jsp selanjutnya mengenerate Java servlet source file. Kemudian mengkompilasi source file ini menjadi class file, hal ini

dilakukan pada saat pertama kali file .jsp dijalankan. Seperti tipe aplikasi java lainnya (Servlet, Applet, Midlet dll), JSP juga bertipe strongtype artinya penggunaan variable pada halaman tersebut harus dideklarasikan terlebih dahulu.

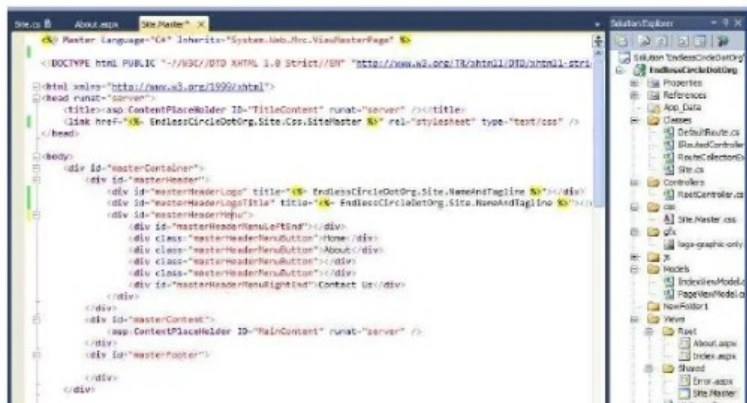
c. Bahasa Pemrograman ASP

ASP merupakan kepanjangan dari Active Server Pages. ASP merupakan suatu bahasa server script (seperti halnya: JSP, Cold Fusion, Perl, dan PHP), yang digunakan untuk menghasilkan halaman-halaman website yang bersifat dinamis. Di Indonesia, sampai saat ini, ASP masih tetap merupakan primadona diantara server script yang lain, walaupun sekarang sudah mulai dijumpai banyak juga programmer-programmer yang mulai menggunakan JSP, PHP dan Cold Fusion.

Untuk bisa menjalankan ASP di komputer, maka terlebih dahulu menginstall web server, dimana contoh web server yang mendukung ASP antara lain: Microsoft Personal Web Server (MS PWS), jika anda menggunakan Windows 9x, atau MS Internet Information Server (MS IIS

4.0, MS IIS 5.0) jika anda menggunakan Win NT atau Win 2000. Jika menggunakan Windows 2000, biasanya IIS 5 sudah terinstall secara default. Kunjungi website www.microsoft.com untuk mengdownload Personal Web Server apabila komputer belum terinstall server jenis ini.

Active Server Pages atau disingkat ASP adalah salah satu bahasa pemrograman web untuk menciptakan halaman web yang dinamis.



Gambar 2.8 Contoh Pemrograman ASP

ASP merupakan salah satu produk teknologi yang disediakan oleh Microsoft. ASPdiproses melalui web

server dan hasil proses ini menghasilkan HTML yang akan dikirimkan melalui penjelajah web (browser). ASP bekerja pada web server dan merupakan server side scripting. Active Server Pages (ASP) ialah pengaturan web yang digunakan untuk membina halaman web yang dinamik dan bersifat interaktif. Active server page adalah fasilitas yang diberikan oleh Microsoft untuk memudahkan pembuatan aplikasi-aplikasi Web Server. Hal ini di mungkinkan dengan menggabungkan komponen-komponen HyperText Markup Language (HTML) dengan Visual Basic Script (VBScript). Sehingga didalam sebuah halaman Web dengan Active Server Page terdapat bagian HTML dan kode VBScript.

ASP diciptakan oleh Microsoft untuk menjawab tantangan pemrograman web dinamis di mana isi dari sebuah website dapat diprogram untuk mendapatkan hasil yang berbeda. Tidak seperti HTML yang hanya menampilkan isi yang statis, ASP mampu menampilkan isi halaman yang berbeda sesuai dengan tujuan pemrogramannya. ASP telah mencapai versi 3.0 yang berjalan pada platform Windows 2000 Professional

atau Server. Versi sebelumnya, yaitu versi 2.0, dapat berjalan pada platform lain seperti Linux, Solaris, OS/2, dan Novell. Bahasa ASP murni Object Oriented Programming (OOP) memiliki struktur bahasa yang sama dengan bahasa Visual Basic yang mudah dimengerti karena sintaksnya menggunakan Bahasa Inggris. Seperti diketahui, bahwa Visual Basic adalah bahasa pemrograman yang paling mudah dimengerti dan dipelajari dibandingkan bahasa pemrograman lainnya. Oleh karena itu ASP sangat mudah dipelajari oleh para pemula sekalipun untuk membuat aplikasi web, disamping kehandalan dan kelengkapan fasilitas yang diberikan.

Kelebihan ASP

- Ada beberapa keuntungan yang diberikan oleh ASP yaitu:
- Mudah dalam mempelajarinya.
- Mudah dikembangkan dari sistem yang telah dimiliki
- Tidak perlu kompilasi.

Mudah menggabungkan dengan teknologi lain seperti CGI, DCOM, ActiveX dan lain-lain. Dengan ActiveX Data Object dan SQL pembuatan Web Data Base menjadi lebih mudah. Memberikan fasilitas keamanan seperti session Object, cookie, dan fasilitas transaksi online yang aman dengan protokol SSL, semua dapat dilakukan dengan sedikit kode tambahan.

d. Bahasa Pemrograman ColdFusion

ColdFusion adalah server aplikasi dan framework pengembangan perangkat lunak yang digunakan untuk mengembangkan perangkat lunak komputer, khususnya situs web dinamis. ColdFusion merupakan produk yang mirip dengan Microsoft ASP.NET, JavaServer Pages dan PHP. ColdFusion adalah teknologi yang pertama kali menyediakan kemampuan membuat situs web dinamis yang terhubung pada basis data backend. ColdFusion pertama kali dikembangkan di Allaire Corporation, oleh kakak beradik JJ dan Jeremy Allaire sampai dengan versi 5 sebelum dibeli oleh Macromedia. Versi yang dijual oleh Macromedia adalah Macromedia

Coldfusion MX 7. Setelah Macromedia dibeli oleh Adobe, perangkat lunak ini diganti namanya menjadi Adobe ColdFusion 8. ColdFusion adalah fitur scripting yang menggunakan, ColdFusion Markup language (CFML). Berikut adalah beberapa fitur lain dari ColdFusion: Hal ini didasarkan pada standar industri arsitektur Java pada umumnya, dan mendukung semua standar penting dan inisiatif, yang merupakan aplikasi cepat, berkat skalabilitasnya dalam, pengkodean multi-threaded arsitektur, layanan berbasis ColdFusion datang dengan semua kait yang diperlukan untuk hyperlink ke hampir semua aplikasi database dan sistem eksternal lainnya itu menyediakan semua peralatan yang diperlukan yang perlu untuk memecahkan masalah dan debug aplikasi, aplikasi sangat mudah karena tidak memiliki proses kompilasi atau menghubungkan langkah yang diperlukan, aplikasi dapat dengan mudah menghasilkan kode berbasis internet pada sisi klien juga mendukung aplikasi consumer server cache manajemen serta clustering server, aplikasi yang mudah untuk menguji dan menggelar sangat mudah, aplikasi mudah untuk

berkembang karena pengkodean kecokelatan yang bebas dengan sederhana tag gaya html aplikasi dapat berkembang pesat dengan akses database disederhanakan, dengan mudah dapat mengambil knowledge dari semua sistem perusahaan umum seperti EnergeticDirectory, LDAP, SMTP, HTTP, POP, FTP, dan lain-lain.

```
<tr>
  <td>
    <cfif strKeyword contains "coldfusion">
      <font size="2" color="#0000ff"><b>
        Keyword anda mengandung kata ColdFusion
      </b></font>
    <cfelse>
      Keyword anda tidak mengandung kata ColdFusion
    </cfif>
  </td>
</tr>
```

Gambar 2.9 Contoh Pemrograman ColdFusion

ColdFusion mendukung bahasa pemrograman lain selain CFML, seperti server-aspectscript Aksi yang dapat ditulis dalam javascript dan dikenal sebagai CFScript.

Fitur lain yang sangat berguna dari ColdFusion adalah integrasi yang mungkin dalam lapisan aplikasi yang berbeda, tetapi mereka semua memiliki fungsi yang sama. Untuk lebih tepat, dapat digunakan untuk layanan remote seperti SOAP, dan SOAP banyak digunakan untuk bertukar informasi terstruktur dalam layanan Web.

e. Bahasa Pemrograman C

Bahasa Pemrograman C adalah sebuah bahasa pemrograman komputer yang bisa digunakan untuk membuat berbagai aplikasi (general-purpose programming language), mulai dari sistem operasi (seperti Windows atau Linux), antivirus, software pengolah gambar (image processing), hingga compiler untuk bahasa pemrograman, dimana C banyak digunakan untuk membuat bahasa pemrograman lain yang salah satunya adalah PHP.

```

#include <stdlib.h>
#include <stdio.h>

int main()
{
    int r;
    float phi, luas;
    phi=3.14;

    printf("Masukan jari-jari lingkaran (cm) :",r);
    scanf("%i",&r);

    luas=phi*r*r;

    printf("Jadi luas lingkaran tersebut adalah :%f \n",luas);
    system ("pause");
    return 0;
}

```

Gambar 2.10. Contoh Pemrograman C

Meskipun termasuk general-purpose programming language, yakni bahasa pemrograman yang bisa membuat berbagai aplikasi, bahasa pemrograman C paling cocok merancang aplikasi yang berhubungan langsung dengan Sistem Operasi dan hardware. Ini tidak terlepas dari tujuan awal bahasa C dikembangkan.

Bahasa pemrograman C dibuat pertama kali oleh Dennis M. Ritchie pada tahun 1972. Saat itu Ritchie

bekerja di Bell Labs, sebuah pusat penelitian yang berlokasi di Murray Hill, New Jersey, Amerika Serikat.

Ritchie membuat bahasa pemrograman C untuk mengembangkan sistem operasi UNIX. Sebelumnya, sistem operasi UNIX dibuat menggunakan bahasa assembly (assembly language). Akan tetapi bahasa assembly sendiri sangat rumit dan susah untuk dikembangkan.

Dengan tujuan mengganti bahasa *assembly*, peneliti di Bell Labs membuat bahasa pemrograman B. Namun bahasa pemrograman B juga memiliki beberapa kekurangan, yang akhirnya di lengkapi oleh bahasa pemrograman C.

Dengan bahasa C inilah sistem operasi UNIX ditulis ulang. Pada gilirannya, UNIX menjadi dasar dari banyak sistem operasi modern saat ini, termasuk Linux, Mac OS (iOS), hingga sistem operasi Android. Berikut beberapa fitur serta keunggulan bahasa pemrograman C jika dibandingkan dengan bahasa pemrograman lain:

- C sebagai bahasa pemrograman prosedural

Konsep pemrograman prosedural adalah sebuah metode pemrograman yang setiap baris perintah diproses secara berurutan dari baris paling atas hingga baris paling bawah. Selain itu bisa terdapat fungsi tambahan (function) yang digunakan untuk menyelesaikan berbagai tugas. Bahasa pemrograman C termasuk ke dalam kelompok ini. Selain konsep prosedural, terdapat juga konsep pemrograman object (object-oriented programming). Di dalam bahasa pemrograman object, setiap tugas akan dijalankan menggunakan class dan object. Contoh bahasa pemrograman object adalah JAVA. Bagi pemula, sangat disarankan untuk mempelajari bahasa pemrograman prosedural terlebih dahulu, baru kemudian masuk ke dalam bahasa pemrograman object. Ini juga menjadi alasan untuk belajar bahasa C terlebih dulu baru kemudian masuk ke bahasa pemrograman object seperti JAVA. Beberapa bahasa pemrograman juga mendukung

konsep prosedural dan objectsekaligus, contohnya bahasa pemrograman C++ dan PHP.

- Bahasa C sangat cepat dan efisien

Aplikasi yang dibuat menggunakan bahasa C bisa dieksekusi dengan sangatcepat serta berukuran kecil. Ini karena C bisa langsung berkomunikasi denganhardware, sebuah fitur yang jarang tersedia di bahasa pemrograman modernseperti JAVA, PHP, maupun Phyton.Akan tetapi, hal ini juga memiliki kelemahan. Bahasa C relatif sederhana dantidak memiliki fitur-fitur modern seperti garbage collection dan dynamic typing.

- C adalah portable language

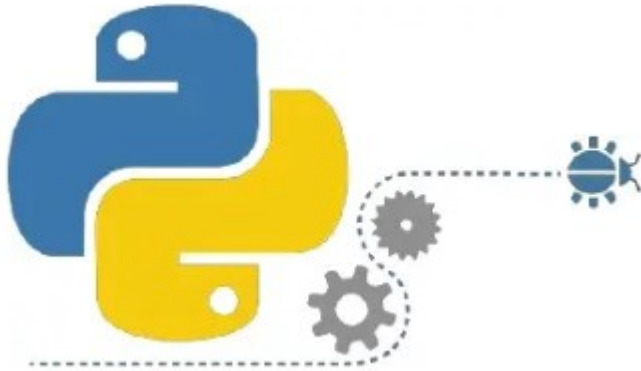
Maksudnya, bahasa pemrograman C bisa di-compile ulang supaya berjalan diberbagai sistem operasi tanpa perlu mengubah kode-kode yang ada. Aplikasiyang dibuat di Windows dengan bahasa C, bisa dipindahkan ke Linux dengansedikit atau tanpa modifikasi.

- C merupakan “induk” dari bahasa pemrograman modern

Bahasa pemrograman C banyak menginspirasi bahasa pemrograman lain, seperti C++, C#, Objective C, PHP, JAVA, JavaScript dan masih banyak lagi. Dengan mempelajari bahasa C, anda akan familiar dan lebih mudah saat berpindah ke bahasa pemrograman lain yang merupakan turunan dari C.

f. Bahasa Pemrograman Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.



Gambar 2.11. Logo Python

Python merupakan salah satu bahasa pemrograman tingkat tinggi (high level language). Python dirancang untuk memberikan kemudahan bagi programmer melalui segi efisiensi waktu, kemudahan dalam pengembangan dan kompatibilitas dengan sistem. Python bisa digunakan untuk membuat aplikasi standalone (berdiri sendiri) dan pemrograman script (scripting programming).

Keunggulan Python

- Mudah digunakan. Python merupakan bahasa pemrograman tingkat tinggi, artinya bahasa

python lebih mendekati bahasa manusia daripada bahasa mesin. Python juga tidak memiliki struktur dasar seperti C++ yang mengharuskan kamu menginclude sistemnya sebelum membuat program. Perintah-perintah yang digunakan di Python menggunakan bahasa inggris yang digunakan sehari-hari seperti print, input, dll.

- Kompabilitas dan Kemampuan Tinggi Python memiliki kemampuan yang tinggi dan mampu menciptakan aplikasi yang sederhana hingga aplikasi yang kompleks. Bahasa pemrograman python mendukung pemrograman berbasis grafis (GUI Programming). Python juga memiliki kelebihan berupa pengalokasian memori secara dinamis.
- Mendukung OOP Python mendukung bahasa pemrograman berorientasi objek (OOP) sehingga programmer semakin mudah dalam menyelesaikan masalah karena pemrograman berorientasi objek merupakan teknik pendekatan

dengan objek-objek nyata yang ada di kehidupan sehari-hari.

Kekurangan Python

Beberapa penugasan terdapat di luar dari jangkauan kemampuan Python, seperti bahasa pemrograman dinamis lainnya, Python tidak secepat atau efisien sebagai statis, tidak seperti bahasa pemrograman C. Disebabkan karena Python merupakan interpreter, Python bukan merupakan perangkat bantu terbaik untuk pengantar komponen kinerja kritis Python tidak dapat digunakan sebagai dasar bahasa pemrograman implementasi untuk beberapa komponen, tetapi dapat bekerja dengan baik sebagai bagian depan script interface Python memberikan tingkat efisiensi dan flexibility trade off by dengan tidak memberikannya secara menyeluruh.

Sejarah Python

Python dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan

dari bahasa pemrograman ABC. Versi terakhir yang dikeluarkan CWI adalah 1.2. Tahun 1995, Guido pindah ke CNRI sambil terus melanjutkan pengembangan Python. Versi terakhir yang dikeluarkan adalah 1.6. Tahun 2000, Guido dan para pengembang inti Python pindah ke BeOpen.com yang merupakan sebuah perusahaan komersial dan membentuk BeOpen Python Labs. Python 2.0 dikeluarkan oleh BeOpen. Setelah mengeluarkan Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke Digital Creations. Saat ini pengembangan Python terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan Python Software Foundation. Python Software Foundation adalah sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual Python sejak versi 2.1 dan dengan demikian mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python sudah mencapai versi 2.6.1 dan versi 3.0. Nama Python dipilih oleh Guido sebagai nama bahasa ciptaannya karena

kecintaan Guido pada acara televisi Monty Python's Flying Circus. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna Python.

g. Bahasa Pemrograman C++

C++ adalah bahasa pemrograman komputer yang di buat oleh Bjarne Stroustrup, yang merupakan perkembangan dari bahasa C dikembangkan di Bell Labs (Dennis Ritchie) pada awal tahun 1970-an, Bahasa itu diturunkan dari bahasa sebelumnya, yaitu B, Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Pada perkembangannya, versi ANSI (American National Standard Institute) Bahasa pemrograman C menjadi versi dominan, Meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded, Bjarne Stroustrup pada Bell Labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi

dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (Object Oriented Programming).

Year	C++ Standard	Informal name
2011	ISO/IEC 14882:2011[4]	C++11
2007	ISO/IEC TR 19768:2007[5]	C++TR1
2003	ISO/IEC 14882:2003[6]	C++03
1998	ISO/IEC 14882:1998[7]	C++98

Gambar 2.12. Standart Komite Pemrograman C++

Pada tahun 1998, C++ Standar Komite (ISO/IEC JTC1/SC22/WG21 WorkingGroup) mengeluarkan Standar Internasional ISO/IEC 14882:1998 yang digunakan selama beberapa tahun. Pada tahun 2003

dirilis versi yang telah dikoreksi, ISO/IEC14882:2003. Pada tahun 2005, Laporan Teknis yang disebut "Laporan Teknis Perpustakaan 1" (sering dikenal sebagai TR1), dirilis. Revisi terbaru dari standar C++ adalah C++11 (sebelumnya dikenal C++0x) telah disetujui oleh ISO/IEC pada tanggal 12 Agustus 2011. Telah diterbitkan sebagai 14882:11.

h. Bahasa Pemrograman Java

Java adalah bahasa pemrograman berorientasi objek murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya (C++, Ada, Simula). Java diciptakan oleh James Gosling, developer dari Sun Microsystems pada tahun 1991.

Karakteristik Java antara lain:

- Sederhana (Simple)
- Berorientasi Objek (Object Oriented)
- Terdistribusi (Distributed)
- Interpreted

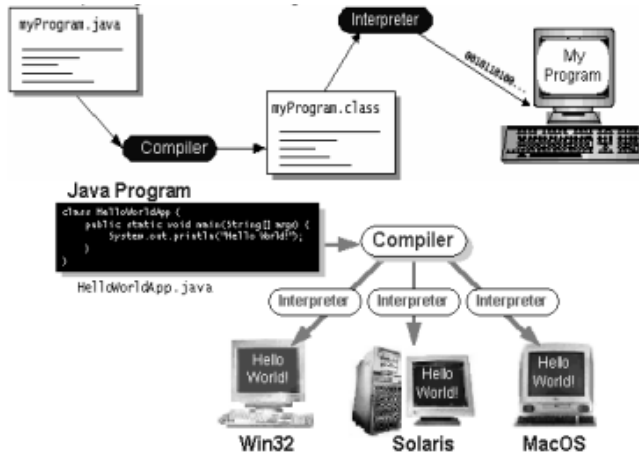
- Robust
- Aman (Secure)
- Architecture Neutral
- Portable
- Performance
- Multithreaded
- Dinamis

Macam-macam Java 2 Software Developer Kit

(J2SDK) antara lain:

- J2SE (Java 2 Standard Edition)
- J2EE (Java 2 Enterprise Edition)
- J2ME (Java 2 Micro Edition)

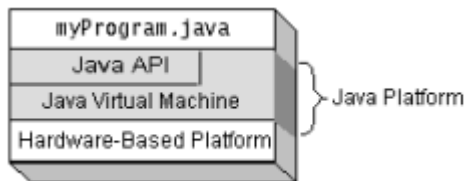
Cara kerja Java dapat digambarkan sebagai berikut:



Gambar 2.13. Cara Kerja Java

Java mempunyai platform yaitu:

- Java Virtual Machine (Java VM)
- Java Application Programming Interface (Java API)



Gambar 2.14. Platform Java

i. Bahasa Pemrograman XML

XML (eXtensible Markup Language) merupakan bahasa web turunan dari SGML (Standart Generalized Markup Language) yang ada sebelumnya. XML hampir sama dengan HTML, dimana keduanya sama-sama turunan dari SGML. Teknologi XML dikembangkan mulai tahun 1966 dan mendapatkan pengakuan dari Worl Wide Web Consortium (W3C) pada bulan Februari 1998. Sedangkan SGML sendiri telah dikembangkan pada awal tahun 1980-an. Pada saat HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting SGML dan dengan berpedoman pada pengembangan HTML menghasilkan bahasa markup yang tidak kalah hebatnya dengan SGML.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa meta-markup yang menyediakan format tertentu untuk

dokumen-dokumen yang mempunyai data terstruktur. Bahasa markup adalah mekanisme untuk mengenal secara terstruktur di dokumen. XML adalah suatu aplikasi profil dari SGML. Seperti yang didefinisikan dari ISO 8879, SGML adalah cara standart dan vendor-independent.

XML tidak mempunyai definisi secara tepat karena ada yang berpendapat bahwa XML bukanlah suatu bahasa pemrograman, melainkan XML merupakan sintaks yang digunakan untuk menjelaskan bahasa markup lain (Dournaee,2002), sehingga dinamakan meta-language. Meskipun demikian pendapat yang XML bukan merupakan bahasa markup, didasarkan bahwa XML merupakan bahasa markup terpisah untuk tujuan terpisah. Selain itu XML bukanlah solusi semua hal untuk tujuan semua user. Sedangkan peran dari markup itu sendiri berupa:

1. Markup dapat menambah maksud arti (semantic) suatu data.
2. Dapat memisahkan data.

3. Dapat mendefinisikan peran data.
4. Dapat mendefinisikan batasan data.
5. Dapat mendefinisikan keterhubungan data.

XML merupakan sebuah himpunan bagian (subset) dari SGML yang bertujuan agar SGML secara generik dapat melayani, menerima, dan memproses di dalam web dengan cara seperti yang dimungkinkan HTML saat ini. XML didesain untuk kemudahan implementasi dan interoperabilitas dengan SGML maupun HTML. XML adalah bahasa markup yang dirancang untuk penyampaian informasi melalui World Wide Web (www) atau sering disebut web saja. (W3C, 2000).

XML merubah cara kita berpikir untuk mengembangkan suatu software terutama aplikasi web. Masalah di era sekarang adalah bagaimana caranya untuk bertukar informasi antar satu aplikasi dengan aplikasi lain. Kadang kolaborasi antara satu aplikasi dengan aplikasi yang lain masih harus ditentukan dengan spesifikasi aplikasi tersebut. Padahal

seharusnya kita hanya perlu mendapatkan informasi data bukan mengerti cara kerja aplikasi lain itu, disinilah visi internet yang belum tercapai. Visi ini adalah dunia internet dimana PC, server, smart devices dan internet-based device dapat berkolaborasi tanpa ada halangan. Bisnis-bisnis akan dapat bertukar data menyediakan customized dan comprehensive solusi kepada customer. Dan yang paling utama adalah informasi yang dibutuhkan dapat diakses dari mana saja dan dengan computing device, platform, atau aplikasi yang kita gunakan. XML dapat memungkinkan pertukaran informasi atau data antar device (server, PCs, smart device, aplikasi, dan situs web). Data ini akan menjadi independent (unlocked), memudahkannya untuk diorganisir, diprogram, dan dirubah, dan ditukar antar situs web atau aplikasi apa saja. Karena kebutuhan ini, maka makin banyak teknologi berbasis XML yang keluar. Contohnya adalah SOAP (Simple Project Acces Protocol) dan UDDI (Universal Description Discovery and Integration).

B. Struktur Dasar Bahasa Pemrograman

Belajar membuat program tentu tidak jauh-jauh dari yang namanya struktur dasar bahasa pemrograman, logika dasar pemrograman, algoritma pemrograman, dan lain sebagainya. Kesemuanya ini tentu harus kita pelajari dengan tepat dan cermat karena belajar pemrograman merupakan sebuah aktivitas membiasakan diri kita untuk membuat program, seperti halnya kita membiasakan diri untuk menggunakan bahasa untuk berbicara.

Belajar bahasa baru menuntut kita untuk sering menggunakan bahasa tersebut dan mempraktikannya sesering dan sebanyak mungkin. Selain agar tidak lupa, pembiasaan ini berguna agar seseorang dapat mengucapkan dengan tepat kata-kata yang sedang dipelajarinya.

Selain itu, ketika seseorang sedang belajar menggunakan bahasa baru, tentu orang tersebut harus belajar untuk menyusun kata-kata yang tepat, agar apa yang disampaikan dapat terkomunikasikan dengan sempurna. Ini pun berlaku bagi orang-orang yang sedang belajar menulis baris-baris program, dengan menggunakan pilihan bahasa yang sesuai dengannya.

Agar dapat menggunakan bahasa pemrograman sesuai dengan apa yang diinginkan, kita harus mengerti struktur dasar bahasa pemrograman yang tepat untuk digunakan. Semakin banyak pilihan bahasa pemrograman yang dipelajari, semakin banyak struktur yang harus dipelajari. Setiap bahasa pemrograman memiliki struktur dasar bahasa pemrograman yang berbeda-beda, akan tetapi struktur tersebut dapat dimasukkan dalam beberapa kategori.

Beberapa kategori struktur dasar bahasa pemrograman yang dapat dipelajari oleh pemula adalah sebagai berikut.

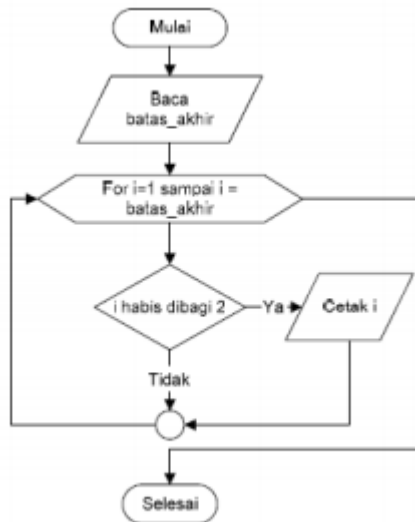
a. Struktur Runtutan

Merupakan struktur algoritma paling dasar yang berisi rangkaian intruksi yang diproses secara sekuensial, yang artinya dikerjakan satu per satu, mulai dari intruksi pertama sampai intruksi terakhir artinya dimulai dari langkah pertama, kemudian kedua, ketiga, dan seterusnya hingga akhir. Pada dasarnya suatu program memang menjalankan suatu proses dari yang dasar seperti struktur seperti ini.

b. Struktur Perulangan

Kadang kala, saat menjalankan sebuah program, kita harus menjalankan sebuah perintah secara berulang-ulang, agar program dapat berjalan secara berkelanjutan.

Struktur ini dapat menjadi pilihan yang tepat untuk membuat sebuah program menjalankan baris-baris perintah secara berulang-ulang.



Gambar 2.15. Contoh Struktur Flowchart Pengulangan

Dengan menggunakan struktur dasar bahasa pemrograman ini, seorang pembuat program dapat

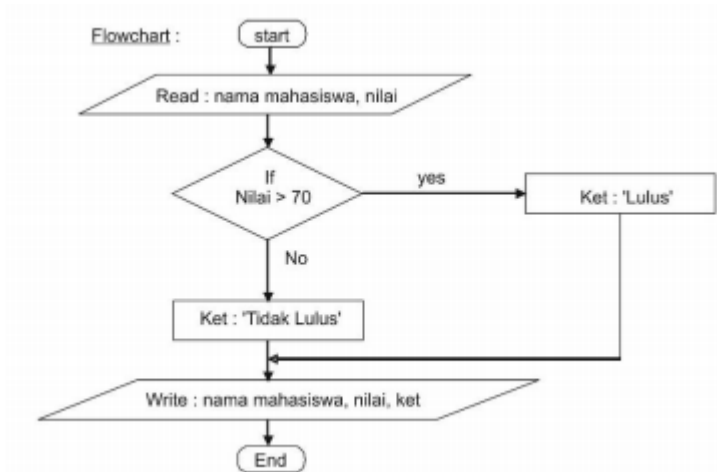
mengatur perintah-perintah dalam programnya agar berjalan secara berulang-ulang hingga pada suatu kondisi dimana syarat untuk perulangannya sudah tidak terpenuhi kembali.

Terdapat beberapa fungsi perulangan yang dapat digunakan untuk membuat struktur program ini, seperti For, While, dan While-do pada bahasa C++. Ketiga fungsi ini dapat digunakan untuk membuat sebuah program untuk menjalankan baris-baris perintah dalam syarat tertentu.

c. Struktur Percabangan

Membuat program juga tidak akan terlepas dari membuat sebuah pilihan dalam menjalankan perintah sesuai dengan syarat tertentu. Pembuat program dapat menggunakan struktur ini untuk membuat sebuah program dengan beberapa pilihan keputusan yang disesuaikan dengan syarat atau kondisi yang dikehendaki.

Sebagai contoh, saat kita ingin melihat apakah lulus atau tidak dalam sebuah mata pelajaran, logikanya adalah jika nilai kita lebih dari 70 maka kita lulus, dan jika kurang dari 70 maka kita tidak lulus. Dalam hal ini, maka kita dapat menggunakan struktur percabangan ini.



Gambar 2.16. Contoh Struktur Flowchart Percabangan

Dalam kondisi di atas, kita dapat menggunakan fungsi If-Else atau Switch Case dalam bahasa pemrograman C untuk board Arduino. Sebagai contoh berikut ini

Contoh struktur percabangan dengan menggunakan fungsi If-Else untuk kasus yang telah disebutkan membuat sebuah program merupakan aktivitas memilih struktur mana yang tepat untuk program tersebut. Sehingga seorang pembuat program harus mengetahui berbagai macam jenis struktur dasar bahasa pemrograman, sebagaimana yang telah dijelaskan di atas.

Setelah mengetahui berbagai macam struktur dan fungsi yang dapat digunakan, pembuat program dapat memilih dan menentukan struktur mana yang tepat untuk program yang akan dibuat. Dengan struktur yang tepat maka program yang dibuat akan sesuai dengan yang diinginkan

C. User Interface dan User Experience Pada Aplikasi Android

1. User Interface

Desain antarmuka (interface) merupakan elemen vital dalam sebuah game edukatif, karena selain bersifat estetik, juga harus fungsional.

User Interface Design atau yang bahasa Indonesianya itu Desain Antarmuka Pengguna adalah Desain antarmuka untuk mesin dan perangkat lunak, seperti komputer, peralatan rumah tangga, perangkat mobile, dan perangkat elektronik lainnya, dengan fokus pada memaksimalkan pengalaman pengguna.

Bahasa gampangnya yaitu UI Design adalah bagaimana suatu website atau aplikasi yang kamu

buat terlihat seperti apa. Orang biasa menyebutnya sebagai tampilan atau desain sebuah website.

2. User Experience

User experience adalah nilai atau value yang di berikan kepada user / pengguna ketika menggunakan produk atau jasa.

User Experience Design (UXD atau UED) adalah proses meningkatkan kepuasan pengguna dengan suatu produk dengan meningkatkan kegunaan, aksesibilitas, dan kepuasan yang disediakan dalam interaksi dengan produk.

Mengembangkan user experience agar pengguna merasa puas akan produk atau jasa yang diberikan bukan hanya tanggung jawab satu orang atau tim, melainkan merupakan visi perusahaan.

“Design is not just what it looks like and feels like. Design is how it works.” — Steve Jobs.

Desain bukan hanya menyangkut masalah penampilan dan perasaan, melainkan cara kerjanya.

Itulah pandangan CEO Apple terhadap pentingnya User Experience dalam visi perusahaannya.

Desain UX yang luar biasa tidak hanya menyoroti dan mempromosikan produk kamu, tetapi juga telah menjadi bagian penting dalam membangun dan menumbuhkan kepercayaan terhadap pelanggan.

Produk atau konten yang menarik tanpa adanya elemen User Experience yang baik dapat memengaruhi kemampuan organisasi untuk mencapai tujuan bisnisnya.

User experience (UX) yang baik selalu menjadi bagian dari proses pengembangan suatu produk. Berinteraksi dengan pengguna untuk membangun kombinasi unik dari struktur, konten, dan pengalaman pengguna agar dapat mencapai tujuan dengan cara yang efisien.

Faktor utama dalam kesuksesan tergantung pada User experience yang pengguna rasakan, maka dari itu sangat penting untuk menjadikan hal tersebut sebagai prioritas dalam mengembangkan desain.

Cara ini mampu membuat desain terlihat rapi, simpel, intuitif, fleksibel, dan menarik serta memberikan pengalaman berbeda kepada pengguna produk atau jasa dan membuat terlihat unik di hadapan kompetitor lain.

Proses Desain User Experience (UX)

Proses desain User Experience adalah metode berulang yang membantu untuk terus meningkatkan dan memoles desain. Dalam prosesnya, dapat melalui tahapan yang berbeda berulang kali sambil mengevaluasi desain pada setiap tahapnya.

Setiap tahap melibatkan stakeholder atau pemegang peran penting yang relevan dalam organisasi yang mengambil bagian dalam proses untuk membuat produk sangat efisien dan memberi manfaat.

Proses desain User Experience melibatkan enam tahap berikut:

1. Understand – Understand ialah memahami apa yang dibutuhkan pengguna

2. Research – Menganalisa pesaing atau kompetitor, Meriset Trend UX terkini, dan memperhatikan aturan-aturan UX yang berlaku
3. Sketch – Mengumpulkan Ide, Menggambarkan sketsa dan wireframes, Serta lakukan evaluasi secara berulang-ulang
4. Design – Desain image atau gambar, Membuat Prototype
5. Implementation – Mengetes fungsi desain, bangun pengalaman
6. Evaluate – Lakukan uji coba kegunaan desain, Buat laporan audit, indentifikasi pengembangan yang ada.

1. Understand (Pemahaman)

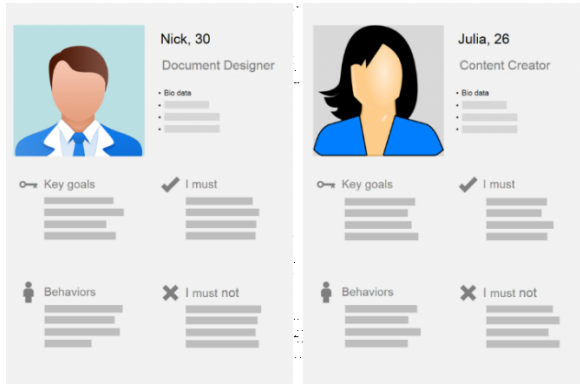
Desain mengatasi masalah, guna memberikan solusi kamu perlu memahami akar permasalahannya. Sebelum membangun desain, ada baiknya agar tim desain kamu memahami apa-apa saja yang dibutuhkan untuk memecahkan masalah yang ada.

Untuk menganalisa hal apa saja yang dibutuhkan, ikuti standar riset yang diterapkan perusahaan kamu baik secara kontekstual maupun interview individu. Di waktu yang sama lakukan observasi kepada pengguna secara nyata.

Lakukan sesi brainstorming dengan user, dan tunjukkan produk atau jasa yang ditawarkan, serta minta pendapat mereka tentang hal tersebut.

Business Manager adalah peran dalam organisasi yang bekerja secara langsung dengan klien dan mendapatkan persyaratan dari mereka. Tim desain dapat bekerja sama dengan Business Manager untuk memahami apa yang pengguna butuhkan.

Pengetahuan tentang pengguna dan lingkungannya bisa membantu kamu dalam membuat desain user experience yang baik.



2. Research (Penelitian)

Riset adalah langkah dasar dalam membangun desain User Experience.

"Butuh beberapa detik untuk menggambar desain User Experience, tapi butuh 34 tahun untuk belajar menggambarnya dalam beberapa detik" - Paula Scher

Tim desain melakukan penelitian untuk mengeksplorasi bagaimana dunia luar bekerja pada fitur-fitur tersebut.

Sherif Amin, Desainer Produk menyebutnya sebagai Analisis Kompetitif UX. Ia mencantumkan tiga tujuan analisis yaitu:

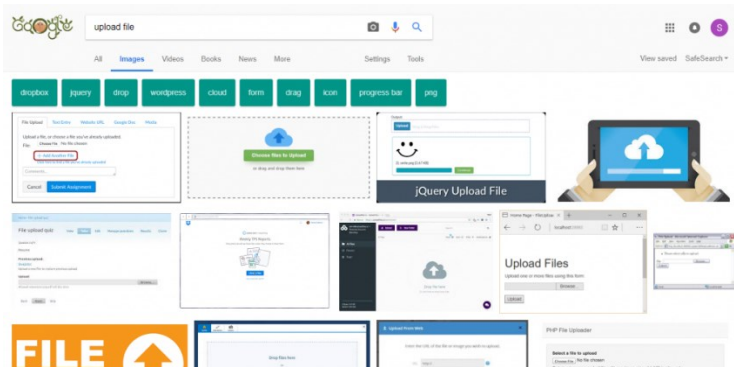
- Pahami Kompetisi Pasar Kamu

- Pelajari Domain produk kamu
- Cari Inspirasi dan Ide dari Kompetitor Kamu

Terus update terhadap trend User Interface yang ada, bangun prinsip, dan panduan User Experience yang ada.

Saat meneliti, coba pikirkan desain dan opsi yang dapat membangun desain user experience yang bagus.

Sebagai contoh, ketika akan membuat desain cara upload file yang ramah UX, bisa mencari terlebih dahulu Desain Interface nya di Google Image:



3. Sketch (Sketsa)

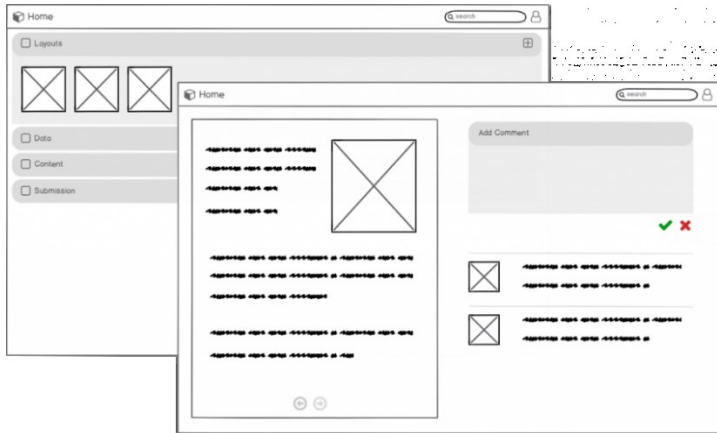
Tahap ini melibatkan definisi UI dari fitur yang diperlukan. Tim desain menggerakkan kegiatan ini yang didasarkan pada dua tahap terakhir dari proses.

Gambar sketsa kertas, Whiteboard Flow, dan gambar kerangka untuk berbagi ide kamu dengan para pemangku kepentingan. Tahap ini sendiri merupakan proses berulang.

“Desain bukanlah sesuatu yang baru saja kamu buat dan mulai menggunakannya. Gambar dan buat konsep, gambar dan cetak secara berulang-ulang, sehingga menciptakan pengalaman yang tak tertandingi.”

Pengujian dan evaluasi wireframes atau kerangka gambar adalah bagian dari tahap ini. Tim desain membangun mockup awal dan membagikannya dengan para pemangku kepentingan untuk mendapatkan saran dari mereka.

Sepanjang proses, penting untuk mengingat tujuan – buat kerangka atau sketsa desain yang dapat digunakan untuk mencapai kepuasan end-user atau pengguna akhir.



4. Design (Desain)

Langkah selanjutnya adalah memberikan sentuhan akhir pada desain UX yang dibuat.

Ubah mockup dan wireframe awal menjadi desain yang tampak hebat dengan memberikan sedikit sentuhan gaya dan tema yang sesuai.

Mempersiapkan dan berbagi spesifikasi desain (gambar, tema, warna, tipografi, ikonografi) ke tim Pengembangan juga merupakan bagian dari tahap ini. Share ide dengan mereka, dan cari tahu desain mana yang dapat memberikan Pengalaman user yang berbeda.

5. Implementasi

Karena orang-orang teknis berpartisipasi dalam tahap awal proses, mereka dapat memulai implementasi saat fase Desain sedang berlangsung.

Tim pengembang membangun fungsionalitas back-end desain terlebih dahulu dan menghubungkannya dengan User Interface.

Lebih baik tim Desain terlibat dalam langkah ini untuk membantu tahap pengembangan. Saat implementasi, mungkin akan ada perubahan kecil yang perlu dilakukan untuk menghasilkan desain UX yang optimal.

6. Evaluate (Evaluasi)

Ketika fitur produk diimplementasikan, produk akhir dievaluasi berdasarkan beberapa faktor berikut:

- Kegunaan sistem
- Sulit tidaknya ketika digunakan end User
- Apakah fleksibel dan mudah diubah?

- Apakah memberikan solusi yang diinginkan untuk mengatasi masalah pengguna?
- Tingkat kredibilitas yang membuat seseorang menggunakan produk kamu

D. Penggunaan Android Studio

1. Pengertian Android Studio

Android studio merupakan sebuah Integrated Development Environment (IDE) untuk platform Android. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers. Android studio bersifat free dibawah Apache License 2.0. Android Studio awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi beta 0.8 yang dirilis pada bulan juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Berbasiskan JetBrains' IntelliJ IDEA, Studio di desain khusus untuk Android Development. Ini sudah bisa di download untuk Windows, Mac OS X, dan Linux.

2. Instalasi Software Android Studio

Untuk dapat menggunakan aplikasi Android Studio ini ada beberapa software yang diperlukan yakni Java Development Kit (JDK) minimum versi 7 atau yang lebih tinggi dan software Android Studio yang dapat di download melalui situs web <http://developer.android.com>.

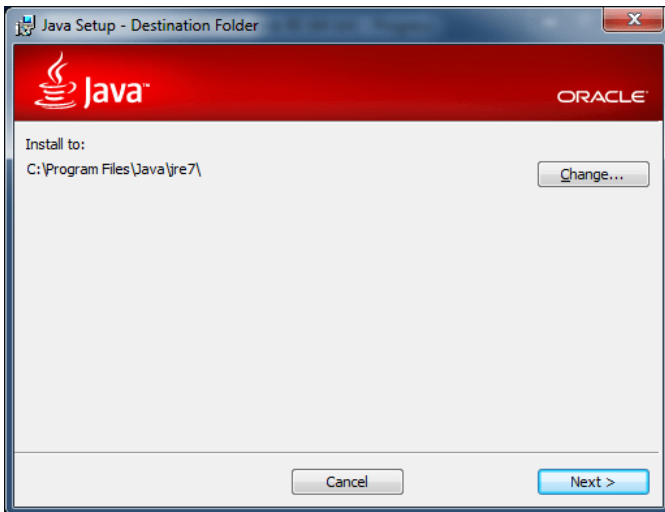
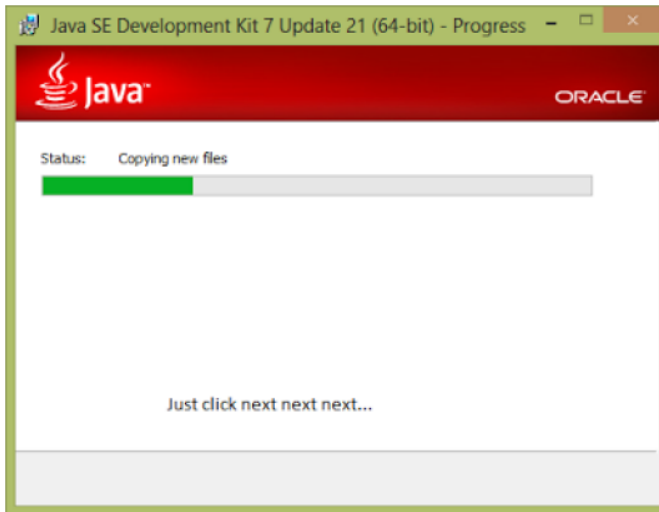
1. Install Java Development Kit (JDK) versi 8 – 64 bit.

Berikut ini langkah – langkah untuk melakukan instalasi software Java Development Kit (JDK) versi 8 – 64 bit pada windows:

a. Download software JDK melalui link berikut:

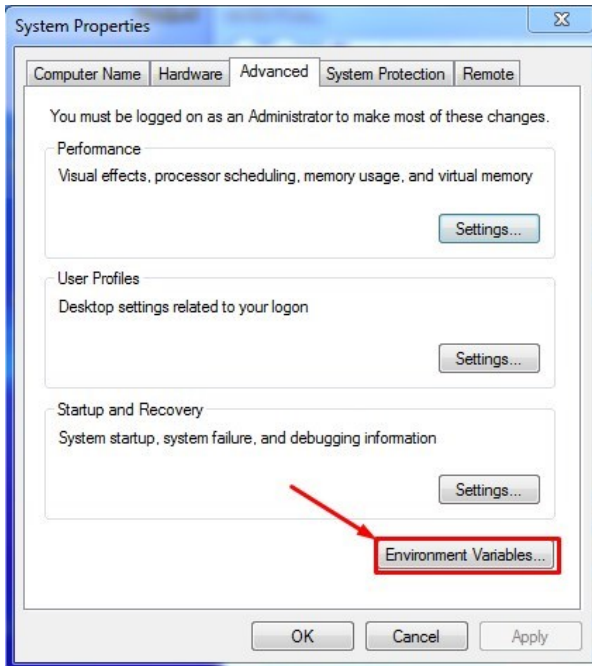
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

b. Install Java JDK dan ingat path dimana file java disimpan.

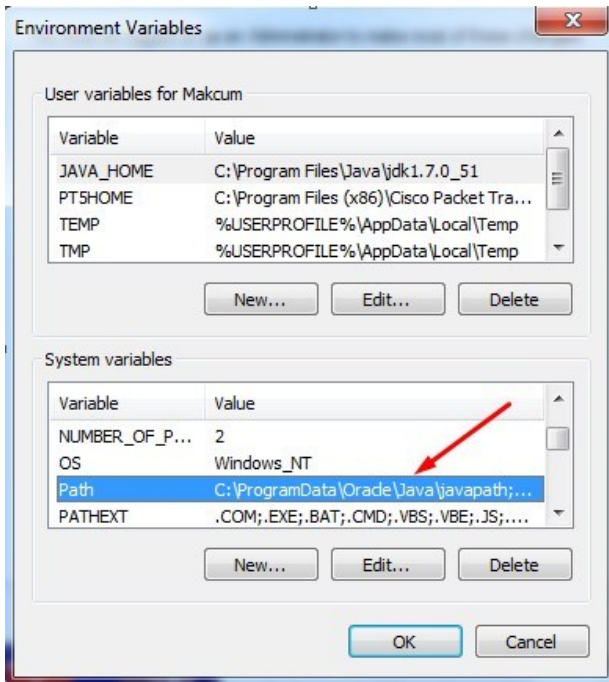


c. Klik tombol windows dan type Environment Variables > klik Edit the system Environment Variables.

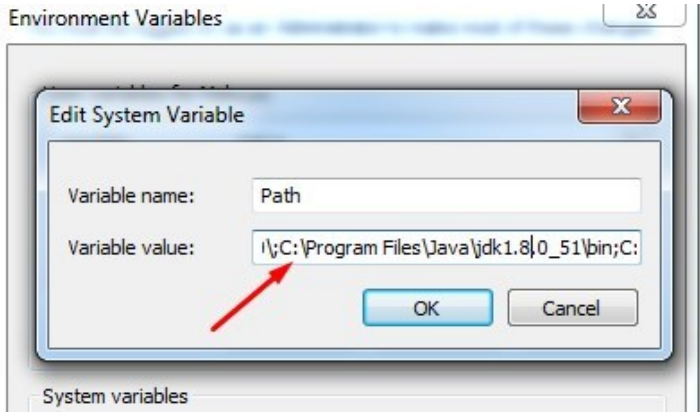
Pada jendela window akan tampil System Properties.
Kemudian klik Environment Variables.



d. Setelah itu klik New



e. Pada kolom pertama tulislah JAVA_HOME. Lihat lokasi dimana path java tadi di install. Dan pada kolom kedua Variable_value isilah path lokasi java tersebut di install.



f. Setelah itu pilih OK.

Pemecahan masalah:

Jika terdapat error pada saat memulai install Android Studio, hal itu disebabkan karena class java nya tidak ditemukan dan terdapat error yang berhubungan dengan path. Hal yang harus dilakukan adalah buka Environment, cari JAVA_HOME yang telah ditambahkan, kemudian masukkan path jre, bukan jdk.

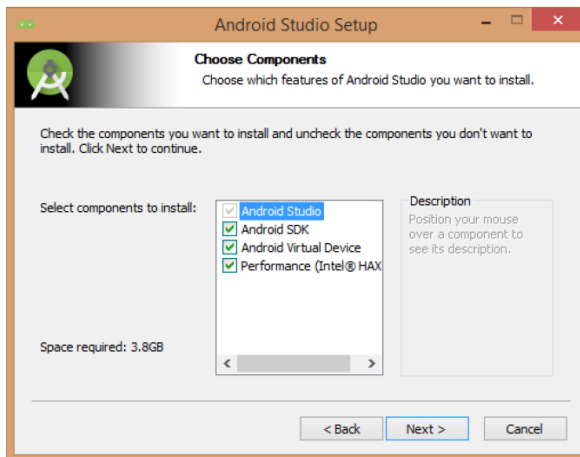
2. Install Android Studio

Setelah berhasil melakukan proses install Java Development Kit (JDK), selanjutnya install software Android Studio seperti langkah – langkah dibawah ini:

- a. Double klik pada file android_studio.exe kemudian pilih NEXT

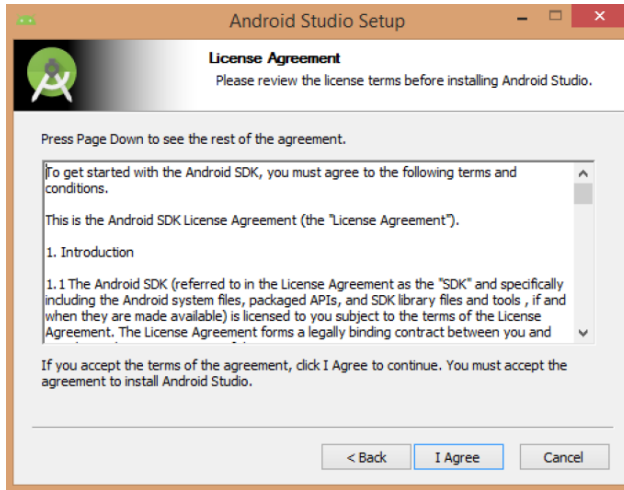


- b. Kemudian muncul tampilan seperti dibawah ini:

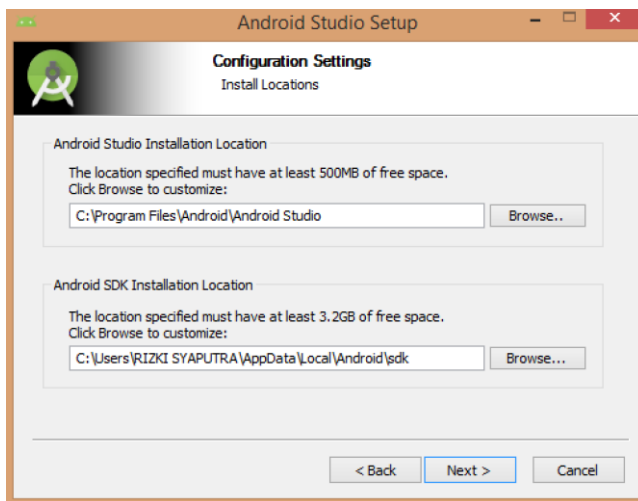


Kemudian pilih NEXT

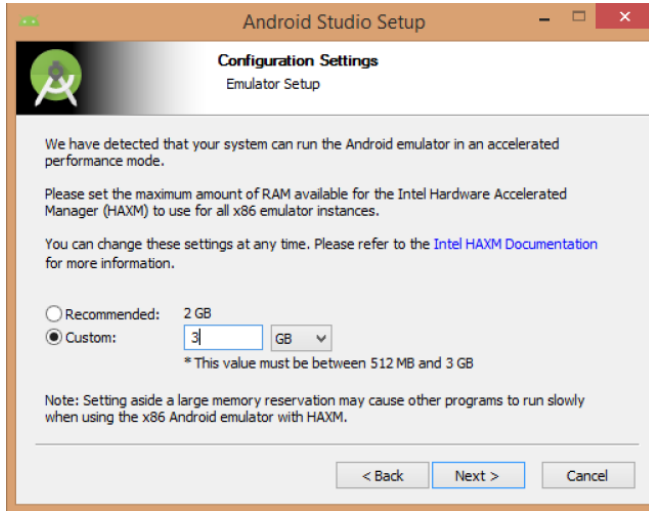
c. Setelah itu pilih Agree



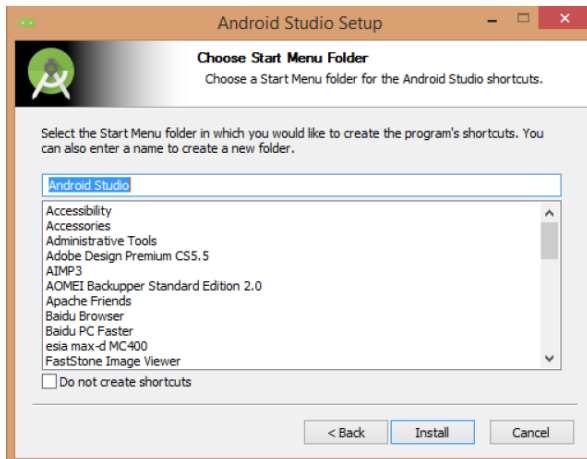
d. Klik NEXT



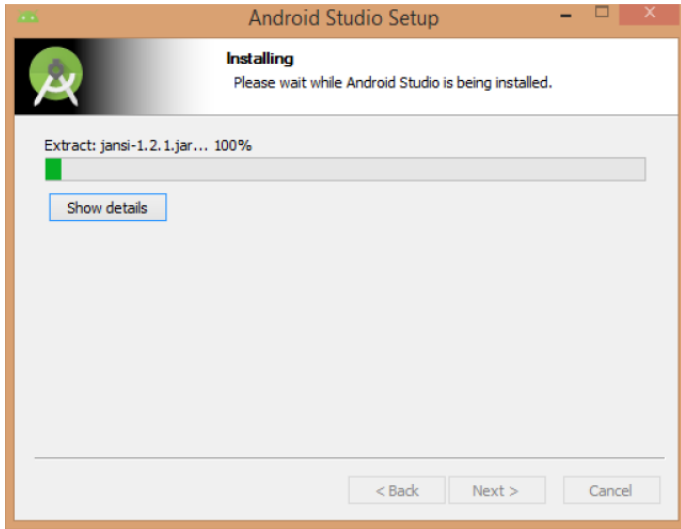
- e. Kemudian pilih alokasi RAM yang akan digunakan misalnya 3 GB. Kemudian pilih NEXT



- f. Klik Install, setelah itu tunggu proses selesai



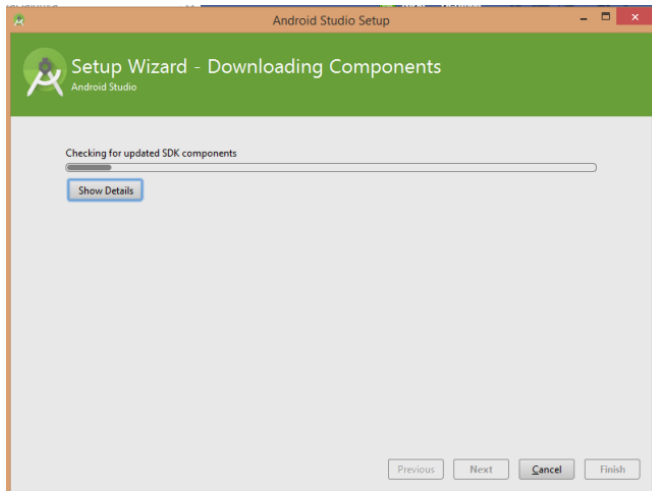
g. Klik Next



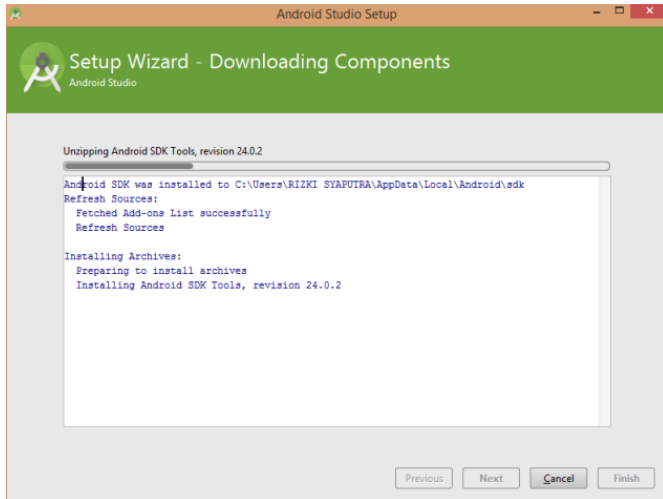
h. Setelah proses selesai KLIK NEXT



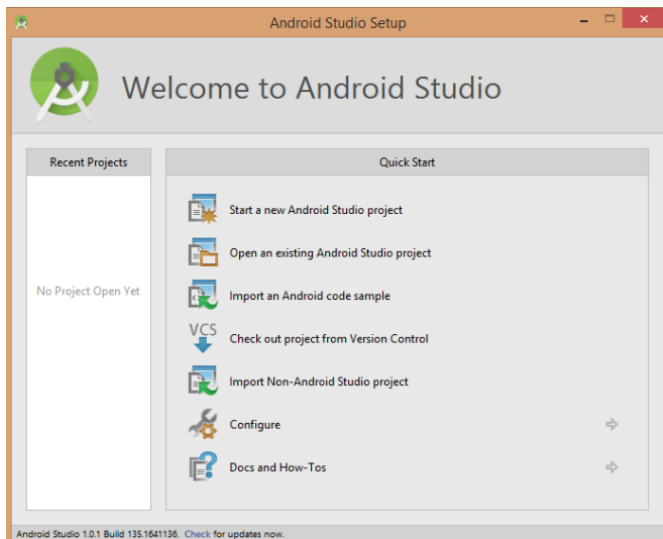
- i. Proses checking for Update SDK components.



- j. Proses check updated SDK components selesai, klik NEXT.



k. Proses instalasi selesai



3. Penggunaan dan Tools Pada Aplikasi Android Studio

Android Studio menyediakan alat untuk pengujian, dan mempublikasikan tahap proses development, serta lingkungan development terpadu untuk membuat aplikasi bagi semua perangkat Android. Lingkungan development menyertakan kode template dengan kode contoh untuk fitur aplikasi umum, alat pengujian dan kerangka kerja yang banyak, dan sistem pembangunan yang fleksibel.

Memulai proyek Android Studio

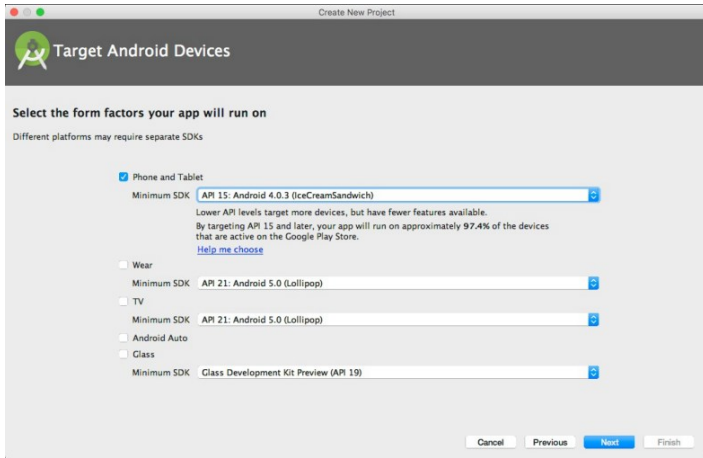
Setelah berhasil memasang Android Studio IDE, klik dua kali ikon aplikasi Android Studio untuk memulainya. Pilih Start a new Android Studio project di jendela Welcome, dan beri nama proyek dengan nama yang sama yang ingin Anda gunakan untuk aplikasi.

Saat memilih Domain Perusahaan unik, ingat bahwa aplikasi yang dipublikasikan ke Google Play

harus memiliki nama paket unik. Karena domain bersifat unik, yang mengawali nama aplikasi dengan nama Anda, atau nama domain perusahaan, sebaiknya sediakan nama paket unik yang memadai. Jika tidak berencana untuk mempublikasikan aplikasi, Anda bisa menerima domain contoh default. Ketahuilah bahwa mengubah nama paket di lain waktu memerlukan kerja tambahan.

Memilih perangkat target dan SDK minimum

Saat memilih Target Android Devices, Phone and Tablet dipilih secara default, seperti yang ditampilkan dalam gambar dibawah ini. Pilihan yang ditampilkan dalam gambar untuk Minimum SDK — API 15: Android 4.0.3 (Ice Cream Sandwich) — membuat aplikasi Anda kompatibel dengan 97% perangkat Android yang aktif di Google Play Store.



Perangkat berbeda menjalankan versi sistem Android yang berbeda, seperti Android 4.0.3 atau Android 4.4. Setiap versi yang berurutan umumnya menambahkan API baru yang tidak tersedia di versi sebelumnya. Untuk menunjukkan rangkaian API yang tersedia, setiap versi menetapkan API level. Misalnya, Android 1.0 adalah API level 1 dan Android 4.0.3 adalah API level 15.

Minimum SDK mendeklarasikan versi Android minimum untuk aplikasi Anda. Setiap versi Android yang berurutan menyediakan kompatibilitas untuk aplikasi yang dibangun menggunakan API dari versi sebelumnya, sehingga aplikasi Anda akan selalu

kompatibel dengan versi Android mendatang sambil menggunakan Android API yang didokumentasikan.

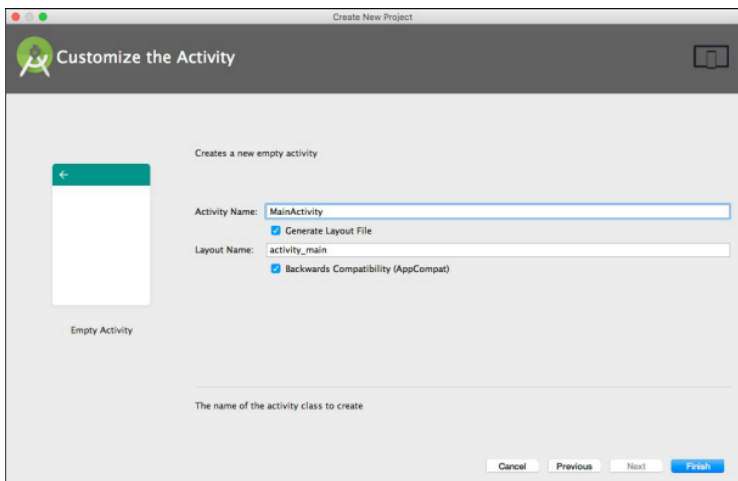
Memilih template

Android Studio mengumpulkan terlebih dulu proyek Anda dengan kode minimum untuk aktivitas dan layout layar berdasarkan template. Tersedia berbagai macam template, mulai dari template kosong virtual (Add No Activity) hingga beragam tipe aktivitas. Anda bisa menyesuaikan aktivitas setelah memilih template. Misalnya, template Empty Activity menyediakan satu aktivitas yang disertai satu sumber daya layout untuk layar. Anda bisa memilih untuk menerima nama yang biasa digunakan untuk aktivitas (seperti Main Activity) atau mengubah nama di layar Customize Activity. Selain itu, jika Anda menggunakan template Empty Activity, pastikan memeriksa yang berikut ini jika belum diperiksa:

- **Generate Layout File:** Biarkan ini dicentang untuk membuat sumber daya layout yang terhubung dengan aktivitas ini, yang biasanya diberi nama

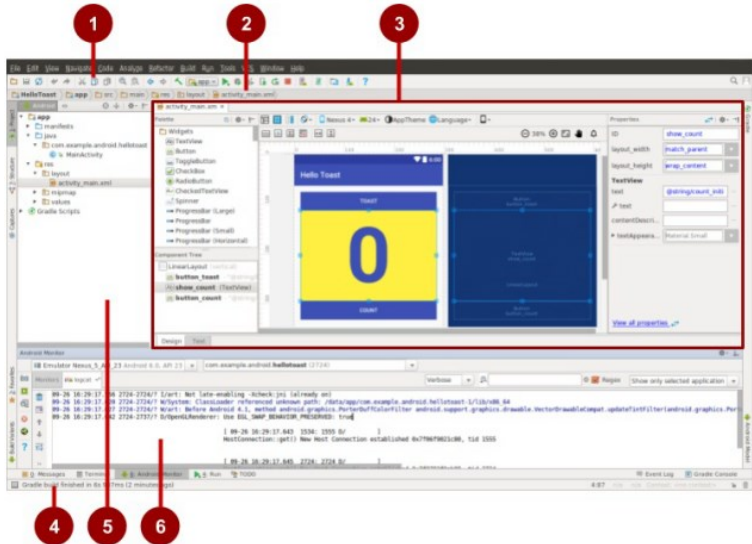
activity_main.xml. Layout mendefinisikan antarmuka pengguna untuk aktivitas.

- Backwards Compatibility (AppCompat): Biarkan ini dicentang untuk menyertakan pustaka AppCompat sehingga aplikasi kompatibel dengan Android versi sebelumnya bahkan jika menggunakan fitur yang hanya ditemukan di versi yang lebih baru.



Android Studio membuat folder untuk proyek yang baru saja dibuat di folder Android Studio Projects pada komputer Anda.

Panel jendela Android Studio. Jendela utama Android Studio terdiri dari sejumlah area logis, atau panel, seperti yang ditampilkan dalam gambar di bawah ini.



Dalam gambar di atas:

1. Bilah Alat. Bilah alat menjalankan beragam tindakan, termasuk menjalankan aplikasi Android dan meluncurkan alat Android.
2. Bilah Navigasi. Bilah navigasi memungkinkan navigasi melalui proyek dan membuka file untuk

pengeditan. Bilah navigasi menyediakan tampilan struktur proyek yang lebih ringkas.

3. Panel Editor. Panel ini menampilkan materi file yang dipilih dalam proyek. Misalnya, setelah memilih layout (seperti yang ditampilkan dalam gambar), panel ini menampilkan editor layout dengan alat untuk mengedit layout. Setelah memilih file kode Java, panel ini menampilkan kode dengan alat untuk mengedit kode.

4. Bilah Status. Bilah status menampilkan status proyek dan Android Studio itu sendiri, serta peringatan atau pesan apapun. Anda bisa mengamati kemajuan pembangunan di bilah status.

5. Panel Project. Panel proyek menampilkan file proyek dan hierarki proyek.

6. Panel Monitor. Panel monitor menawarkan akses ke daftar TODO untuk mengelola tugas, Android Monitor untuk memantau eksekusi aplikasi (ditampilkan dalam gambar), logcat untuk menampilkan pesan log, dan aplikasi Terminal untuk melakukan aktivitas Terminal.

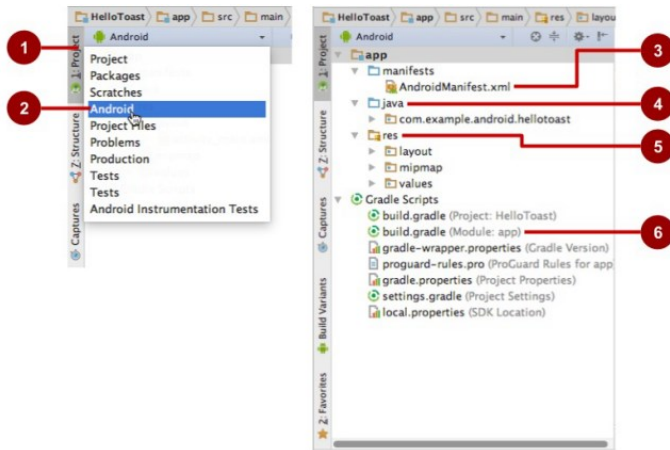
Menjelajahi proyek

Setiap proyek di Android Studio berisi file Android Manifest.xml, file kode sumber komponen, dan file sumber daya terkait.

Secara default, Android Studio mengatur file proyek Anda berdasarkan jenis file, dan menampilkannya dalam proyek:

Tampilan Android di panel alat (bantu) kiri, seperti yang ditampilkan di bawah ini. Tampilan menyediakan akses cepat ke file kunci proyek Anda.

Untuk beralih kembali ke tampilan ini dari tampilan lainnya, klik tab Project vertical di kolom paling kiri panel Project, dan pilih Android dari menu munculan di bagian atas panel Project, seperti yang ditampilkan dalam gambar di bawah ini.



Dalam gambar di atas:

1. Tab Project. Klik untuk menampilkan tampilan proyek.
2. Pilihan Android di menu tarik-turun proyek.
3. File Android Manifest.xml. Digunakan untuk menetapkan informasi tentang aplikasi bagi lingkungan waktu proses Android. Template yang Anda pilih membuat file ini.
4. Folder java. Folder ini menyertakan aktivitas, pengujian, dan komponen lain di kode sumber Java. Setiap aktivitas, layanan, dan komponen lain didefinisikan sebagai kelas Java, biasanya dalam file miliknya. Nama aktivitas pertama (layar) yang

pengguna lihat, yang juga melakukan inisialisasi sumber daya seluruh aplikasi, biasanya adalah Main Activity.

5. Folder res. Folder ini menyimpan sumber daya, seperti layout XML, string UI, dan gambar. Aktivitas biasanya dikaitkan dengan file sumber daya XML yang menetapkan layout tampilannya. File ini biasanya diberi nama setelah aktivitas atau fungsinya.
6. File `**build.gradle` (Module: App
7. `)*`. File ini menetapkan konfigurasi pembangunan modul. Template yang Anda pilih membuat file ini, yang mendefinisikan konfigurasi pembangunan, termasuk atribut `min Sdk Version` yang mendeklarasikan versi minimum untuk aplikasi, dan atribut `target SdkVersion` yang mendeklarasikan versi tertinggi (terbaru) untuk aplikasi yang dioptimalkan. File ini juga menyertakan daftar dependensi*, yaitu pustaka yang diperlukan oleh kode – seperti pustaka App Compat untuk mendukung beragam versi Android.

BAB III

PERANCANGAN DAN

DESAIN APLIKASI

ANDROID

BAB III

PERANCANGAN DAN DESAIN APLIKASI ANDROID

A. Perancangan Proses Dalam Pembuatan Aplikasi

1. Definisi Perancangan

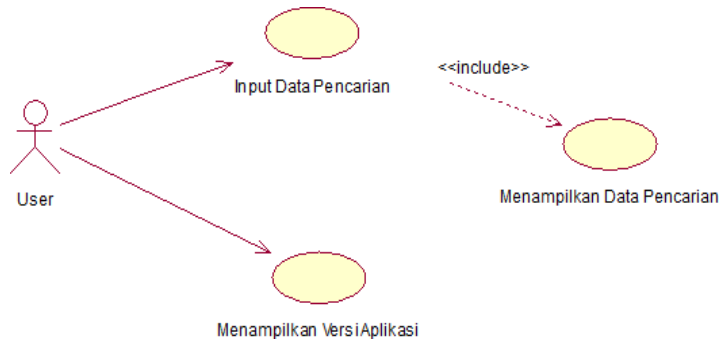
Menurut Ladjamudin (2013:39) menjelaskan bahwa Tahapan Perancangan (desain) memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik, kegiatan yang dilakukan dalam tahap perancangan ini meliputi perancangan output, input, dan file. Berdasarkan definisi di atas, penulis menarik kesimpulan bahwa perancangan merupakan suatu pola yang dibuat untuk mengatasi masalah yang dihadapi perusahaan atau organisasi setelah melakukan analisis terlebih dahulu.

2. Perancangan Aplikasi

Perancangan akan dibagi menjadi dua bagian yaitu terdiri dari perancangan proses dan perancangan interface. Berikut dibawah ini merupakan penjelasan dan contoh dari bagian perancangan proses pada perancangan aplikasi pencarian musik dengan menggunakan algoritma genetika.

a. Use Case Diagram

Langkah pertama yang dilakukan untuk membuat kebutuhan pengguna adalah dengan cara memodelkan sistem, hal tersebut dapat digambarkan dengan menggunakan sebuah use case diagram. Use case pada aplikasi ini dapat dilihat pada gambar dibawah ini.



Gambar 3.1 Use Case Sistem Aplikasi

b. Use Case Description

Tabel 3.1 Use Case Sistem Aplikasi

Use Case	Keterangan
Description	<i>Use case</i> menjalankan aplikasi
Actor	<i>User</i>
Pra Condition	Aplikasi belum berjalan
Action	<ol style="list-style-type: none"> 1. <i>User</i> menginstal aplikasi pencarian pada <i>smartphone</i> android 2. <i>User</i> menjalankan aplikasi pencarian
Pasca	Jika aplikasi ini telah berjalan, maka

Use Case	Keterangan
Condition	<i>user</i> dapat langsung menjalankan aplikasi pencarian data

Tabel 3.2 Skenario Use Case Input Data pencarian

Use Case	Keterangan
Description	<i>Use case</i> input data dari user
Actor	<i>User</i>
Pra Condition	Aplikasi sudah berjalan
Action	<ol style="list-style-type: none"> 1. <i>User</i> menjalankan aplikasi pencarian data 2. <i>User</i> melakukan input data pada kolom pencarian 3. Sistem mengecek apakah ada proses data yang diinput
Pasca Condition	Jika <i>use case</i> telah berhasil melakukan pencarian, maka <i>user</i> dapat melihat data yang telah dicari

Tabel 3.3 Skenario Use Case Menampilkan Data Pencarian

Use Case	Keterangan
Description	<i>Use case</i> menampilkan hasil pencarian
Actor	<i>User</i>
Pra Condition	Aplikasi sudah berjalan
Action	<ol style="list-style-type: none"> 1. <i>User</i> menunggu proses pencarian data 2. Sistem melakukan pencarian data baik di internal maupun eksternal memory 3. <i>User</i> mendapatkan hasil pencarian data yang diinginkan 4. Aplikasi berhasil melakukan pencarian data pada internal maupun eksternal memory
Pasca Condition	Jika <i>use case</i> telah berhasil melakukan pencarian, maka <i>user</i> dapat melihat data yang diinginkan

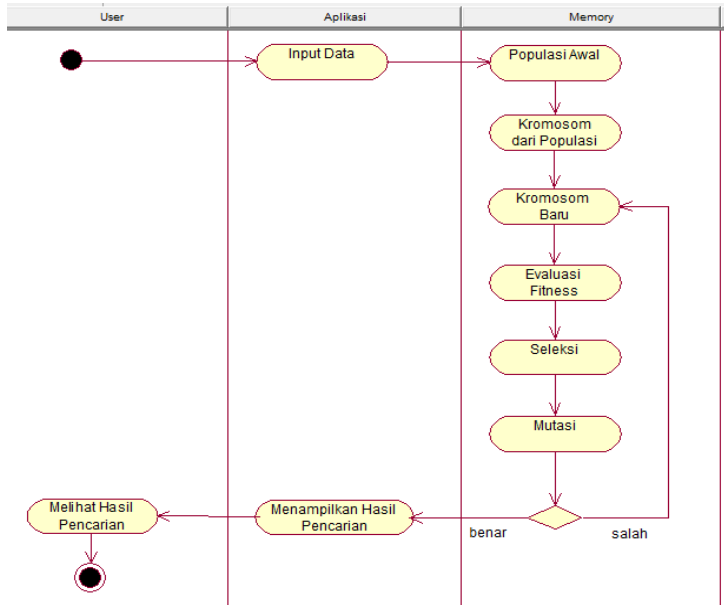
Tabel 3.4 Skenario Use Case Menampilkan Versi Aplikasi

Use Case	Keterangan
Description	<i>Use case</i> menampilkan versi aplikasi
Actor	<i>User</i>
Pra Condition	Aplikasi sudah berjalan
Action	<ol style="list-style-type: none"> 1. <i>User</i> mendapatkan versi aplikasi dari sistem 2. Sistem mengecek apakah ada proses data yang diinput 3. Aplikasi berhasil menampilkan versi aplikasi
Pasca Condition	Jika <i>use case</i> telah berhasil melakukan pencarian, maka <i>user</i> dapat melihat versi aplikasi

c. Activity Diagram Pembentukan Populasi

Activity Diagram merupakan Salah satu aktifitas yang dapat dilakukan oleh user. Activity

diagram digunakan untuk memodelkan aspek dinamis dari sistem. Activity diagram secara esensial mirip dengan diagram alir (flowchart).



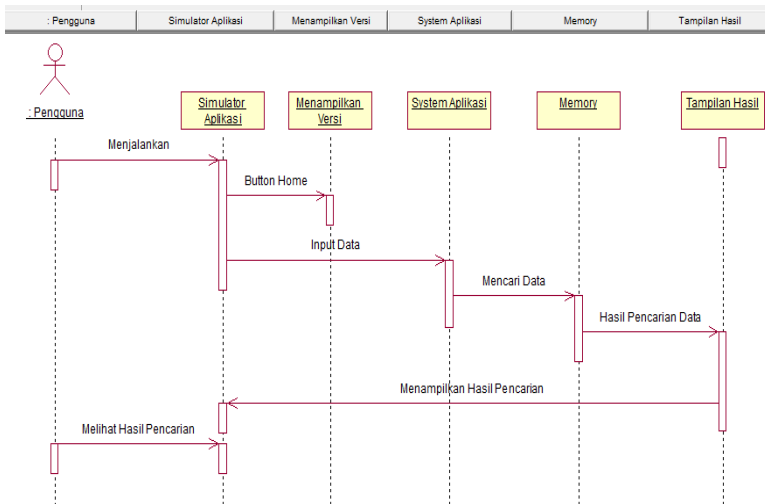
Gambar 3.2 Activity Diagram Pencarian Dokumen Musik

Activity diagram merupakan gambaran alur proses algoritma genetik. Proses tersebut dimulai dari data input. Pertama dengan melakukan inisialisasi kromosom terhadap tiap data input,

sehingga didapat nilai evaluasi fitness tiap-tiap kromosom. Dari nilai fitness yang diperoleh selanjutnya di seleksi dan mutasi. Jika hasil belum optimal maka dilakukan lagi evaluasi kromosom dan jika hasilnya telah memenuhi syarat yang menghasilkan generasi baru.

d. Sequence Diagram

Sequence diagram menjelaskan interaksi object yang disusun dalam suatu burutan waktu. Diagram ini secara khusus berasosiasi dengan use case. Sequence diagram memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case.



Gambar 3.3 Sequence Diagram Aplikasi Pencarian Dokumen Musik

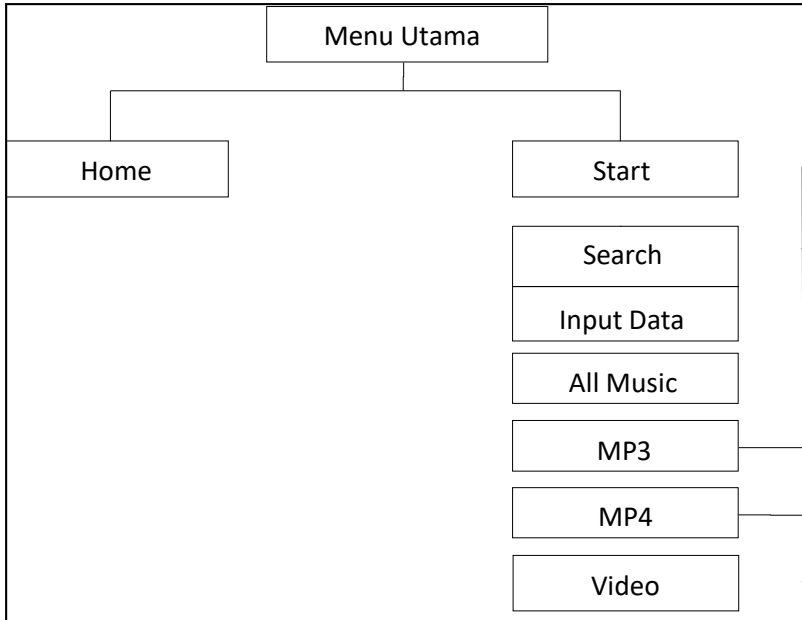
Pada gambar diatas menggambarkan sequence diagram dalam menampilkan hasil pencarian, yang sebelumnya telah dilakukan dengan menjalankan aplikasi, input data dan kemudian mendapatkan hasil pencarian. Sequence diagram tersebut menggambarkan tahapan yang terjadi pada proses algoritma genetik. Setelah dilakukan proses genetik dipilih individu dan nilai fitness secara acak. Dari proses

tersebut nantinya menghasilkan tampilan dari pencarian data dari data hasil inputan.

e. Struktur Navigasi

Struktur navigasi infomusik yang terdapat pada gambar 3.4 menggunakan struktur navigasi yang merupakan gabungan dari ke tiga struktur sebelumnya yaitu satu arah, tidak berurut dan bercabang. Dengan menggunakan struktur navigasi campuran memungkinkan pengguna untuk bisa mengakses halaman lain yang terdapat pada menu tanpa harus kembali ke halaman utama. Pada struktur navigasi ini ketika mulai menjalankan program ini, terlebih dahulu akan masuk ke menu home dan start. Jika memilih menu home maka akan muncul penjelasan mengenai aplikasi infomusik dan versi aplikasi infomusik. Sebelum menggunakan mesin pencarian dibutuhkan proses untuk memilih menu start dengan memilih menu tersebut, maka akan muncul submenu yaitu All music, MP3, MP4 dan Video.

Dibawah ini merupakan struktur dari menu infomusik.



Gambar 3.4 Struktur Menu Infomusik

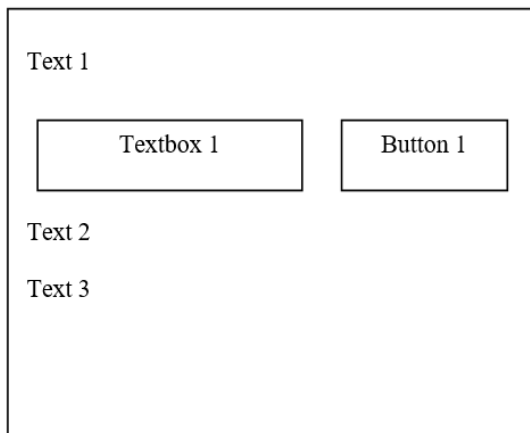
e.1 Rancangan Interface

Pada tahap ini akan dimulai perancangan aplikasi, Rancangan aplikasi dapat dikelompokkan menjadi tiga bagian yaitu rancangan tampilan menu home, rancangan tampilan menu start dan rancangan

tampilan menu hasil. Rancangan tampilan menu terdiri dari dua buah menu, diantaranya adalah tampilan home dan tampilan start.

e.2 Rancangan Tampilan Menu Home

Pada saat program pertama kali dijalankan, program akan masuk ke tampilan menu utama. Menu utama ini berisi menu home dan start. Jika masuk ke aplikasi ini secara otomatis akan langsung menampilkan menu kedua menu tersebut. Di bawah ini adalah contoh tampilan menu home.



Gambar 3.5. Rancangan Tampilan Menu Home

Keterangan:

Text 1 : 'Logo' menampilkan logo aplikasi

Text 2 : 'Versi aplikasi' menampilkan versi dari aplikasi

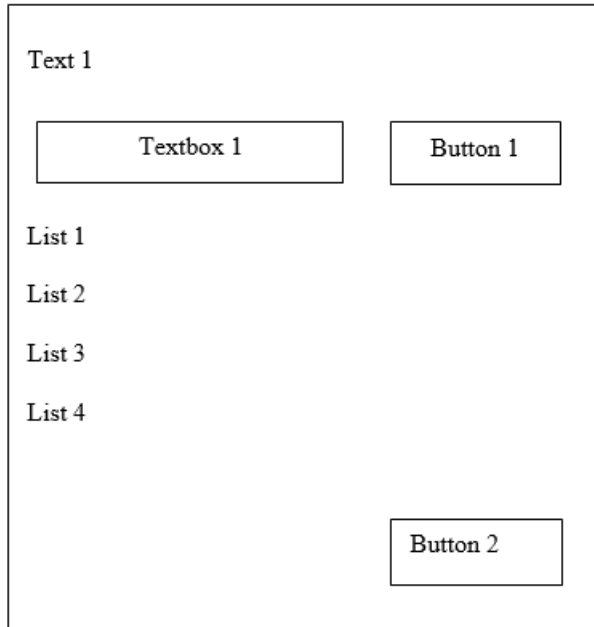
Text 3 : 'Penjelasan aplikasi' menampilkan penjelasan tentang aplikasi

Textbox 1: 'Input data' untuk menginput data pencarian

Button 1 : 'Start' merupakan proses untuk memulai system pencarian

e.3 Rancangan Tampilan Menu Start

Setelah menu home selanjutnya menampilkan menu start. Di dalam menu ini terdapat submenu yaitu All Music, MP3, MP4 dan Video. Sebelum memilih submenu tersebut kita bisa menginput data yang ini dicari pada menu input data. Di bawah ini merupakan contoh dari tampilan menu start.



Gambar 3.6 Rancangan Tampilan Menu Start

Keterangan:

List 1 : 'All Music' menampilkan semua file musik

List 2 : 'MP3' menampilkan file musik MP3

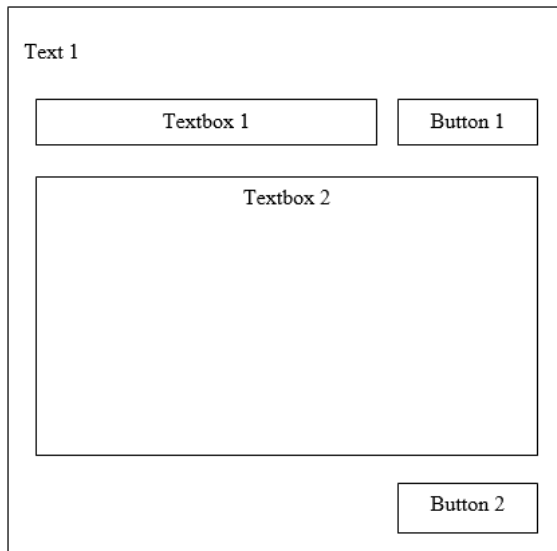
List 3 : 'MP4' menampilkan file musik MP4

List 4 : 'Video' menampilkan file musik video

Button 2 : 'Back' merupakan tombol kembali ke menu sebelumnya

e.4 Rancangan Tampilan Menu Hasil

Setelah menu start dilakukan maka akan dapat diketahui hasil dari pencarian tersebut. Berikut ini adalah tampilan dari menu hasil.



Gambar 3.7 Rancangan Tampilan Menu Hasil

Keterangan:

Textbox 2 : Hasil pencarian berdasarkan inputan data

f. Pengumpulan Bahan

Pada aplikasi ini dibutuhkan beberapa bahan pendukung yang berfungsi untuk dijadikan sebuah alat yang akan memperindah dan menguatkan aplikasi ini, bahan tersebut diantaranya gambar dan teori untuk penggunaan aplikasi Android Developer Tools sebagai bahasa pemrograman pembuatan aplikasi ini.

f.1 Gambar

Tujuan penulis menambahkan gambar pada bahan pendukung pembuatan aplikasi ini yaitu agar maksud yang disampaikan oleh penulis dapat dipahami dan dengan adanya gambar aplikasi tersebut dapat dimengerti dengan baik.

f.2 Teori

Penulis akan menjabarkan beberapa teori yang terkait dengan aplikasi ini, teori tersebut adalah sebagai berikut:

f.2.1 Proses Pembentukan Populasi Awal

Proses pembentukan populasi awal adalah proses pertama yang dijalankan dengan cara melakukan penginputan karakter pada kolom txtSearch oleh user. Setelah itu karakter yang telah diinput akan dibaca oleh sistem agar dapat diproses dalam algoritma. Semua lagu akan di simpan pada sdcard, hal ini dilakukan agar mudah digunakan dalam poses pencarian data tersebut. Untuk mencari sebuah data lagu cukup menginput judul atau penyanyi yang ingin dicari maka dengan itu sistem akan mencari data yang diinginkan. Pada setiap list lagu akan memiliki banyaknya pilihan pencarian yang berbeda. Bisa jadi ada list lagu yang memiliki tiga pilihan lagu dari segi penyanyi dan judul lagunya. Dengan adanya list lagu yang berisikan banyaknya tingkat pilihan pencarian dari setiap list lagu yang ada serta telah diurutkan dari list lagu yang paling sedikit pilihan pencariannya hingga list lagu yang terbanyak pilihan pencariannya.

Dari proses ini diharapkan tidak ada list lagu yang tidak dapat teralokasikan pencariannya dikarenakan pada pencarian yang memungkinkan bagi list lagu tersebut telah digunakan oleh list lagu lainnya. list lagu disini terdiri dari : All Music, MP3, MP4, dan Video.

1. Kolom input data

```
txtSearch = (EditText) findViewById(R.id.txtSearch)
```

2. Membaca data karakter inputan pada proses

```
pencarian File f = new File(memory);
```

```
(f.exists() && f.canRead())
```

f.2.2 Proses Pembuatan Kromosom dari Populasi

Berdasarkan urutan dari list lagu, setiap lagu akan dijadwalkan ke dalam list lagu secara acak. Agar diketahui apakah pada waktu tersebut user dapat menemukan hasil dari pencarian yang diinginkan, maka list lagu tersebut harus dipetakan terlebih dahulu, algoritma ini akan mengecek terlebih dahulu kepada list lagu untuk mengetahui data lagu yang di cari adalah lagu yang dimaksud,

jika bukan algoritma ini akan terus mencari sampai titik akhir.

Proses pembuatan kromosom dari populasi merupakan suatu proses membangkitkan sejumlah individu yang dilakukan secara acak atau random dari populasi melalui prosedur tertentu, sehingga sebuah siklus operasional genetik akan berhasil apabila kromosom yang disebut parent (orang tua) dipilih melalui tahapan penyeleksian yang spesifik. Syarat-syarat yang harus dipenuhi untuk menunjukkan suatu solusi harus benar-benar diperhatikan dalam pembangkitan setiap individunya.

Dalam aplikasi ini proses pembuatan kromosom yaitu sama dengan proses pembentukan penyimpanan data baik di internal memory atau di sdcard. Penyimpanan data akan tersimpan secara acak. Prosedur yang dilakukan pada waktu proses pembentukan kromosom dari populasi didalam aplikasi yaitu sebagai berikut :

1. Perubahan karakter inputan

```
if(nama.toLowerCase().
```

2. Menentukan bentuk dari karakter dari inputan pada proses pencarian `IndexOf(".") != -1`)

3. Penyimpanan perubahan karakter

```
vPopulate.addElement(memory+nama);
```

```
vUrl.addElement(memory + nama);
```

```
vHasil.addElement(nama);
```

Pada langkah diatas, pertama kali per-karakter akan dilakukan pemilihan dan pembentukan karakter yang baru diubah menjadi LowerCase (huruf kecil). Kemudian hasil identifikasi karakter dikelompokkan pada nama folder atau nama file. Selanjutnya setiap perubahan per-karakter akan disimpan pada vPopulate, untuk data yang sudah spesifik akan diinput pada vUrl dan untuk data yang belum spesifik akan diinput pada vHasil sebagai hasil dari populasi tersebut.

f.2.3 Proses Evaluasi Fitness

Sebelum algoritma ini dijalankan masalah apa yang ingin dioptimalkan itu harus dinyatakan dalam sebuah fungsi tujuan, yang dikenal dengan fungsi fitness. Jika nilai fitness semakin besar, maka sistem yang dihasilkan semakin baik. Walaupun pada awalnya semua nilai fitness kemungkinan sangat kecil (karena algoritma ini menghasilkannya secara random), sebagian akan lebih tinggi dari yang lain. Kromosom dengan nilai fitness yang tinggi ini akan memberikan probabilitas yang tinggi untuk bereproduksi pada generasi selanjutnya. Sehingga untuk setiap generasi pada proses evolusi fungsi fitness yang mensimulasikan seleksi alam akan menekan populasi kearah fitness yang meningkat.

Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai kepatasan setiap kromosom hingga kriteria berhenti terpenuhi. Bila kriteria berhenti tetapi belum terpenuhi, maka akan dibentuk lagi generasi baru. Beberapa kriteria

berhenti yang umum digunakan adalah, sebagai berikut:

- berhenti pada generasi tertentu,
- berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai fitness,
- tertinggi/terendah (tergantung persoalan) tidak berubah,
- berhenti bila dalam n generasi berikutnya tidak diperoleh nilai fitness yang lebih tinggi/rendah.

Generasi yang memiliki nilai kepastian terbaik diharapkan merupakan solusi optimal yang diinginkan. Untuk melihat prosedur dari proses evaluasi fitness didalam aplikasi ini yaitu sebagai berikut:

1. Melakukan inisialisasi kromosom dari populasi di `vPopulateCopy`

```
if (nama.endsWith("."))
```

langkah pertama dilakukan adalah menginisialisasi pada setiap kromosom yang berada di `vPopulateCopy`, selanjutnya inisialisasi dilakukan

kembali untuk memastikan bila ada perubahan pada karakter inputan.

f.2.4 Proses Seleksi

Proses seleksi dilakukan dengan cara membuat kromosom yang mempunyai fungsi objektif terkecil/terbesar mempunyai kemungkinan terpilih yang besar atau mempunyai nilai probabilitas yang tinggi.

Untuk mendapatkan solusi yang terbaik, maka program harus menyeleksi solusi yang memiliki nilai fitness yang tergolong rendah. Seleksi menggunakan metode good fitness yaitu setengah dari jumlah populasi yang memiliki harga fitness yang terendah akan dihilangkan sehingga akan hanya selalu tersisa sekelompok solusi yang terbaik yang pernah diperoleh oleh program. Solusi yang tersisa hasil seleksi ini dikenal dengan nama populasi induk. Agar jumlah populasi tetap, maka perlu dibangkitkan solusi

baru sebanyak setengah dari jumlah populasi yang ada.

Dalam program ini, cara yang digunakan untuk membangkitkan solusi baru menggunakan dua cara yaitu reproduksi kromosom baru dan cara mutasi dari solusi induk. Tujuan pembangkitan solusi baru ini untuk menemukan alternatif solusi yang lebih baik dari solusi - solusi yang sudah diperoleh.

Dibawah ini merupakan prosedur dari proses seleksi dalam aplikasi:

1. Inisialisasi bentuk kromosom dari populasi pada `vPopulateCopy`
`if(nama.toLowerCase().indexOf(txtSearch.getText().toString().toLowerCase().trim()) != -1)`
2. Penyimpanan hasil inisialisasi berdasarkan dari bentuk
`vUrl.addElement(memory + nama);`
`vHasil.addElement(nama);`
langkah pertama dilakukan inisialisasi pada bentuk sebuah kromosom, apakah itu bentuk

sebuah folder ataukah nama file musik. Kemudian setelah diketahui bentuk kromosom tersebut dilakukannya penyimpanan kromosom, jika kromosom itu berbentuk file maka akan tersimpan di vUrl sedangkan jika kromosom tersebut berbentuk nama folder akan tersimpan di vHasil.

f.2.5 Proses Reproduksi Kromosom Baru

Setengah dari jumlah populasi baru akan dibangkitkan dengan cara reproduksi kromosom baru, yaitu penyusunan alternatif solusi penjadwalan secara acak kembali untuk setiap lagu. Dengan proses ini maka akan dihasilkan sekelompok populasi baru yang benar-benar berbeda dengan populasi induknya.

Prosedur yang dilakukan pada waktu proses reproduksi kromosom baru didalam aplikasi ini yaitu sebagai berikut:

1. Melakukan kembali inisialisasi pada kromosom karakter


```
for(int i=0;i<vPopulate.size();i++)
```

```
f = new  
File(vPopulate.elementAt(i).toString());
```

2. Melakukan penyimpanan sementara pada kromosom karakter baru

```
if(nama.toLowerCase().indexOf(".") != -1)  
vPopulateCopy.addElement(vPopulate.ele  
mentAt(i).toString()+"/"+nama);
```

Pertama hal yang dilakukan adalah melakukan inisialisasi kembali pada setiap kromosom karakter, dengan melakukan pemilahan dan pembentukan kromosom baru dari kromosom yang sudah ada sebagai hasil disimpan pada vPopulate. Setelah terbentuknya kromosom baru kemudian dilakukan penyimpanan sementara untuk semua element yang berada di vPopulate kedalam vPopulateCopy.

f.2.6 Proses Mutasi

Adapun setengah populasi baru lainnya akan dibangkitkan dengan cara mutasi. Proses mutasi merupakan proses dimana setengah dari populasi induk akan dipilih untuk diduplikasi. Pemilihan dapat dilakukan dengan metode good fitness, random maupun roulette wheel. Pada hasil duplikasi ini akan dilakukan sedikit percobaan terhadap posisi pencarian sebuah lagu. Proses mutasi ini adalah suatu proses eksploitasi terhadap kemungkinan - kemungkinan modifikasi pada list lagu yang telah ada.

Perubahan posisi beberapa lagu tersebut (mutasi) dapat membuat solusi duplikasi ini menjadi memiliki nilai fitness yang lebih rendah maupun lebih tinggi daripada solusi induknya. Jika ternyata diperoleh solusi yang memiliki fitness yang lebih tinggi maka hal itulah yang diharapkan. Tetapi jika diperoleh solusi dengan nilai fitness yang lebih rendah maka bisa jadi pada iterasi berikutnya diperoleh solusi hasil mutasi yang lebih baik nilai fitnessnya daripada solusi induknya.

Maksudnya adalah tidak menjadi masalah jika solusi hasil mutasi ini mengalami penurunan nilai fitness daripada solusi induknya, tetaplah memberikan perhatian pada solusi terbaik yang telah dicapai. Pada proses mutasi ini, program akan memilih beberapa format list lagu secara acak untuk dikeluarkan dari solusi hasil penggandaan tersebut untuk dijadwalkan kembali pada posisi yang berbeda.

Sebelum list lagu yang dikeluarkan tersebut dijadwalkan kembali, maka program harus memproses data input terlebih dahulu dari beberapa format list lagu tersebut. Proses ini sama dengan proses pada langkah kedua dari algoritma pemrograman ini. Hal ini dilakukan untuk menghindari list lagu yang memiliki pilihan pencarian yang sedikit menjadi tidak bisa diposisikan kembali dalam pencarian karena pilihan waktu pencariannya telah terisi oleh list lagu yang dikeluarkan lainnya.

Mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berfungsi untuk menggantikan gen yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal. Sehingga mutasi akan meningkatkan variasi populasi. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh parameter laju mutasi. Proses mutasi dilakukan dengan cara mengganti satu gen yang terpilih secara acak dengan suatu nilai baru yang didapat secara acak.

Dibawah ini merupakan prosedur dari proses mutasi didalam aplikasi:

1. Melakukan identifikasi besaran data pada
vPopulateCopy
if(vPopulateCopy.size()>0)
for(int i=0;i<vPopulateCopy.size();i++)
2. Melakukan duplikasi data dari vPopulateCopy
kedalam
vPopulate

```
vPopulate.addElement(vPopulateCopy.elementAt(i).toString());
```

3. Melakukan penghapusan data di vPopulateCopy

```
vPopulateCopy.removeAllElements();
```

Proses yang dilakukan pertama kali yaitu melakukan pengecekan jumlah besaran data yang terdapat pada vPopulateCopy, jika data yang terdapat pada vPopulateCopy lebih besar dari "0" maka akan masuk pada tahap kedua yaitu melakukan duplikasi elemen data dari vPopulateCopy kedalam vPopulate. Setelah semua elemen data vPopulateCopy masuk kedalam vPopulate, terakhir dilakukan penghapusan data yang terdapat didalam vPopulateCopy

f.2.7 Proses Cek Kondisi

Prosedur yang dilakukan pada proses cek kondisi didalam aplikasi ini yaitu sebagai berikut:

1. Melakukan identifikasi besaran data yang terdapat pada vPopulate

```
if(vPopulateCopy.size())>0)
```

2. Melakukan pemberhentian pencarian break;

Pada tahap ini yang pertama dilakukan adalah mengecek jumlah data yang terdapat pada vPopulate, jika jumlah data lebih besar atau sama dengan "0" maka proses pencarian akan masuk ke tahap selanjutnya yaitu break; atau berhenti kemudian data tersebut akan masuk ke vUrl, tetapi jika data yang berada pada vPopulate tidak memenuhi syarat maka akan kembali ke proses evaluasi fitness untuk dilakukan identifikasi ulang.

f.2.8 Proses Kondisi Selesai

Terdapat tiga kondisi selesai yang dapat menghentikan proses algoritma pemrograman ini, yaitu:

- 1) Jika setelah beberapa generasi berturut-turut nilai fitness terbaik dari populasi tidak mengalami perubahan kembali

2) Jika jumlah generasi atau iterasi maksimum telah tercapai.

3) Jika nilai fitness terbaik minimal telah tercapai. Jika salah satu kondisi di atas telah diperoleh maka iterasi akan dihentikan, dan jika salah satu kondisi selesai ini belum tercapai maka program akan mengulang kembali proses ini / iterasi dari langkah keempat yaitu evaluasi fitness terhadap populasi baru tadi.

B. Desain Wireframe Aplikasi Android

1. Pengertian Wireframe

Mockup atau wireframe adalah rancangan atau gambaran rangka awal suatu desain web atau aplikasi android yang dibuat dengan cara manual menggunakan software edit gambar seperti software pencil evolus atau software online pengedit gambar lainnya. Mockup atau wireframe juga adalah sebuah gambaran model dan prototipe pada halaman web atau aplikasi android secara detail. Wireframe

digunakan agar mempermudah konsep dalam perancangan dan membangun sebuah aplikasi.

Sebuah wireframe adalah struktur dasar dari sebuah website/aplikasi android. Dalam tahap pengembangan, wireframe dapat menetapkan fungsi (sliders, tabs, dan lainnya) tanpa warna atau desain apapun, hanya structural guidelinesnya saja. Wireframing merupakan tahap yang penting dalam proses perancangan. Wireframe memungkinkan untuk mendefinisikan information hierarchy dari desain, sehingga memudahkan dalam perencanaan layout sesuai dengan yang direncanakan tentang bagaimana user dapat memproses informasi. Wireframe dapat membantu untuk mengorganisir dan menyederhanakan elemen dan isi di dalam website/aplikasi android.



Gambar 3.8 Wireframe Aplikasi Android

2. Manfaat Desain Wireframe

Kegunaan wireframe dapat memberikan informasi dan penjelasan beberapa hal, di antaranya:

- Apa isi yang ada dalam halaman website/aplikasi
- Bagaimana isi tersebut terorganisir dalam halaman website/aplikasi
- Isi mana yang paling penting dalam halaman website/aplikasi
- Kemana user akan pergi dari halaman website/aplikasi

- Dimana halaman ini dalam website/aplikasi
- Bagaimana user akan bergerak di sekitar website/aplikasi

Keuntungan dalam membuat wireframe adalah:

- Memberikan tampilan awal dari website/aplikasi (atau desain ulang)
- Menginspirasi desainer
- Memberikan gambaran jelas dari elemen tentang bagaimana developer akan melakukan coding
- Memudahkan pengadaptasian dan memperlihatkan layout dari banyak bagian dari website/aplikasi

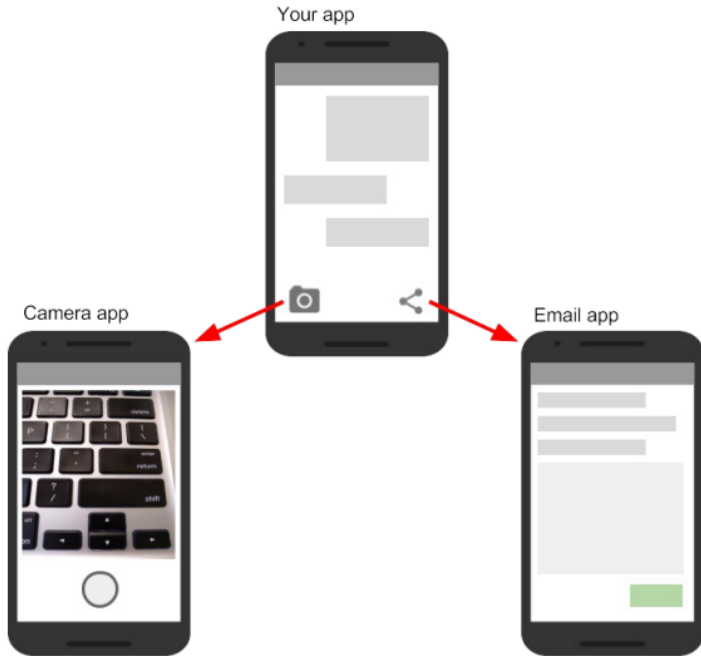
C. Aktivitas dan Maksud pada Android Studio

1. Tentang aktivitas

Aktivitas menyatakan layar tunggal di aplikasi Anda dengan antarmuka yang bisa digunakan pengguna untuk berinteraksi. Misalnya, aplikasi email mungkin memiliki satu aktivitas yang menampilkan

daftar email baru, aktivitas lain untuk menulis email, dan aktivitas lainnya lagi untuk membaca pesan satu persatu. Aplikasi Anda adalah koleksi aktivitas yang dibuat sendiri atau yang digunakan kembali dari aplikasi lain.

Meskipun aktivitas di aplikasi Anda bekerja sama membentuk pengalaman pengguna yang kohesif di aplikasi, setiap aplikasi tidak saling bergantung. Proses ini memungkinkan aplikasi memulai aktivitas di aplikasi lainnya, dan aplikasi lainnya bisa memulai aktivitas Anda (jika aplikasi mengizinkannya). Misalnya, aplikasi perpesanan yang Anda tulis bisa memulai aktivitas di aplikasi kamera untuk mengambil gambar, kemudian memulai aktivitas di aplikasi email untuk memungkinkan pengguna berbagi gambar itu di email.



Gambar 3.9 Contoh Aktivitas Aplikasi Pemesanan

Umumnya, satu aktivitas di aplikasi ditetapkan sebagai aktivitas "utama", yang disajikan kepada pengguna saat membuka aplikasi untuk pertama kali. Kemudian setiap aktivitas bisa memulai aktivitas lainnya untuk melakukan tindakan yang berbeda.

Setiap kali aktivitas baru dimulai, aktivitas sebelumnya akan dihentikan, namun sistem

mempertahankan aktivitas dalam tumpukan ("back-stack"). Bila pengguna selesai dengan aktivitas saat ini dan menekan tombol Kembali, aktivitas akan muncul dari tumpukan (dan dimusnahkan) dan aktivitas sebelumnya dilanjutkan.

Bila aktivitas dihentikan karena aktivitas baru dimulai, aktivitas pertama akan diberi tahu tentang perubahan tersebut dengan metode callback daur hidup aktivitas. Daur hidup Aktivitas adalah serangkaian keadaan aktivitas, mulai dari pertama kali dibuat, hingga setiap kali dihentikan atau dilanjutkan, hingga bila sistem memusnahkannya.

Membuat aktivitas

Untuk mengimplementasikan aktivitas di aplikasi Anda, lakukan yang berikut ini:

- Buat kelas Java aktivitas.
- Implementasikan antarmuka pengguna untuk aktivitas itu.

- Deklarasikan aktivitas baru itu di manifest aplikasi.

Bila Anda membuat proyek baru untuk aplikasi atau menambahkan aktivitas baru ke aplikasi, di Android Studio (dengan File > New > Activity), kode template untuk setiap tugas ini akan disediakan untuk Anda.

Buat kelas aktivitas

Aktivitas adalah subkelas dari kelas Activity atau salah satu dari subkelasnya. Jika Anda membuat proyek baru di Android Studio, aktivitas tersebut secara default menjadi subkelas dari kelas AppCompatActivity. Kelas AppCompatActivity adalah subkelas Activity yang memungkinkan Anda menggunakan fitur aplikasi Android terbaru seperti bilah aksi dan desain material, sementara tetap memungkinkan aplikasi tersebut kompatibel dengan perangkat yang menjalankan Android versi lama.

Ini adalah subkelas kerangka dari aktivitas AppCompatActivity:

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Tugas pertama Anda di subkelas aktivitas adalah mengimplementasikan metode callback daur hidup aktivitas standar (seperti `onCreate()`) untuk menangani perubahan keadaan aktivitas. Perubahan keadaan ini meliputi hal-hal seperti kapan aktivitas dibuat, dihentikan, dilanjutkan, atau dimusnahkan. Anda akan mengetahui selengkapnya tentang daur hidup aktivitas dan callback daur hidup di bab berikutnya.

Salah satu callback yang diperlukan dan harus diimplementasikan oleh aplikasi Anda adalah metode `onCreate()`. Sistem memanggil metode ini bila membuat aktivitas, dan semua komponen penting aktivitas Anda harus melakukan diinisialisasi di sini. Yang paling penting, metode `onCreate()` memanggil

`setContentView()` untuk membuat layout utama aktivitas.

Anda biasanya mendefinisikan antarmuka pengguna untuk aktivitas di satu atau beberapa file layout XML. Bila metode `setContentView()` dipanggil dengan jalur ke file layout, sistem akan membuat semua tampilan awal dari layout yang ditetapkan dan menambahkannya ke aktivitas Anda. Hal ini sering kali disebut sebagai memekarkan layout.

Sering kali Anda mungkin juga ingin mengimplementasikan metode `onPause()` di kelas aktivitas. Sistem memanggil metode ini sebagai indikasi pertama bahwa pengguna meninggalkan aktivitas Anda (walaupun tidak selalu berarti aktivitas akan dimusnahkan). Di sinilah biasanya Anda harus mengikat perubahan yang harus dipertahankan di luar sesi pengguna saat ini (karena pengguna mungkin tidak akan kembali). Anda akan mengetahui selengkapnya tentang callback `onPause()` dan semua callback daur hidup lainnya di bab berikutnya.

Selain daur hidup callback, Anda juga bisa mengimplementasikan metode di aktivitas untuk menangani perilaku lainnya seperti masukan pengguna atau klik tombol. Implementasikan antarmuka pengguna Antarmuka pengguna untuk aktivitas disediakan menurut hierarki tampilan, yang mengontrol ruang tertentu dalam jendela aktivitas dan bisa merespons interaksi pengguna.

Cara yang paling umum untuk mendefinisikan antarmuka pengguna yang menggunakan tampilan adalah dengan file layout XML yang disimpan sebagai bagian dari sumber daya aplikasi Anda. Mendefinisikan layout di XML memungkinkan Anda untuk mengelola desain antarmuka pengguna secara terpisah dari kode sumber yang mendefinisikan perilaku aktivitas.

Anda juga bisa membuat tampilan baru secara langsung di kode aktivitas Anda dengan menyisipkan objek tampilan baru ke dalam ViewGroup, kemudian meneruskan ViewGroup akar ke setContentView(). Setelah layout dimekarkan – apa pun sumbernya --

Anda bisa menambahkan lebih banyak tampilan di Java di mana saja dalam hierarki tampilan.

Deklarasikan aktivitas di manifest

Setiap aktivitas di aplikasi Anda harus dideklarasikan di manifest aplikasi Android bersama elemen `<activity>`, di dalam `<application>`. Bila Anda membuat proyek baru atau menambahkan aktivitas baru ke proyek di Android Studio, manifest akan dibuat atau diperbarui untuk menyertakan deklarasi aktivitas kerangka bagi setiap aktivitas. Inilah deklarasi untuk aktivitas utama.

```
<activity android:name=".MainActivity" >
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Elemen `<activity>` menyertakan sejumlah atribut untuk mendefinisikan property aktivitas seperti label, ikon, atau tema.

Satu-satunya atribut yang diperlukan adalah `android:name`, yang menetapkan nama kelas aktivitas (seperti "Main Activity").

Lihat referensi elemen `<activity>` untuk informasi selengkapnya tentang deklarasi aktivitas. Elemen `<activity>` juga bisa menyertakan deklarasi untuk filter maksud. Filter maksud menetapkan jenis maksud yang akan diterima aktivitas Anda.

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Filter maksud harus menyertakan setidaknya satu elemen, dan juga bisa menyertakan sebuah dan opsional. Aktivitas utama untuk aplikasi Anda memerlukan filter maksud yang mendefinisikan aksi "main" dan kategori "launcher" agar sistem bisa meluncurkan aplikasi. Android Studio membuat filter maksud ini untuk aktivitas utama di proyek Anda:

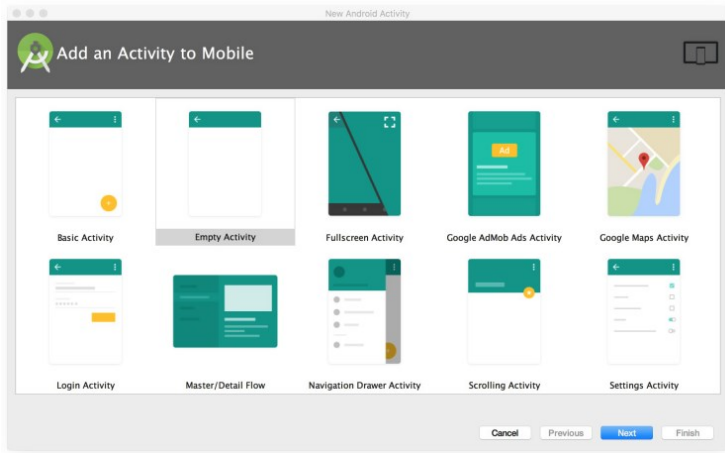
Elemen menetapkan bahwa ini adalah titik masuk "utama" ke aplikasi. Elemen menetapkan bahwa

aktivitas ini harus tercantum dalam peluncur aplikasi sistem (untuk memungkinkan pengguna meluncurkan aktivitas ini).

Aktivitas lain di aplikasi Anda juga bisa mendeklarasikan filter maksud, namun hanya aktivitas utama yang harus menyertakan aksi "utama". Anda akan mengetahui selengkapnya tentang maksud implisit dan filter maksud dalam bab berikutnya.

Tambahkan lebih banyak aktivitas ke proyek Anda

Aktivitas utama untuk aplikasi Anda dan file layout terkait disertakan bersama proyek bila Anda membuatnya. Anda bisa menambahkan aktivitas baru ke proyek di Android Studio dengan menu File > New > Activity. Pilih template aktivitas yang ingin Anda gunakan, atau buka Gallery untuk melihat semua template yang tersedia.



Bila memilih sebuah template aktivitas, Anda akan melihat serangkaian layar yang sama untuk membuat aktivitas baru yang dilakukan saat membuat proyek di awal. Android Studio menyediakan tiga hal ini untuk setiap aktivitas baru di aplikasi Anda:

- File Java untuk aktivitas baru dengan definisi kelas kerangka dan metode `onCreate()`. Aktivitas baru, seperti aktivitas utama, adalah subclass `AppCompatActivity`.
- File XML yang berisi layout untuk aktivitas baru. Perhatikan, metode `setContentView()` di kelas aktivitas akan memerkan layout baru ini.

- Elemen `<activity>` tambahan di manifest Android yang menetapkan aktivitas baru. Definisi aktivitas kedua tidak menyertakan filter maksud apa pun. Jika Anda ingin menggunakan aktivitas ini hanya dalam aplikasi (dan tidak memungkinkan aktivitas tersebut dimulai oleh aplikasi lain), maka tidak perlu menambahkan filter.

Tentang maksud

Semua aktivitas Android dimulai atau diaktifkan dengan maksud. Maksud adalah objek pesan yang membuat permintaan yang akan digunakan oleh waktu proses Android untuk memulai aktivitas atau komponen aplikasi lainnya di aplikasi Anda atau di beberapa aplikasi lainnya. Anda tidak bisa memulai aktivitas itu sendiri;

Bila aplikasi pertama kali dimulai dari layar utama perangkat, waktu proses Android akan mengirimkan maksud ke aplikasi Anda untuk memulai aktivitas utama aplikasi (yang didefinisikan dengan

aksi MAIN dan kategori LAUNCHER di Manifest Android). Untuk memulai aktivitas lain di aplikasi Anda, atau meminta tindakan untuk dilakukan oleh beberapa aktivitas lain yang tersedia diperangkat, bangunlah maksud sendiri dengan kelas Intent dan panggil metode startActivity() untuk mengirim maksud itu.

Selain untuk memulai aktivitas, maksud juga digunakan untuk meneruskan data di antara aktivitas. Bila membuat maksud untuk memulai aktivitas baru, Anda bisa menyertakan informasi tentang data yang diinginkan untuk mengoperasikan aktivitas baru itu. Jadi, misalnya, aktivitas email yang menampilkan daftar pesan bisa mengirim maksud ke aktivitas yang menampilkan pesan itu. Aktivitas tampilan memerlukan data tentang pesan yang akan ditampilkan, dan Anda bisa menyertakan data itu di maksud.

Tipe maksud

Ada dua tipe maksud di Android:

- Maksud eksplisit menetapkan aktivitas penerima (atau komponen lainnya) melalui nama kelas yang benar-benar memenuhi syarat di aktivitas tersebut. Gunakan maksud eksplisit untuk memulai komponen di aplikasi milik sendiri (misalnya, untuk beralih layar di antarmuka pengguna), karena Anda sudah mengetahui paket dan nama kelas komponen itu.
- Maksud implisit tidak menetapkan aktivitas tertentu atau komponen lainnya untuk menerima maksud. Sebagai gantinya, Anda mendeklarasikan aksi umum yang dilakukan di maksud tersebut. Sistem Android mencocokkan permintaan Anda dengan aktivitas atau komponen lainnya yang bisa menangani aksi permintaan tersebut. Anda akan mengetahui selengkapnya tentang maksud implisit dalam bab berikutnya.

Objek dan bidang Intent

Objek Intent adalah instance kelas Intent. Untuk maksud eksplisit, bidang kunci suatu maksud menyertakan yang berikut ini:

- Kelas aktivitas (untuk maksud eksplisit). Ini adalah nama kelas aktivitas atau komponen lainnya yang akan menerima maksud, misalnya, `com.example.SampleActivity.class`.
- Gunakan konstruktor maksud atau metode `setComponent()`, `setComponentName()` atau `setClassName()` maksud untuk menetapkan kelas.
- Data maksud. Bidang data maksud berisi referensi ke data yang Anda inginkan untuk mengoperasikan aktivitas penerima, sebagai objek Uri.
- Ekstra maksud. Ini adalah pasangan nilai-kunci yang membawa informasi yang diperlukan aktivitas penerima untuk melakukan aksi yang diminta.
- Flag maksud. Ini adalah bit metadata tambahan, yang didefinisikan oleh kelas Intent. Flag dapat menginstruksikan sistem Android tentang cara

meluncurkan aktivitas atau cara memperlakukan aktivitas setelah diluncurkan.

Untuk maksud implisit, Anda juga mungkin perlu mendefinisikan kategori dan aksi maksud. Anda akan mengetahui selengkapnya tentang kategori dan aksi maksud di bagian 2.3.

Memulai aktivitas dengan maksud eksplisit

Untuk memulai aktivitas tertentu dari aktivitas lain, gunakan maksud eksplisit dan metode `startActivity()`. Maksud eksplisit menyertakan nama kelas yang benar-benar memenuhi syarat untuk aktivitas atau komponen lainnya di objek `Intent`.

Semua bidang maksud lainnya bersifat opsional, dan nol secara default. Misalnya, jika Anda ingin memulai `ShowMessageActivity` untuk menampilkan pesan tertentu di aplikasi email, gunakan kode seperti ini.

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
startActivity(messageIntent);
```

Konstruktor Intent menggunakan dua argument untuk maksud eksplisit.

Konteks aplikasi. Dalam contoh ini, kelas aktivitas menyediakan materi (di sini, this).

Komponen tertentu untuk dimulai (ShowMessageActivity.class).

Gunakan metode `startActivity()` bersama objek maksud baru sebagai satu-satunya argumen. Metode `startActivity()` mengirim maksud ke sistem Android, yang meluncurkan kelas `ShowMessageActivity` atas nama aplikasi Anda. Aktivitas baru muncul pada layar dan aktivitas pembuatnya dihentikan sementara.

Aktivitas yang dimulai tetap pada layar hingga pengguna mengetuk tombol kembali pada perangkat, pada saat itu aktivitas ditutup dan diklaim kembali oleh sistem, dan aktivitas yang menghasilkannya akan dilanjutkan. Anda juga bisa menutup aktivitas yang dimulai secara manual sebagai respons terhadap aksi pengguna (seperti klik tombol) dengan metode `finish()`:

```
public void closeActivity (View view) {  
    finish();  
}
```

Meneruskan data diantara aktivitas dengan maksud

Selain untuk memulai satu aktivitas dari aktivitas lain, Anda juga menggunakan maksud untuk meneruskan informasi di antara aktivitas. Objek maksud yang Anda gunakan untuk memulai aktivitas bisa menyertakan data maksud (URI objek untuk bertindak), atau ekstra maksud, yang merupakan bit data tambahan yang mungkin diperlukan aktivitas.

Di aktivitas (pengirim) pertama:

1. Buat objek Intent.
2. Masukkan data atau ekstra ke dalam maksud itu.
3. Mulailah aktivitas baru dengan `startActivity()`.

Di aktivitas (penerima) kedua:

1. Dapatkan objek maksud yang digunakan memulai aktivitas.
2. Ambil data atau ekstra dari objek Intent.

Waktu untuk menggunakan data maksud atau ekstra maksud

Anda bisa menggunakan data maksud dan ekstra maksud untuk meneruskan data di antara aktivitas. Ada sejumlah perbedaan utama antara data dan ekstra yang menentukan mana yang harus Anda gunakan.

Data maksud hanya bisa menyimpan satu bagian informasi. URI yang menyatakan lokasi data yang ingin Anda gunakan untuk mengoperasikan. URI tersebut bisa berupa URL laman web (<http://>), nomor telepon (<tel://>), lokasi geografis (<geo://>), atau URI khusus lainnya yang Anda definisikan.

Gunakan bidang maksud data:

Bila Anda hanya memiliki satu bagian informasi yang perlu dikirim ke aktivitas yang telah dimulai.

Bila informasi itu adalah lokasi data yang bisa dinyatakan dengan URI. Ekstra maksud adalah untuk data arbitrer lainnya yang ingin Anda teruskan ke aktivitas yang telah dimulai. Ekstra maksud disimpan di objek Bundle sebagai pasangan kunci dan nilai. Bundle

adalah peta, yang dioptimalkan untuk Android, dengan tombol berupa string, dan nilai-nilainya bisa berupa tipe objek atau primitive apapun (objek harus mengimplementasikan antarmuka Parcelable). Untuk memasukkan data ke dalam ekstra maksud, Anda bisa menggunakan salah satu metode `putExtra()` kelas `Intent`, atau membuat bundle sendiri dan memasukkannya ke dalam maksud dengan `putExtras()`. Gunakan ekstra maksud:

Jika Anda ingin meneruskan lebih dari satu bagian informasi ke aktivitas yang telah dimulai. Jika informasi yang ingin Anda teruskan tidak bisa dinyatakan melalui URI.

Data dan ekstra maksud tidak eksklusif; Anda bisa menggunakan data URI dan ekstra untuk informasi tambahan yang diperlukan aktivitas yang dimulai untuk memproses data dalam URI itu.

Tambahkan data ke maksud

Untuk menambahkan data ke maksud eksplisit dari aktivitas pembuatnya, buat objek maksud seperti yang Anda lakukan sebelumnya:

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
```

Gunakan metode `setData()` bersama objek Uri untuk menambahkan URI itu ke maksud. Beberapa contoh penggunaan `setData()` bersama URI:

```
// A web page URL
messageIntent.setData(Uri.parse("http://www.google.com"));
// a Sample file URI
messageIntent.setData(Uri.fromFile(new File("/sdcard/sample.jpg")));
// A sample content: URI for your app's data model
messageIntent.setData(Uri.parse("content://mysample.provider/data"));
// Custom URI
messageIntent.setData(Uri.parse("custom:" + dataID + buttonId));
```

Perlu diingat bahwa bidang data hanya bisa berisi URI tunggal; jika Anda memanggil `setData()` beberapa kali, hanya nilai terakhir yang akan digunakan. Gunakan ekstra maksud untuk menyertakan informasi tambahan (termasuk URI.) Setelah menambahkan data, Anda bisa memulai aktivitas dengan maksud seperti biasanya.

```
startActivity(messageIntent);
```

Tambahkan ekstra ke maksud

Untuk menambahkan ekstra maksud ke maksud eksplisit dari aktivitas pembuatnya:

1. Tentukan kunci yang akan digunakan untuk informasi yang ingin dimasukkan ke dalam ekstra, atau definisikan sendiri. Setiap bagian informasi memerlukan kunci unik.
2. Gunakan metode `putExtra()` untuk menambahkan pasangan kunci/nilai ke ekstra maksud. Secara opsional Anda bisa membuat objek `Bundle`, menambahkan data ke bundel, kemudian menambahkan bundle ke maksud.

Kelas `Intent` menyertakan sejumlah kunci ekstra maksud yang bisa digunakan, didefinisikan sebagai konstanta yang dimulai dengan kata `EXTRA_`. Misalnya, Anda bisa menggunakan `Intent.EXTRA_EMAIL` untuk menunjukkan larik alamat email (sebagai string), atau `Intent.EXTRA_REFERRER` untuk menetapkan informasi

tentang aktivitas pembuat yang mengirim maksud tersebut.

Anda juga bisa mendefinisikan kunci ekstra maksud milik sendiri. Secara konvensional Anda mendefinisikan kunci ekstra maksud sebagai variabel-variabel statis dengan nama yang dimulai kata EXTRA_. Untuk menjamin kunci tersebut unik, nilai string kunci itu sendiri harus diawali dengan nama kelas yang benar-benar memenuhi syarat aplikasi. Misalnya:

```
public final static String EXTRA_MESSAGE = "com.example.mysampleapp.MESSAGE";  
public final static String EXTRA_POSITION_X = "com.example.mysampleapp.X";  
public final static String EXTRA_POSITION_Y = "com.example.mysampleapp.Y";
```

Buat objek maksud (jika belum ada):

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
```

Gunakan metode `putExtra()` bersama kunci untuk memasukkan data ke dalam ekstra maksud. Kelas `Intent` mendefinisikan banyak metode `putExtra()` untuk jenis data yang berbeda:

```
messageIntent.putExtra(EXTRA_MESSAGE, "this is my message");  
messageIntent.putExtra(EXTRA_POSITION_X, 100);  
messageIntent.putExtra(EXTRA_POSITION_Y, 500);
```

Atau, Anda bisa membuat bundle baru dan mengisinya dengan ekstra maksud. Bundel mendefinisikan banyak metode "put" untuk jenis data primitive yang berbeda serta objek yang mengimplementasikan antarmuka Parcelable Android atau Serializable Java.

```
Bundle extras = new Bundle();  
extras.putString(EXTRA_MESSAGE, "this is my message");  
extras.putInt(EXTRA_POSITION_X, 100);  
extras.putInt(EXTRA_POSITION_Y, 500);
```

Setelah Anda mengisi bundle tersebut, tambahkan ke maksud dengan metode `putExtras()` (perhatikan "s" di Extras):

```
messageIntent.putExtras(extras);
```

Mulai aktivitas dengan maksud seperti biasa:

```
startActivity(messageIntent);
```

Ambil data dari maksud di aktivitas yang dimulai

Bila Anda memulai aktivitas bersama maksud, aktivitas yang telah dimulai akan memiliki akses ke maksud dan data yang dimuatnya.

Untuk mengambil maksud yang digunakan untuk memulai aktivitas (atau komponen lain), gunakan metode `getIntent()`:

```
Intent intent = getIntent();
```

Gunakan `getData()` untuk mendapatkan URI dari maksud itu:

```
Uri locationUri = getData();
```

Untuk mendapatkan ekstra dari maksud, Anda perlu mengetahui kunci untuk pasangan kunci/nilai. Anda bisa menggunakan ekstra Intent standar jika

telah menggunakannya, atau bisa menggunakan kunci yang didefinisikan di aktivitas pembuatnya (jika didefinisikan sebagai publik.)

Gunakan salah satu metode `getStringExtra()` untuk mengekstrak data ekstra dari objek maksud:

```
String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);  
int positionX = intent.getIntExtra(MainActivity.EXTRA_POSITION_X);  
int positionY = intent.getIntExtra(MainActivity.EXTRA_POSITION_Y);
```

Atau Anda bisa mendapatkan seluruh bundle ekstra dari maksud dan mengekstrak nilai dengan beragam metode `Bundle`:

```
Bundle extras = intent.getExtras();  
String message = extras.getString(MainActivity.EXTRA_MESSAGE);
```

Mendapatkan data kembali dari aktivitas

Bila Anda memulai aktivitas bersama sebuah maksud, aktivitas pembuatnya akan dihentikan sementara, dan aktivitas baru tetap di layar hingga pengguna mengklik tombol kembali, atau panggil metode `finish()` di handle klik atau fungsi lainnya yang mengakhiri keterlibatan pengguna pada aktivitas ini.

Kadang-kadang bila mengirim data ke aktivitas bersama sebuah maksud, Anda juga ingin mendapatkan kembali data dari maksud itu. Misalnya, Anda mungkin memulai aktivitas galeri foto yang memungkinkan pengguna memilih foto. Dalam hal ini aktivitas asli Anda perlu menerima informasi tentang foto yang dipilih pengguna dari aktivitas yang diluncurkan.

Untuk meluncurkan aktivitas baru dan mendapatkan kembali hasilnya, lakukan langkah-langkah berikut di aktivitas pembuatnya:

1. Sebagai ganti meluncurkan aktivitas bersama `startActivity()`, panggil `startActivityForResult()` bersama maksud dan kode permintaan.
2. Buat maksud baru di aktivitas yang diluncurkan dan tambahkan data yang dikembalikan ke maksud itu.
3. Implementasikan `onActivityResult()` di aktivitas pembuatnya untuk memproses data yang dikembalikan.

Anda akan mempelajari tentang setiap langkah di bagian berikut ini.

Gunakan `startActivityForResult()` untuk meluncurkan aktivitas

Untuk mendapatkan kembali data dari aktivitas yang diluncurkan, mulailah aktivitas itu bersama metode `startActivityForResult()` sebagai ganti `startActivity()`.

```
startActivityForResult(messageIntent, TEXT_REQUEST);
```

Metode `startActivityForResult()`, seperti `startActivity()`, mengambil argument maksud yang berisi informasi tentang aktivitas yang diluncurkan dan data yang dikirim ke aktivitas itu. Akan tetapi, metode `startActivityForResult()` juga memerlukan kode permintaan.

Kode permintaan adalah integer yang mengidentifikasi permintaan dan bisa digunakan untuk membedakan hasil bila Anda memproses data yang dikembalikan. Misalnya, jika meluncurkan satu aktivitas untuk mengambil foto dan aktivitas lain untuk memilih foto dari galeri, Anda akan memerlukan kode permintaan yang berbeda untuk mengidentifikasi

permintaan yang menjadi pemilik data yang dikembalikan.

Secara konvensional, Anda mendefinisikan kode permintaan sebagai variabel-variabel integer statis dengan nama yang menyertakan REQUEST. Gunakan integer yang berbeda untuk setiap kode. Misalnya:

```
public static final int PHOTO_REQUEST = 1;
public static final int PHOTO_PICK_REQUEST = 2;
public static final int TEXT_REQUEST = 3;
```

Kembalikan respons dari aktivitas yang diluncurkan

Data respons dari aktivitas yang diluncurkan kembali ke aktivitas pembuatnya akan dikirim dalam maksud, baik dalam data maupun ekstra. Anda membentuk maksud yang dikembalikan ini dan memasukkan data ke dalamnya menggunakan cara yang sangat mirip dengan yang Anda lakukan untuk maksud yang mengirimnya. Biasanya aktivitas yang diluncurkan akan memiliki metode `onClick` atau

metode callback masukan pengguna lain yang Anda gunakan untuk memproses aksi pengguna dan menutup aktivitas. Di sini juga Anda membentuk respons.

Untuk mengembalikan data dari aktivitas yang diluncurkan, buat objek maksud kosong yang baru.

```
Intent returnIntent = new Intent();
```

Catatan: Untuk menghindari kebingungan atas data yang dikirim dengan data yang dikembalikan, gunakan objek maksud baru, bukan menggunakan kembali objek maksud pengirim asal.

Maksud hasil yang dikembalikan tidak memerlukan kelas atau nama komponen berakhir di tempat yang tepat. Sistem Android akan mengarahkan respons kembali ke aktivitas pembuatnya untuk Anda.

Tambahkan data atau ekstra ke maksud dengan cara sama seperti yang Anda lakukan pada maksud asal. Anda mungkin perlu mendefinisikan kunci untuk ekstra maksud yang dikembalikan pada awal kelas.


```
public final static String EXTRA_RETURN_MESSAGE =  
    "com.example.mysampleapp.RETURN_MESSAGE";
```

Selanjutnya masukkan data yang dikembalikan ke maksud seperti biasa. Di sini, pesan yang dikembalikan adalah ekstra maksud dengan kunci EXTRA_RETURN_MESSAGE.

```
messageIntent.putExtra(EXTRA_RETURN_MESSAGE, mMessage);
```

Gunakan metode setResult() bersama kode respons dan maksud dengan data respons:

```
setResult(RESULT_OK, replyIntent);
```

Kode respons didefinisikan oleh kelas Activity, dan bisa berupa

RESULT_OK: permintaan berhasil.

RESULT_CANCELED: pengguna membatalkan operasi.

RESULT_FIRST_USER: untuk mendefinisikan kode hasil milik Anda.

Anda akan menggunakan kode hasil di aktivitas pembuatnya.

Terakhir, panggil `finish()` untuk menutup aktivitas dan melanjutkan aktivitas pembuatnya:

```
finish();
```

Baca data respons di `onActivityResult()`

Karena sekarang aktivitas yang diluncurkan telah mengirimkan data kembali ke aktivitas pembuatnya bersama maksud, aktivitas pertama itu harus menangani data tersebut. Untuk menangani data yang dikembalikan di aktivitas pembuatnya, implementasikan metode callback `onActivityResult()`. Inilah sebuah contoh sederhana.

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == TEXT_REQUEST) {
        if (resultCode == RESULT_OK) {
            String reply =
                data.getStringExtra(SecondActivity.EXTRA_RETURN_MESSAGE);
            // process data
        }
    }
}
```

Tiga argument untuk `onActivityResult()` berisi semua informasi yang Anda perlukan untuk menangani data yang dikembalikan.

1. Kode permintaan. Kode permintaan yang Anda setel saat meluncurkan aktivitas bersama `startActivityForResult()`. Jika Anda meluncurkan aktivitas yang berbeda untuk melakukan operasi yang berbeda, gunakan kode ini untuk mengidentifikasi data tertentu yang Anda dapatkan kembali.
2. Kode hasil: kode hasil yang disetel di aktivitas yang diluncurkan, biasanya salah satu dari `RESULT_OK` atau `RESULT_CANCELED`.
3. Data maksud. Maksud berisi data yang dikembalikan dari aktivitas peluncur.

Metode contoh yang ditampilkan di atas menampilkan logika tipikal untuk menangani kode permintaan dan kode respons.

Pengujian pertama adalah untuk permintaan `TEXT_REQUEST`, dan bahwa hasilnya adalah berhasil. Di dalam isi pengujian itu, ekstrak informasi yang

dikembalikan dari maksud. Gunakan `getData()` untuk mendapatkan data maksud, atau `getExtra()` untuk mengambil nilai dari ekstra maksud bersama kunci tertentu.

Navigasi aktivitas


Setiap aplikasi dengan kompleksitas apapun yang Anda bangun akan menyertakan beberapa aktivitas, baik yang didesain dan diimplementasikan oleh Anda, maupun yang berpotensi di aplikasi lain. Karena pengguna Anda berpindah-pindah dalam aplikasi dan di antara aktivitas, navigasi yang konsisten menjadi semakin penting untuk pengalaman pengguna aplikasi. Beberapa hal lebih mengganggu pengguna daripada navigasi dasar yang berperilaku tidak konsisten dan tak terduga. Mendesain navigasi aplikasi Anda dengan bijak akan membuat penggunaan aplikasi tersebut bisa diprediksi dan bisa diandalkan pengguna.

Sistem Android mendukung dua bentuk strategi navigasi yang berbeda untuk aplikasi Anda.

Navigasi Sementara atau Kembali, yang disediakan melalui tombol kembali di perangkat, dan back-stack.

Navigasi leluhur, atau Naik, yang disediakan oleh Anda sebagai opsi di bilah aksi aplikasi.

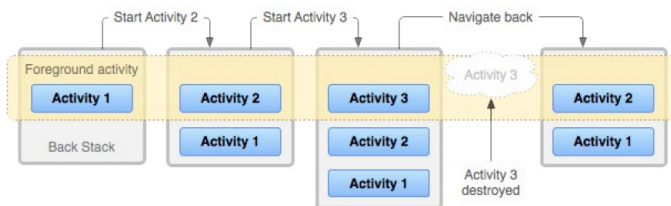
Navigasi kembali, tugas, dan back-stack


Navigasi Kembali memungkinkan pengguna Anda untuk kembali ke aktivitas sebelumnya dengan mengetuk tombol kembali di perangkat . Navigasi kembali juga disebut navigasi sementara karena tombol kembali menyusuri riwayat layar yang baru saja ditampilkan, dalam urutan kronologi terbalik.

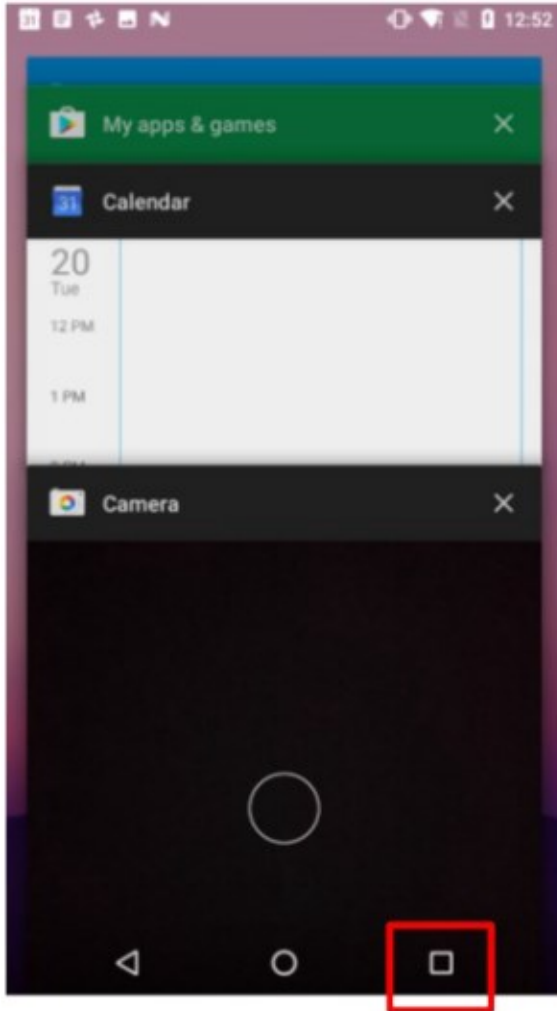
Back-stack adalah serangkaian aktivitas yang telah dikunjungi pengguna dan bisa dikunjungi kembali oleh pengguna dengan tombol kembali. Setiap kali aktivitas baru dimulai, aktivitas akan didorong ke back-stack dan mengambil focus

pengguna. Aktivitas sebelumnya akan dihentikan, namun tetap tersedia di back-stack. Back-stack beroperasi berdasarkan mekanisme "masuk terakhir, keluar pertama", jadi bila pengguna selesai dengan aktivitas saat ini dan menekan tombol Kembali, aktivitas tersebut akan dimunculkan dari tumpukan (serta dimusnahkan) dan aktivitas sebelumnya dilanjutkan.

Karena aplikasi bisa memulai aktivitas baik di dalam maupun di luar satu aplikasi, back-stack berisi semua aktivitas yang telah diluncurkan oleh pengguna dalam urutan terbalik. Setiap kali pengguna menekan tombol Kembali, setiap aktivitas di tumpukan akan dimunculkan untuk membuka aktivitas sebelumnya, hingga pengguna kembali ke layar Utama.



Android menyediakan back-stack untuk setiap tugas. Tugas adalah konsep penyusunan semua aktivitas yang berinteraksi dengan pengguna saat melakukan operasi, baik di dalam aplikasi maupun di beberapa aplikasi. Sebagian besar tugas dimulai dari layar utama Android, dan mengetuk ikon aplikasi akan memulai tugas (serta back-stack baru) untuk aplikasi itu. Jika pengguna menggunakan aplikasi sebentar, mengetuk beranda, dan memulai aplikasi, aplikasi baru itu akan diluncurkan di tugasnya sendiri dan memiliki back-stack sendiri. Jika pengguna kembali ke aplikasi pertama, back-stack tugas pertama akan kembali. Menyusuri dengan tombol kembali hanya akan mengembalikan ke aktivitas di tugas saat ini, bukan untuk semua tugas yang berjalan pada perangkat. Android memungkinkan pengguna menyusuri berbagai tugas bersama ringkasan atau layar tugas saat ini, yang bisa diakses dengan tombol segi empat di sudut kanan bawah perangkat .



Dalam sebagian besar kasus, Anda tidak perlu khawatir tentang pengelolaan tugas atau back-stack

untuk aplikasi—sistem terus melacak hal ini untuk Anda, dan tombol kembali selalu tersedia pada perangkat.

Akan tetapi, mungkin ada saatnya Anda ingin mengganti perilaku default untuk tugas atau untuk back-stack. Misalnya, jika layar Anda berisi browser web yang disematkan yang memungkinkan pengguna menyusuri laman web, Anda mungkin ingin menggunakan perilaku kembali default di browser bila pengguna menekan tombol Kembali di perangkat, daripada mengembalikan ke aktivitas sebelumnya. Anda mungkin juga perlu mengubah perilaku default aplikasi dalam kasus khusus lainnya seperti pada notifikasi atau widget, sehingga aktivitas yang berada jauh di dalam aplikasi Anda mungkin diluncurkan sebagai tugasnya sendiri, tanpa back-stack sama sekali. Anda akan mengetahui selengkapnya tentang pengelolaan tugas dan back-stack dibagian berikutnya.

Navigasi naik



Misalnya, jika aktivitas utama di aplikasi email adalah daftar semua pesan, memilih sebuah pesan akan meluncurkan aktivitas kedua untuk menampilkan satu email itu. Dalam hal ini, aktivitas pesan akan menyediakan tombol Naik yang mengembalikan ke daftar pesan.

Perilaku tombol Naik didefinisikan oleh Anda di setiap aktivitas berdasarkan cara mendesain navigasi aplikasi. Dalam banyak kasus, navigasi Naik dan Kembali mungkin menyediakan perilaku yang sama:

Cuma mengembalikan ke aktivitas sebelumnya. Misalnya, aktivitas Setelan mungkin tersedia dari aktivitas apapun di aplikasi Anda, sehingga "naik" sama dengan kembali – Cuma mengembalikan pengguna ke tempat sebelumnya di hierarki.

Menyediakan perilaku Naik untuk aplikasi Anda adalah hal yang opsional, namun praktik desain yang baik, adalah menyediakan navigasi yang konsisten untuk berbagai aktivitas di aplikasi Anda.

Implementasikan navigasi naik bersama aktivitas induk

Dengan proyek template standar di Android Studio, mudah untuk mengimplementasikan navigasi Naik. Jika satu aktivitas adalah anak aktivitas lain di hierarki aktivitas aplikasi, tetapkan induk aktivitas itu di Manifes Android.

Mulai di Android 4.1 (API level 16), deklarasikan induk logis setiap aktivitas dengan menetapkan atribut `android:parentActivityName` di elemen `<activity>`. Untuk mendukung Android versi lama, sertakan

informasi `<meta-data>` untuk mendefinisikan aktivitas induk secara eksplisit. Gunakan kedua metode agar kompatibel mundur dengan semua versi Android.

Ini adalah definisi kerangka untuk kedua aktivitas utama (induk) dan aktivitas kedua (anak):

```
<application ... >
  <!-- The main/home activity (it has no parent activity) -->
  <activity
    android:name=".MainActivity" ...>
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

  </activity>
  <!-- A child of the main activity -->
  <activity android:name=".SecondActivity"
    android:label="@string/activity2_name"
    android:parentActivityName=".MainActivity">
    <meta-data
      android:name="android.support.PARENT_ACTIVITY"
      android:value="com.example.android.twoactivities.MainActivity" />
  </activity>
</application>
```

BAB IV

BANGUN APLIKASI

ANDROID

BAB IV

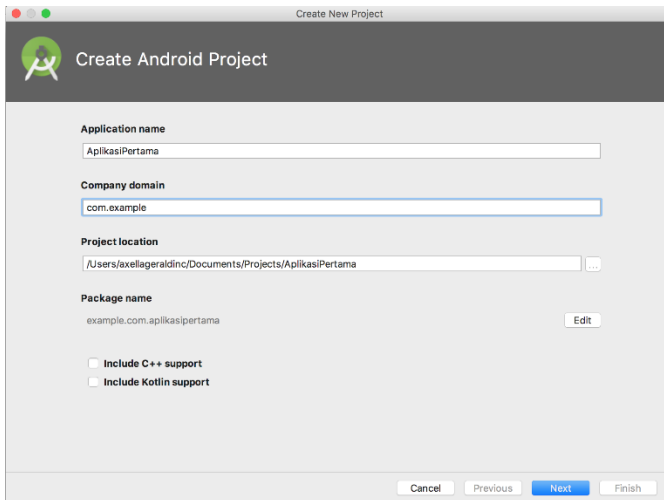
BANGUN APLIKASI ANDROID

A. Membuat Aplikasi Android Pertama

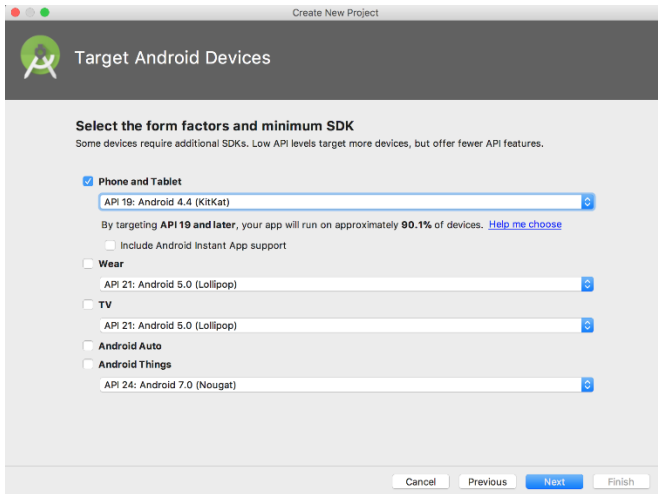
Dalam membuat aplikasi android menggunakan android studio maka harus melakukan tahap-tahap sebagai berikut.

1. Create New Project Android

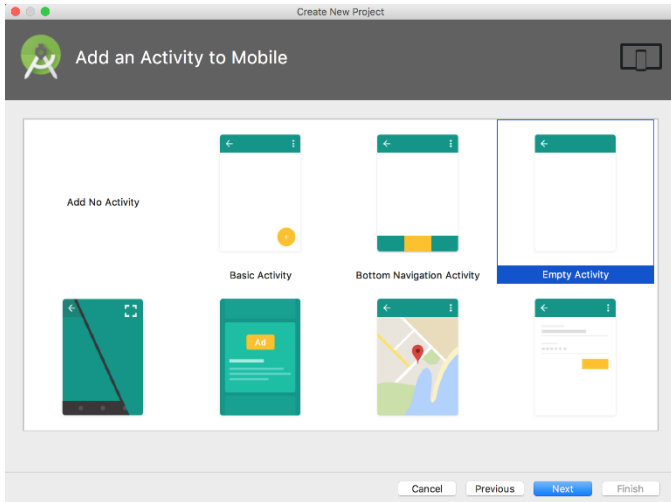
- a. Buka Android Studio dan pilih Start a new Android Studio project, lalu akan muncul halaman seperti di bawah ini.



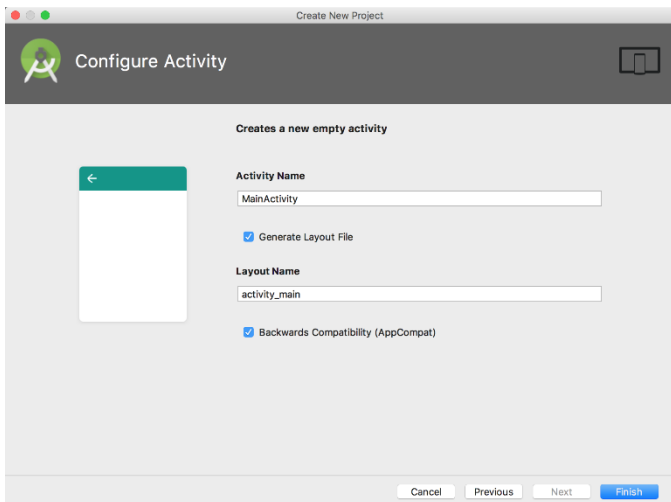
- b. Isikan nama aplikasinya dan pilih lokasi penyimpanan project android studionya.



- c. Pilih minimum OS Android yang dibutuhkan untuk menjalankan aplikasimu. Disini aku pilih minimum OS nya adalah Android 4.4 (Kitkat).

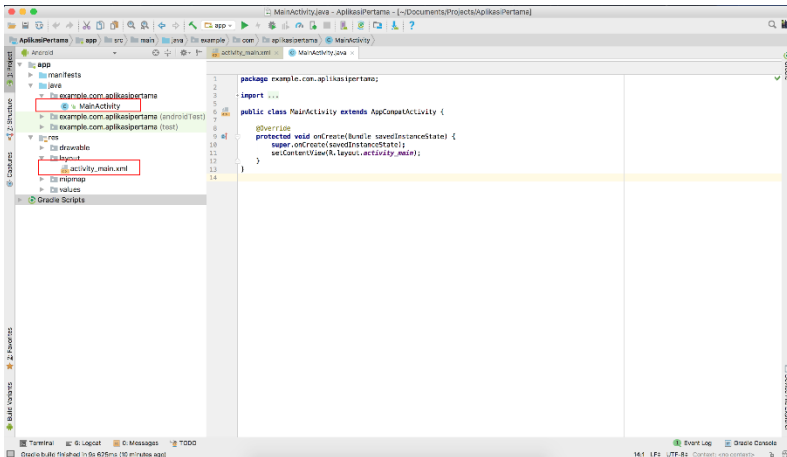


d. Memilih activity. Pilih saja empty activity.



e. Memberi nama activity yang akan dibuat, lalu klik FINISH.

2. Tampilan Environment Android Studio



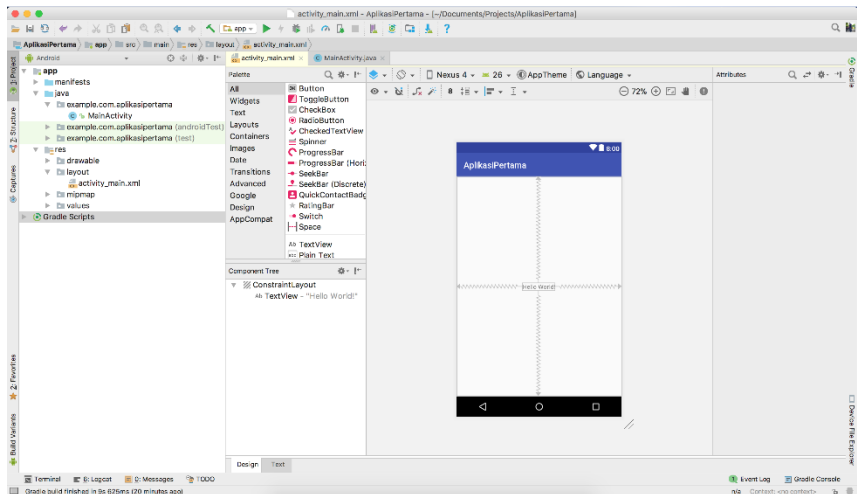
Gambar 4.1. Tampilan Android Studio saat membuat project baru

Berikut penjelasan dari kedua kotak merah dari gambar diatas.

1. **MainActivity:** Adalah class java untuk activity yang sudah dibuat tadi. Di file inilah kita akan coding untuk logic/proses aplikasi kita.
2. **activity_main.xml:** Adalah file xml untuk class MainActivity. Jadi, apabila file MainActivity tadi itu berisi logic/proses dari aplikasi, file xml ini adalah tampilannya.

3. Melihat tampilan android

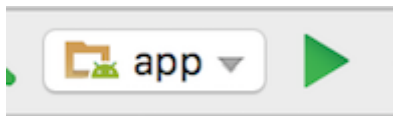
Buka file **activity_main.xml**. Maka akan terlihat seperti di bawah ini:



Pada **activity_main.xml** ini berisi tampilan dari activity yang akan ditampilkan di layar android nantinya.

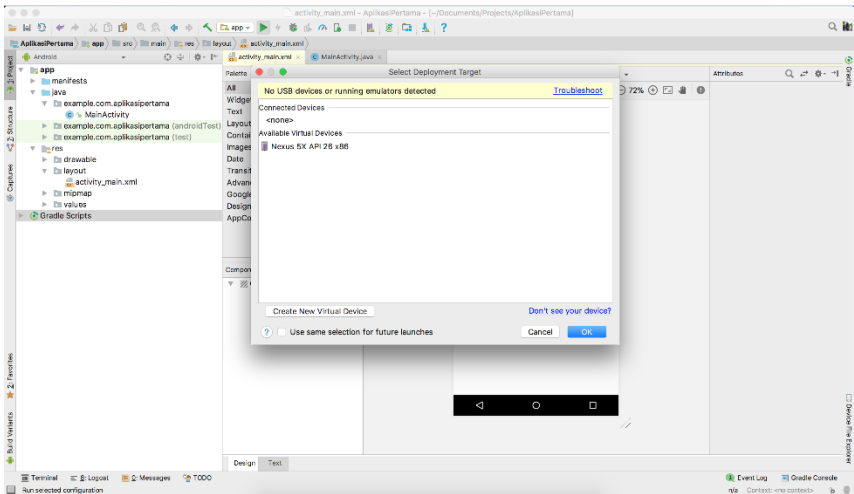
4. Jalankan aplikasi android

Jalankan aplikasi android kita ini dengan cara menekan tombol play berwarna hijau.



Tombol play berwarna hijau untuk menjalankan aplikasi android.

Setelah itu akan muncul dialog permintaan untuk memilih untuk menjalankan aplikasi android.

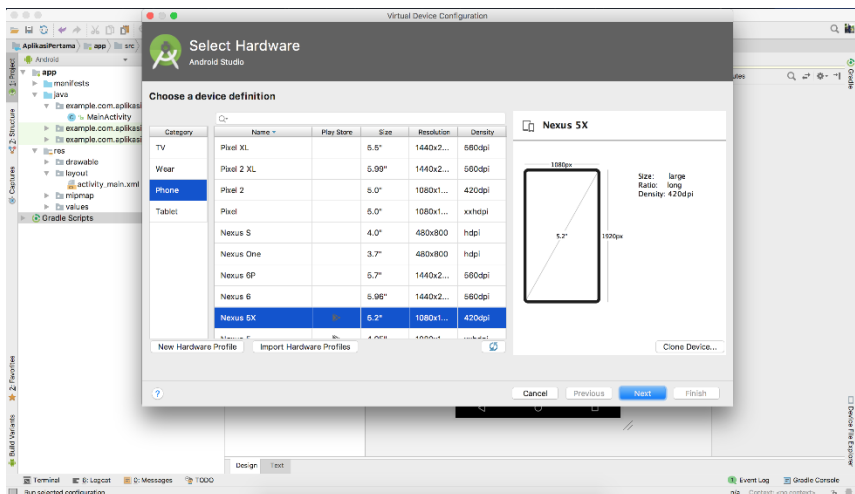


Memilih device untuk menjalankan aplikasi android

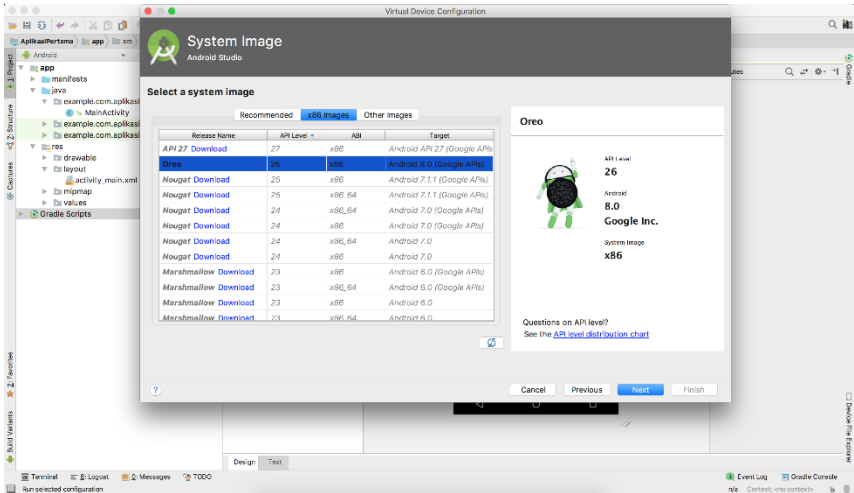
1. Untuk yang punya smartphone android, pergi ke developer options di menu settings android-nya → Lalu aktifkan opsi USB Debugging → Tancapkan smartphone ke komputermu via kabel USB → Jika muncul peringatan di android mengenai perizinan USB Debugging, klik saja ALLOW → Install driver smartphone android di komputermu → (Kembali ke Android Studio) → Maka

device android-mu akan muncul di dialog tadi di bagian Connected Devices. Klik device android-mu lalu klik OK, tunggu hingga proses compiling selesai dan aplikasi android-mu akan berjalan di smartphone-mu.

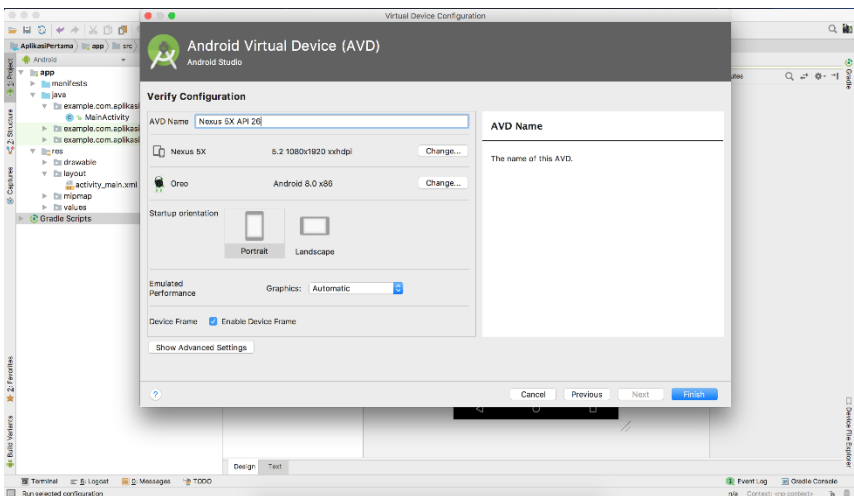
2. Untuk yang ingin menjalankan aplikasi di Emulator, klik pada Create New Virtual Device pada dialog di atas.



Pilih ukuran layar dari emulator yang akan dibuat

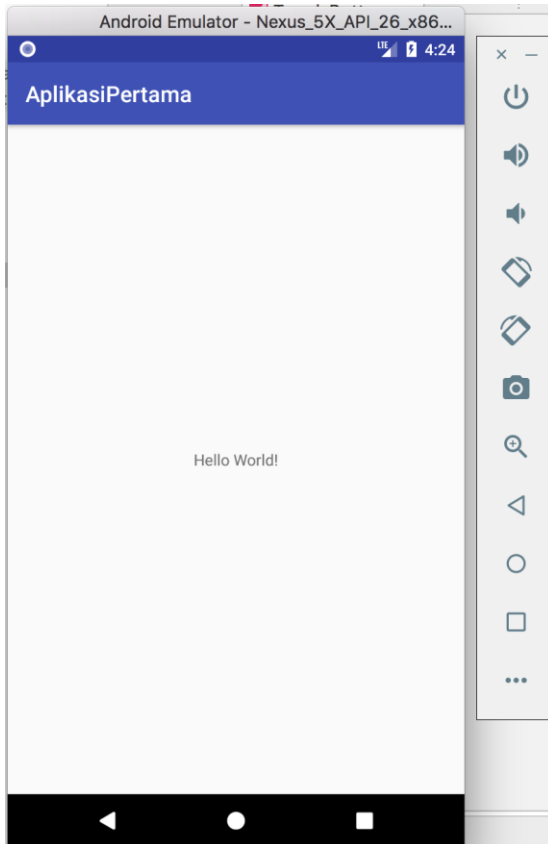


Memilih versi OS Android yang akan diinstall pada emulator. Sebagai contoh Android Oreo.



Memberi nama emulatorenya dan konfigurasi lain. Setelah selesai, klik FINISH.

Ketika proses pembuatan emulator sudah selesai, maka emulator-nya akan muncul di dialog tadi, klik emulatoanya dan klik OK. Tunggu hingga proses booting emulator selesai dan proses compiling program selesai.



Aplikasi berhasil berjalan di emulator android yang telah dibuat.

B. Perancangan User Interface dan User Experience

Pada Aplikasi

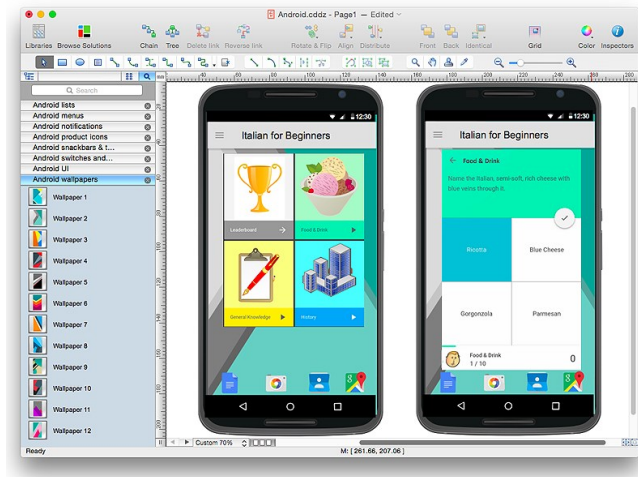
1. Perancangan User Interface

Perancangan UI (User Interface) merupakan proses menciptakan media komunikasi yang efektif antara manusia dan komputer, atau pada saat ini aplikasi mobile. Istilah lainnya, UI menjadi penghubung secara langsung antara sistem dengan pengguna. Perancangan UI melalui suatu proses yang kompleks, karena itu UI harus dibuat dengan benar sebab akan membentuk persepsi para pengguna terhadap suatu perangkat lunak. Desain UI harus dibuat dengan memperhatikan kemudahan penggunaan agar dapat diterima oleh masyarakat. Jika pengguna merasa UI yang dibuat tidak menarik, sulit dimengerti, dan dapat menyebabkan kebosanan, maka dapat berakibat kegagalan pada sebuah aplikasi. Saat ini, penelitian yang fokus tentang desain UI aplikasi mobile smart grid tidak ditemukan. Penelitian yang telah dilakukan mengarah pada soal teknis proses sistem smart grid dapat dilakukan pemantauan, pengaturan, dan

penjadwalan konsumsi energi pada aplikasi mobile saja. Sementara itu, untuk proses menjual dan membeli daya listrik belum ada, padahal proses ini merupakan salah satu keunggulan dari smart grid. Karenanya, perlu adanya penelitian lebih lanjut mengenai proses menjual dan membeli pada perancangan desain UI aplikasi mobile smart grid. Pada proses perancangan UI, sangat penting untuk memperhatikan perilaku calon pengguna. Agar mudah digunakan pengguna, cara yang umum digunakan untuk merancang desain UI pada aplikasi mobile Android adalah menggunakan MDfA atau desain template UI standar Android dan mengadopsi CES atau desain UI aplikasi mobile yang populer. MDfA digunakan pada aplikasi Gmail atau aplikasi bawaan dari Google lainnya. Karena aplikasi ini arahnya lebih kepada sebuah e-commerce, maka aplikasi ini dapat mengadopsi CES desain UI dari aplikasi mobile e-commerce yang populer di Indonesia. Pada makalah ini, perancangan UI aplikasi mobile smart grid menggabungkan dua cara, yaitu penggunaan MDfA dengan penggunaan desain yang mengadopsi CES. Penggabungan keduanya selain

bertujuan untuk mendapatkan hasil rancangan desain terbaik, juga didasari oleh hasil penelitian sebelumnya. Dari hasil pengujian, perbedaan skor antara kedua cara tersebut hanya selisih sedikit saja. Selain itu, penelitian lain merekomendasikan penggunaan MDfA karena persentase yang tinggi menunjukkan bahwa kebanyakan file XML UI menggunakan beberapa MDfA (misalnya Relative Layout, Frame Layout, dan LinearLayout). Untuk mengetahui mudah digunakan dan diterima atau tidaknya sebuah desain UI, perlu dilakukan penelitian dengan pengujian. Metode yang umumnya digunakan saat ini untuk mengukur kemudahan digunakan adalah uji usability. Usability merupakan derajat kemampuan perangkat lunak untuk membantu pengguna menyelesaikan tugas tertentu. Sebuah aplikasi perlu memperhatikan aspek usability sebagai kunci keberhasilan dan syarat penerimaan pengguna terhadap aplikasi mobile. UI yang baru perlu melakukan pengujian usability sebelum diberikan pada pengguna umum. Namun, di sisi lain, pengujian usability hanya fokus pada kemudahan penggunaan pada aplikasi. Pada dasarnya,

semua aplikasi pasti dibuat dengan tujuan dapat dan mudah digunakan untuk menyelesaikan tujuan tertentu. Untuk dapat bersaing dengan aplikasi sejenisnya, aplikasi mobile harus mempunyai daya tarik, agar digunakan terus menerus oleh pengguna. Oleh karena itu, pengujian pada UI sebuah desain UI aplikasi mobile tidaklah cukup hanya dengan aspek usability, tetapi perlu adanya pengujian lain seperti pengalaman pengguna. Pengalaman pengguna atau user experience (UX) merupakan faktor penting untuk menentukan suatu informasi sudah cukup memadai, penerimaan oleh penggunanya, atau belum. Selain uji usability, uji UX juga perlu dilakukan untuk kesuksesan sebuah desain UI. Untuk melakukan pengujian UX dengan cepat dan sederhana, dapat digunakan UEQ. Pada makalah ini, pengujian desain prototipe UI aplikasi mobile smart grid menggabungkan uji usability dan UX. Untuk pengujian usability digunakan SUS dan untuk pengujian UX digunakan UEQ.

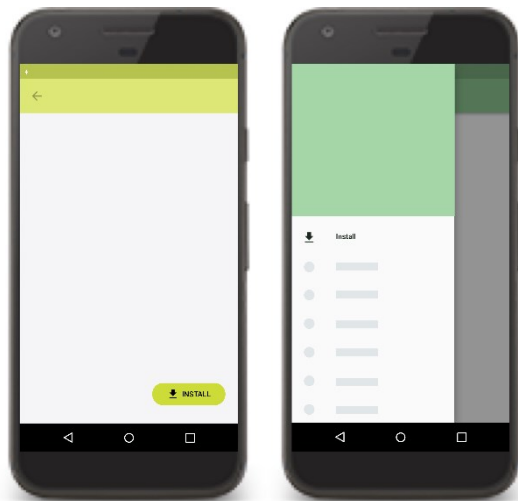


Gambar 4.2 Contoh Perancangan User Interface

2. Perancangan User Experience

Pengalaman pengguna atau user experience (UX) merupakan faktor penting untuk menentukan suatu informasi sudah cukup memadai, penerimaan oleh penggunanya, atau belum. Selain uji usability, uji UX juga perlu dilakukan untuk kesuksesan sebuah desain UI. Untuk melakukan pengujian UX dengan cepat dan sederhana, dapat digunakan UEQ. Pengujian UX menggunakan UEQ dapat dilakukan dengan cepat dan sederhana, karena UEQ merupakan alat atau kuesioner

yang mudah dan efisien untuk mengukur pengalaman pengguna. UEQ berisi enam aspek, yaitu daya tarik, kejelasan, efisiensi, ketepatan, stimulasi, dan kebaruan.



Gambar 4.3 Contoh Perancangan User Experience

C. Menguji dan Mendebug Aplikasi

Android Studio menyediakan debugger yang memungkinkan Anda melakukan hal-hal berikut ini dan banyak lagi:

- Memilih perangkat untuk men-debug aplikasi Anda.
- Menetapkan titik henti sementara dalam kode Java, Kotlin, dan C/C++.

- Memeriksa variabel dan mengevaluasi ekspresi pada saat runtime.

Mengaktifkan proses debug

Sebelum mulai men-debug, Anda perlu mempersiapkan hal-hal berikut:

- Aktifkan proses debug di perangkat Anda:
Jika Anda menggunakan emulator, proses ini diaktifkan secara default. Namun, untuk perangkat yang terhubung, Anda perlu mengaktifkan proses debug dalam opsi developer perangkat.
- Jalankan varian build yang dapat di-debug:
Anda harus menggunakan varian build yang menyertakan `debuggable true` dalam konfigurasi build. Biasanya, Anda cukup memilih varian "debug" default yang disertakan dalam setiap project Android Studio (meskipun tidak terlihat dalam file `build.gradle`). Tetapi, jika Anda menentukan tipe build baru yang seharusnya dapat di-debug, Anda harus

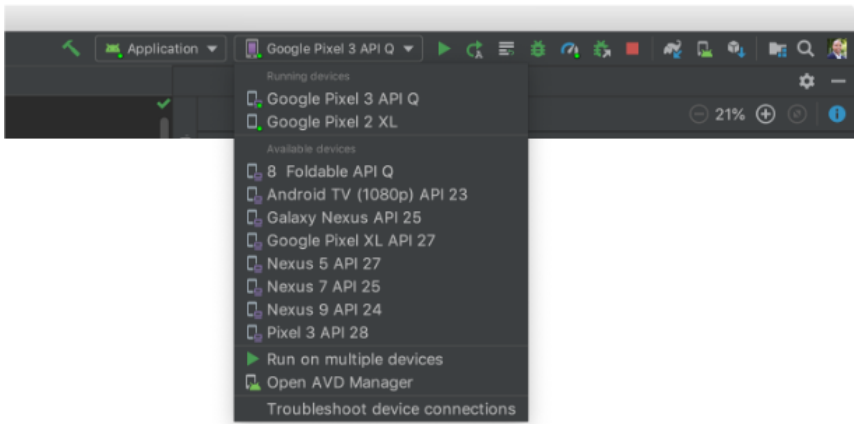
menambahkan `debuggable true` ke tipe build tersebut:

```
android {  
    buildTypes {  
        customDebugType {  
            debuggable true  
            ...  
        }  
    }  
}
```


Memulai proses debug

Anda dapat memulai sesi proses debug dengan langkah berikut:

1. Tetapkan beberapa titik henti sementara dalam kode aplikasi.
2. Pada toolbar, pilih perangkat untuk men-debug aplikasi Anda dari menu drop-down perangkat target




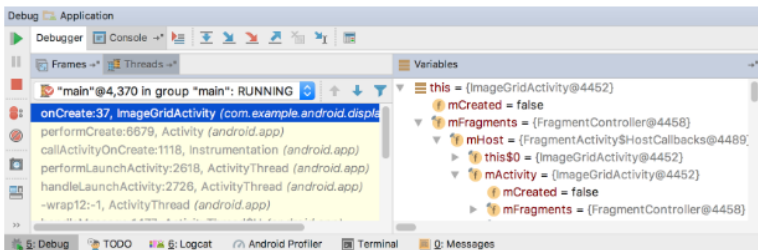
Jika tidak memiliki perangkat yang dikonfigurasi, Anda harus menghubungkan perangkat melalui USB atau membuat AVD untuk menggunakan Android Emulator.

3. Di toolbar, klik Debug .

Jika Anda melihat dialog yang menanyakan apakah Anda ingin "switch from Run to Debug" (beralih dari Run ke Debug), berarti aplikasi Anda sudah berjalan di perangkat dan aplikasi akan dimulai ulang untuk memulai proses debug. Jika Anda ingin instance aplikasi yang sama tetap berjalan, klik Cancel Debug dan pasang debugger ke aplikasi yang sedang berjalan.

Jika tidak, Android Studio membuat APK, menandatangani dengan kunci debug, menginstalnya di perangkat yang Anda pilih, dan menjalankannya. Jika Anda menambahkan kode C dan C++ ke project Anda, Android Studio juga akan menjalankan debugger LLDB di jendela Debug untuk men-debug kode native Anda.


4. Jika jendela Debug tidak terbuka, pilih View > Tool Windows > Debug (atau klik Debug  di jendela toolbar), lalu klik tab Debugger, seperti yang ditunjukkan pada gambar 1.



Gambar 4.4. Jendela Debugger, menampilkan thread saat ini dan pohon objek untuk variabel

Memasang debugger ke aplikasi yang sedang berjalan

Jika aplikasi Anda sudah berjalan di perangkat, Anda dapat memulai proses debug tanpa memulai ulang aplikasi dengan cara berikut:

1. Klik Attach debugger to Android process  .
2. Pada dialog Choose Process, pilih proses yang Anda ingin dipasangkan debugger.

Jika Anda menggunakan emulator atau perangkat yang telah di-root, Anda dapat mencentang Show all processes untuk melihat semua proses.

Dari menu drop-down Debugger, Anda bisa memilih tipe debug lainnya. Secara default, Android Studio menggunakan tipe debug Auto untuk memilih opsi debugger terbaik berdasarkan apakah project Anda menyertakan kode Java atau C/C++.

3. Klik OK.
Jendela Debug akan muncul.

BAB V

PENUTUP

BAB V

PENUTUP

5.1 Kesimpulan

Aplikasi Android dirancang sebagai kombinasi dari beragam komponen yang bisa dipanggil satu per satu. Misalnya, aktivitas adalah jenis komponen aplikasi yang menyediakan antarmuka pengguna (UI).

Android memungkinkan Anda menyediakan resource yang berbeda untuk perangkat yang berbeda. Misalnya, Anda bisa membuat tata letak yang berbeda untuk ukuran layar yang berbeda. Sistem akan menentukan tata letak mana yang akan digunakan berdasarkan ukuran layar perangkat yang sedang digunakan.

Jika salah satu fitur aplikasi Anda memerlukan hardware tertentu, seperti kamera, Anda dapat mengkueri pada saat runtime terlepas dari apakah perangkat memiliki hardware itu atau tidak, lalu menonaktifkan fitur tersebut jika tidak. Anda dapat menentukan bahwa aplikasi Anda memerlukan hardware tertentu sehingga Google Play tidak

akan mengizinkan aplikasi untuk diinstal pada perangkat tanpa hardware tersebut.

5.2 Kritik dan Saran

Jika ada kesalahan dalam penulisan dan pengeditan tulisan ini penulis dengan sangat membutuhkan kritik dan saran yang membangun.

DAFTAR REFERENSI

DAFTAR REFERENSI

- Android Developers. Arsitektur Platform. Diakses pada 06-03-2020 di <https://developer.android.com/guide/platform?hl=id>
- B. Laugwitz, T. Held, dan M. Schrepp, "Construction and Evaluation of a User Experience Questionnaire," HCI Usability Educ. Work, vol. 5298, hal. 63–76, 2008.
- C++ (2016, Juli 24). Diambil kembali dari Wikipedia Bahasa Indonesia:<https://id.wikipedia.org/wiki/C%2B%2B>
- ColdFusion. (2013, April 5). Diambil kembali dari FENDIK DANA ISWARA:<https://fendikcancer.wordpress.com/2013/04/05/coldfusion>
- Deokar, R. S., Uchade, C. S., Tayde, A. K., Shelke, J. B., & Jawandhiya, P. M. (2017, April). Web Application Languages:A review. International Journal of Research in

Advent Technology (IJRAT), Special Issue National Conference "CONVERGENCE 2017", pp. 72-78.

H. B. Santoso, M. Schrepp, R. Yugo Kartono Isal, Y. Utomo, dan B. Priyogi, "Measuring User Experience of the Student-Centered eLearning Environment," J. Educ. Online-JEO, Vol. 13, No. 1, hal. 142–166, 2016.

H. B. Santoso, R. Y. K. Isal, T. Basaruddin, L. Sadita, dan M. Schrepp, "Research-in-progress: User Experience Evaluation of Student Centered E-Learning Environment for Computer Science Program," 2014 3rd International Conference on User Science and Engineering (i-USEr), 2014, dan. 52–55.

M. Schrepp, (2015) "User Experience Questionnaire Handbook," [Online], https://www.researchgate.net/publication/281973617_User_Experience_Questionnaire_Handbook, tanggal akses: 19-Jan-2018.

Sophan, Mochammad Kautsar., Kurniawati, Arik. 2018. PERANCANGAN APLIKASI LEARNING BY DOING

INTERAKTIF UNTUK Mendukung Pembelajaran Bahasa Pemrograman. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 02(05) 163-170.

Tim Pelatihan Developer Google. 2016. Android Developer Fundamentals Course – Concept Reference. Android Developer Fundamental

Tim Pelatihan Developer Google. 2016. Android Developer Fundamentals Course - Practical Workbook. Android Developer Fundamental.

Staff, W., 2009. Using Wireframes to Streamline Your Development Process. [Online] Available at: <https://www.webdesignerdepot.com/2009/07/using-wireframes-to-streamline-your-development-process/> [Accessed 27 April 2017].

Ward, T., 2010. UX 101: The Wireframe. [Online] Available at: <https://www.viget.com/articles/ux-101-the-wireframe1> [Accessed 27 April 2017].

Wibawanto, Wandah, Nugrahani, Rahina. 2018. *DESAIN ANTARMUKA (USER INTERFACE) PADA GAME EDUKASI*. Jurnal Imajinasi, 02(12), 57-64.

Wu, Bian & Wang, Alf Inge. 2012. A Guideline for Game Development-Based Learning: A Literature Review. International Journal of Computer Games Technology Volume 2012, Article ID 103710.

BIODATA PENULIS

BIODATA PENULIS

Penulis 1



Yeka Hendriyani, S.Kom., M.Kom. Lahir di Tungkal Kabupaten Bengkulu Selatan pada tanggal 20 Mei 1984. Menyelesaikan SD tahun 1996 di Tungkal Bengkulu Selatan, SLTP tahun 1999 di kota Bengkulu dan SLTA tahun 2002 di Pino Bengkulu Selatan. Kemudian menyelesaikan Sarjana Komputer di Universitas Putera Indonesia "YPTK" Padang tahun 2006 dan tahun 2009 menyelesaikan pendidikan S2 Ilmu Komputer di kampus yang sama. Diangkat menjadi dosen PNS tahun 2010 di Universitas Negeri Padang hingga sekarang. Buku pertama yang telah dipublikasikan adalah pemrograman Jaringannya dengan Java.

Penulis 2



Karmila Suryani lahir di Padang pada tanggal 28 April 1982. Menyelesaikan SD tahun 1994 di Pariaman, SLTP tahun 1997 di Pariaman dan SLTA tahun 200 di Bukittinggi. Gelar Sarjana Sistem Komputer diperoleh Tahun 2005 di Universitas Putera Indonesia “YPTK” Padang. Tahun 2010 memperoleh gelar Magister Teknologi Informasi di universitas yang sama. Staf pengajar di Program studi Pendidikan Teknologi Informasi dan Komputer di Fakultas Keguruan dan Ilmu Pendidikan Universitas Bung Hatta sejak tahun 2010 sampai sekarang.