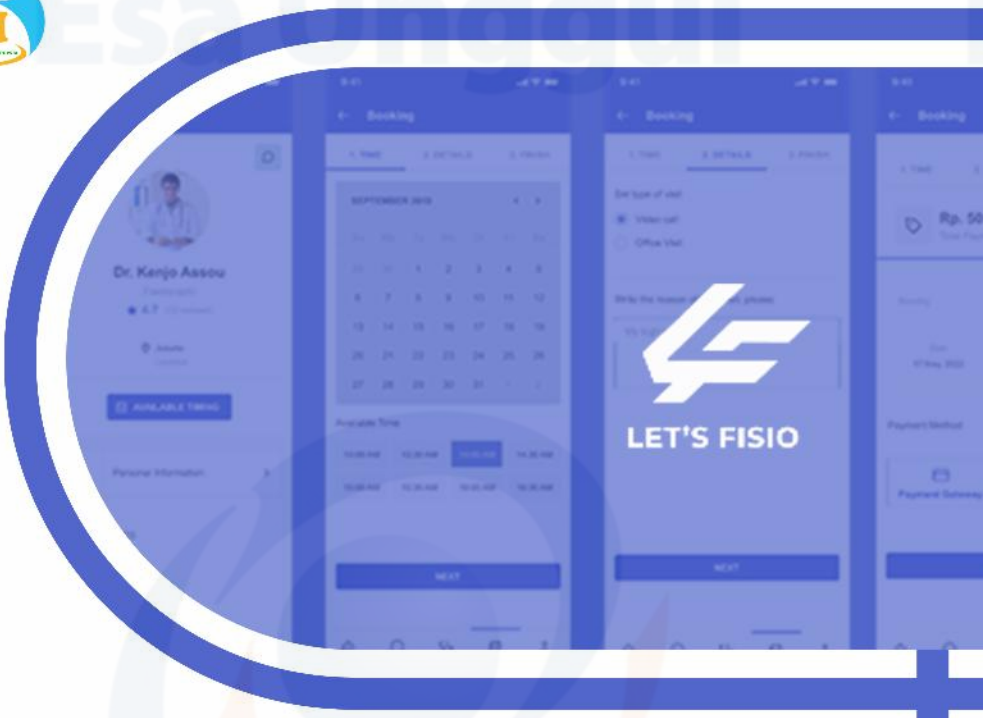


APLIKASI
LETSFISIO

Ir. Nizirwan Anwar, MT, MET, IPM.
Dr. (Cand) Dewanto Rosian Adhy, ST, MT.
Dr. Jerry Maratis, S.Ft, M.Fis.

APLIKASI LETSFISIO

Ir. Nizirwan Anwar, MT, MET, IPM.

Dr. (Cand) Dewanto Rosian Adhy, ST, MT.

Dr. Jerry Maratis, S.Ft, M.Fis.

Palembang © 2022, Aplikasi Letsfisio

Ir. Nizirwan Anwar, MT, MET, IPM.
Dr. (Cand) Dewanto Rosian Adhy, ST, MT.
Dr. Jerry Maratis, S.Ft, M.Fis.

Editor : Dr. Febrianty, SE, M.Si
Perancang Sampul : Jenri Ambarita, M.Pd.K.
Layouter : Rizki Amalia, A.Md, Ak

Diterbitkan oleh
Perkumpulan Rumah Cemerlang Indonesia
ANGGOTA IKAPI JAWA BARAT
Pondok Karisma Residence Jalan Raflesia VI D.151
Panglayungan, Cipedes Tasikmalaya – 085223186009

Website : www.rcipress.rcipublisher.org
E-mail : rumahcemerlangindonesia@gmail.com

Referensi | Non Fiksi | R/D. ; 15,5 x 23 cm

No ISBN : 978-623-448-393-2

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah semata yang telah memberikan sekian banyak nikmat dan karunia-Nya kepada penulis, hingga penulis tidak mampu untuk menghitungnya. Atas berkat nikmat dan karunia tersebut, penulis dapat menyelesaikan penulisan sebuah buku dengan judul "Letfisio".

Buku ini berisi Empat bab yakni terdiri atas bab pertama mengenai Platform Android, bab dua membahas mengenai trend platform sistem operasi android, bab ketiga yakni berisi tentang Aplikasi android studio. Bab Keempat yakni manual *User Interface* (UI) Aplikasi Let'sfisio.

Penulis menyadari bahwa buku ini masih terdapat kelemahan dan kekurangan. Oleh karena itu, saran dan kritik membangun akan diterima dengan tangan terbuka. Akhir kata, penulis berharap semoga buku ini dapat menjadi sebuah sumbangan yang berarti bagi ilmu pengetahuan dan bermanfaat bagi banyak pihak.

Jakarta, Desember 2022

Penulis

DAFTAR ISI

KATA PENGANTAR III
DAFTAR ISI IV
BAB I PLATFORM ANDROID 1
1.1 Sekilas Android 1
1.2 Sejarah Singkat 3
1.3 Arsitektur Frame-Work Android 9
BAB II TREND PLATFORM SISTEM OPERASI ANDROID 14
2.1 Perkembangan Android 14
2.1.1 Awal 14
2.1.2 Lanjutan 16
2.2 Evolusi Desain Android 20
2.3 Arsitektur Android 23
2.4 Binder IPC 26
2. 5 Dalvik 29
2.6 Proses Model Life-Cycle 32
2.7 Aplikasi Android 37
BAB III APLIKASI ANDROID STUDIO 39
3.1 Unduh Aplikasi Android Studio 39
3.2 Requirements 41
3.2.1 MacOS® 41
3.2.2 Windows 41
3.2.3 Linux 41
3.3 Kompatibilitas plugin Android Gradle 42
3.3.1 Mengupdate plugin Android Gradle 42
3.3.2 Mengupdate Gradle 43
3.4 Kompatiblitas 45
3.5 Probis Dasar Build Aplikasi Android 45
BAB IV MANUAL USER INTERFACE (UI) APLIKASI LET'SFISIO 47
DAFTAR PUSTAKA 61
GLOSARIUM 68
PROFIL PENULIS 72

BAB I
PLATFORM ANDROID

1.1 Sekilas Android

Android adalah sistem operasi seluler sumber terbuka yang terutama dibuat untuk perangkat seluler layar sentuh seperti smartphone dan tablet. Itu dibangun di atas versi modifikasi dari kernel Linux. Open Handset Alliance, sekelompok pengembang, membuat Android, yang didukung secara finansial oleh Google. HTC Dream, perangkat Android komersial pertama, memulai debutnya pada September 2008 setelah terungkap pada November 2007.

Sebagian besar versi Android adalah hak milik. Bagian utamanya berasal dari **Android Open Source Project (AOSP),** sebuah proyek perangkat lunak bebas dan sumber terbuka (**FOSS**) yang pada prinsipnya tercakup dalam Lisensi Apache. Android biasanya diinstal pada perangkat dengan batasan pada kemampuan untuk memodifikasi perangkat lunak bebas dan sumber terbuka, baik dengan tidak memberikan kode sumber yang cocok atau dengan memblokir penginstalan ulang melalui tindakan teknis, menjadikan versi yang diinstal sebagai hak milik. Sebagian besar smartphone Android sudah diinstal sebelumnya dengan perangkat lunak berpemilik tambahan[4], terutama Google Mobile Services (GMS), yang terdiri dari aplikasi penting seperti Google Chrome, toko online Google Play, dan pengembangan Layanan Google Play terkait. peron.

Ekosistem Google digunakan oleh lebih dari 70% smartphone yang diberdayakan Android, beberapa di antaranya juga menyertakan UI khusus vendor dan app store seperti TouchWiz dan selanjutnya One UI oleh Samsung dan HTC Sense[6]. Ekosistem dan garpu Android yang bersaing termasuk Fire OS yang dikembangkan Amazon, OPPO ColorOS, Vivo OriginOS, Honor MagicUI, dan

ROM khusus seperti LineageOS. Nama dan logo "Android", di sisi lain, adalah merek dagang Google, yang telah menetapkan kriteria untuk mencegah perangkat "tidak bersertifikat" menggunakan branding Android di luar ekosistemnya.

Dengan bantuan kode sumber, variasi Android telah dibuat untuk berbagai perangkat elektronik lainnya, termasuk konsol game, kamera digital, pemutar media portabel, dan PC. Masing-masing perangkat ini memiliki antarmuka pengguna yang unik. Dua versi terkenal buatan Google adalah Wear OS untuk perangkat yang dapat dikenakan dan Android TV untuk televisi. Paket perangkat lunak Android, yang menggunakan **format file APK**, biasanya dirilis melalui platform sumber tertutup seperti Aptoide atau F-Droid atau etalase sumber terbuka seperti Google Play Store, Amazon Appstore (yang menyertakan versi untuk Windows 11), Samsung Galaxy Store, Huawei AppGallery, dan GetJar.

Sejak 2011 dan sejak 2013, Android telah menjadi sistem operasi (OS) paling populer untuk smartphone dan tablet secara global. Basis terinstal terbesar dari semua sistem operasi per Mei 2021 mencakup lebih dari tiga miliar pengguna aktif bulanan, dan per Januari 2021, Google Play Store menawarkan lebih dari tiga juta aplikasi. Versi terbaru Android, Android 13, dirilis pada 15 Agustus 2022, dan Android 12.1/12L, yang baru saja diluncurkan, memiliki penyempurnaan yang dirancang khusus untuk ponsel lipat, tablet, layar berukuran desktop, dan Chromebook.

Gambar 1.1 Logo Android Pertama (2007–2014)

Gambar 1.2. Logo Android Kedua (2014–2015)

Gambar 1.3. Logo Android Ketiga (2015 – 2019)

Gambar 1.4. Logo Android Keempat (2019 – sekarang)

1.2 Sejarah Singkat

Andy Rubin , Rich Miner, Nick Sears, dan Chris White memulai Android Inc. di Palo Alto, California, pada Oktober 2003. Proyek Android, menurut Rubin, "memiliki janji besar untuk menciptakan perangkat seluler yang lebih cerdas yang lebih sadar akan lokasi dan preferensi pemiliknya." Tujuan awal perusahaan adalah untuk menciptakan sistem operasi mutakhir untuk kamera digital, dan ini berfungsi sebagai dasar penawarannya kepada investor pada bulan April 2004. Setelah menentukan bahwa pasar kamera tidak mencukupi untuk mencapai tujuannya, perusahaan mengalihkan fokusnya, dan lima bulan kemudian, ia mempromosikan Android sebagai sistem operasi handset yang akan bersaing dengan Symbian dan Microsoft Windows Mobile. Sejak awal, Rubin berjuang untuk mendapatkan pendukung, dan Android akan diusir dari ruang kantornya. Seorang teman

dekat Rubin bernama Steve Perlman mengiriminya uang tunai $10.000 dalam sebuah amplop, dan tak lama setelah itu, dia mengirimkan sejumlah uang lain sebagai uang awal. Perlman mengungkapkan bahwa dia menolak saham dalam bisnis tersebut karena dia "percaya pada produknya" dan ingin mendukung Andy.

Rubin berusaha untuk mencapai kesepakatan dengan Samsung dan HTC pada tahun 2005. Tak lama setelah itu, Google membeli bisnis tersebut pada bulan Juli tahun itu dengan harga sedikitnya $50 juta. David Lawee , wakil presiden pengembangan perusahaan Google saat itu , menyatakan pada tahun 2010 bahwa ini adalah "penawaran terbesar yang pernah dilakukan perusahaan sebagai bagian dari akuisisi, karyawan kunci Android Rubin, Miner, Sears, dan White bergabung dengan Google.

Pada saat itu, tidak ada yang diketahui tentang Android Inc. yang tertutup; perusahaan hanya menyatakan bahwa mereka memproduksi perangkat lunak untuk perangkat seluler. Tim yang dipimpin Rubin di Google menciptakan platform perangkat seluler berbasis kernel Linux. Google mengiklankan platform tersebut kepada operator dan produsen perangkat sebagai sistem yang fleksibel dan dapat ditingkatkan. Google telah "mengumpulkan sejumlah mitra untuk perangkat lunak dan perangkat keras dan memberi tahu operator bahwa mereka terbuka untuk berbagai tingkat kerja sama".

Hingga Desember 2006, spekulasi tentang niat Google untuk memasuki pasar komunikasi seluler berkembang. Tanpa layar sentuh dan keyboard QWERTY fisik, prototipe awal Android sangat mirip dengan ponsel BlackBerry. Namun, ketika Apple iPhone memulai debutnya pada tahun

2007, Android "harus kembali ke papan gambar". Terlepas dari kenyataan bahwa "Produk dikembangkan dengan adanya tombol fisik diskrit sebagai asumsi, sehingga layar sentuh tidak dapat sepenuhnya menggantikan tombol fisik," Google kemudian mengubah spesifikasi spesifikasi Android mereka untuk mengklarifikasi bahwa "Layar sentuh akan didukung". IPhone 3G ditantang oleh smartphone berbasis sentuh pada tahun 2008, dan Android akhirnya mempersempit fokusnya menjadi hanya layar sentuh. HTC Dream, sebelumnya dikenal sebagai T-Mobile G1, diresmikan pada tanggal 23 September 2008, dan merupakan ponsel cerdas bertenaga Android pertama yang dapat diakses public.

Perangkat Android pertama yang dirilis secara resmi adalah HTC Dream atau T-Mobile G1 (2008) Pada tanggal 5 November 2007, Open Handset Alliance, sekelompok perusahaan teknologi dengan misi menciptakan "platform pertama yang benar-benar terbuka dan komprehensif untuk perangkat seluler," diperkenalkan. Anggota aliansi termasuk Google, HTC, Motorola, dan Samsung, serta penyedia layanan nirkabel Sprint dan T-Mobile. Symbian Foundation dan LiMo Foundation, yang terakhir juga bekerja pada sistem operasi seluler berbasis Linux seperti Google, menghadirkan Open Handset Alliance dengan dua pesaing open source tambahan dalam setahun. Google telah mengajukan sejumlah aplikasi paten di bidang telepon seluler, menurut sebuah studi oleh Evalueserve yang diterbitkan InformationWeek pada September 2007.

Sejak tahun 2008, Android telah menerima sejumlah pembaruan yang secara bertahap menyempurnakan sistem operasi dengan memperkenalkan kemampuan baru dan

mengatasi kekurangan pada versi sebelumnya. Beberapa versi Android pertama diberi nama "Cupcake", "Donut", "Eclair", dan "Froyo", dalam urutan itu. Setiap rilis utama diberi nama setelah makanan manis atau makanan penutup. Setiap versi Android diberi nama makanan penutup, klaim Google saat mengumumkan Android KitKat pada 2013; namun, perwakilan Google memberi tahu CNN dalam sebuah wawancara bahwa "Ini seperti masalah tim internal, dan kami lebih suka sedikit - bagaimana saya harus mengatakan-sedikit tidak dapat dipahami dalam masalah ini". Untuk membuat perangkat baru dan menyediakan versi Android yang diperbarui, Google bekerja sama dengan berbagai produsen perangkat untuk membuat rangkaian gadget Nexus-nya pada tahun 2010. Seri ini terkenal karena perangkat lunaknya yang "bebas gembung" dan "pembaruan... tepat waktu", dan dikreditkan dengan "memainkan bagian penting dalam sejarah Android dengan memperkenalkan iterasi perangkat lunak baru dan standar perangkat keras secara menyeluruh". Pada konferensi pengembangnya pada Mei 2013, Google meluncurkan model Samsung Galaxy S4 yang disesuaikan, menjanjikan pembaruan sistem yang cepat dan perangkat lunak "stok Android" daripada penyesuaian Android milik Samsung sendiri. Handset akan meluncurkan program edisi Google Play, dan lebih banyak perangkat, seperti edisi HTC One Google Play dan edisi Google Play Moto G, akan tiba setelahnya. Ponsel Android edisi Google Play terakhir di etalase online Google terdaftar sebagai "tidak lagi tersedia untuk dijual" awal pekan ini. Sekarang setelah semuanya hilang, tampaknya program tersebut telah selesai, menurut artikel tahun 2015 oleh Ars Technica. Hugo Barra, Andy

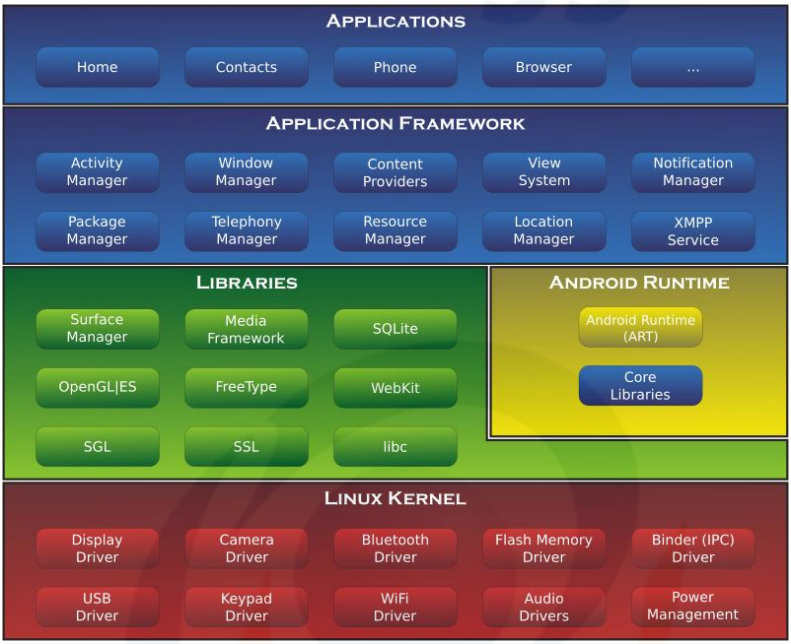
Rubin, dan Eric Schmidt pada konferensi pers tahun 2012 di mana mereka mengumumkan tablet Google Nexus 7 Hugo Barra mewakili Android sebagai juru bicara produknya dari tahun 2008 hingga 2013, menghadiri acara pers dan Google I/O, konferensi pengembang tahunan perusahaan. Pada Agustus 2013, dia keluar dari Google untuk bergabung dengan produsen smartphone Cina Xiaomi. Kurang dari enam bulan sebelumnya, CEO Google saat itu Larry Page mengungkapkan dalam sebuah posting blog bahwa Sundar Pichai akan mengambil alih sebagai pemimpin Android baru dan Andy Rubin telah meninggalkan divisi Android untuk mengerjakan proyek baru di Google. Setelah Google direorganisasi menjadi konglomerat Alfabet pada Agustus 2015. Pichai sendiri akhirnya berganti peran, menjadi CEO baru perusahaan dan menunjuk Hiroshi Lockheimer sebagai kepala baru Android. Pada Android 4.4 Kit Kat, hanya direktori khusus dengan nama paket yang sesuai, ditemukan di dalam Android/data/, tetap dapat ditulisi ke kartu memori MicroSD untuk program yang diinstal pengguna. Dengan Android 5 Lollipop, akses menulis telah dipulihkan melalui antarmuka Google Storage Access Framework, yang tidak kompatibel dengan versi Android sebelumnya. Untuk pengguna di negara berkembang, Google meluncurkan Android One pada Juni 2014 sebagai kumpulan "model referensi perangkat keras" yang akan "memungkinkan [pembuat perangkat] membuat ponsel berkualitas tinggi dengan harga murah dengan cepat." Google mengungkapkan batch pertama ponsel Android One yang akan tersedia di India pada bulan September. Proyek ini, bagaimanapun, dianggap " kekecewaan " oleh Recode pada bulan Juni 2015 karena "konsumen dan mitra manufaktur yang enggan" dan

"macet dari perusahaan pencari yang tidak pernah memahami perangkat keras sepenuhnya". Gagasan untuk menghidupkan kembali Android One pertama kali muncul pada Agustus 2015, dan seminggu kemudian terungkap bahwa Afrika akan menjadi benua operasi program berikutnya. Google dilaporkan memperluas inisiatif Android One yang terjangkau ke Amerika Serikat, menurut laporan dari The Information pada Januari 2017, sementara The Verge menunjukkan bahwa perusahaan kemungkinan tidak akan membuat perangkat yang sebenarnya. Pada bulan Oktober 2016, Google meluncurkan smartphone Pixel dan Pixel XL, yang diiklankan sebagai perangkat pertama perusahaan dan memiliki rilis terbatas beberapa fitur perangkat lunak, termasuk Asisten Google. Generasi baru ponsel Pixel diperkenalkan pada Oktober 2017 untuk menggantikan seri Nexus. Sistem operasi tersebut terlibat dalam konflik perdagangan antara China dan AS termasuk Huawei pada Mei 2019. Huawei, seperti banyak perusahaan IT lainnya, semakin bergantung pada akses ke platform Android. Huawei mengatakan pada musim panas 2019 bahwa mereka akan mengembangkan Harmony OS, sistem operasi yang bersaing dengan Android, dan telah mengajukan hak kekayaan intelektual di beberapa pasar internasional yang penting. Karena Harmony OS pada awalnya dibuat untuk perangkat internet of things daripada smartphone dan tablet, Huawei memiliki niat jangka panjang untuk mengganti Android pada tahun 2022 dengan sistem operasi baru. Android "Q" akan secara resmi dipasarkan sebagai Android 10, diumumkan pada 22 Agustus 2019, secara resmi menghilangkan praktik historis penamaan versi utama setelah makanan penutup. Nama-nama ini tidak

"inklusif", menurut Google, untuk pengguna dari negara lain (karena makanan yang disebutkan di atas tidak dikenal secara internasional, atau sulit diucapkan dalam beberapa Bahasa. Pada hari yang sama, Android Police mengungkapkan bahwa Google telah membayar pemasangan patung "10" besar di lobi kantor baru pengembang. Pada 3 September 2019, ponsel Google Pixel menjadi yang pertama menerima Android 10. Beberapa pelanggan melaporkan mengalami masalah saat menelepon layanan darurat pada paruh kedua tahun 2021. Masalah ini disebabkan oleh kumpulan kesalahan dalam perangkat lunak Microsoft Teams dan sistem operasi Android; kedua perusahaan mengeluarkan rekomendasi untuk memperbaiki masalah.

1.3 Arsitektur Frame-Work Android

Beberapa pakar google juga menyebut arsitektur android sebagai stack android (Tumpukan). Bisa disebutkan sebagai stack karena aplikasi android memiliki berbagai lapisan didalamnya serta memiliki tugasnya masing- masing.



Gambar 5. Arsitektur Android
(http://www.eazytutz.comandroidandroid-architecture)

Arsitektur Android adalah lapisan komponen software untuk memenuhi kebutuhan perangkat mobile. Terdapat 5 (lima) lapisan utama dalam arsitektur android antara lain yakni Application, Android Framework, Android Runtime, Platform Library, dan Linux Kernel.

a. Application

   Urutan dari Android Architecture yang teratas ditempati oleh Apllication. userdapat menulis aplikasinya untuk diinstal pada lapisan ini saja. Layer Application berjalan dengan Android runtime yang menggunakan class-class dan services yang tersedia dari application framework.

b. Android Framework.

   Layer Android Framewrk menyediakan class dan interface untuk Pengembangan aplikasi Android dan services ke aplikasi dalam bentuk class Java. layanan telepon, layanan lokasi, manajer notifikasi, layanan NFC, sistem tampilan, merupakan service yang terdapat didalam Android Framework.

c. Android Runtime

   Android Runtime merupakan lapisan ketiga dari Android Architecture. Android Runtime adalah mesin yang menggerakkan aplikasi kita bersama dengan libraries dan membentuk dasar untuk framework aplikasi. Pada Android Runtime menyediakan komponen utama yaitu Dalvik Virtual Machine (DVM) yang merupakan sejenis Mesin Virtual Java yang dirancang dan dioptimalkan khusus untuk Android.

d. Platform Libraries

   Platform Libraries mencangkup berbagai macam C/C++ core libraries dan Java-based libraries seperti SSL, libc, Graphics, SQLite, Webkit, Media, Surface Manger, OpenGL. Libraries juga bertanggung jawab untuk memutar dan merekam format audio dan video, FreeType untuk dukungan font, WebKit untuk dukungan browser, SQLite untuk database, SSL untuk keamanan Internet, dll.

e. Linux Kernel

   Kernel Linux adalah komponen terpenting di Android yang memiliki tugas menyediakan fungsionalitas sistem operasi ke mobile yang bertanggung jawab atas aplikasi ini. Kernel mempunyai beberapa tugas yaitu bertanggung jawab untuk berbagai driver

perangkat seperti driver kamera/tampilan/ Bluetooth/keypad, Manajemen memori, Manajemen proses, Manajemen daya, dll. Kernel juga menangani segala sesuatu di Linux yang terkait dengan jaringan, driver perangkat, dan membantu interface agar berjalan dengan benar.

Dalam Android Runtime terdapat 2 (dua) bagian utama, diantaranya:

a. Core Libraries
   Android dikembangkan melalui bahasa pemrograman Java, tetapi Android Runtime bukanlah mesin virtual Java dan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus android. Core Libraries biasa dikatakan sebagai sebuah kamus,yang berfungsi sebagai penerjemah (interpreter) bahasa Java/C.
b. Mesin Virtual Dalvik
   Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format Dalvik Executable (*.dex). Dengan formatini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatanm emori pada file yang dieksekusi.

Libraries merupakan layer tempat fitur-fitur Android berada. Pada umumnya libraries di-akses untuk menjalankan aplikasi. Android menggunakan beberapa paket pustakayang terdapat pada C/C++ dengan standar Berkeley Software Distribution (BSD) hanya setengah dari

yang aslinya untk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

1) Media Library untuk memutar dan merekam berbagai macam format audio dan video,
2) Surface Manager untuk mengatur hak akses layer dari berbagai aplikasi,
3) Graphic Library termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan3D,
4) SQLite untuk mengatur relasi database yang digunakan pada aplikasi,
5) SSl dan WebKit untuk browser dan keamanan internet.

BAB II
TREND PLATFORM SISTEM OPERASI ANDROID

2.1 Perkembangan Android

2.1.1 Awal

Android, Inc. adalah perusahaan perangkat lunak yang didirikan untuk membuat perangkat lunak guna membuat perangkat seluler yang lebih cerdas. Awalnya melihat kamera, visi segera beralih ke smartphone karena potensi pasarnya yang lebih besar. Tujuan awal tersebut berkembang untuk mengatasi kesulitan saat ini dalam mengembangkan perangkat seluler, dengan menghadirkan platform terbuka yang dibangun di atas Linux yang dapat digunakan secara luas. Selama periode ini, prototipe untuk antarmuka pengguna platform diimplementasikan untuk mendemonstrasikan ide di baliknya. Platform itu sendiri menargetkan tiga bahasa utama, JavaScript, Java, dan C++, untuk mendukung lingkungan pengembangan aplikasi yang kaya. Google mengakuisisi Android pada Juli 2005, menyediakan sumber daya yang diperlukan dan dukungan layanan cloud untuk melanjutkan pengembangan Android sebagai produk lengkap. Sekelompok kecil para pakar bekerja sama secara intensif dan konstruktif serta ber-inovasi, mulai mengembangkan infrastruktur inti untuk platform dan fondasi untuk pengembangan aplikasi tingkat tinggi.

Pada awal tahun 2006, terjadi perubahan rencana yang signifikan: alih-alih mendukung banyak bahasa pemrograman, platform akan berfokus sepenuhnya pada bahasa pemrograman Java untuk pengembangan

aplikasinya. Ini adalah perubahan yang sulit, karena pendekatan multi-bahasa asli lo-level language membuat semua orang senang dengan "yang terbaik dari semua dunia"; berfokus pada satu bahasa terasa seperti langkah mundur bagi para insinyur yang lebih menyukai bahasa lain. Namun, mencoba membuat semua orang bahagia, dapat dengan mudah membuat tidak ada yang bahagia. Membangun 3 (tiga) set API bahasa yang berbeda akan membutuhkan lebih banyak usaha daripada berfokus pada satu bahasa, sangat mengurangi kualitas masing-masing. Keputusan untuk fokus pada bahasa Java sangat penting untuk kualitas akhir dari platform dan kemampuan tim pengembangan untuk memenuhi tenggat waktu yang penting.

Seiring kemajuan pengembangan, platform Android dikembangkan erat dengan aplikasi yang pada akhirnya akan dikirimkan di atasnya. Google sudah memiliki berbagai macam layanan - termasuk Gmail, Maps, Kalender, YouTube, dan tentu saja pencarian- yang akan dihadirkan dalam platform Android. Pengetahuan yang diperoleh dari penerapan aplikasi ini di atas platform awal dimasukkan kembali ke dalam desainnya. Proses berulang dengan aplikasi ini memungkinkan banyak kelemahan desain pada platform untuk rekonstruksi di awal pengembangannya.

Sebagian besar pengembangan aplikasi awal dilakukan dengan sedikit platform dasar yang benar-benar tersedia untuk para pengembang. Platform ini biasanya berjalan dalam satu proses, melalui "simulator" yang menjalankan semua sistem dan aplikasi sebagai proses tunggal pada komputer host. Nyatanya masih ada sisa-sisa dari implementasi lama ini hingga saat ini, dengan hal-hal seperti

metode Applica tion.onTerminate masih ada di SDK (Software Dev elopment Kit), yang digunakan programmer Android untuk menulis (console) aplikasi.

Pada bulan Juni 2006, dua perangkat keras dipilih sebagai target pengembangan perangkat lunak untuk produk yang direncanakan. Yang pertama, dengan nama kode "Sooner", didasarkan pada smartphone yang sudah ada dengan keyboard QWERTY dan layar tanpa input sentuh. Tujuan perangkat ini adalah mengeluarkan produk awal secepat mungkin, dengan memanfaatkan perangkat keras yang ada. Perangkat target kedua, dengan nama kode "Mimpi", dirancang khusus untuk Android, untuk menjalankannya seperti yang dibayangkan sepenuhnya. Itu termasuk layar sentuh besar (untuk waktu itu), keyboard QWERTY slide-out, radio 3G (untuk penjelajahan laman yang lebih cepat), akselerometer, GPS dan kompas (untuk mendukung Google Maps), dll.

Saat jadwal perangkat lunak menjadi lebih fokus, menjadi jelas bahwa kedua jadwal perangkat keras tidak masuk akal. Pada saat dimungkinkan untuk merilis Sooner, perangkat keras itu sudah ketinggalan zaman, dan upaya yang dilakukan. Sooner mendorong keluar perangkat Dream yang lebih penting. Untuk mengatasi hal ini, diputuskan untuk menghentikan Sooner sebagai perangkat target (meskipun pengembangan pada perangkat keras tersebut berlanjut selama beberapa waktu hingga perangkat keras yang lebih baru siap) dan berfokus sepenuhnya pada Dream.

2.1.2 Lanjutan

Ketersediaan publik pertama platform Android adalah pratinjau SDK yang dirilis pada November 2007. Ini terdiri

dari emulator perangkat keras yang menjalankan image sistem perangkat Android lengkap dan aplikasi inti, dokumentasi API, dan lingkungan pengembangan. Pada titik ini desain dan implementasi inti sudah ada, dan dalam banyak hal sangat mirip dengan arsitektur sistem Android modern yang akan kita diskusikan. Pengumuman tersebut mencakup demo video dari bentuk plat yang berjalan di atas perangkat keras Sooner dan Dream.

Pengembangan awal Android telah dilakukan di bawah serangkaian tonggak demo tri-wulanan untuk mendorong dan menunjukkan proses yang berkelanjutan. Rilis SDK adalah rilis pertama yang lebih formal untuk platform tersebut. Dibutuhkan mengambil semua bagian yang telah disatukan sejauh ini untuk pengembangan aplikasi, membersihkannya, mendokumentasikannya, dan menciptakan lingkungan pengembangan yang kohesif untuk pengembang pihak ketiga.

Pengembangan sekarang berjalan dalam dua jalur: menerima umpan balik tentang SDK untuk menyempurnakan dan menyelesaikan API lebih lanjut, serta menyelesaikan dan menstabilkan implementasi yang diperlukan untuk mengirimkan perangkat Dream. Sejumlah pemutakhiran publik ke SDK terjadi selama ini, yang berpuncak pada rilis 0.9 pada Agustus 2008 yang berisi API yang hampir final.

Platform itu sendiri telah mengalami perkembangan pesat, dan pada musim semi tahun 2008 fokusnya beralih ke stabilisasi sehingga Dream dapat diluncurkan. Android pada saat ini berisi sejumlah besar kode yang belum pernah dikirimkan sebagai produk komersial, mulai dari bagian

pustaka C, hingga penerjemah Dalvik (yang menjalankan aplikasi), sistem, dan aplikasi.

Android juga berisi beberapa ide desain baru yang belum pernah dilakukan sebelumnya, dan tidak jelas bagaimana ide tersebut akan berjalan dengan baik. Ini semua perlu disatukan sebagai produk yang stabil, dan tim menghabiskan beberapa bulan yang menegangkan untuk bertanya- tanya apakah semua hal ini benar-benar akan bersatu dan berfungsi sebagaimana mestinya. Akhirnya, pada bulan Agustus 2008, perangkat lunak sudah stabil dan siap dikirim. Build masuk ke pabrik dan mulai di-flash ke perangkat. Pada bulan September Android 1.0 diluncurkan pada perangkat Dream, sekarang disebut T-Mobile G1. Setelah rilis Android 1.0, pengembangan berlanjut dengan sangat cepat. Terdapat sekitar 15 pembaruan besar pada platform selama 5 tahun berikutnya, menambahkan berbagai macam fitur dan peningkatan baru dari rilis awal 1.0.

Dokumen Definisi Kompatibilitas asli pada dasarnya hanya diperbolehkan untuk perangkat yang kompatibel yang sangat mirip dengan T-Mobile G1. Selama tahun-tahun berikutnya, jangkauan perangkat yang kompatibel akan sangat berkembang. Poin-poin penting dari proses ini adalah:

Selama tahun 2009, Android versi 1.5 hingga 2.0 memperkenalkan keyboard lunak untuk menghapus persyaratan keyboard fisik, dukungan layar yang jauh lebih luas (baik ukuran dan kerapatan piksel) untuk perangkat QVGA kelas bawah dan perangkat baru yang lebih besar dan berkepadatan lebih tinggi seperti WVGA Motorola Droid, dan fasilitas "fitur sistem" baru bagi perangkat untuk

melaporkan fitur perangkat keras apa yang mereka dukung dan aplikasi untuk menunjukkan fitur perangkat keras mana yang mereka perlukan. Yang terakhir adalah mekanisme utama yang digunakan Google Play untuk menentukan kompatibilitas aplikasi dengan perangkat tertentu.

Selama tahun 2011, Android versi 3.0 hingga 4.0 memperkenalkan dukungan inti baru di platform untuk tablet 10 inci dan lebih besar; bentuk plat inti sekarang sepenuhnya mendukung ukuran layar perangkat di mana saja mulai dari ponsel QVGA kecil, hingga smartphone dan phablet yang lebih besar, tablet 7 inci dan tablet yang lebih besar hingga lebih dari 10 inci.

Karena platform menyediakan dukungan bawaan untuk perangkat keras yang lebih beragam, tidak hanya layar yang lebih besar tetapi juga perangkat non-sentuh dengan atau tanpa mouse, semakin banyak jenis perangkat Android yang muncul. Ini termasuk perangkat TV seperti Google TV, perangkat game, notebook, kamera, dll. Pekerjaan pengembangan yang signifikan juga mengarah pada sesuatu yang tidak terlihat: pemisahan yang lebih bersih antara layanan milik Google dari platform sumber terbuka Android.

Untuk Android 1.0, pekerjaan signifikan telah dilakukan untuk memiliki API aplikasi pihak ketiga yang bersih dan platform sumber terbuka tanpa ketergantungan pada hak milik. Kode Google. Namun, penerapan kode hak milik Google seringkali belum dibersihkan, memiliki ketergantungan pada bagian internal platform. Seringkali platform tersebut bahkan tidak memiliki fasilitas yang diperlukan oleh kode hak milik Google agar dapat terintegrasi dengan baik. Serangkaian proyek segera dilakukan untuk mengatasi masalah ini:

Pada tahun 2009, Android versi 2.0 memperkenalkan arsitektur untuk pihak ketiga untuk menyambungkan adaptor sinkronisasi mereka sendiri ke API platform seperti database kontak. Kode Google untuk menyinkronkan berbagai data dipindahkan ke API SDK yang terdefinisi dengan baik ini.

Pada tahun 2010, Android versi 2.2 menyertakan pekerjaan pada desain internal dan penerapan kode hak milik Google. "Unbundling hebat" ini menerapkan banyak layanan inti Google dengan rapi, mulai dari memberikan pembaruan perangkat lunak sistem berbasis cloud hingga "perpesanan cloud-ke-perangkat" dan layanan latar belakang lainnya, sehingga dapat dikirimkan dan diperbarui secara terpisah dari peron.

Pada tahun 2012, aplikasi layanan Google Play baru dikirimkan ke perangkat, berisi fitur yang diperbarui dan baru untuk layanan nonaplikasi milik Google. Ini adalah hasil dari pekerjaan unbundling pada tahun 2010, yang memungkinkan API berpemilik seperti pengiriman pesan cloud-ke-perangkat dan peta dikirimkan dan diperbarui sepenuhnya oleh Google.

2.2 Evolusi Desain Android

Sejumlah tujuan desain utama untuk platform Android berkembang selama pengembangannya:

Menyediakan platform sumber terbuka yang lengkap untuk perangkat seluler. Bagian open-source Android adalah tumpukan sistem operasi dari bawah ke atas, termasuk berbagai aplikasi, yang dapat dikirimkan sebagai produk yang lengkap.

Sangat mendukung aplikasi pihak ketiga berpemilik dengan API yang kuat dan stabil. Seperti yang telah dibahas sebelumnya, mempertahankan platform yang benar-benar open-source dan juga cukup stabil untuk aplikasi pihak ketiga berpemilik merupakan tantangan tersendiri. Android menggunakan perpaduan solusi teknis (menentukan SDK yang terdefinisi dengan sangat baik dan pembagian antara API publik dan implementasi internal) dan persyaratan kebijakan (melalui CDD) untuk mengatasi hal ini.

Izinkan semua aplikasi pihak ketiga, termasuk dari Google, untuk bersaing di arena permainan yang setara. Kode sumber terbuka Android adalah dirancang senetral mungkin untuk fitur sistem tingkat tinggi yang dibangun di atasnya, dari akses ke layanan cloud (seperti sinkronisasi data atau API perpesanan cloud-ke-perangkat), hingga perpustakaan (seperti perpustakaan pemetaan Google) dan layanan seperti toko aplikasi.

Sediakan model keamanan aplikasi di mana pengguna tidak perlu terlalu mempercayai aplikasi pihak ketiga. Sistem operasi harus melindungi pengguna dari perilaku buruk aplikasi, tidak hanya aplikasi buggy yang dapat menyebabkannya macet, tetapi penyalahgunaan perangkat dan data pengguna yang lebih halus di dalamnya. Semakin sedikit pengguna yang perlu mempercayai aplikasi, semakin banyak kebebasan yang mereka miliki untuk mencoba dan menginstalnya.

Mendukung interaksi pengguna seluler biasa: menghabiskan waktu singkat di banyak aplikasi. Pengalaman seluler cenderung melibatkan interaksi singkat dengan aplikasi: melihat email baru yang diterima, menerima dan mengirim pesan SMS atau IM, membuka

kontak untuk melakukan panggilan, dll. Sistem perlu dioptimalkan untuk kasus ini dengan peluncuran aplikasi yang cepat dan beralih kali; tujuan

untuk Android umumnya adalah 200 mdtk untuk memulai aplikasi dasar hingga menampilkan UI interaktif penuh.

Mengelola proses aplikasi untuk pengguna, menyederhanakan pengalaman pengguna seputar aplikasi sehingga pengguna tidak perlu khawatir menutup aplikasi setelah selesai menggunakannya. Perangkat seluler juga cenderung berjalan tanpa ruang swap yang memungkinkan sistem operasi gagal dengan lebih mulus ketika kumpulan aplikasi yang berjalan saat ini membutuhkan lebih banyak RAM daripada yang tersedia secara fisik. Untuk mengatasi kedua persyaratan ini, sistem perlu mengambil sikap yang lebih proaktif dalam mengelola proses dan memutuskan kapan harus dimulai dan dihentikan.

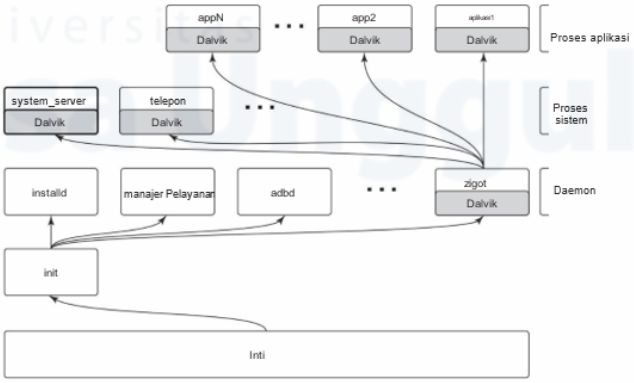
Dorong aplikasi untuk saling beroperasi dan berkolaborasi dengan cara yang kaya dan aman. Aplikasi seluler dalam beberapa hal adalah kembali ke perintah shell: daripada desain aplikasi desktop monolitik yang semakin besar, mereka ditargetkan dan difokuskan untuk kebutuhan khusus. Untuk membantu mendukung ini, sistem operasi harus menyediakan jenis fasilitas baru untuk aplikasi ini untuk berkolaborasi bersama untuk membuat keseluruhan yang lebih besar.

Buat sistem operasi serba guna yang lengkap. Perangkat seluler adalah ekspresi baru dari komputasi tujuan umum, bukan sesuatu yang lebih sederhana daripada sistem operasi desktop tradisional kami. Rancangan Android harus cukup

kaya sehingga dapat tumbuh setidaknya mampu sebagai sistem operasi tradisional..

2.3 Arsitektur Android

Android dibangun di atas kernel Linux standar, dengan hanya beberapa ekstensi signifikan pada kernel itu sendiri yang akan dibahas nanti. Namun, begitu berada di ruang pengguna, implementasinya sangat berbeda dari distribusi Linux tradisional dan menggunakan banyak fitur Linux yang sudah Anda pahami dengan cara yang sangat berbeda. Dalam sistem Linux tradisional, proses ruang pengguna pertama Android adalah init, yang merupakan akar dari semua proses lainnya. Proses init daemon Android dimulai berbeda, namun, lebih berfokus pada detail tingkat rendah (mengelola sistem file dan akses perangkat keras) daripada fasilitas pengguna tingkat tinggi seperti menjadwalkan pekerjaan cron. Android juga memiliki lapisan proses tambahan, yang menjalankan lingkungan bahasa Java Dalvik, yang bertanggung jawab untuk mengeksekusi semua bagian sistem yang diimplementasikan di Java.

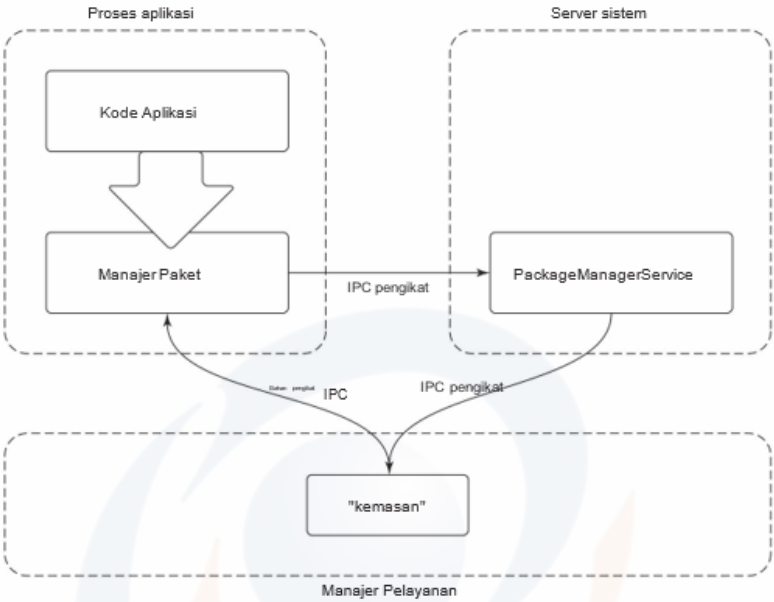


Gambar 2.1 Hirarki proses Android

Gambar 2.1 diatas, mengilustrasikan struktur proses dasar Android. Pertama adalah proses init, yang memunculkan sejumlah proses daemon tingkat rendah. Salah satunya adalah zigot, yang merupakan akar dari proses bahasa Java tingkat tinggi. Init Android tidak menjalankan shell dengan cara tradisional, karena perangkat Android pada umumnya tidak memiliki konsol lokal untuk akses shell. Alih-alih, proses daemon adbd mendengarkan koneksi jarak jauh (seperti melalui USB) yang meminta akses shell, membagi proses shell untuk mereka sesuai kebutuhan.

Karena sebagian besar Android ditulis dalam bahasa Java, daemon zyangote dan proses yang dimulainya merupakan inti dari sistem. Proses pertama zigot selalu dimulai disebut server sistem , yang berisi semua layanan inti sistem operasi. Bagian penting dari ini adalah power manager, package manager, window manager, dan activity manager. Proses lain akan dibuat dari zigot sesuai kebutuhan. Beberapa di antaranya adalah proses "terus-menerus" yang merupakan bagian dari sistem operasi dasar, seperti tumpukan telepon dalam proses telepon, yang harus selalu berjalan. Proses aplikasi tambahan akan dibuat dan dihentikan sesuai kebutuhan saat sistem sedang berjalan.

Aplikasi berinteraksi dengan sistem operasi melalui panggilan ke perpustakaan yang disediakan olehnya, yang bersama-sama menyusun kerangka kerja Android. Beberapa perpustakaan ini dapat melakukan pekerjaan mereka dalam proses itu, tetapi banyak yang perlu melakukan komunikasi antarproses dengan proses lain, seringkali layanan dalam proses server sistem.



Gambar 2.2 Proses Aplikasi berinteraksi Manajemen Layanan

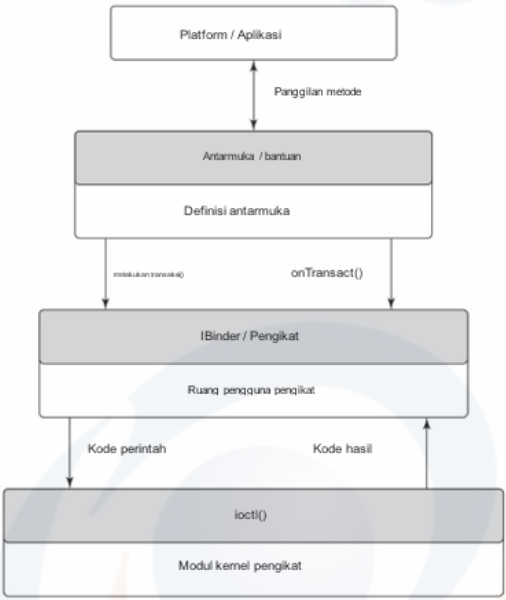
Gambar 2.2 menunjukkan desain umum API kerangka kerja Android yang berinteraksi dengan layanan sistem, dalam hal ini pengelola paket. Manajer paket menyediakan kerangka kerja API untuk memanggil aplikasi dalam proses lokal mereka, di sini kelas PackageManager . Secara internal, kelas ini harus mendapatkan koneksi ke layanan corres ponding di server sistem. Untuk melakukannya, saat boot server sistem menerbitkan setiap layanan dengan nama yang terdefinisi dengan baik di manajer layanan, sebuah daemon yang dijalankan oleh init. PackageManager dalam proses aplikasi mengambil koneksi dari manajer layanan ke layanan sistemnya dengan menggunakan feature yang sama. Setelah PackageManager terhubung dengan layanan

sistemnya, ia dapat melakukan panggilan padanya. Sebagian besar panggilan aplikasi ke Package Manager diimplementasikan sebagai komunikasi antar proses menggunakan mekanisme **Binder IPC Android**, dalam hal ini melakukan panggilan (call) ke implementasi Package Manager Service di server sistem. PackageManagerService menengahi interaksi di semua aplikasi klien dan mempertahankan status yang akan dibutuhkan oleh beberapa aplikasi.

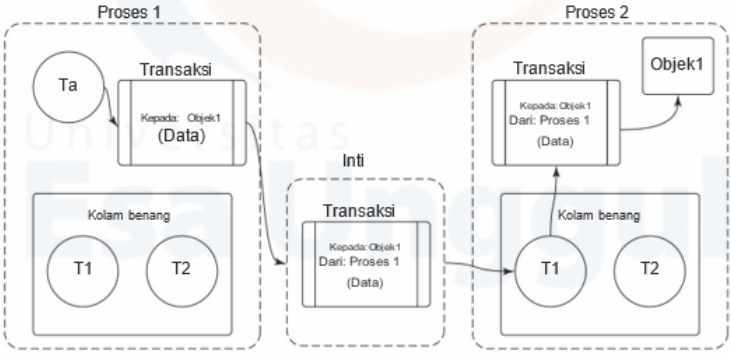
2.4 Binder IPC

Desain sistem Android berputar secara signifikan di sekitar isolasi proses, baik di antara aplikasi maupun di antara berbagai bagian sistem itu sendiri. Ini membutuhkan sejumlah besar komunikasi antarproses untuk mengoordinasikan antara proses yang berbeda, yang dapat membutuhkan banyak pekerjaan untuk diterapkan dan diperoleh. Baik. Mekanisme komunikasi antarproses Binder Android adalah fasilitas IPC tujuan umum yang kaya di mana sebagian besar sistem Android dibangun di atasnya. Arsitektur Binder dibagi menjadi tiga lapisan, ditunjukkan pada Gambar. 7. Di bagian bawah tumpukan adalah modul kernel yang mengimplementasikan interaksi lintas proses aktual dan memaparkannya melalui fungsi ioctl kernel . (ioctl adalah panggilan kernel serba guna untuk mengirim perintah khusus ke driver dan modul kernel.) Di atas modul kernel adalah API ruang pengguna berorientasi objek dasar, memungkinkan aplikasi untuk membuat dan berinteraksi dengan titik akhir IPC melalui kelas IBinder dan Binder . Di bagian atas adalah model pemrograman berbasis antarmuka di mana aplikasi mendeklarasikan antarmuka IPC mereka dan sebaliknya tidak perlu khawatir tentang detail

bagaimana IPC terjadi di lapisan bawah, Modul Kernel Binder Daripada menggunakan fasilitas IPC Linux yang sudah ada seperti pipa, Binder menyertakan modul kernel khusus yang mengimplementasikan mekanisme IPC-nya sendiri. Model Binder IPC cukup berbeda dari mekanisme Linux tradisional sehingga tidak dapat diimplementasikan secara efisien di atasnya hanya di ruang pengguna. Selain itu, Android tidak mendukung sebagian besar primitif System V untuk interaksi lintas proses (semafor, segmen memori bersama, antrean pesan) karena mereka tidak menyediakan semantik yang kuat untuk membersihkan sumber daya mereka dari aplikasi bermasalah atau berbahaya. Model IPC dasar yang digunakan Binder adalah RPC (panggilan prosedur jarak jauh). Artinya, proses pengiriman mengirimkan operasi IPC lengkap ke kernel, yaitu dieksekusi dalam proses penerimaan; pengirim dapat memblokir sementara penerima mengeksekusi, memungkinkan hasil dikembalikan dari panggilan. (Pengirim secara opsional dapat menentukan mereka tidak boleh memblokir, melanjutkan eksekusi mereka secara paralel dengan penerima.) Binder IPC dengan demikian berbasis pesan, seperti antrian pesan Sistem V, bukan berbasis aliran seperti pada pipa Linux. Pesan di Binder disebut sebagai transaksi, dan pada tingkat yang lebih tinggi dapat dilihat sebagai panggilan fungsi lintas pro esses. Setiap transaksi yang dikirimkan ruang pengguna ke kernel adalah operasi lengkap: ini mengidentifikasi target operasi dan identitas pengirim serta data lengkap yang dikirimkan. Kernel menentukan proses yang sesuai untuk menerima transaksi itu, mengirimkannya ke thread yang menunggu dalam proses.



Gambar 2.3 Arsitektur Binder IPC



Gambar 2.4 Transaksi Binder IPC dasar

Perhatikan bahwa setiap proses pada Gambar 2.4 memiliki "thread pool". Ini adalah satu atau lebih thread yang dibuat oleh ruang pengguna untuk menangani

transaksi yang masuk. Kernel akan menambal setiap transaksi yang masuk ke pool yang saat ini sedang menunggu pekerjaan di kumpulan utas proses itu. Pemanggilan ke dalam kernel dari proses pengiriman tidak perlu datang dari kumpulan thread - setiap thread dalam proses bebas untuk memulai transaksi, seperti Ta pada gambar 2.4

2. 5 Dalvik

Dalvik mengimplementasikan lingkungan bahasa Java di Android yang bertanggung jawab untuk menjalankan aplikasi serta sebagian besar kode sistemnya. Hampir semua hal dalam proses layanan sistem —mulai dari manajer paket, melalui manajer jendela, hingga manajer aktivitas— diimplementasikan dengan kode bahasa Java yang dijalankan oleh Dalvik. Namun, Android bukanlah platform berbahasa Java dalam pengertian tradisional. Kode Java dalam aplikasi Android disediakan dalam format bytecode Dalvik, berbasis di sekitar mesin register daripada bytecode berbasis stack tradisional Java.

Format bytecode Dalvik memungkinkan interpretasi yang lebih cepat, sambil tetap mendukung kompilasi JIT (Just-in-Time). Bytecode Dalvik juga lebih hemat ruang, baik di disk maupun di RAM, melalui penggunaan string pooling dan teknik lainnya. Saat menulis aplikasi Android, kode sumber ditulis dalam Java dan kemudian dikompilasi menjadi bytecode Java standar menggunakan alat Java tradisional. Android kemudian memperkenalkan langkah baru: mengonversi bytecode Java tersebut menjadi representasi bytecode Dalvik yang lebih ringkas. Ini adalah versi kode byte Dalvik dari aplikasi yang dikemas sebagai biner aplikasi final dan akhirnya diinstal pada perangkat.

Arsitektur sistem Android sangat bergantung pada Linux untuk primitif sistem, termasuk manajemen memori, keamanan, dan komunikasi lintas batasan keamanan. Ia tidak menggunakan bahasa Java untuk konsep inti sistem operasi—ada sedikit upaya untuk mengabstraksikan aspek-aspek penting dari sistem operasi Linux yang mendasarinya.

Catatan khusus adalah penggunaan proses oleh Android. Desain Android tidak bergantung pada bahasa Java untuk isolasi antara aplikasi dan sistem, melainkan mengambil pendekatan proses isolasi sistem operasi tradisional. Ini berarti bahwa setiap aplikasi berjalan dalam proses Linuxnya sendiri dengan lingkungan Dalviknya sendiri, begitu pula server sistem dan bagian inti lainnya dari platform yang ditulis dalam Java. Menggunakan proses untuk isolasi ini memungkinkan Android memanfaatkan semua fitur Linux untuk mengelola proses, mulai dari isolasi memori hingga membersihkan semua sumber daya yang terkait dengan proses saat proses tersebut hilang. Selain proses, alih-alih menggunakan arsitektur SecurityManager Java, Android hanya mengandalkan fitur keamanan Linux. Penggunaan proses dan keamanan Linux sangat menyederhanakan lingkungan Dalvik, karena tidak lagi bertanggung jawab atas aspek penting dari stabilitas dan ketahanan sistem. Bukan kebetulan, itu juga memungkinkan aplikasi untuk secara bebas menggunakan kode asli dalam implementasinya, yang sangat penting untuk game yang biasanya dibangun dengan mesin berbasis C++. Proses pencampuran dan bahasa Java seperti ini memang menimbulkan beberapa tantangan. Memunculkan lingkungan berbahasa Java yang segar dapat memakan waktu sebentar, bahkan pada perangkat keras seluler

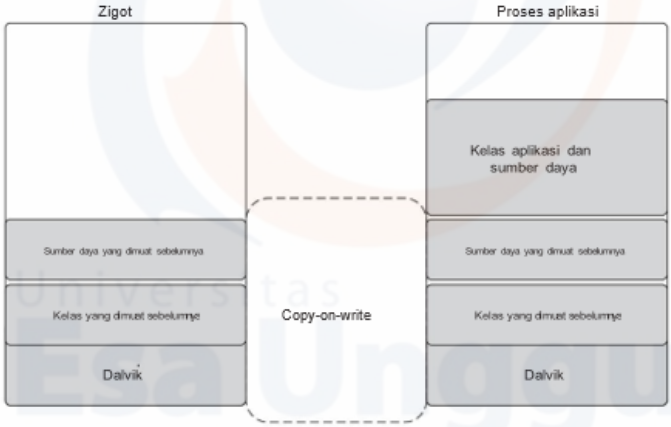
modern. Ingat salah satu tujuan desain Android, untuk dapat meluncurkan aplikasi dengan cepat, dengan target 200 msec.

Mengharuskan proses Dalvik baru untuk aplikasi baru ini akan jauh melampaui anggaran itu. Peluncuran 200 mdtk sulit dicapai pada perangkat keras seluler, bahkan tanpa perlu menginisialisasi lingkungan berbahasa Java yang baru. Solusi untuk masalah ini adalah zyangote native daemon yang telah kami sebutkan secara singkat dari proses sebelumnya. Zyangote bertanggung jawab untuk memunculkan dan menginisialisasi Dalvik, ke titik di mana ia siap untuk mulai menjalankan sistem atau kode aplikasi yang ditulis di Java. Semua proses baru berbasis Dalvik (sistem atau aplikasi) bercabang dari zyangote, memungkinkan mereka untuk memulai eksekusi dengan lingkungan yang sudah siap digunakan.

Bukan hanya Dalvik yang dimunculkan oleh zigot . Zyangote juga melakukan preload banyak bagian framework Android yang biasa digunakan dalam sistem dan aplikasi, serta memuat resource dan hal-hal lain yang sering dibutuhkan. Perhatikan bahwa membuat proses baru dari zyangote melibatkan garpu Linux, tetapi tidak ada panggilan exec . Proses baru adalah replika dari proses zigot asli , dengan semua keadaan prainisialisasinya sudah diatur dan siap digunakan. Gambar 9., mengilustrasikan bagaimana proses aplikasi Java baru terkait dengan proses zigot asli . Setelah fork, proses baru memiliki lingkungan Dalvik sendiri yang terpisah, meskipun berbagi semua data yang dimuat sebelumnya dan diinisialisasi dengan zyangote melalui halaman copy-on-write. Semua yang tersisa untuk menyiapkan proses baru yang sedang berjalan adalah memberinya identitas yang benar (UID, dll.), menyelesaikan

semua inisialisasi Dalvik yang memerlukan utas awal, dan memuat aplikasi atau kode sistem untuk dijalankan.

Selain kecepatan peluncuran, ada manfaat lain yang dibawa oleh zigot . Karena hanya garpu yang digunakan untuk membuat proses darinya, sejumlah besar halaman RAM kotor yang diperlukan untuk menginisialisasi Dalvik dan preload kelas dan sumber daya dapat dibagi antara zyangote dan semua proses anaknya. Berbagi ini sangat penting untuk lingkungan Android, di mana swap tidak tersedia; halaman permintaan halaman bersih (seperti kode yang dapat dieksekusi) dari ''disk'' (memori flash) tersedia. Namun setiap halaman kotor harus tetap terkunci di RAM; mereka tidak dapat di-page ke ''disk”.

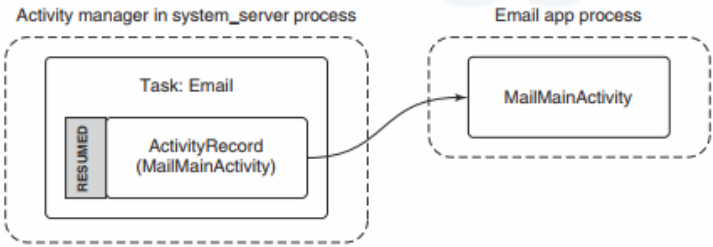


Gambar 2.5 Membuat proses Dalvik baru dari zigot

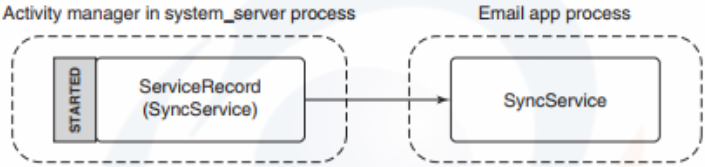
2.6 Proses Model Life-Cycle

Model proses tradisional di Linux adalah fork untuk membuat proses baru, diikuti oleh exec untuk menginisialisasi proses tersebut dengan kode yang akan dijalankan dan kemudian memulai eksekusinya. Shell

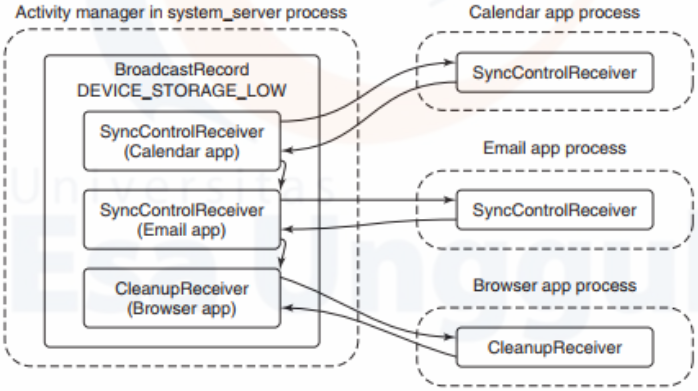
bertanggung jawab untuk menjalankan eksekusi ini, proses forking dan eksekusi yang diperlukan untuk menjalankan perintah shell. Ketika perintah itu keluar, prosesnya dihapus oleh Linux. Android menggunakan proses yang agak berbeda. Seperti yang telah dibahas pada bagian sebelumnya tentang aplikasi, pengelola aktivitas adalah bagian dari Android yang bertanggung jawab untuk mengelola aplikasi yang sedang berjalan. Ini mengoordinasikan peluncuran proses aplikasi baru, menentukan apa yang akan berjalan di dalamnya, dan kapan tidak lagi Untuk meluncurkan proses baru, pengelola aktivitas harus berkomunikasi dengan zigot. Saat pengelola aktivitas pertama kali dimulai, ia membuat soket khusus dengan zyangote, yang melaluinya ia mengirimkan perintah saat diperlukan untuk memulai proses. Perintah tersebut terutama menjelaskan sandbox yang akan dibuat: UID tempat proses baru harus dijalankan dan batasan keamanan lainnya yang akan diterapkan padanya. Oleh karena itu, Zyangote harus dijalankan sebagai root: ketika bercabang, Zyangote melakukan pengaturan yang sesuai untuk UID yang akan dijalankannya, akhirnya menghapus hak akses root dan mengubah proses ke UID yang diinginkan. Ingat dalam diskusi kita sebelumnya tentang aplikasi Android bahwa pengelola aktivitas menyimpan informasi dinamis tentang pelaksanaan aktivitas (gambar 2.6), layanan (gambar 2.7), siaran (ke penerima seperti pada gambar 2.8) , dan penyedia konten (gambar 2.9).



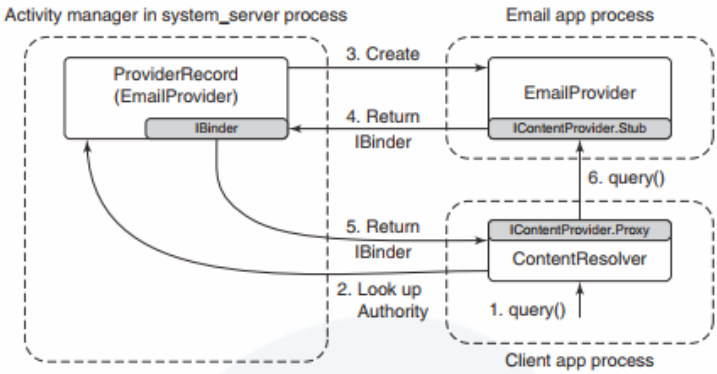
Gambar 2.7 Memulai aktivitas utama aplikasi e-mail



Gambar 2.8 Memulai proses layanan aplikasi



Gambar 2.9 Mengirim siaran ke penerima aplikasi



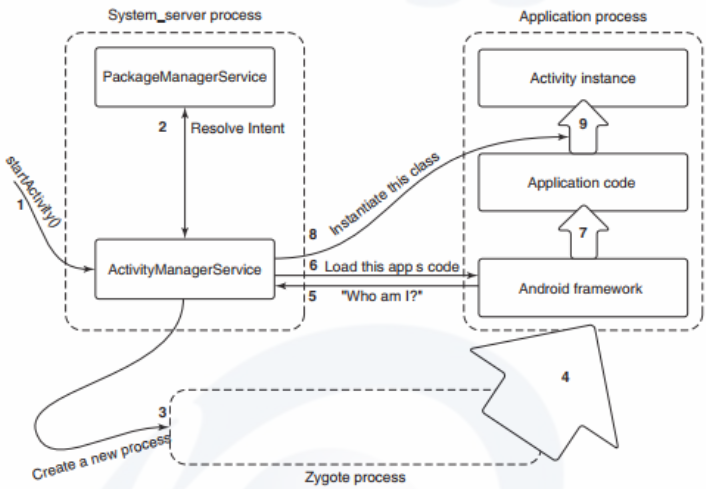
Gambar 2.10 Mengirim siaran ke penerima aplikasi

Dalam menggunakan informasi ini untuk mendorong pembuatan dan pengelolaan proses aplikasi. Misalnya, saat peluncur aplikasi memanggil sistem dengan maksud baru untuk memulai aktivitas seperti yang kita lihat di gambar 10, manajer aktivitaslah yang bertanggung jawab untuk menjalankan aplikasi baru tersebut. Alur untuk memulai aktivitas dalam proses baru ditunjukkan pada gambar 2.10 Itu rincian setiap langkah dalam ilustrasi adalah:

a. Beberapa proses yang ada (seperti peluncur aplikasi) memanggil pengelola aktivitas dengan maksud menjelaskan aktivitas baru yang ingin dimulainya.
b. Manajer aktivitas meminta manajer paket untuk menyelesaikan maksud ke komponen eksplisit.
c. Manajer aktivitas menentukan bahwa proses aplikasi belum siap berjalan, dan kemudian meminta zyangote untuk proses baru dari UID yang sesuai. Zyangote melakukan fork, membuat proses baru yang merupakan tiruan dari dirinya sendiri,

melepaskan hak istimewa dan menyetel UID-nya dengan tepat untuk sandbox aplikasi, dan menyelesaikan inisialisasi Dalvik dalam proses tersebut sehingga waktu proses Java berjalan sepenuhnya. Misalnya, ia harus memulai utas seperti pengumpul sampah setelah bercabang.

d. Proses baru, sekarang klon zigot dengan lingkungan Java sepenuhnya aktif dan berjalan, menelepon kembali ke pengelola aktivitas, menanyakan ''Apa yang harus saya lakukan?''
e. Pengelola aktivitas mengembalikan informasi lengkap tentang aplikasi tion itu dimulai, seperti di mana menemukan kodenya.
f. Proses baru memuat kode untuk aplikasi yang sedang dijalankan.
g. Manajer aktivitas mengirimkan ke proses baru setiap operasi yang tertunda, dalam hal ini ''memulai aktivitas X.''
h. Proses baru menerima perintah untuk memulai suatu aktivitas, memberi contoh kelas Java yang sesuai, dan mengeksekusinya



Gambar 2.11 Langkah-langkah dalam meluncurkan proses aplikasi baru

2.7 Aplikasi Android

Android menyediakan model aplikasi yang sangat berbeda dari lingkungan baris perintah normal di shell Linux atau bahkan aplikasi yang diluncurkan dari antarmuka pengguna grafis. Aplikasi bukanlah file yang dapat dieksekusi dengan titik masuk utama; itu adalah wadah dari segala sesuatu yang membentuk aplikasi itu: kodenya, sumber daya grafis, deklarasi tentang apa itu untuk sistem, dan data lainnya.
paket android.os antarmuka I Service Manager {IBinder get Service (Str ing name); void add Service (Str ing name, pengikat IBinder) }

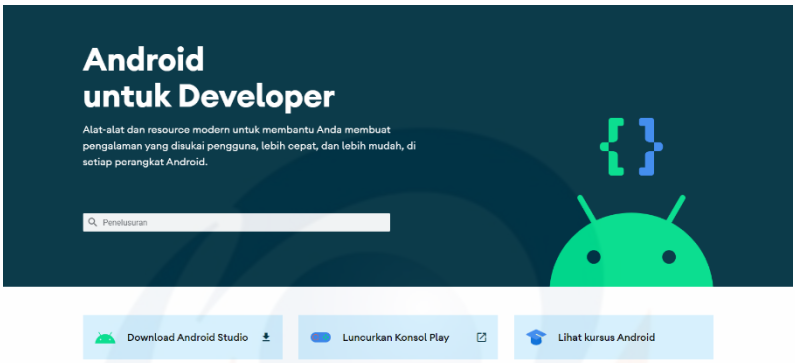
Aplikasi Android menurut konvensi adalah file dengan ekstensi apk untuk Paket Android. File ini sebenarnya adalah arsip zip biasa berisi segala hal tentang aplikasi.

Konten penting dari sebuah apk adalah Sebuah manifes yang menjelaskan apa itu aplikasi, apa fungsinya, dan bagaimana menjalankannya. Manifes harus memberikan nama paket untuk aplikasi, string cakupan gaya Java (seperti com.android.app.calculator), yang mengidentifikasinya secara unik.

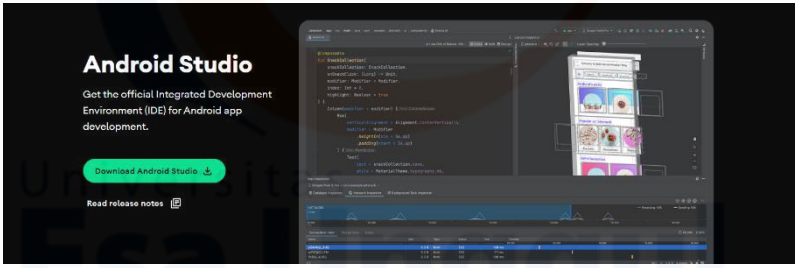
Sumber daya yang dibutuhkan oleh aplikasi, termasuk string yang ditampilkannya kepada pengguna, data XML untuk tata letak dan deskripsi lainnya, peta bit grafis, dll. Kode itu sendiri, yang mungkin merupakan bytecode Dalvik serta li asli kode bray. Menandatangani informasi, mengidentifikasi penulis dengan aman.

BAB III
APLIKASI ANDROID STUDIO

3.1 Unduh Aplikasi Android Studio



Gambar 3.1 Link unduh aplikasi
(https://developer.android.com/studio?hl=id)



Gambar 3.1 Link unduh aplikasi
(https://developer.android.com/studio?hl=id)

Android Studio adalah IDE resmi untuk pengembangan Android, dan dilengkapi semua hal yang Anda perlukan untuk mem-build aplikasi Android. Halaman ini mencantumkan fitur baru dan peningkatan dalam versi terbaru di saluran stabil, Android Studio Dolphin. Anda dapat mendownloadnya di sini atau mengupdatenya di

dalam Android Studio dengan mengklik **Help > Check for updates (Android Studio > Check for updates** di macOS). Untuk melihat catatan rilis Android Studio versi lama/preview. (https://developer.android.com/studio/releases/past-releases).

Tabel 3.1. Android Studio downloads

Platform	Android Studio Package	Sie	SHA-256 checksum
Windows (64-bit)	android-studio-2021.3.1.17-windows.exe	957.3 MB	dd176791e15e921d4a3b3c9a251c61e5cfd28d75588fd717971dfbac030cd497
	Recommended		
Windows (64-bit)	android-studio-2021.3.1.17-windows.zip	959.6 MB	bdce14643efee37a4d892994b3329496460062f9c65ed870ff61a80267cb206a
	No .exe installer		
Mac (64-bit)	android-studio-2021.3.1.17-mac.dmg	1.0 GB	4e10799559efc3445d61fb12bbf68e0a9801607a6114c6783bb26a93784d3150
Mac (64-bit, ARM)	android-studio-2021.3.1.17-mac_arm.dmg	1.0 GB	0adbbddfa1e0e52e7bf21a5b560f60f8982ef82c0677db2d2ff7a2bd73ab156f
Linux (64-bit)	android-studio-2021.3.1.17-linux.tar.gz	982.7 MB	89adb0ce0ffa46b7894e7bfedb142b1f5d52c43c171e6a6cb9a95a49f77756ca

Sumber : Android Studio downloads*
(https://developer.android.com/studio)

(*) Untuk lebih lengkap anda dapat meng-unduh link https://developer.android.com/studio/archive dan untuk emlator Android Emulator.
https://developer.android.com/studio/emulator_archive .

Tabel 3.2. Command line tools only

Platform	Android Studio Package	Sie	SHA-256 checksum
Windows	commandlinetools-win-9123335_latest.zip	120.9 MB	8a90e6a3deb2fa13229b2e335efd07687dcc8a55a3c544da9f40b41404993e7d
Mac	commandlinetools-mac-9123335_latest.zip	120.9 MB	d0192807f7e1cd4a001d13bb1e5904fc287b691211648877258aa44d1fa88275
Linux	commandlinetools-linux-9123335_latest.zip	120.9 MB	0bebf59339eaa534f4217f8aa0972d14dc49e7207be225511073c661ae01da0a

Sumber : Command line tools
(https://developer.android.com/studio)

3.2 Requirements

3.2.1 MacOS®

1) MacOS® 10.14 (Mojave) or higher
2) ARM-based chips, or 2nd generation Intel Core or newer with support for Hypervisor.Framework
3) 8 GB RAM or more
4) 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
5) 1280 x 800 minimum screen resolution

3.2.2 Windows

1) 64-bit Microsoft® Windows® 8/10
2) x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor.Framework
3) 8 GB RAM or more
4) 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
5) 1280 x 800 minimum screen resolution

3.2.3 Linux

1) Any 64-bit Linux distribution that supports Gnome, KDE, or Unity DE; GNU C Library (glibc) 2.31 or later.
2) x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD processor with support for AMD Virtualization (AMD-V) and SSSE3
3) 8 GB RAM or more
4) 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
5) 1280 x 800 minimum screen resolution

3.3 Kompatibilitas plugin Android Gradle

Sistem build Android Studio didasarkan pada Gradle. Plugin Android Gradle menambahkan beberapa fitur yang dikhususkan untuk mem-build aplikasi Android. Tabel berikut mencantumkan versi plugin Android Gradle yang diperlukan untuk setiap versi Android Studio. Pada android rilis sebelumnya, Sistem build Android Studio didasarkan pada Gradle. Plugin Android Gradle menambahkan beberapa fitur yang dikhususkan untuk mem-build aplikasi Android. Tabel berikut menyanjikan versi plugin Android Gradle yang diperlukan untuk setiap versi Android Studio.

Tabel 3.3. Versi Plugin Android Gradle

Versi Android Studio	Versi plugin yang diperlukan
Flamingo | 2022.2.1	3.2-8.0
Electric Eel | 2022.1.1	3.2-7.4
Dolphin | 2021.3.1	3.2-7.3
Chipmunk | 2021.2.1	3.2-7.2
Bumblebee | 2021.1.1	3.2-7.1
Arctic Fox | 2020.3.1	3.1-7.0

Sumber : Kompatibilitas Android Studio dan plugin Android Gradle(https://developer.android.com/studio/releases/past-releases)

3.3.1 Mengupdate plugin Android Gradle

Saat meng-update Android Studio, saudara akan menerima notifikasi request untuk otomatis meng-update plugin Android Gradle ke versi terbaru yang tersedia. Anda dapat memilih untuk menerima update atau menentukan versi secara manual berdasarkan persyaratan build project. Dalam menentukan versi plugin di menu **File > Project**

Structure > Project di Android Studio, atau dalam file build.gradle level teratas. Versi plugin berlaku untuk semua modul yang dibuat di project Android Studio tersebut. Contoh berikut menetapkan plugin ke versi 4.2.0 dari file build.gradle:

Console #Groovy

```
buildscript { repositories { // Gradle 4.1 and higher include support for Google's Maven repo using // the google() method. And you need to include this repo to download // Android Gradle plugin 3.0.0 or higher. google()  ... } dependencies{classpath 'com.android.tools.build:gradle:4.2.0'}
```

Console #Kotlin

```
buildscript { repositories {  // Gradle 4.1 and higher include support for Google's Maven repo using // the google() method. And you need to include this repo to download // Android Gradle plugin 3.0.0 or higher. google().. } dependencies{classpath("com.android.tools.build:gradle:4.2.0}
```

3.3.2 Mengupdate Gradle

Saat mengupdate Android Studio, Anda mungkin menerima permintaan untuk mengupdate juga Gradle ke versi terbaru yang tersedia. Anda dapat memilih untuk menerima update atau menentukan versi secara manual berdasarkan persyaratan build project. Tabel berikut mencantumkan versi Gradle yang diperlukan untuk setiap versi plugin Android Gradle. Agar performanya optimal, Anda harus menggunakan Gradle dan plugin versi terbaru.

Tabel 3.4. Sinkronisasi Versi Plugin Android Gradle

Versi Plugin	Versi Gradle yang diperlukan
1.0.0 - 1.1.3	2.2.1 - 2.3
1.2.0 - 1.3.1	2.2.1 - 2.9
1.5.0	2.2.1 - 2.13
2.0.0 - 2.1.2	2.10 - 2.13
2.1.3 - 2.2.3	2.14.1 - 3.5
2.3.0+	3.3+
3.0.0+	4.1+
3.1.0+	4.4+
3.2.0 - 3.2.1	4.6+
3.3.0 - 3.3.3	4.10.1+
3.4.0 - 3.4.3	5.1.1+
3.5.0 - 3.5.4	5.4.1+
3.6.0 - 3.6.4	5.6.4+
4.0.0+	6.1.1+
4.1.0+	6.5+
4.2.0+	6.7.1+
7.0	7.0+

Sumber : Update Gradle (https://developer.android.com/studio/releases/gradle-plugin#groovy)

3.4 Kompatiblitas

Tabel 3.5. Kompatibiltas Gradle

	Version Minimum	Version Default	Keterangan
Gradle	7.0.2	7.0.2	Untuk mempelajari lebih lanjut, lihat mengupdate Gradle.
SDK Build Tools	30.0.2	30.0.2	Instal atau konfigurasi SDK Build Tools.
NDK	T/A	21.4.7075529	Instal atau konfigurasi versi lain dari NDK.
JDK	11	11	Untuk mempelajari lebih lanjut, lihat menyetel versi JDK.

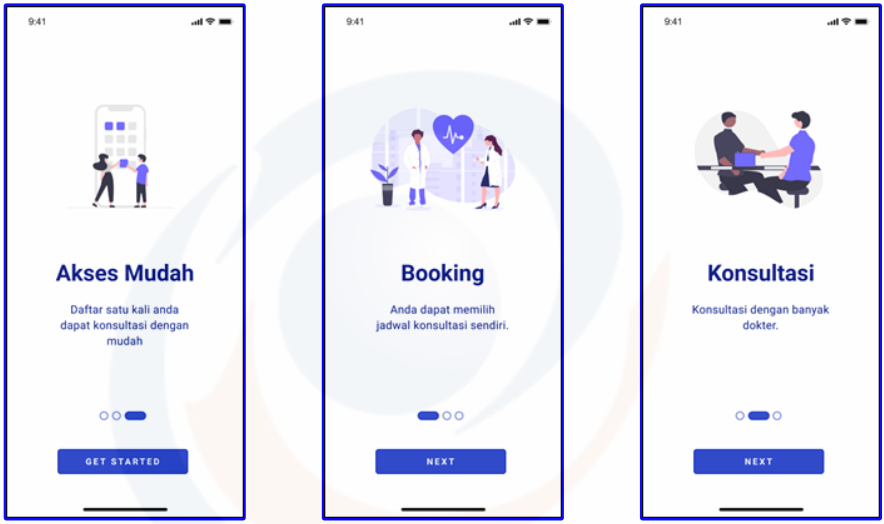
3.5 Probis Dasar Build Aplikasi Android

Secara sederhana dan ringkasan proses untuk membuat aplikasi Android dan menyertakan link ke beberapa fitur Android Studio yang perlu Anda gunakan selama setiap tahap pengembangan.

Gambar 3.3 Tahapan proses bisnis membangun aplikasi (https://developer.android.com/studio/workflow)

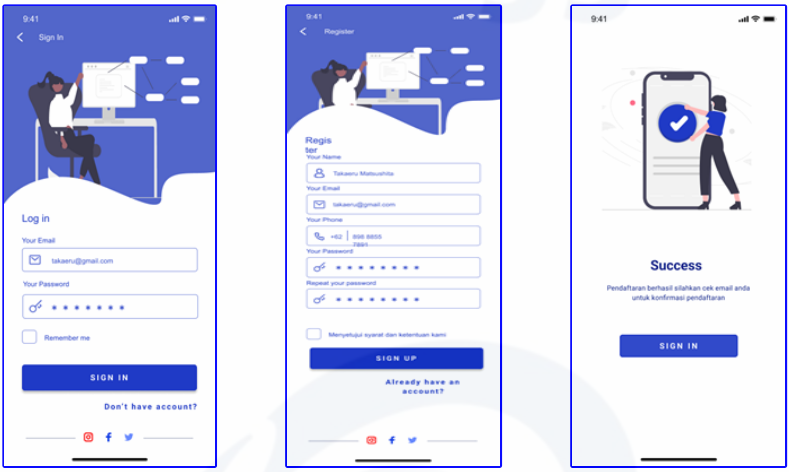
BAB IV
MANUAL USER INTERFACE (UI) APLIKASI LET'SFISIO

Pada Saat Aplikasi dibuka akan menampilkan *Splash Screen* Seperti Berikut ini :



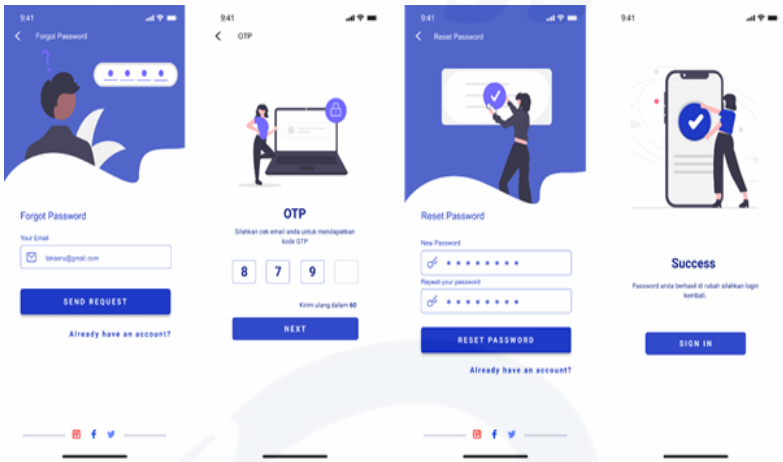
Gambar 4.1 Splash Screen

Selanjutnya, akan Pengguna Akan diminta untuk Login dengan Username dan Password yang telah terdaftar sebelumnya, Jika Belum Memiliki Akun Klik "Don't Have Account" Untuk menuju Formulir Registrasi Akun pada aplikasi. Jika Berhasil Mendaftar Akan muncul Menu dengan bertuliskan Success seperti pada gambar dibawah ini.

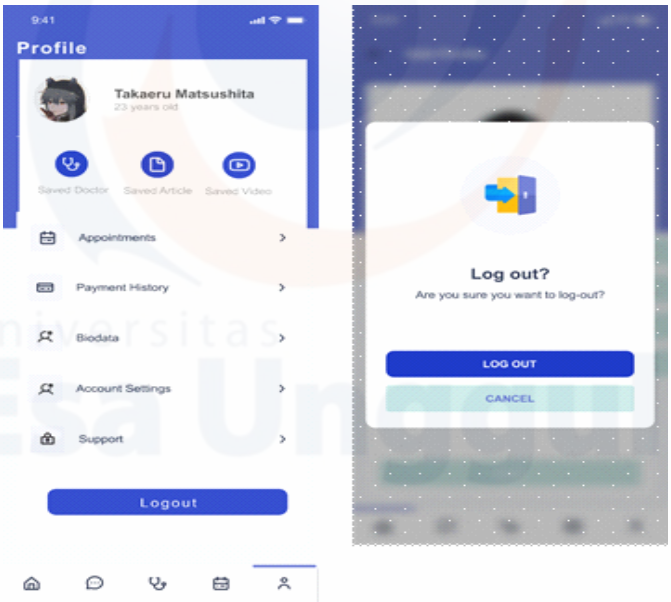


Gambar 4.2 Tampilan Login dan Registrasi Akun

Jika Pengguna Pernah membuat akun dan lupa terhadap akun yang pernah dibuat, pengguna dapat mereset password dengan *forgot Password*, Pengguna dapat memberikan informasi email dan dapat memasukan kode Otp yang telah dikirim ke email pengguna. Setelah berhasil, pengguna akan diminta untuk memasukan password baru dan melakukan login pada aplikasi Lets Fisio.



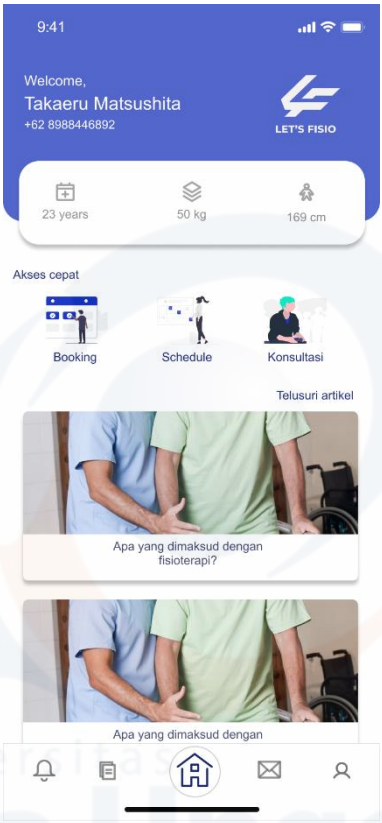
Gambar 4.3 Tampilan Reset Password



Gambar 4.4 Tampilan Menu Logout Account

Jika pengguna selesai menggunakan Lets Fisio, pengguna dapat logout dengan masuk ke Menu Profile dan Mengklik Tombol Logout, Selanjutnya pengguna diminta

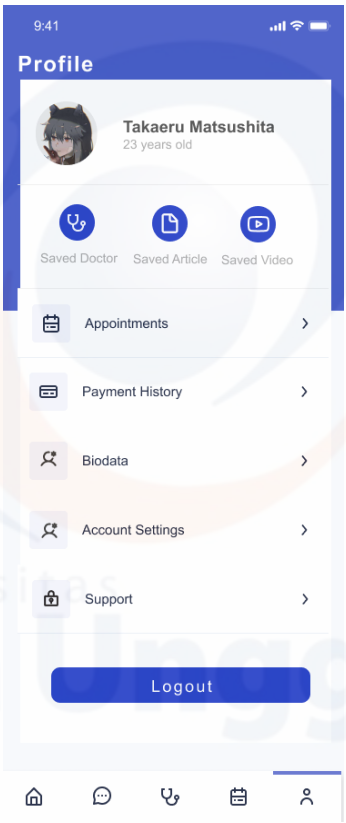
untuk memilih Pop Up Log out untuk Kembali kemenu Login. Setelah berhasil login, Pengguna akan dialihkan pada menu Dashboard Atau Home dengan tampilan seperti berikut ini :



Gambar 4.5 Tampilan Beranda / Home

Dalam menu Dashboard atau Home ini terdiri dari beberapa Informasi salah satunya Informasi Kontak Pribadi (Nama dan No Telepon), Tinggi Badan, Usia, dan Berat badan. Selain itu, Terdapat Akses Cepat untuk Pengguna yang ingin melaksanakan Booking Fisioterapis, atau Memeriksa Schedule atau Janji dengan Fisioterapis sampai

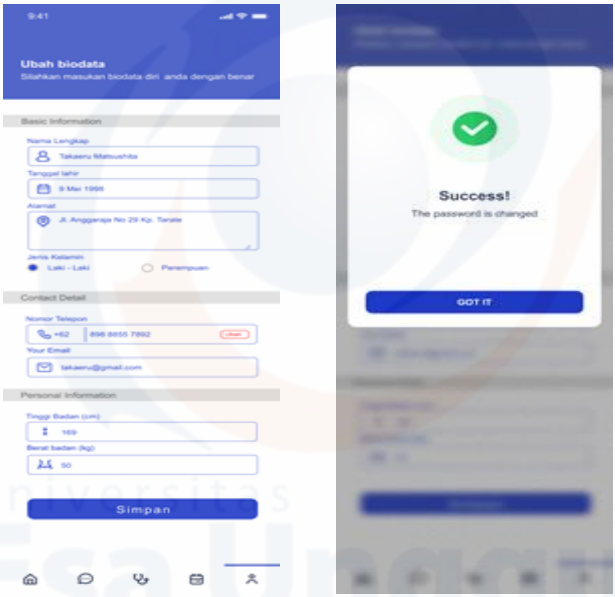
pada Konsultasi Melalui chat atau Video Conference yang telah disediakan. Dalam menu Home ini juga, pengguna dapat menerima informasi Artikel terupdate yang ditulis oleh admin atau fisioterapis, Update Video tutorial terbaru dan menu lainnya. Pada menu paling bawah tersedia beberapa Icon Akses Cepat seperti menu pesan, profile, Notifikasi dan Informasi Lainnya



Gambar 4.6 Tampilan Menu Profile

Selanjutnya, Dari Menu Home, Pengguna Bisa memilih menuju Profile . Pada Menu Profile terdapat beberapa pilihan Informasi yang diakses oleh Pengguna. Contohnya

seperti History Fisioterapis yang pernah berhubungan, Artikel dan Video yang simpan. Selain itu Pengguna juga dapat melihat Janji Atau Appointments yang telah dibuat atau yang pernah dilakukan. Menu Payment History untuk melihat Transaksi yang pernah dilakukan, Akun Setting dan Biodata Untuk Melihat Informasi Akun dan Biodata yang tersedia pada Akun pengguna , Support Untuk Melihat Informasi mengenai Aplikasi dan Bantuan Lainnya.

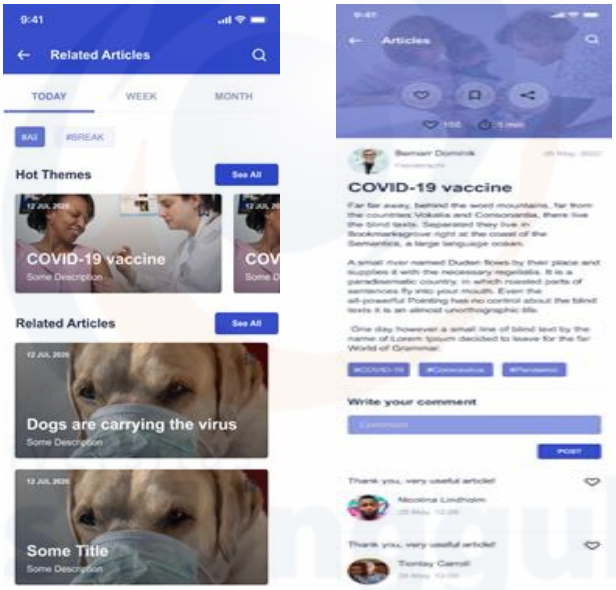


Gambar 4.7 Tampilan Menu Ubah Biodata Pengguna

Dari Menu Profile Jika kita Klik Menu Ubah Biodata, Maka akan tersedia Formulir untuk pengguna Mengubah informasi mengenai Biodata Pengguna Seperti Nama Lengkap, Tanggal dan Bulan Lahir, Tahun Lahir, Alamat dan Jenis Kelamin.

Dalam menu Biodata ini Alamat sangat berperan penting untuk mengetahui fisioterapis yang tersedia di sekitar pengguna. Jadi Ketika pengguna berpindah lokasi maka alamat ini juga harus diubah agar aplikasi dapat melacak fisioterapis terdekat dari lokasi pengguna.

Selain itu, tersedia juga Contact Detail yang dapat diubah pengguna dan Informasi pribadi terkait dengan data Tinggi dan Berat badan, Data – data ini digunakan oleh fisioterapis yang nantinya akan berhubungan dengan pengguna.



Gambar 4.8 Tampilan Menu Related Artikel

Selanjutnya Pengguna juga dapat mengakses Informasi Artikel yang tersedia yang ditulis oleh fisioterapis atau oleh administrator, Pengguna dapat memilih informasi artikel yang dibutuhkan seputar Kesehatan terutama mengenai fisioterapis. Jika terdapat artikel yang dibutuhkan pengguna

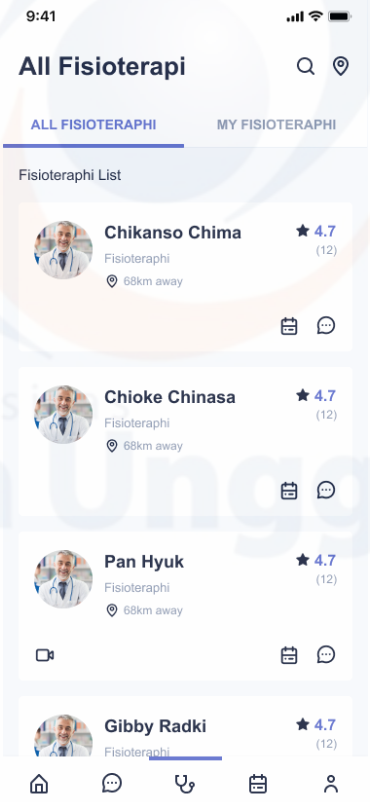
dapat membaca artikel secara utuh dan dapat berkomentar atau berkonsultasi dengan penulis melalui Kolom komentar yang telah tersedia pada aplikasi. Pengguna juga dapat Menyukai artikel serta menyimpan artikel tersebut untuk dibaca pada waktu yang lain dan dapat membagikan artikel tersebut pada platform aplikasi lainnya.

Gambar 4.9 Tampilan Menu Deskripsi Artikel

Pengguna juga dapat mengakses Menu Video untuk mencari informasi mengenai video – video perawatan diri

yang tersedia pada platform. Video dibuat oleh fisioterapis dan administrator, pengguna dapat memilih beberapa video yang related dengan kondisi pengguna atau yang direkomendasikan oleh fisioterapis.

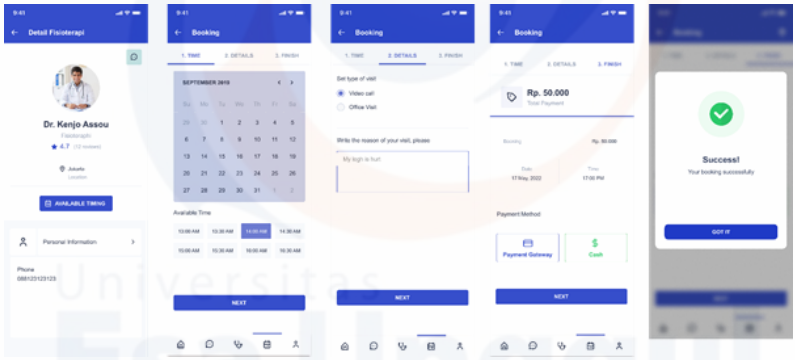
Jika terdapat video yang dibutuhkan pengguna dapat meilhat video secara utuh dan dapat berkomentar atau berkonsultasi dengan penyedia video melalui Kolom komentar yang telah tersedia pada aplikasi. Pengguna juga dapat Menyukai artikel serta menyimpan artikel tersebut untuk dibaca pada waktu yang lain dan dapat membagikan artikel tersebut pada platform aplikasi lainnya.



Gambar 4.10 Tampilan Menu List All Fisiotherapi

Selanjutnya adalah menu Fisioterapis, Pengguna dapat melihat fisioterpis yang tersedia pada alamat terdekat pengguna, Dalam menu ini tersedia Maps Untuk Melihat Lokasi fisioterapis, Menu Cari untuk mencari Nama fisioterapis yang di inginkan, Menu List Fisioterapis yang terdaftar pada aplikasi dan menu My Fisioterapis untuk Melihat fisioterapis yang pernah di booking atau pernah berhubungan sebelumnya.

Pengguna juga dapat melihat Ratting Fisioterapis yang diberikan pengguna lain dan pengguna dapat melakukan Booking serta Konsultasi melalui Chat atau Video interaktif dengan Seluruh Fisioterapis yang telah terdaftar pada Aplikasi.



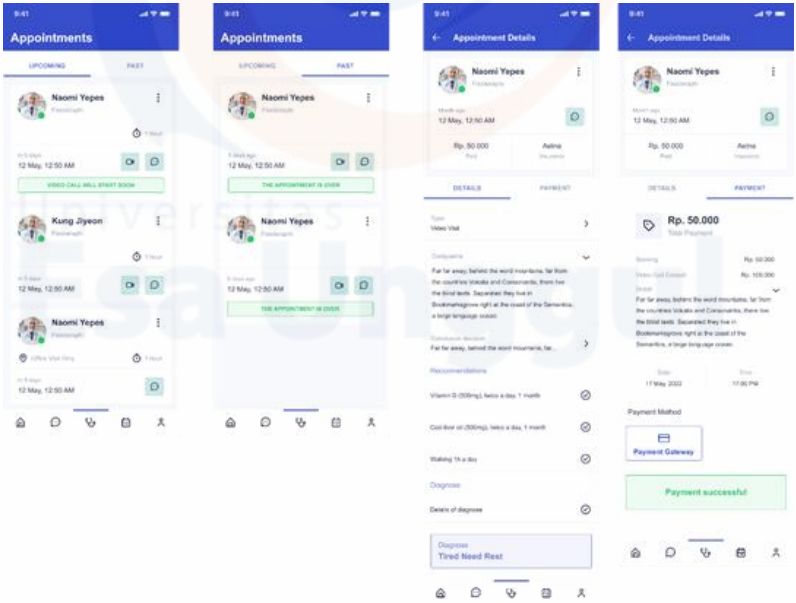
Gambar 4.11 Tampilan Menu Booking Fisiotherapi

Untuk melakukan Booking Fisioterapis dapat dilakukan dengan cara sebagi berikut :

- Pengguna Memilih Fisioterapis yang di inginkan Maka akan muncul Data Detail dari Fisioterapis yang dipilih
- Pengguna dapat menentukan Janji Pertemuan dengan fisioterapis dengan memilih Tanggal dan Jam Janji.

- Pengguna dapat memilih Cara untuk pertemuan dengan Fisioterapis apakah menggunakan Videocall atau bertemu langsung di Kantor Fisioterapis
- Pengguna Akan mendapatkan tagihan mengenai janji pertemuan yang telah ditentukan. Pengguna dapat memilih Metode pembayaran, Menggunakan Payment Gateway tersedia pada aplikasi atau secara cash Ketika bertemu dengan fisioterapis
- Jika ke empat proses tersebut telah berhasil dilaksanakan, maka akan muncul Popup Sukses dalam pemesanan. Selanjutnya pengguna dapat memeriksa schedule dan menu Myfisioterapis untuk memastikan Pemesanan booking berhasil.

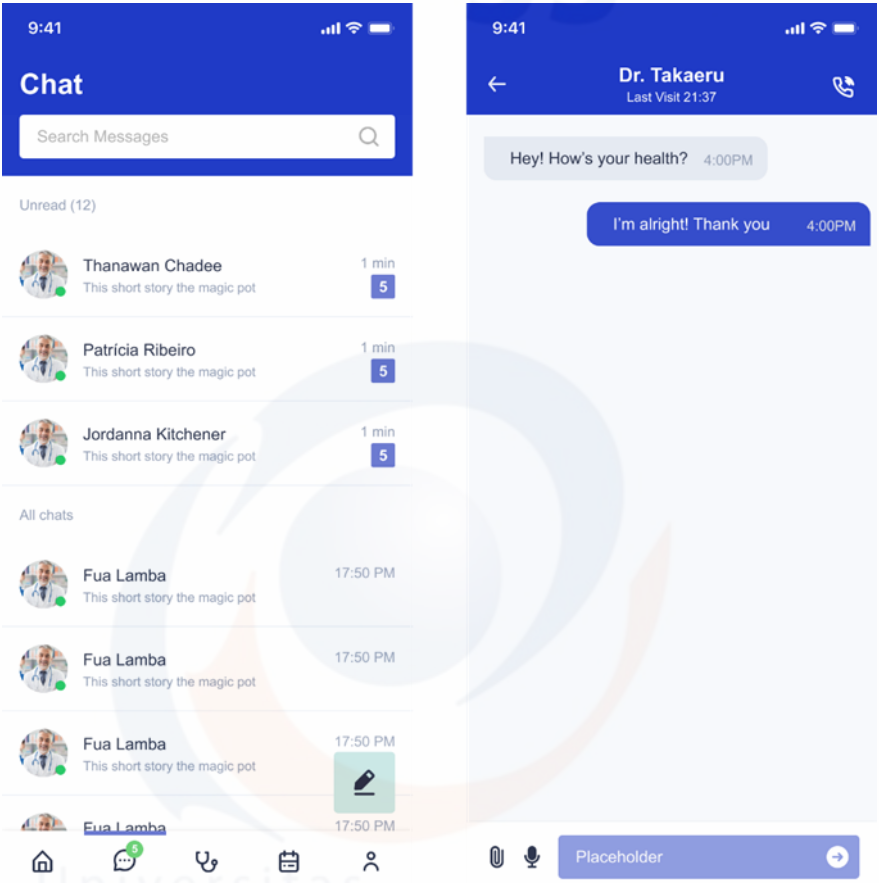
Untuk melihat informasi pemesanan atau Booking dapat dilihat juga seperti pada menu berikut:



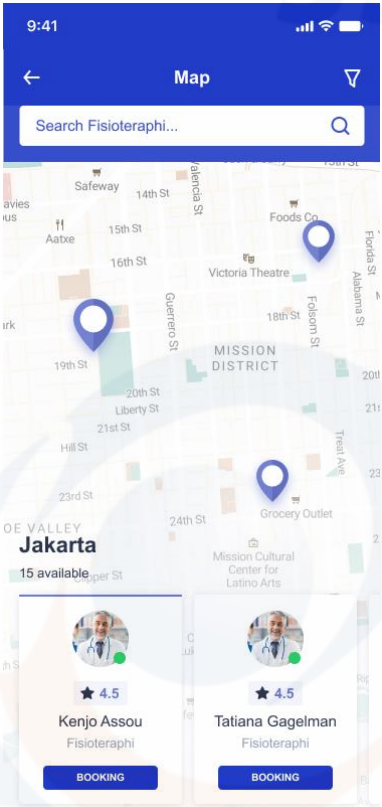
Gambar 4.12 Tampilan Informasi Pemesanan Fisiotherapi

Menu disamping adalah Menu Map yang dapat digunakan oleh pengguna untuk memeriksa lokasi fisioterapis yang tersedia di lokasi pengguna. Maps ini juga dapat diakses oleh pengguna untuk petunjuk menuju lokasi fisioterapis yang telah di booking sebelumnya.

Dibawah ini adalah menu Konsultasi atau Komunikasi Interaktif antara Pengguna Dengan fisioterapis. Pengguna dapat Melakukan Komunikasi dengan menggunakan Media (Chat, Voice Call dan Video Call) Dalam menu ini pengguna dapat melihat informasi status dari semua fisioterapis (Aktif, Sedang Tidak Aktif atau Informasi Aktif Terakhir). Semua informasi chat yang telah dilaksanakan akan disimpan pada history Chat pada Menu ini.



Gambar 4.13 Tampilan Menu Urgent



Gambar 4.14 Tampilan Menu Maps

DAFTAR PUSTAKA

"Android One smartphones released in India". BBC News. September 15, 2014. (2014). Archived from the original on March 13, 2017. Retrieved March 12, 2017.

"Android's iconic dessert names are going away, starting with Android 10". (2019). Android Police. August 22, 2019. Retrieved August 22, 2019.

"Google's Android OS: Past, Present, and Future". (2017). PhoneArena. August 18, 2011. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

"Industry Leaders Announce Open Platform for Mobile Devices". (2017). Open Handset Alliance. November 5, 2007. Archived from the original on March 9, 2012. Retrieved March 12, 2017.

"Number of Android applications on the Google Play store". (2020). AppBrain. Retrieved August 12, 2020.

"This important Microsoft Teams for Android update fixes the strange 911 calling bug". (2022). ZDNET. Retrieved December 7, 2022.

Amadeo, Ron (January 5, 2022). "Google fixes nightmare Android bug that stopped user from calling 911". Ars Technica. Retrieved December 7, 2022.

Amsden, Z., Arai, D., Hecht, D., Holler, A., And Subrahmanyam, P. (2006). ''VMI: An Interface for Paravirtualization,'' Proc. 2006 Linux Symp., 2006.

Android Mobile App Developer Tools – Android Developers. (n.d.). Android Developers. Retrieved December 19, 2022, from https://developer.android.com/

Arensman, William L., Whipple, J. G., & Boler, Marian Starr (2009). A public safety application of GPS-enabled smartphones and the android operating system., https://doi.org/10.1109/ICSMC.2009.5346390

Barr, K., Bungale, P., Deasy, S., Gyuris, V., Hung, P., Newell, C., Tuch, H., and Zoppis, B. (2010). ''The VMware Mobile Virtualization Platform: Is That a Hypervisor in Your Pocket?'' ACM SIGOPS Operating Systems Rev., vol. 44, pp. 124-135, Dec.2010

Block, Ryan (August 28, 2007). "Google is working on a mobile OS, and it's due out shortly". Engadget. AOL. Archived from the original on March 12, 2017. Retrieved March 11, 2017.

Bohn, Dieter (August 22, 2019). (2019). "Google deserts desserts: Android 10 is the official name for Android Q". The Verge. Retrieved August 22, 2019.

Chang, G., Tan, Chunguang, Li, Guanhua, & Zhu, Chuan (2008). Developing Mobile Applications on the Android Platform., https://doi.org/10.1007/978-3-642-12349-8_15

Chavez, Chris (January 21, 2015). "Google kills off the last remaining Google Play Edition device in the Play Store". Phandroid. Archived from the original on November 29, 2016. Retrieved March 12, 2017.

Cooper, Daniel (August 18, 2015). "Google brings Android One devices to Africa". Engadget. AOL. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

Cranz, Alex (May 18, 2021). "There are over 3 billion active Android devices". The Verge. Retrieved March 24, 2022.

Cunningham, Andrew (January 14, 2014). "Moto G Google Play edition replaces near-stock Android with stock Android". Ars Technica. Condé Nast. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

Cunningham, Andrew (January 25, 2015). "Don't cry for the Google Play edition program; it was already dead". Ars Technica. Condé Nast. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

Cunningham, Andrew (July 4, 2013). "Review: The HTC One Google Play edition offers the best of both worlds". Ars Technica. Condé Nast. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

England, Jason (June 14, 2019). (2019). "Huawei begins trademarking its Android replacement OS—HongMeng". Android Central. Retrieved August 10, 2019. The trademark's been filed in Canada, the European Union, Mexico, and more.

Google's Android OS: Past, Present, and Future. (2017). PhoneArena. August 18, 2011. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

Hughes, Terry (July 28, 2014). "Google and Android Are Not the Same... and That's a Good Thing". App Developer Magazine. Retrieved July 29, 2020.

Jie, Yang; Strumpf, Dan (May 24, 2019). (2019). "Who Needs Google's Android? Huawei Trademarks Its Own Smartphone OS". Tech. The Wall Street Journal. Retrieved August 10, 2019. Chinese tech giant plans to launch its own operating system this year as access to U.S. software is hit by export ban

Junhom, Kantiya, Semkham, Sirada, Lumlert, Paphawee, Niampoonthong, Phapan, & Visoottiviseth, V. (2014). Cloudbroid: An Android Mobile Application for CloudStack Management System., https://doi.org/10.1109/ICT-ISPC.2014.6923232

Kalkov, Igor, Franke, Dominik, Schommer, John F., & Kowalewski, S. (2012). A real-time extension to the Android platform., https://doi.org/10.1145/2388936.2388955

Lomas, Natasha (August 18, 2015). "Google Pushes Android One To Africa". TechCrunch. AOL. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

Ma, Li, Gu, Lei, & Wang, Jin (2014). Research and Development of Mobile Application for Android Platform., https://doi.org/10.14257/IJMUE.2014.9.4.20

McAfee, Andrew; Brynjolfsson, Erik. (2017). Machine, Platform, Crowd : Harnessing Our Digital Future. New York. p. 166. ISBN 978-0-393-25429-7. OCLC 987909505.

Memon, R. A., & Ryu, Yeonseung (2011). A Measurement Study of the Linux Kernel for Android Mobile Computer., https://doi.org/10.1007/978-3-642-20975-8_22

Modern Operating Systems - Wikipedia. (2017, February 1). Modern Operating Systems - Wikipedia. Retrieved December 19, 2022, from https://en.wikipedia.org/wiki/Modern_Operating_Systems

Omer, M. A., Zeebaree, Subhi R. M., Sadeeq, M. A., Salim, B. W., Mohsin, Sanaa x, Rashid, Z., & Haji, Lailan M. (2021). Efficiency of Malware Detection in Android System: A Survey., https://doi.org/10.9734/AJRCOS/2021/V7I430189

Pearce, James Quintana (September 20, 2007). "Google's Strong Mobile-Related Patent Portfolio". Gigaom. Knowingly, Corp. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

Pichai, Sundar (September 15, 2014). (2014). "For the next five billion: Android One". Official Google Blog. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

Porter, Jon (August 9, 2019). (2019). "Huawei's new operating system is called HarmonyOS". The Verge. Retrieved August 9, 2019.

Programming Android with Kotlin. (2021, December 1). O'Reilly Online Learning. Retrieved December 19, 2022, from https://www.oreilly.com/library/view/programming-android-with/9781492062998/

Reichert, Corinne (June 14, 2019). (2019). "Huawei moves to trademark its own OS while objecting to US ban". Tech News. CNET. Retrieved August 10, 2019. Huawei is moving to trademark the name of its operating system, "Hongmeng," in Peru.

Savov, Vlad (October 4, 2016). (2016). "Pixel 'phone by Google' announced". The Verge. Vox Media. Archived from the original on October 5, 2016. Retrieved March 13, 2017.

Seifert, Dan (June 26, 2014). (2014). "With Android One, Google is poised to own the entire world". The Verge. Vox Media. Archived from the original on March 13, 2017. Retrieved March 12, 2017.

Silva, Pablo, Amorim, V. J., Ribeiro, Filipe Nunes, & Muzetti, Igor (2015). PrivacyMod: Controlling and Monitoring Abuse of Privacy-Related Data by Android Applications., https://doi.org/10.1109/SBESC.2015.15

Tanenbaum, A. S., & Bos, H. (n.d.). Modern Operating Systems -- Rental Edition, 5th Edition | InformIT. Modern Operating Systems -- Rental Edition, 5th Edition | InformIT. Retrieved December 19, 2022, from https://www.informit.com/store/modern-operating-systems-rental-edition-9780137618873

Tanenbaum, Andrew S.; Woodhull, Albert S. (January 2006). Operating Systems: Design and Implementation. ISBN 9780131429383.

Welch, Chris (April 16, 2013). "Before it took over smartphones, Android was originally destined for cameras". The Verge. Vox Media. Archived from

the original on April 29, 2017. Retrieved May 9, 2017.

Wu, Yonghong, Luo, Jianchao, & Luo, Lei (2010). Porting Mobile Web Application Engine to the Android Platform., https://doi.org/10.1109/CIT.2010.369

Zhang, Yuan, Dai, Jiarun, Zhang, Xiaohan, Huang, S., Yang, Zhemin, Yang, Min, & Chen, Hao (2018). Detecting third-party libraries in Android applications with high precision and recall., https://doi.org/10.1109/SANER.2018.8330204

GLOSARIUM

Aktivitas	: segala sesuatu, benda atau orang mengenai hal-hal yang telah dikerjakan atau dilakukan
Android	: sistem operasi berbasis linux yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk ponsel, smartphone dan juga pc tablet
Aplikasi	: paket perangkat lunak komputer yang melakukan fungsi tertentu secara langsung untuk pengguna akhir atau untuk aplikasi lain
Arsitektur	: ilmu seni teknik sipil atau praktik perancangan dan pembangunan struktur dan konstruksi bangunan
Artikel	: karangan faktual secara lengkap dengan panjang tertentu yang dibuat untuk dipublikasikan di media daring maupun cetak dan bertujuan menyampaikan gagasan dan fakta yang dapat meyakinkan, mendidik, dan menghibur.
Desain	: suatu perencanaan atau perancangan yang dilakukan sebelum pembuatan suatu objek, sistem, komponen, atau struktur.
Desktop	: komputer pribadi yang ditujukan untuk penggunaan secara umum di satu lokasi yang berlawanan dengan komputer jinjing atau komputer portabel
Detail	: bagian yang kecil-kecil (yang sangat terperinci) merupakan sesuatu yang

menjelaskan suatu topik secara mendalam yang disusun secara rinci,logis dan padat materinya.

Digital : penggambaran dari suatu keadaan bilangan yang terdiri dari angka 0 dan 1 atau off dan on.

Dinamis : istilah umum yang merujuk kepada segala sesuatu atau kondisi yang terus-menerus berubah, bergerak secara aktif dan mengalami perkembangan berarti

Evolusi : perubahan pada sifat-sifat terwariskan suatu populasi organisme dari satu generasi ke generasi berikutnya

Fisiotherapi : salah satu profesi di bidang kesehatan

Formulir : sebuah kertas yang berisi beberapa pertanyaan formal yang harus diisi

Frame : bingkai, kerangka, rangka, tubuh, lis, kusen, tulang, bangunan, bagan, dan keadaan badan

Hirarki : suatu susunan hal di mana hal-hal tersebut dikemukakan sebagai berada di "atas," "bawah," atau "pada tingkat yang sama" dengan yang lainnya

Interaksi : berarti saling mempengaruhi, saling menarik, saling meminta, dan memberi.

Konsultasi : Seorang Tenaga Profesional Yang Menyediakan Jasa Kepenasihatan Dalam Bidang Keahlian Tertentu

Layanan : Suatu Perbuatan Atau Penampilan Yang Dapat Ditawarkan Oleh Suatu Partai Kepada

Yang Lain Yang Tidak Berwujud Dan Tidak Menghasilkan Kepemilikan Apapun

Manajemen : Suatu Proses Bekerja Sama Dengan Dan Melalui Lainnya Untuk Mencapai Tujuan Organisasi Dengan Efektif Dan Secara Efisien Menggunakan Sumber Daya Yang Terbatas Di Lingkungan Yang Berubah-Ubah

Media : Alat Saluran Komunikasi Atau Perantara Yakni Sumber Pesan Dengan Penerima Pesan

Platfrom : Dasar Dari Sebuah Sistem Teknologi Yang Berupa Software Maupun Hardware.

RAM : Kepanjangan Dari Random Access Memory. Secara Teori, RAM Memiliki Prinsip Kerja Yang Sama Dengan Memori Internal (Storage) Yang Biasa Dipakai Untuk Menyimpan Berbagai Foto, Video, Aplikasi, Dan Lain Sebagainya Di Ponsel.

Registrasi : Proses, Cara, Perbuatan Mendaftar (Mendaftarkan); Pencatatan Nama, Alamat, Dan Sebagainya Kedalam Daftar

Service : Setiap Kegiatan Yang Diperuntukkan Dan Ditujukan Untuk Member Kepuasan Melalui Pelayanan Yang Diberikan Seseorang Secara Memuaskan

Siaran : Pesan Atau Rangkaian Pesan Dalam Bentuk Suara, Gambar, Atau Suara Dan Gambar Atau Yang Berbentuk Grafis, Karakter, Baik Yang Bersifat Interaktif

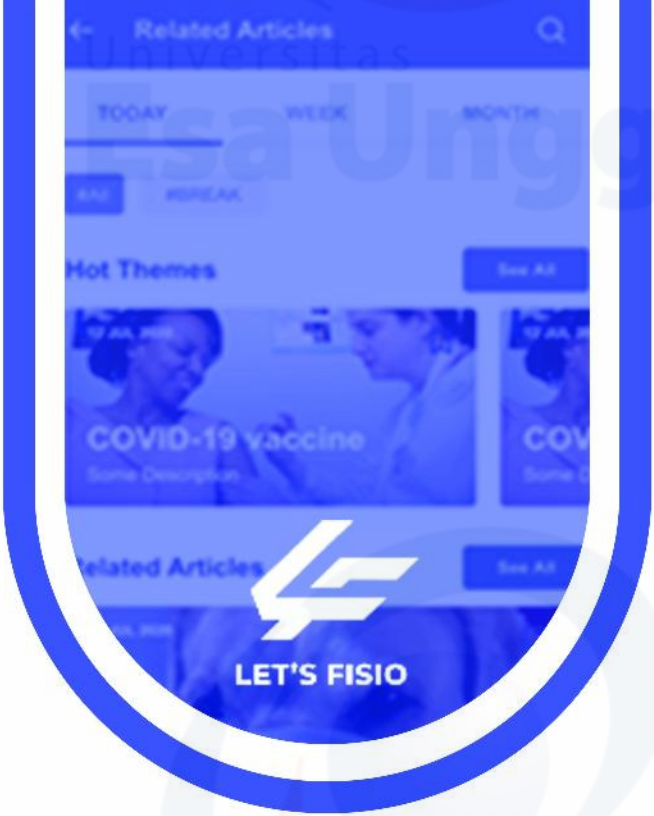
	Maupun Tidak, Yang Dapat Diterima Melalui Perangkat Penerima Siara
Sinkronisasi	: Proses Pengaturan Jalannya Beberapa Proses Pada Saat Yang Bersamaan.
Studio	: Suatu Tempat Di Mana Seorang Seniman Bekerja
Teknologi	: Sebagai Ilmu Pengetahuan Yang Mempelajari Mengenai Keterampilan Dalam Menciptakan Alat Hingga Metode Pengolahan Guna Membantu Menyelesaikan Berbagai Pekerjaan Manusia.
Update	: Atau Peningkatan Merupakan Proses Menggantikan Produk Dengan Versi Yang Lebih Baru Dari Produk Yang Sama.

PROFIL PENULIS

Ir. Nizirwan Anwar, MT, MET, IPM, AER (on-process) adalah Dosen Tetap Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Esa Unggul dibawah naungan Yayasan Pendidikan Kemala Bangsa (YPKB). Penulis lahir di Kota Bandung tanggal 24 Juli 1964, menyelesaikan jenjang S1 dari Program Studi Fisika Fakultas MIPA Universitas Padjadjaran Bandung tahun 1989 dan melanjutkan Program Studi Teknik Elektro Fakultas Teknik (dh. Program Studi Pascasarjana) Universitas Indonesia Jakarta menyelesaikan studi tahun 1995. Riwayat singkat penulis berkarir sebagai pendidik (dosen) dalam melaksanakan dan mengabdikan Tri Dharma Perguruan Tinggi 33 tahun, Alhamdulillah telah menghasilkan beberapa karya ilmiah publikasi pada tingkat nasional dan internasional bereputasi serta ter-indeks (focus area literature: Internet of Things, Kriptografi, Bibliometric dan Data Science). Hingga saat ini tercatat aktif dalam beberapa organisasi profesi (IAII, ADI, ACM, PII, APTIKOM, IAENG), assessor BKD dan sebagai reviewer pada jurnal terakreditasi Arjuna, Komite Ilmiah (Nasional/ Internasional).

Dr. (Cand) Dewanto Rosian Adhy, ST, MT., adalah Dosen Tetap Program Studi Teknik Informatika Sekolah Tinggi Teknologi YBS Internasional dibawah naungan Yayasan Islam Bojong. Penulis lahir di Kota Solo tanggal 4 November 1971 1964, menyelesaikan jenjang S1 dari Departemen Teknik Elektro Institut Teknlogi Bandung tahun 1994 dan melanjutkan Departemen Teknik Elektro Institut Teknlogi Bandung tahun 1994. Riwayat singkat penulis berkarir sebagai pendidik (dosen) dalam melaksanakan dan mengabdikan Tri Dharma Perguruan Tinggi 22 tahun, Alhamdulillah telah menghasilkan beberapa karya ilmiah publikasi pada tingkat nasional dan internasional bereputasi dan ter-indeks (Internet of Things, Kriptografi, dan Data Science). Hingga saat ini tercatat aktif dalam beberapa organisasi profesi dan sebagai reviewer pada jurnal terakreditasi Arjuna, Komite Ilmiah (Nasional/Internasional).

Dr. Jerry Maratis, S.Ft, M.Fis. adalah Dosen Tetap Program Studi Fisioterapi Fakultas Fisioterapi Universitas Esa Unggul dibawah naungan Yayasan Pendidikan Kemala Bangsa (YPKB). Penulis lahir di Kota Palembang tanggal 17 Agustus 1977, menyelesaikan jenjang S1 dari Program Studi Fisioterapi Fakultas Fisioterapi Universitas Esa Unggul Jakarta tahun 2013, jenjang S2 Program Studi Fisioterapi Fakultas Fisioterapi Universitas Udayana Jakarta tahun 2015, dan tahun 1989 dan melanjutkan Program Studi Teknik Elektro Fakultas Teknik (dh. Program Studi Pascasarjana) Universitas Indonesia Jakarta menyelesaikan studi tahun 1995. Riwayat singkat penulis berkarir sebagai pendidik (dosen) dalam melaksanakan dan mengabdikan Tri Dharma Perguruan Tinggi 33 tahun, Alhamdulillah telah menghasilkan beberapa karya ilmiah publikasi pada tingkat nasional dan internasional bereputasi serta ter-indeks (focus area literature: Internet of Things, Kriptografi, Bibliometric dan Data Science). Hingga saat ini tercatat aktif dalam beberapa organisasi profesi (IAII, ADI, ACM, PII, APTIKOM, IAENG), assessor BKD dan sebagai reviewer pada jurnal terakreditasi Arjuna, Komite Ilmiah (Nasional/Internasional).



Perkumpulan Rumah Cemerlang Indonesia
Pondok Karisma Residence
Jalan Raflesia VI D.151
Panglayungan, Cipedes Tasikmalaya - 085223186009
http://rcipress.rcipublisher.org/

ISBN 978-623-448-393-2 (PDF)