

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 TINJAUAN PUSTAKA

Tinjauan pustaka merupakan acuan utama pada penelitian ini berupa beberapa studi yang pernah dilakukan yang berkaitan dengan penelitian ini. Penelitian tersebut diantaranya sebagai berikut

Penelitian tentang Aplikasi *daily note* pengingat kegiatan penting yang telah dilakukan oleh Nur Hikmah Maulida (2021) aplikasi untuk mencatat segala kegiatan dan dalam aplikasi Daily Note ini Juga terdapat menu alarm, kalender, *note* dan *about*. Metodologi penelitian yang digunakan adalah metode *waterfall* sebagai metodologi pengembangan aplikasi. Mulai dari tahapan *Requirement Analysis, Design, Implementation, Verification dan Maintenance*.

Penelitian tentang *Reminder* Pengontrolan Perawatan Gigi Berbasis *Android* yang telah dilakukan oleh A.Astuty Indriani dkk (2019). merancang aplikasi berbasis *android* untuk membantu dokter maupun pasien untuk mengingat jadwal pengontrolan perawatan gigi dengan menggunakan fitur notifikasi.

Penelitian tentang *Perancangan Aplikasi Todolist* Berbasis *Android* Menggunakan *Flutter Sdk* Dan *Database Sqlite* yang dilakukan oleh Rifky Azmi, Rahmawati Rahmawati pada tahun 2021, aplikasi *To-Do-List* untuk mengimplementasikan kemudahan yang akan didapat oleh setiap pengguna *smartphone* salah satunya pembuatan catatan agenda penjadwalan yang akan menampilkan notifikasi pada waktu yang telah ditentukan, sistem ini

menggunakan *Android Studio* dan *Visual Studio Code* sebagai *integrated Development Environment(IDE)* dan basis data *SQLite*.

Penelitian tentang Perancangan dan Pembuatan Aplikasi *Student Reminder* Berbasis *Android* yang di lakukan oleh Robertus Pambudi, Kristo Radion (2018), aplikasi ini menggunakan *android studio developer* dan menggunakan bahasa pemrograman *java*. *Software android* digunakan untuk membuat tampilan pada aplikasi, membuat fungsi untuk mengambil data dari PHP Dan membuat *auto synchronize* ke *google* kalender. PHP digunakan untuk mengambil nilai dari *database* dan menampilkan nilai tersebut di tampilan *web*. *Json* digunakan untuk membaca data dari *php* ke *java* di *android*. *SQL* digunakan untuk membuat tabel database dan relasinya dengan tabel lainnya

Penelitian tentang Aplikasi Pengingat Jadwal Imunisasi Pada Puskesmas Kebun Handil Kota Jambi Berbasis *Android* yang di lakukan oleh Achmad Tasyrif Havaso dkk(2019), aplikasi pengingat jadwal imunisasi pada puskesmas Kebun Handil kota Jambi membantu untuk mengingat jadwal imunisasi serta dapat menampilkan informasi tumbuh kembang anak vaksin , dan informasi tentang KIPPI (Kejadian Ikutan Pasca Imunisasi) serta memberikan petunjuk tentang praktek dokter terdekat dengan lokasi orang tua. Dan juga informasi yang dihasilkan pada aplikasi ini adalah jadwal.

Tabel 2. 1 Perbandingan penelitian

Penulis	Objek	Teknologi	Interfac e	Hasil	Perbedaan
Nur Hikmah Maulida (2021)	Aplikasi daily note peringat kegiatan penting	Android, java	Aplikasi Android	Aplikasi daily note peringat kegiatan penting untuk mempermudah untuk menyimpan catatan penting nya dan melihat calendernya dalam satu aplikasi	Untuk mengingat kegiatan penting sehari- hari

A.Astuty Indriani dkk (2019)	Reminder Pengontrolan Perawatan Gigi Berbasis Android	Android, JavaScript, Dan Firebase	Aplikasi Android	Aplikasi reminder untuk pengontrolan perawatan gigi berbasis android	Menggunakan SQLite sebagai database dan sebagai aplikasi pengingat minum obat
Rifky Azmi, Rahmawati Rahmawati (2021)	Perancangan Aplikasi Todo List Berbasis Android Menggunakan Flutter Sdk Dan Database Sqlite	Flutter, SQLite	Android	aplikasi pencatatan dan penjadwalan agenda pengguna	Menggunakan flutter di peruntukan untuk pencatatan agenda sehari-hari
Robertus Pambudi, Kristo Radion (2018)	Perancangan dan Pembuatan Aplikasi	Java, PHP, Mysql	Android, Website	Aplikasi student reminder yang menggunakan android	pembuatan aplikasi ini menggunakan android studio developer dan

	Student Reminder Berbasis Android			dan juga website, website digunakan untuk admin menginput jadwal dan android digunakan oleh para siswa menerima pengingat	menggunakan bahasa pemrograman java. membuat fungsi untuk mengambil data dari PHP Dan membuat auto synchronize ke google kalender kemudian Json digunakan untuk membaca data dari php ke java di android.
Achmad Tasyrif Havaso (2019)	Aplikasi Pengingat Jadwal Imunisasi Pada Puskesmas Kebun	Android	Aplikasi Android, website	Aplikasi pengingat jadwal imunisasi berbasis android yang digunakan di	Pengingat jadwal imunisasi yang dimana menggunakan web sebagai halaman untuk

	Handil Kota Jambi Berbasis Android			Puskesmas Handil Kota Medan	admin menginput jadwal dan android untuk para orang tua menerima jadwal
Marsel Lilipory (2022)	Aplikasi pengingat jadwal minum obat berbasis android	android	Kotlin, SQLite	Aplikasi pengingat jadwal minum obat	Aplikasi pengingat jadwal minum obat menggunakan kotlin

2.2 DASAR TEORI

2.2.1 Penjadwalan minum obat

Obat umumnya diberikan dengan beberapa jenis aturan pakai, misalnya 3 kali dalam sehari, dengan pembagian waktu yang tepat. Hal ini berarti obat diminum setiap 8 jam sekali dalam 1 hari, sebagai contoh, bila dosis terakhir pada jam 11 malam. Minumlah obat pada waktu yang sama setiap harinya.

Penting untuk menghabiskan obat yang sudah diresepkan oleh dokter dan jangan berhenti untuk mengkonsumsinya tanpa seizin dokter, meskipun sudah merasa lebih sehat, hal ini dilakukan untuk mencegah gejala penyakit muncul Kembali. (dr. Kevin Adrian – 12 februari 2020).

2.2.2 KISS “Keep It Simple, Short”, Principle

KISS adalah prinsip desain yang berfokus pada proses desain produk yang sederhana dan tidak rumit, sehingga desain produk yang lebih baik dan mudah dimengerti, baik itu untuk user maupun tim yang mengembangkannya dan juga meningkatnya *user experience*.

Karena persyaratan produk yang sederhana biasanya juga memiliki komponen yang tak begitu rumit, alhasil pasti lebih mudah bagi para end-user untuk menggunakannya. Dengan desain yang mudah dipahami dan mementingkan apa yang fundamental saja,

secara otomatis tentu saja pengalaman user saat membuka aplikasi atau website pasti jadi lebih menyenangkan.

2.2.3 SOLID

Solid mengacu kepada *keep it simple*, yang mana adalah sebuah akronim dari lima prinsip *object-oriented design* (OOD) . Prinsip ini biasa diterapkan pada saat berkecimpung dalam pemrograman berorientasi objek.

Kelima prinsip ini adalah praktek dalam mengembangkan sebuah program dengan mempertimbangkan pemeliharaan serta pengembangan lebih lanjut agar kode mudah dirawat, mudah dimengerti serta fleksibel.

1. *S - Solid-Responsibility Principle*

Setiap kelas hanya punya satu tugas. Sehingga kelas itu cuman punya satu alasan untuk berubah,yakni merubah satu tugas yang diberikan padanya.

2. *O - Open-Closed Principle*

Kelas yang sudah ditulis dan disepakati behaviornya harus gampang diwariskan , dan tidak boleh diubah-ubah lagi.

Cara mengubahnya dengan mewariskan ke kelas turunan yang punya behavior baru.

3. *L - Liskov Substitution Principle*

Kelas turunan harus bisa menjadi pengganti kelas parent-nya. Tidak boleh methodnya di *override* lalu dikosongkan/lempar eksepsi (*degenerate implementation*).

4. *I - Interface Segregation Principle*

satu *interface* Tidak boleh menyediakan semua *service (god class)* bagi semua jenis client objek. Satu interface mempunyai tugas spesifik untuk tiap klien nya

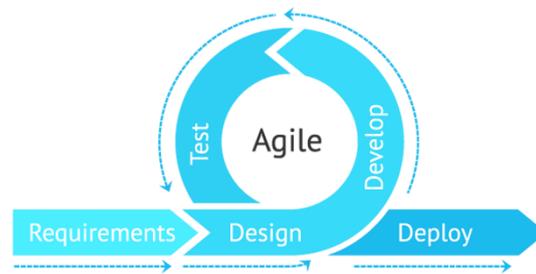
5. *D - Dependency Inversion Principle*

Modul level-tinggi tidak bergantung pada modul level rendah. Keduanya bergantung pada abstraksi. Sementara abstraksi tidak bergantung kepada detail. Malah detail yang harus bergantung pada abstraksi.

Mengadopsi prinsip ini dapat membantu dalam menghindari bad code, membantu dalam refactoring code serta pengembangan aplikasi secara Agile atau Adaptive. (Robert C. Martin)

2.2.4 Metode Agile

Short mengacu pada metode agile, dimana agile adalah metode pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun.



- a. Requirments
- b. Design
- c. Development
- d. Test
- e. Deployment

2.2.5 Android

Menurut Hendra Nugraha Lengkong, *Android* merupakan salah satu *platform* dari perangkat *Smartphone*. Salah satu keutamaan dari *Android* yaitu lisensinya bersifat terbuka (*open source*) dan gratis (*free*) sehingga bebas untuk dikembangkan karena tidak ada biaya royalti maupun didistribusikan dalam bentuk apapun. Hal ini memudahkan para *programmer* untuk membuat aplikasi baru di dalamnya. Berikut merupakan perkembangan android dari awal pertama kemunculan sampe dengan sekarang

1. Android Astro 1.0 (Alpha)

Versi Android 1.0 ini dirilis pada 23 September 2008 dengan nama Alpha yang digunakan pada ponsel jenis HTC Dream.

2. Android Bender 1.1 (Beta)

Pada versi 1.1 rilisan 9 Februari 2009, google play store meluncur dengan nama Android market.

3. Android Cupcake 1.5

Pada 27 April 2009, Android versi 1.5 dengan nama cupcake dirilis secara komersil bersama fitur on-screen keyboard.

4. Android Donut 1.6

Android merilis versi donut 1.6 pada 15 September 2009 dengan mendukung teknologi CDMA/EVDO, 802.1 x, VPNs.

5. Android Eclair 2.0 – 2.1

Kemunculan Eclair 2.0 – 2.1 pada 26 Oktober 2009, menggeser fungsi peta konvensional dengan fitur navigasi Google maps.

6. Android Froyo 2.2

Versi froyo atau frozen yoghurt rilis pada 20 Mei 2010 dengan berbagai fitur perbaikan pada sistem yang sudah ada.

7. Android Gingerbread 2.3

Gingerbread 2.3 rilis pada 6 Desember 2010 dengan mengutamakan beberapa pembaharuan.

8. Android Honeycomb 3.0/3.1

Android meluncurkan versi honeycomb 3.0/3.1 pada 22 Februari 2011 untuk penggunaan OS Android pada tablet.

9. Android Ice Cream Sandwich 4.0

Fitur yang ada pada versi tablet dimasukkan dalam Ice Cream Sandwich 4.0 yang diluncurkan pada 19 Oktober 2011.

10. Android Jelly Bean 4.1/ 4.2/ 4.3

Google now yang berfungsi untuk voice assistant diperkenalkan bersama dengan peluncuran versi Jelly Bean 4.1 di tahun 2012.

Sedangkan fitur photo sphere, daydream dan lain-lain ada di versi 4.2, yang mana semua versi dimutakhirkan pada versi 4.3

11. Android Kitkat 4.4

Key Lime Pie atau kitkat 4.4 yang rilis pada 31 Oktober 2013 hanya akan berjalan optimal pada perangkat dengan RAM minimal 512 MB.

12. Android Lollipop 5.0

12 November 2014 adalah waktu peluncuran Lollipop 5.0 yang membuat perubahan pada desain User Interface.

13. Android Marshmallow 6.0

Marshmallow 6.0 muncul bersama fitur canggih seperti search bar, sensor sidik jari dan sebagainya pada 5 Oktober 2015.

14. Android Nougat 7.0

Nougat 7.0 membuat perubahan besar sejak muncul pada 23 Agustus 2016 dengan 63 emoji baru dan multi-window.

15. Android Oreo 8.0

Oreo 8.0 hadir pada 21 Agustus 2017 dengan fitur-fitur yang multitasking serta penampilan UI yang lebih rapi.

16. Android Pie 9.0Pie 9.0

dirilis pada 6 Agustus 2018 dengan membawa smart reply, digital wellbeing, adaptive battery dan lain-lain.

17. Android 10

Salah satu alasan versi ke 10 dinamakan Android 10 dikarenakan tidak adanya nama makanan yang berawalan Q.

Untuk versi Android 10 keatas sudah tidak menggunakan nama-nama makanan seperti versi sebelumnya.

Android versi 10 diluncurkan pada 3 September 2019, fitur barunya berupa navigasi layar utama, izin untuk akses media, akses ke background dll.

18. Android 11

Android 11 diluncurkan pada tanggal 8 September 2020 dengan membawa fitur yang lebih lengkap dari generasi sebelumnya.

Contoh fitur yang dimilikinya seperti chat bubbles, perekaman layar, notifications, kontrol izin dan sebagainya.

19. Android 12

Android 12 resmi meluncur pada tanggal 4 Oktober 2021, setelah pertama kali diumumkan Google pada tanggal 18 Februari 2021.

Fitur baru yang dihadirkan dalam versi Android terbaru ini seperti kemudahan berbagi WiFi, support gambar AVIF, Material You, tangkapan layar bergulir, mode satu tangan, zoom area, extra dim dan masih banyak lainnya.

20. Android 12L

Android 12L direncanakan meluncur pada awal tahun 2022, updatenya khusus untuk perangkat smartphone lipat, tablet, layar besar dan chromebook. Versi Android 12L ini akan lebih mudah menyesuaikan dengan layar pengguna yang lebih besar

2.2.6 Aplikasi *mobile*

Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi *mobile* juga dikenal sebagai aplikasi yang dapat diunduh dan memiliki fungsi tertentu sehingga menambah

fungsionalitas dari perangkat *mobile* itu sendiri. Aplikasi *mobile* kini banyak digemari oleh masyarakat, tentunya dikalangan para entrepreneur dalam memanfaatkan teknologi. Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemari oleh hampir 70% pengguna telepon seluler, karena dengan memanfaatkan adanya fitur *game*, musik *player*, sampai *video player* membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun.

Untuk mendapatkan *mobile application* yang diinginkan, user dapat mengunduhnya melalui situs tertentu sesuai dengan sistem operasi yang dimiliki. *Google Play* dan *iTunes* merupakan beberapa contoh dari situs yang menyediakan beragam aplikasi bagi pengguna *Android* dan *IOS* untuk mengunduh aplikasi yang diinginkan. Pengembangan aplikasi *mobile* bisa jadi cepat dan bisa jadi lambat, tergantung kerumitan dari kebutuhan aplikasi tersebut, harga juga sangat bervariasi namun tentu aplikasi *mobile* jauh lebih mahal dibanding dengan aplikasi web. Aplikasi ini dapat diakses melalui 2 perangkat nirkabel seperti pager, telepon seluler dan PDA, smartphone, Smartwatch, atau tablet PC. (Iwan Ady Prabowo, M.Kom)

2.2.7 UML

UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML

diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997.

UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Hal itu termasuk faktor-faktor *scalability*, *robustness*, *security*, dan sebagainya.

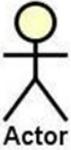
1. Menyatukan informasi-informasi terbaik yang ada dalam pemodelan.
2. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
3. Tidak hanya menggambarkan model sistem *software* saja, namun dapat memodelkan sistem berorientasi objek.
4. Mempermudah pengguna untuk membaca suatu sistem.
5. Berguna sebagai *blueprint*, jelas ini nantinya menjelaskan informasi yang lebih detail dalam perancangan berupa koding suatu program.

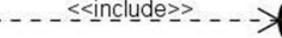
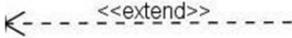
Diagram UML

Setiap sistem yang kompleks seharusnya bisa dipandang dari sudut yang berbeda – beda sehingga bisa mendapatkan pemahaman secara menyeluruh . Untuk upaya tersebut UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya statis atau dinamis. Diagram sering digunakan dalam UML itu adalah :

1. Use case diagram

Diagram ini bersifat statis. Diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

Simbol	Deskripsi
<p data-bbox="507 842 628 875"><i>Use case</i></p> 	<p data-bbox="839 842 1353 1240"><i>Use Case</i> fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja</p>
<p data-bbox="507 1352 679 1386">Aktor / actor</p> 	<p data-bbox="839 1308 1350 1924"><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i></p>

<p><i>Asosiasi/ association</i></p> 	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p>
<p><i>Include</i></p> 	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> contohnya adalah pemanggilan sebuah fungsi program</p>
<p><i>Extend</i></p> 	<p><i>Extend</i>, merupakan perluasan dari usecase lain jika kondisi atau syarat terpenuhi</p>

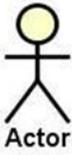
2. Sequence diagram

Sequence diagram adalah sebuah diagram yang digunakan untuk menjelaskan, dan menunjukkan urutan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Tujuan utama dari pembuatan diagram urutan adalah untuk mengetahui

urutan kejadian yang dapat menghasilkan output yang diinginkan, dan juga menggambarkan aliran data lebih detail, termasuk data atau perilaku yang diterima atau dikirimkan.

Ada beberapa komponen-komponen utama yang digunakan dalam pembuatan sequence diagram, yaitu,

a. Actor

	<p>Komponen pertama yaitu <i>actor</i>, komponen ini menggambarkan seorang pengguna(user) yang berada diluar sistem dan sedang berinteraksi dengan sistem. Dalam sequence diagram aktor biasanya digambarkan dengan simbol <i>stick figure</i></p>
<p>Activation box</p> 	<p>Komponen <i>activation box</i> ini mempresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugasnya, semakin lama waktu yang diperlukan, maka secara otomatis activation boxnya akan lebih Panjang. Komponen</p>

	ini digambarkan dengan persegi Panjang
<p>Lifeline</p> 	Komponen ini digambarkan dengan bentuk <i>lifeline</i> . Ini biasanya memiliki kotak yang berisi objek yang memiliki fungsi untuk menggambarkan aktivitas dari objek
<p>Object</p> 	kotak yang berisikan nama dari objek dengan garis bawah. Biasanya objek berfungsi untuk mendokumentasikan perilaku sebuah objek pada sebuah sistem
<i>messages</i>	<i>messages</i> atau pesan ini menggambarkan komunikasi antar objek. Messages ini biasanya muncul secara berurutan pada lifeline. Komponen messages ini direpresentasikan dengan anak panah. Inti dari sebuah diagram

	urutan terdapat pada komponen lifeline dan messages ini.
--	--

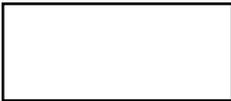
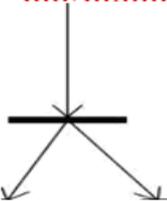
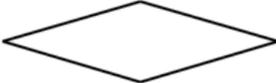
3. *Activity* Diagram

Activity diagram atau diagram aktivitas adalah diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan dari sebuah proses pada sebuah sistem yang digambarkan secara vertikal. *Activity* diagram merupakan pengembangan dari *use case* yang memiliki alur aktivitas dan juga memiliki komponen-komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah, panah tersebut mengarah ke-urutan aktivitas yang terjadi dari awal hingga akhir.

Fungsi *activity* diagram yaitu

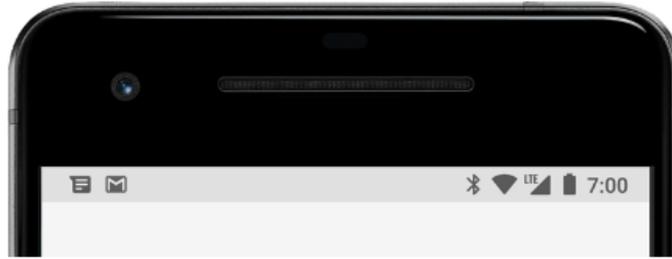
1. Menampilkan urutan aktivitas proses pada sistem
2. Membantu memahami proses secara keseluruhan
3. *Activity* diagram dibuat berdasarkan *use case*
4. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses

Berikut merupakan simbol-simbol yang digunakan dalam *activity* diagram

Symbol	Keterangan
Start point 	Start Point, diletakkan pada pojok kiri atas dan merupakan awal aktivitas
End point 	<i>End Point</i> , akhir aktivitas
Activities 	Activities, menggambarkan suatu proses atau kegiatan bisnis
Fork atau Percabangan 	Fork atau percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel untuk menggabungkan dua kegiatan paralel menjadi satu
Decision Points 	Decision points, menggambarkan pilihan untuk pengambilan keputusan, true atau false

2.2.8 Notifikasi

Notifikasi adalah pesan yang ditampilkan oleh Android di luar UI aplikasi Anda untuk memberikan pengingat, komunikasi dari orang lain, atau informasi tepat waktu lainnya dari aplikasi. Pengguna dapat mengetuk notifikasi untuk membuka aplikasi Anda atau melakukan tindakan langsung dari notifikasi (google developer)



gambar 2. 1 Status bar notification

Pengguna dapat menggeser status bar ke bawah untuk membuka panel samping notifikasi, dan di sini pengguna dapat melihat detail selengkapnya dan mengambil tindakan terkait notifikasi.

2.2.9 Kotlin

Kotlin adalah Bahasa pemrograman modern, disajikan secara statis yang berjalan di *platform java virtual machine(JMV)*. *Kotlin* menggunakan *compiler LLVM* yang artinya, dapat dikompilasi ke dalam kode *JavaScript*, *Kotlin* merupakan aplikasi yang cocok bagi para pengembang aplikasi dalam membangun aplikasi *android*, *kotlin* merupakan Bahasa pemrograman yang bisa dikatakan mudah karena dilengkapi dengan fitur-fitur yang menarik, seperti :

1. Concise

Kode atau Bahasa penulisan yang digunakan *kotlin* lebih sederhana dibandingkan dengan *java*

2. Versatile

Kotlin adalah Bahasa pemrograman yang diturunkan dari *java*, maka bisa digunakan untuk mengembangkan aplikasi baik *mobile* maupun *website*.

3. *Safe*

Setiap kode yang diproses terjamin keamanannya dan bebas *error*, seperti bebas dari NPE.

4. *Interoperable*

Kotlin memang berbeda dari Bahasa pemrograman lainnya yang merupakan turunan dari java.

2.2.10 SQLite

SQLite adalah perpustakaan perangkat lunak yang menerapkan engine database SQL secara mandiri, tanpa perlu melakukan konfigurasi, dan bersifat transaksional. SQLite adalah *engine database* SQL yang paling banyak digunakan didunia. SQLite merupakan proyek yang bersifat *public* domain yang dikerjakan oleh D. Richard Hipp. (SQLite Organization, 2000).

