

BAB II

LANDASAN TEORI

2.1 Kalender

Kalender adalah sebuah sistem untuk memberi nama pada sebuah periode waktu seperti hari dan bulan. kalender dipergunakan sebagai alat yang menjadi penanda perubahan yang sehari-hari kita kenal sebagai tanggal. Tanggal ini bisa didasarkan dari gerakan-gerakan benda angkasa seperti matahari dan bulan. Kalender juga dapat mengacu kepada alat yang mengilustrasikan sistem tersebut.

Kalender yang sering digunakan masyarakat pada umumnya antara lain kalender masehi dan kalender hijriyah. Kalender masehi merupakan kalender yang menjadi acuan internasional untuk menentukan tanggal penting. Sedangkan kalender hijriyah merupakan kalender yang digunakan umat islam untuk menentukan tanggal-tanggal penting umat islam.

2.1.1 Kalender Hijriyah

Kalender Hijriyah merupakan kalender yang perhitungannya didasarkan pada pergerakan bulan atau Qomariyah. Setelah wafatnya Nabi Muhammad, diusulkan kapan dimulainya Tahun 1 Kalender Islam (Habib, 2008).

Pada tahun 638 M, khalifah Umar bin Khatab menetapkan awal patokan penanggalan Islam adalah tahun dimana hijrahnya Nabi Muhammad dari Mekkah ke Madinah (Noer, 2007). Penentuan awal patokan ini dilakukan setelah menghilangkan seluruh bulan-bulan tambahan dalam periode 9 tahun. Tanggal 1 Muharam Tahun 1 Hijriah bertepatan dengan tanggal 16 Juli 622, dan tanggal ini bukan berarti tanggal hijrahnya Nabi Muhammad (Syaputra, 2012). Peristiwa hijrahnya Nabi Muhammad

terjadi bulan September 622. Dokumen tertua yang menggunakan sistem Kalender Hijriah adalah papirus di Mesir pada tahun 22 H.

Kalender Hijriyah memiliki 12 bulan dalam setahun. Dengan menggunakan siklus sinodik bulan, bilangan hari dalam satu tahunnya adalah $12 \times 29,53059$ hari = 354,36708 hari. Hal inilah yang menjelaskan 1 tahun Kalender Hijriah lebih pendek sekitar 11 hari dibanding dengan 1 tahun Kalender Masehi.

Jumlah hari dalam satu bulan dalam Kalender Hijriah bergantung pada posisi bulan, bumi dan matahari. Usia bulan yang mencapai 30 hari bersesuaian dengan terjadinya bulan baru (*new moon*) di titik apoooge, yaitu jarak terjauh antara bulan dan bumi, dan pada saat yang bersamaan, bumi berada pada jarak terdekatnya dengan matahari. Sementara itu, satu bulan yang berlangsung 29 hari bertepatan dengan saat terjadinya bulan baru di jarak terdekat bulan dengan bumi berada di titik terjauhnya dari matahari. Dari sini terlihat bahwa usia bulan tidak tetap melainkan berubah-ubah antara 29 sampai 30 hari sesuai dengan kedudukan bulan, bumi dan matahari.

Jumlah hari dalam satu tahun hijriah adalah 354 hari atau 355 hari. Pada bulan dzulhijah bisa 29 atau 30 hari tergantung apakah tahun kabisat atau tidak. Adapun tahun-tahun yang mendapatkan tambahan satu hari dalam periode 30 tahun itu adalah tahun-tahun yang angkanya merupakan kelipatan 30 ditambah 2, 5, 7, 10, 13, 15, 18, 21, 24, 26 dan 29 (Ajim, 2015). Pada dasarnya lama hari pada kalender hijriyah seperti pada tabel 2.1 di bawah ini.

Tabel 2.1 Kalender Hijriyah

No	Penanggalan Hijriyah	Lama Hari
1	Muharram	30
2	Safar	29
3	Rabiul Awal	30
4	Rabiul Akhir	29
5	Jumadil Awal	30

6	Jumadil Akhir	29
7	Rajab	30
8	Sya'ban	29
9	Ramadhan	30
10	Syawal	29
11	Dzulkaidah	30
12	Dzulhijjah	29 (30)
Total		354/355

Menurut hisab urfi, dalam kalender hijriah ada 354 hari. Namun sebenarnya, perputaran bulan hakiki selama satu tahun adalah 354,367 hari atau 354 hari 8 jam 44 menit 35 detik ($\text{Periode sideris } 29,53059 \times 12 = 354,367$). Tentunya manusia tidak mungkin menggunakan kalender dengan sisa 0,367 hari tersebut (Musa, 2010). Untuk menyasati hal ini, maka:

1. Peredaran bulan sinodis: 29 hari 12 jam 44 menit 2,8 detik. Angka 2,8 detik diabaikan karena sangat kecil sehingga tidak berarti. Dengan demikian, rata-rata hari dalam satu tahun adalah: $29,5 \text{ hari} \times 12 = 354 \text{ hari } 44 \text{ menit} \times 12 = 528 \text{ menit}$ Jadi, dalam setahun ada 354 hari 528 menit.
2. Berhubung manusia tidak mungkin menggunakan kalender dengan jumlah hari 0,5 maka untuk menyasatinya bilangan pecahan 29,5 hari tersebut dikalikan dengan 2 sehingga menjadi 59 hari (hitungan 2 bulan). 30 hari diberikan kepada bulan ganjil, 29 hari diberikan kepada bulan genap. Sehingga, dalam satu tahun ada 6 bulan yang berjumlah hari 29 dan 6 bulan yang berjumlah hari 30. Apabila dijumlahkan maka akan didapatkan angka 354 hari (jumlah hari dalam satu tahun hisab urfi).
3. Terdapat sisa 44 menit setiap bulan yang akan menjad 528 menit setiap tahun. Dalam waktu 3 tahun, jumlah ini akan menjadi 1 hari lebih ($528 \times 3 = 1548 \text{ menit}$, 1 hari = 1440 menit). Dalam siklus 1 daur (30 tahun) -1 daur dipilih 30

tahun karena apabila 0,367 hari yang merupakan sisa hari setiap tahun dikalikan dengan 30 tahun akan menghasilkan 11,01 hari (dengan angka di belakang koma terkecil)- akan menjadi 15480 menit atau genap 11 hari ($15480 : 1440 = 11$). Sisa 11 hari tersebut didistribusikan ke dalam tahun-tahun selama 1 daur (30 tahun). Masing-masing akan mendapatkan 1 tahun.

Dalam kalender hijriah, intervalnya memang terlihat tidak teratur, namun ada metode tersendiri dalam menetapkan tahun kabisat, yaitu dengan mengalikan bilangan urutan tahun tersebut dengan sisa 0,367. Apabila sisanya lebih dari 0,5 (hari) maka tahun tersebut adalah tahun kabisat. Apabila sisanya kurang dari 0,5 hari, maka tahun tersebut adalah tahun basitah. Sebagai contoh:

- Tahun ke-1 x 0,367 = 0,367 (kurang dari 0,5 maka tahun basitah)
- Tahun ke-2 x 0,367 = 0,734 (lebih dari 0,5 maka tahun kabisat)
- Tahun ke-3 x 0,367 = 1,101 (berhubung 1 hari sudah dipakai di tahun kedua, maka menjadi 0,101, karena kurang dari 0,5 maka basitah)
- Tahun ke-4 x 0,367 = 1,468 (berhubung 1 hari sudah dipakai di tahun kedua, maka menjadi 0,468, karena kurang dari 0,5 maka basitah)
- Tahun ke-5 x 0,367 = 1,835 (berhubung 1 hari sudah dipakai di tahun kedua, maka menjadi 0,835, karena lebih dari 0,5 maka kabisat)

Untuk mengetahui apakah suatu tahun itu kabisat atau basitah, caranya dengan membagi bilangan tahun dengan 30 (1 daur), sisa pembagiannya apabila terdapat pada salah satu angka di atas, maka ia kabisat. Misalkan tahun $1359 : 30 = 45$ dengan sisa 9 tahun, berarti 1359 merupakan tahun basitah. Tahun $1431 : 30 = 47$ dengan sisa 21 tahun, berarti, 1431 merupakan tahun kabisat.

2.1.2 Kalender Masehi

Berbeda dengan kalender hijriyah, perhitungan Kalender Masehi mengacu pada perputaran matahari. Kata Masehi digunakan oleh umat Kristen awal untuk menetapkan hari kelahiran Yesus yang dalam bahasa latin disebut *Anno Domini* (AD) yang berarti “Tahun Tuhan Kita” atau *Common Era/CE* (Era Umum) untuk era Masehi, dan *Before Christ/BC* (sebelum [kelahiran Kristus) atau *Before Common Era / BCE* (Sebelum Era Umum).

Sistem penanggalan yang merujuk pada awal tahun Masehi ini mulai diadopsi di Eropa Barat selama abad ke-8. Awalnya biarawan Katolik, Dionisius Exogous pada tahun 527 M ditugaskan pimpinan Gereja untuk membuat perhitungan tahun dengan titik tolak tahun kelahiran Nabi Isa as. Dan mula-mula dipergunakan untuk menghitung tanggal Paskah berdasarkan tahun pendirian Roma.

Kalender Masehi awalnya hanya berumur 10 bulan yaitu: *Martius* (Maret), *Aprilis* (April), *Maius* (Mei), *Junius* (Juni), *Quintilis* (Juli), *Sextilis* (Agustus), *September* (September), *October* (Oktober), *November* (Nopember), *December* (Desember).

Penanggalan yang terdiri atas 10 bulan kemudian berkembang menjadi 12 bulan. Berarti ada tambahan 2 bulan, yaitu *Januarius* dan *Februarius*. *Januarius* adalah nama dewa *Janus*. Dewa ini berwajah dua, menghadap ke muka dan ke belakang, hingga dapat memandang masa lalu dan masa depan. Karenanya *Januarius* ditetapkan sebagai bulan pertama. *Februarius* diambil dari upacara *Februa*, yaitu upacara semacam bersih kampung atau ruwatan untuk menyambut kedatangan musim semi. Jadilah penanggalan kalender masehi dengan lama hari yang sudah ditentukan seperti tabel 2.2.

Tabel 2.2 Kalender Masehi

No	Penanggalan Masehi	Lama Hari
1	Januari	31
2	Februari	28/29
3	Maret	31
4	April	30
5	Mei	31
6	Juni	30
7	Juli	31
8	Agustus	31
9	September	30
10	Oktober	31
11	November	30
12	Desember	31
Total		365/366

1 tahun masehi sama dengan 365 hari jika tahun itu merupakan tahun basithoh dengan februari berjumlah 28 hari atau 366 hari jika tahun itu merupakan tahun kabisat dengan februari berjumlah 29 hari.

2.1 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang

bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Berikut Fitur Android sehingga tampil beda dari *mobile* biasa

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin virtual Dalvik dioptimalkan untuk perangkat *mobile*
3. Integrated browser berdasarkan engine open *source WebKit*
4. Grafis yang dioptimalkan dan didukung oleh perpustakaan grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
5. SQLite untuk penyimpanan data
6. *Media Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG,PNG, GIF)
7. GSM *Telephony* (tergantung hardware)
8. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
9. Kamera, GPS, kompas, dan *accelerometer* (tergantung hardware)
10. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk Eclipse IDE

Android sendiri, sudah dari awal perancangan telah terpasang pada perangkat *mobile touchscreen* seperti *smartphone* dan komputer tablet. Jadi Android adalah suatu sistem operasi yang berjalan pada smatphone saat ini dan menyesuaikan spesifikasi di kelas *low-end* hingga *high-end*. Hampir semua vendor saat ini mengembangkan produknya dengan sistem operasi Android, karena peminatnya yang semakin meningkat tajam.

2.3 Aplikasi Pendukung

2.3.1 Firebase

Firebase adalah penyedia layanan *cloud* dengan *back-end* sebagai servis yang berbasis di San Fransisco, California. Firebase membuat sejumlah produk untuk pengembangan aplikasi *mobile* ataupun *web*. Firebase di dirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan *cloud database* secara realtime di tahun 2012 (Evangelist, 2015).

Produk utama dari Firebase yakni suatu *database* yang menyediakan API untuk memungkinkan pengembang menyimpan dan mensinkronisasi data lewat *multiple client*. Perusahaan ini diakusisi oleh Google pada Oktober 2014.

Firebase menyediakan realtime *database* dan *backend* sebagai layanan. Suatu aplikasi layanan yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada cloudnya Firebase.

Firebase memiliki banyak library yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, iOS, *Javascript*, *Java*, Objective-C dan Node.JS. *Database* Firebase juga bersifat bisa diakses lewat REST API dan data binding untuk beberapa *framework Javascript* seperti halnya AngularJS, ReactJS, Ember.JS, dan Backbone.JS. REST API tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima push notification dari server. Pengembang juga bisa menggunakan *database* ini untuk mengamankan data mereka menggunakan server Firebase dengan rules yang ada.

Untuk mengakses firebase dapat diperoleh dengan mengunjungi situs <https://www.firebase.com>.

2.3.2 Android Studio

Android Studio adalah sebuah *Integrated Development Environment* (IDE) yang bisa digunakan untuk pengembangan aplikasi Android, dan dikembangkan oleh

Google (Herdi, 2014). Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE *Java* populer, yaitu IntelliJ IDEA. Android Studio direncanakan untuk menggantikan Eclipse ke depannya sebagai IDE resmi untuk pengembangan aplikasi Android (Abdulloh, 2015).

Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai *build environment*. Fitur-fitur lainnya adalah sebagai berikut :

- Menggunakan *Gradle-based build system* yang fleksibel.
- Bisa melakukan *building multiple APK* .
- *Template support* untuk *Google Services* dan berbagai macam tipe perangkat.
- *Layout editor* yang lebih bagus.
- *Built-in support* untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
- *Import library* langsung dari *Maven repository*.

Android studio dapat di download untuk *developer-developer* yang ingin membuat aplikasi android secara gratis dengan mengunjungi link <http://developer.android.com/tools/studio/index.html>

2.3.3 Ionic Framework

Ionic Framework merupakan *framework* aplikasi *mobile* HTML5 lanjutan yang dibangun di atas AngularJS dan Apache Cordova. Ionic dapat digunakan untuk mengembangkan aplikasi *mobile hybrid*. Ionic menggunakan AngularJS, Node.js, SASS sebagai *engine*-nya. Seperti *mobile framework* kebanyakan, Ionic juga memiliki komponen/elemen CSS standar digunakan untuk *smartphone*, seperti button, list, card, form, range, tabs, grid dan lainnya (Ilhami, 2015).

Ionic *Framework* dibuat dan dikembangkan oleh para designer dan *developer* yang memiliki passion tentang teknologi *web*. Dengan memanfaatkan keunggulan AngularJS, Ionic mampu membuat proses development *mobile* app dengan HTML5 menjadi lebih mudah.

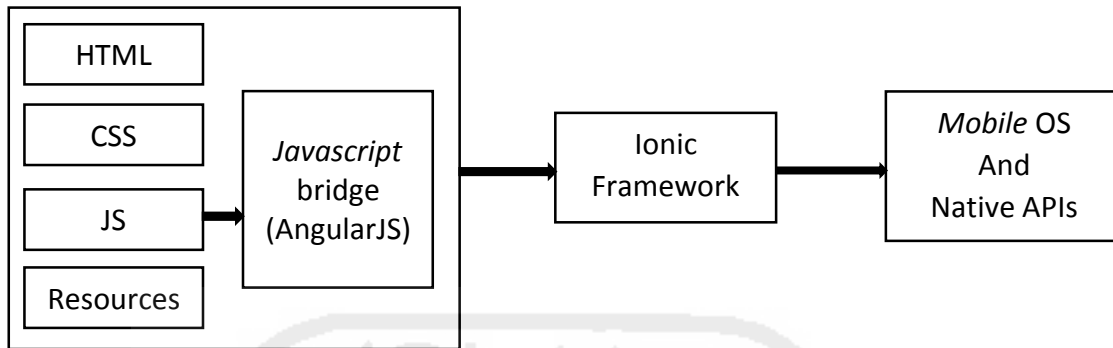
Ionic menyediakan semua fungsi pada development SDK untuk pembuatan aplikasi *native mobile*. Pengguna dapat membangun aplikasi untuk Android maupun iOS, dan dapat disebarluaskan melalui Cordova. Ionic meliputi *mobile component*, *typography*, paradigma interaktif, dan basis tema *extensible*.

Ionic menyediakan komponen kustom dan metode untuk berinteraksi menggunakan Angular. Salah satu komponen tersebut dapat memungkinkan *user* untuk menelusuri daftar ribuan item tanpa mempengaruhi kinerja. Komponen lainnya adalah *scroll-view*, menciptakan *container* yang *scrollable* dimana *user* dapat berinteraksi menggunakan *native-influenced delegate system*.

Selain SDK, Ionic juga menyediakan layanan yang dapat digunakan *developer* untuk mengaktifkan fitur, seperti *push notifications*, *A/B testing*, *analytics*, *code deploys*, dan *automated builds*.

Ion juga menyediakan *Command-Line Interface* (CLI) yang powerful, sehingga para *developer* dapat memulai dan membuat proyek dengan perintah sederhana. CLI juga memungkinkan pengembang untuk menambahkan *plugin* Cordova dan paket *front-end* tambahan, mengaktifkan *push notifications*, *generate app Icons* dan *Splash screens*, serta membangun *binary native*.

Untuk petunjuk cara menggunakan dan melakukan instalasi *framework* ionic dengan cara mengunjungi <http://ionicframework.com/docs/guide/>. Implementasi penggunaan ionic *framework* dapat dilihat pada gambar 2.1 di bawah ini.



Gambar 2.1 Ionic Framework

2.3.4 Node JS

Node.js adalah sebuah perangkat lunak yang didisain untuk mengembangkan aplikasi berbasis *Web*. Node.js dieksekusi sebagai aplikasi *server*. *Platform* ini menggunakan bahasa pemrograman *javascript* dan menggunakan teknik *non-bloking* untuk mempercepat proses. Teknik *non-bloking* adalah sebuah metode penyelesaian sebuah fungsi (Maqrikisandi, 2015).

Javascript merupakan bahasa pemrograman yang lengkap hanya saja selama ini di pakai sebagai bahasa untuk pengembangan aplikasi *web* yang berjalan pada sisi *client* atau browser saja. Tetapi sejak ditemukannya Node.js oleh Ryan Dhal pada tahun 2009, *Javascript* bisa digunakan sebagai bahasa pemrograman di sisi server sekelas dengan PHP, ASP, C#, Ruby dll dengan kata lain Node.js menyediakan *platform* untuk membuat aplikasi *Javascript* dapat dijalankan di sisi server.

Sekarang dengan berjalannya *Javascript* di server lalu apa keuntungan yang di peroleh dengan mempelajari Node.js, kurang lebih seperti berikut ini :

1. Pengembang hanya memakai satu bahasa untuk mengembangkan aplikasi lengkap *client* & server sehingga mengurangi *Learning Curve* untuk mempelajari bahasa server yang lain.
2. Sharing kode antara *client* dan server atau istilahnya *code reuse*.

3. *Javascript* secara *native* mendukung JSON yang merupakan standar transfer data yang banyak dipakai saat ini sehingga untuk mengkonsumsi data-data dari pihak ketiga pemrosesan di Node.js akan sangat mudah sekali.
4. *Database* NoSQL seperti MongoDB dan CouchDB mendukung langsung *Javascript* sehingga *interfacing* dengan *database* ini akan jauh lebih mudah.
5. Node.js memakai V8 yang selalu mengikuti perkembangan standar ECMAScript, jadi tidak perlu ada kekhawatiran bahwa browser tidak akan mendukung fitur-fitur di Node.js.

Karena pada dasarnya Ionic dibuat dengan *javascript*, maka kita juga butuh Node.js. Node.js ini nanti digunakan untuk menjalankan perintah-perintah Ionic seperti *SCSS compiling*, *serve app* untuk *testing*, *generate icon resources*, dan banyak lainnya.

2.3.5 Android SDK

Android-SDK merupakan *tools* bagi para programmer yang ingin mengembangkan aplikasi berbasis google android (Haidibarasa, 2013). Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, handset emulator dokumentasi, contoh kode, dan tutorial.

Untuk membuat aplikasi android, instalasi SDK Android juga diperlukan. Android SDK menyediakan API *library* dan alat-alat pengembang yang diperlukan untuk membangun, menguji, dan *debug* aplikasi Android. Untuk versi yang sekarang, dianjurkan untuk menginstall API 22. Walaupun dianjurkan install versi 22, namun kita tetap bisa menjalankannya di Android dengan versi minimal 16.

2.3.6 Java JDK 7

JDK adalah singkatan dari *Java Development Kit* yaitu software yang digunakan untuk membangun aplikasi-aplikasi java. Tanpa JDK kita tidak akan bisa membangun atau membuat berbagai macam aplikasi java. JDK berisi sekumpulan

command line *tool* untuk menciptakan program *java*. JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis *java*.

Berikut adalah beberapa komponen utama JDK :

1. Kompulator (javac)
2. Interpreter program java (java)
3. *Applet viewer* (appletviewer)
4. *Debugger* (jdb)
5. *Class file disassembler* (javap)
6. *Java Archive* (jar)
7. *Documentation generator* (javadoc)
8. Applet demo
9. Kode sumber *Java* API

JDK berisi *Java Runtime Environment* (JRE) dan semua alat yang diperlukan untuk mengkompilasi aplikasi Java. Hal ini diperlukan untuk mengkompilasi dan menjalankan Java, yang diperlukan untuk menjalankan android.

Java JDK 7 dapat di download dengan mengunjungi situs <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>.

2.3.7 Apache Ant 1.9.6

Apache Ant adalah software berbasis *Java* yang digunakan untuk keperluan build *tool*. Sebagai build *tool*, tugas utama dari Ant adalah untuk menyediakan sumber daya dan melaksanakan proses yang memungkinkan membangun suatu software dari bentuk *source code* menjadi aplikasi yang siap didistribusikan atau bahkan telah diinstall pada *remote server*.

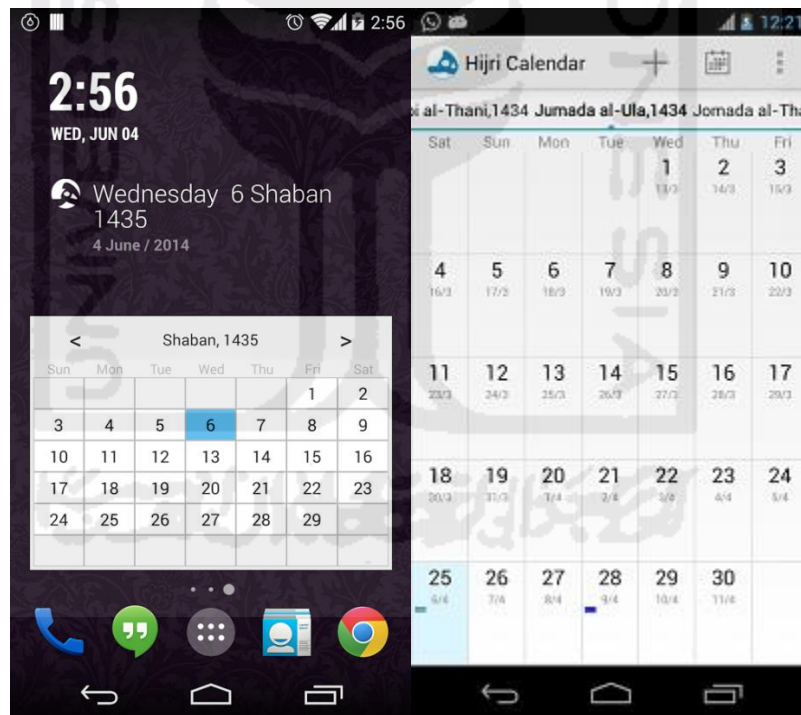
Apache ant 1.9.6 dapat di download secara gratis untuk umum dengan cara mengunjungi situs <http://ant.apache.org/bindownload.cgi>.

2.4. Aplikasi Sejenis

2.4.1 Hijri Calendar by Alwan Azahrani

Hijri *calendar* yang dibuat oleh Alwan Azahrani merupakan sebuah aplikasi kalender hijriyah yang dapat mengkonversi tanggal pada kalender hijriyah menjadi masehi. Aplikasi ini juga dapat menambahkan *event* sesuai keinginan *user* dan juga sudah bisa sinkron ke *google calendar*. Namun *event* yang terdapat pada aplikasi ini tidak menyesuaikan dengan *event* yang ada di indonesia. *Event* islam juga tidak menyinggung ibadah-ibadah sunah.

Gambar tampilan untuk aplikasi Hijri *Calendar* yang dibuat oleh Alwan Azahrani dapat dilihat pada gambar 2.2 di bawah ini.



Gambar 2.2 Hijri *Calendar* by Alwan Azahrani

2.4.2 Hijri Calendar by Kodelokus Cipta Aplikasi

Hijri Calendar yang dibuat oleh Kodelokus Cipta Aplikasi adalah aplikasi untuk konversi antara tanggal masehi dan hijriyah, berfungsi juga sebagai kalender Islam yang menampilkan hari-hari besar Islam. Fitur-fitur yang terdapat pada aplikasi ini antara lain: Slide untuk bulan-bulan sebelumnya dan selanjutnya, menampilkan tanggal tertentu, dan bersifat estimasi secara perhitungan bulan (hisab), untuk penentuan hari raya biasanya menunggu keputusan Ru'yatul Hilal.

Gambar tampilan untuk aplikasi Hijri Calendar yang dibuat oleh Kodelokus Cipta Aplikasi dapat dilihat pada gambar 2.3 di bawah ini.

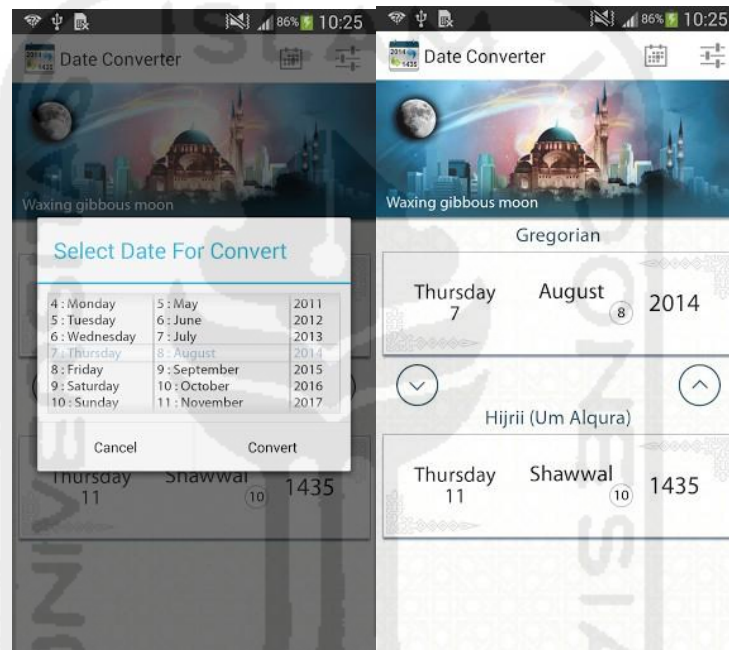


Gambar 2.3 Hijri Calendar by Kodelokus Cipta Aplikasi

2.4.3 Date Converter by RevoTeam

Date Converter yang dibangun oleh RevoTeam adalah sebuah aplikasi yang berguna untuk mengkonversi antara kalender hijriyah ke kalender masehi. Aplikasi ini hanya menampilkan hasil dari tanggal yang diminta.

Gambar tampilan untuk aplikasi Date Converter yang dibuat oleh RevoTeam dapat dilihat pada gambar 2.4 di bawah ini.



Gambar 2.4 Date Converter by RevoTeam

2.4.4 Islamic Calendar by ImranQureshi.com

Aplikasi *Islamic Calendar* yang dibuat oleh ImranQureshi.com ini dapat melihat tanggal sekarang pada kalender hijriyah dan menyesuaikan dengan tanggal pada kalender masehi. Aplikasi ini difokuskan untuk umat muslim mengetahui hari penting dalam kalender hijriyah lebih cepat.

Gambar tampilan untuk aplikasi *Islamic Calendar* yang dibuat oleh ImranQureshi.com dapat dilihat pada gambar 2.5 di bawah ini.



Gambar 2.5 *Islamic Calendar* by ImranQureshi.com