

**KALENDER BUDDHIS BERBASIS ANDROID**

**(Skripsi)**

**Disusun Oleh**  
**Cynthia Dewi Tandoyo**



**JURUSAN ILMU KOMPUTER**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS LAMPUNG**  
**2017**

## **ABSTRAK**

### **KALENDER BUDDHIS BERBASIS ANDROID**

Disusun Oleh

Cynthia Dewi Tandoyo

Terdapat 6 (enam) agama di Indonesia, salah satunya adalah agama Buddha. Tiap agama memiliki hari keagamaannya masing-masing, tak terkecuali agama Buddha. Untuk melihat hari-hari keagamaan tersebut, umat Buddha menggunakan kombinasi *Solar Calendar* dan *Lunisolar/Chinese Calendar*. Tidak Banyak orang yang hafal hari-hari keagamaan tersebut. Mereka akan menyimpan *file* yang berisi hari-hari keagamaan di dalam komputer, *flash disk* atau dalam bentuk *print out*. Dalam penelitian ini, peneliti akan mengembangkan sebuah aplikasi kalender yang tisk hanya dapat mengkonversi *Chinese Calendar* kedalam *Solar Calendar*, tetapi juga menampilkan hari keagamaan pada hari tersebut, dan hari raya keagamaan dalam 1 (satu) tahun. Dalam aplikasi ini, SQLite database akan digunakan untuk menampung seluruh hari keagamaan.

## **ABSTRACT**

### **BUDDHIS CALENDAR BASED ON ANDROID**

Created By

Cynthia Dewi Tandoyo

There are six religions in Indonesia, one of them is Buddha. Every religions have their own religious days, including Buddha. To see the religious days, a Buddhist uses the combination of both Solar Calendar and Lunisolar/Chinese Calendar. There are not many people can memorize all religious days. They will save the file contain the list of religious days in computer, flash disk or a printout file. In this research, the researcher will develop an android calendar application that not only can convert Chinese Calendar into Solar Calendar, but also show the religious days on that day, and the religious days in a year. In this application, SQLite database will be used to record all the religious days.

**KALENDER BUDDHIS BERBASIS ANDROID**

**Oleh:**

**Cynthia Dewi Tandoyo**

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar

**SARJANA KOMPUTER**

Pada

Jurusan Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam



**JURUSAN ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS LAMPUNG**

**2017**

**Judul Skripsi**

**: KALENDER BUDDHIS BERBASIS  
ANDROID**

**Nama Mahasiswa**

**: Cynthia Dewi Tandoyo**

**Nomor Pokok Mahasiswa**

**: 1117032020**

**Jurusan**

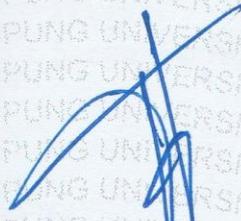
**: Ilmu Komputer**

**Fakultas**

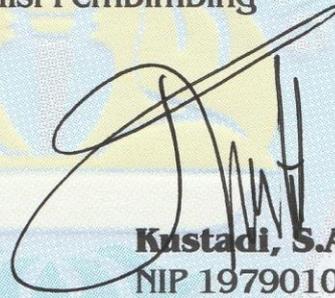
**: Matematika dan Ilmu Pengetahuan Alam**

**MENYETUJUI**

**1. Komisi Pembimbing**



**Didik Kurniawan, S.Si., M.T.**  
**NIP 19800419 200501 1 004**



**Kustadi, S.AG.**  
**NIP 19790108 200501 1 002**

**2. Ketua Jurusan Ilmu Komputer**



**Dr. Ir. Kurnia Muludi, M.S.Sc.**  
**NIP 19640616 198902 1 001**

**MENGESAHKAN**

**1. Tim Penguji**

**Ketua**

**: Didik Kurniawan, S.Si., M.T.** .....

**Sekretaris**

**: Kustadi, S.AG.** .....

**Penguji**

**Bukan Pembimbing**

**: Dwi Sakethi, S.Si., M.Kom.** .....

**2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam**



**Prof. Warsito, S.Si., D.E.A., Ph.D.**

**NIP 19710212 199512 1 001**

**Tanggal Lulus Ujian Skripsi : 12 Oktober 2017**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul “Kalender Buddhis Berbasis Android” merupakan karya saya sendiri dan bukan hasil karya orang lain. Semua hasil tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya ini merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 12 Oktober 2017



**Cynthia Dewi Tanooyo**

NPM. 1117032020

## **RIWAYAT HIDUP**

Penulis dilahirkan di Bandar Lampung pada tanggal 30 September 1993 sebagai anak pertama dari tiga bersaudara dari pasangan Fairley Leonardo Tandoyo dan Cendrawasih.

Penulis memasuki Taman Kanak – Kanak (TK) pada tahun 1997 di TK Xaverius Teluk Betung, Kecamatan Teluk Betung Utara, Bandar Lampung. Penulis memasuki tingkat Sekolah Dasar (SD) pada tahun 1999 di SD. Xaverius Teluk Betung, Kecamatan Teluk Betung Utara, Bandar Lampung dan lulus pada tahun 2005. Penulis melanjutkan pendidikan ke jenjang Sekolah Menengah Pertama (SMP) di SMP. Xaverius I, Kecamatan Teluk Betung Utara, Bandar Lampung dan lulus pada tahun 2008. Kemudian, penulis melanjutkan pendidikan ke tingkat Sekolah Menengah Atas (SMA) di SMA. Xaverius Pahoman, Kecamatan Rawa Laut, Bandar Lampung dan lulus pada tahun 2011.

Penulis mendaftarkan diri di SNMPTN tertulis pada tahun 2011 dan akhirnya lolos seleksi dan menjadi mahasiswa di Program Studi Ilmu Komputer Jurusan

Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam. Penulis cukup aktif berorganisasi, diantaranya:

1. Koordinator Divisi Kesekretariatan UKM Buddha Universitas Lampung periode 2012-2013.
2. Bendahara Umum UKM Buddha Universitas Lampung periode 2013-2014.
3. Anggota Bidang Keilmun Himpunan Ilmu Komputer (HIMAKOM) periode 2013-2014.
4. Kerja Praktek di Lab Komputer Jurusan Ilmu Komputer FMIPA Universitas Lampung pada tahun 2014.
5. Kuliah Kerja Nyata di Desa Suka Marga, Kecamatan Pulau Pisang, Kabupaten Pesisir Barat, Provinsi Lampung.
6. *Exchange Program Student* ke Tokyo University of Agriculture and Technology (TUAT), Koganei-shi, Tokyo, Japan.
7. Anggota *group* belajar bahasa Jepang '*Minna no Nihongo (MNN)*'.

## PERSEMBAHAN

Puji dan Syukur kepada Tuhan **DMC** atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan sebaik-baiknya.

Kupersembahkan karya ini kepada:

Dapa dan Mama yang telah mendidik, dan membesarkanku serta memberikan doa dan dukungan agar anak-anaknya dapat mencapai kesuksesan. Terima kasih atas semua kasih sayang yang telah kalian berikan serta pengorbanan dan perjuangan yang telah kalian lakukan untukku.

Adik-adikku yang kusayangi **Jessica Tandoyo** dan **Florence Thie Tandoyo**, serta keluarga besarku yang kukasih.

Keluarga Besar Ilmu Komputer 2011,

Serta Almamaterku Universitas Lampung

## MOTTO

*Sesuatu yang terlihat mustahil bukanlah tidak mungkin untuk bisa  
dilakukan*

*Jangan pernah menyerah dalam melakukan sesuatu*

*Setiap persoalan yang ada, pasti memiliki jalan keluarnya*

## SANWACANA

Puji dan syukur kepada Tuhan YME yang telah menyertai penulis hingga dapat menyelesaikan penelitian sebagai pemenuhan Tugas Akhir yang merupakan syarat bagi penulis untuk mendapatkan gelar Sarjana Komputer di Jurusan Ilmu Komputer Universitas Lampung dengan judul “Kalender Buddhis Berbasis Android”.

Dalam penyusunan Tugas Akhir ini, penulis banyak menghadapi kesulitan.

Namun berkat bantuan, pengarahan, dan dukungan dari berbagai pihak, akhirnya penulis dapat menyelesaikan Tugas Akhir ini. Karenanya, penulis ingin

berterimakasih kepada:

1. Kedua orang tuaku dan adik-adikku yang telah mendoakan akan keberhasilan penulis, dan juga memberi dukungan baik dari segi mental maupun material
2. Bapak Didik Kurniawan, S.Si., M.T., sebagai Sekertaris Jurusan Ilmu Komputer sekaligus Pembimbing I atas pengarahan dan masukannya dalam penelitian serta dorongan untuk segera menyelesaikan skripsi ini

3. Bapak Kustadi, S.AG., sebagai Pembimbing II yang telah memberikan masukan sehingga penulis dapat menyelesaikan skripsi ini.
4. Bapak Dwi Sakethi, M.Kom., sebagai Pembahas yang telah memberikan saran dan masukan yang bermanfaat dalam penyelesaian skripsi ini.
5. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc., selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
6. Bapak Prof. Warsito, S.Si., DEA., Ph.D selaku Dekan FMIPA Universitas Lampung.
7. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang tiada lelah dalam membagikan ilmunya kepada penulis.
8. Teman-teman Minna no Nihonggo, UKM Buddha Unila, teman-teman Ilkom 2011 serta kakak dan adik tingkat di Jurusan Ilmu Komputer Univeritas Lampung yang selalu memberikan dukungan dan menghibur penulis sepanjang penelitian.
9. Seluruh pihak yang tidak dapat disebutkan satu per satu.

Semoga Tugas Akhir ini dapat bermanfaat bagi mereka yang membutuhkan.

Akhir kata, tidak ada manusia yang sempurna begitupula penulis. Tugas Akhir ini juga masih belum sempurna. Karenanya, penulis meminta maaf jika terdapat

penulisan yang menyinggung ataupun yang kurang berkenan di hati pembaca.

Kritik dan saran yang membangun sangat diharapkan guna memperbaiki

penulisan di masa depan.

Bandar Lampung, 12 Oktober 2017

**Cynthia Dewi Tandoyo**

## DAFTAR ISI

|   | Halaman |
|---|---------|
| DAFTAR ISI.....                                     | xii     |
| DAFTAR GAMBAR .....                                 | xv      |
| DAFTAR TABEL.....                                   | xvi     |
| <br>  |         |
| BAB I PENDAHULUAN                                   |         |
| 1.1 Latar Belakang .....                            | 1       |
| 1.2 Rumusan Masalah .....                           | 3       |
| 1.3 Tujuan .....                                    | 3       |
| 1.4 Manfaat .....                                   | 3       |
| 1.5 Batasan Masalah.....                            | 4       |
| <br>  |         |
| BAB II TINJAUAN PUSTAKA                             |         |
| 2.1 Kalender .....                                  | 5       |
| 2.2 Kalender Buddha.....                            | 7       |
| 2.3 Definisi Buddha, Buddhis, dan Buddhisme .....   | 10      |
| 2.4 Android .....                                   | 11      |
| 2.4.1 Arsitektur Android .....                      | 11      |
| 2.4.2 <i>Android Versions</i> .....                 | 14      |
| 2.4.2.1 <i>Éclair</i> (Versi 2.0/2.1) .....         | 15      |
| 2.4.2.2 <i>Froyo</i> (Versi 2.2).....               | 15      |
| 2.4.2.3 <i>Gingerbread</i> (Versi 2.3).....         | 16      |
| 2.4.2.4 <i>Honeycomb</i> (Versi 3.0) .....          | 16      |
| 2.4.2.5 <i>Ice Cream Sandwich</i> (Versi 4.0) ..... | 16      |
| 2.4.2.6 <i>Jellybean</i> (Versi 4.1) .....          | 17      |

|   |    |
|---|----|
| 2.4.2.7 <i>Kitkat</i> (Versi 4.4) .....                     | 17 |
| 2.4.2.8 <i>Lollipop</i> (Versi 5.0).....                    | 18 |
| 2.4.3 Android SDK .....                                     | 18 |
| 2.4.4 API ( <i>Application Programming Interface</i> )..... | 18 |
| 2.5 Metodologi Pembuatan Aplikasi.....                      | 19 |
| 2.5.1 <i>Agile Methods Extreme Programming</i> .....        | 19 |
| 2.5.2 UML ( <i>Unified Modeling Language</i> ) .....        | 20 |
| 2.6 SQLite .....  | 26 |
| 2.7 Penelitian Sejenis .....                                | 26 |
| 2.7.1 Android –Lunar View .....                             | 26 |
| 2.8 Metode Pengujian .....                                  | 27 |
| 2.8.1 <i>Black Box Testing</i> .....                        | 27 |
| 2.8.2 <i>Equivalent Partitioning</i> (EP) .....             | 28 |
| 2.8.3 Skala Likert.....                                     | 28 |

### BAB III METODOLOGI PENELITIAN

|  |    |
|--|----|
| 3.1 Waktu dan Tempat Penelitian .....                            | 30 |
| 3.2 Alat Pendukung Penelitian.....                               | 30 |
| 3.3 Tahapan Penelitian.....                                      | 31 |
| 3.4 Desain Sistem.....   | 33 |
| 3.4.1 Analisis Kebutuhan Sistem .....                            | 33 |
| 3.4.2 Perancangan Sistem .....                                   | 33 |
| 3.4.3 ERD ( <i>Entity Relationship Diagram</i> ).....            | 33 |
| 3.4.4 Perancangan UML ( <i>Unified Modeling Language</i> ) ..... | 34 |
| 3.4.5 Perancangan Antar Muka.....                                | 39 |
| 3.5 Data Hari Keagamaan .....                                    | 41 |
| 3.6 Pengujian Sistem.....  | 42 |
| 3.6.1 Pengujian Fungsional.....                                  | 42 |
| 3.6.2 Pengujian Non-Fungsional.....                              | 44 |

### BAB IV HASIL DAN PEMBAHASAN

|                        |    |
|------------------------|----|
| 4.1 Implementasi ..... | 45 |
|------------------------|----|

|   |    |
|---|----|
| 4.2 Tampilan Aplikasi.....                      | 45 |
| 4.2.1 Tampilan Awal.....                        | 45 |
| 4.2.2 Tampilan <i>List</i> .....                | 46 |
| 4.2.3 Tampilan <i>About</i> .....               | 47 |
| 4.3 Pengujian Aplikasi .....                    | 48 |
| 4.3.1 Pengujian Fungsional .....                | 48 |
| 4.3.1.1 Pengujian Terhadap Hasil Konversi ..... | 51 |
| 4.3.2 Pengujian Non-Fungsional.....             | 51 |
| 4.4 Upload Aplikasi .....                       | 56 |

## BAB V KESIMPULAN DAN SARAN

|                      |    |
|----------------------|----|
| 5.1 Kesimpulan ..... | 57 |
| 5.2 Saran .....      | 57 |

## DAFTAR PUSTAKA

## LAMPIRAN

## DAFTAR GAMBAR

|  | Halaman |
|--|---------|
| Gambar 2.4.1.1 Arsitektur Android .....                                  | 11      |
| Gambar 2.4.2.1 Sejarah Perilisan Android .....                           | 15      |
| Gambar 2.5.1.1 Tahapan <i>eXtreme Programming</i> (Pressman, 2010) ..... | 20      |
| Gambar 3.3.1 Alur Pembuatan Sistem .....                                 | 31      |
| Gambar 3.4.3.1 ERD ( <i>Entity Relationship Diagram</i> ) .....          | 34      |
| Gambar 3.4.4.1 <i>Use Case Diagram</i> .....                             | 34      |
| Gambar 3.4.4.2 <i>Activity Diagram</i> Menampilkan Tampilan Awal .....   | 35      |
| Gambar 3.4.4.3 <i>Activity Diagram</i> Akses Tombol <i>List</i> .....    | 36      |
| Gambar 3.4.4.4 <i>Activity Diagram</i> Akses Tombol <i>About</i> .....   | 36      |
| Gambar 3.4.4.5 <i>Sequence Diagram</i> Menampilkan Tampilan Awal .....   | 37      |
| Gambar 3.4.4.6 <i>Sequence Diagram</i> Akses Tombol <i>List</i> .....    | 38      |
| Gambar 3.4.4.7 <i>Sequence Diagram</i> Akses Tombol <i>About</i> .....   | 39      |
| Gambar 3.4.5.1 <i>Layout</i> Tampilan Awal .....                         | 40      |
| Gambar 3.4.5.2 <i>Layout List</i> .....                                  | 40      |
| Gambar 3.4.5.3 <i>Layout About</i> .....                                 | 41      |
| Gambar 4.2.1.1 Tampilan Awal yang Ada Hari Keagamaan .....               | 46      |
| Gambar 4.2.1.2 Tampilan Awal yang Tidak Ada Hari Keagamaan .....         | 46      |
| Gambar 4.2.2.1 Tampilan <i>List</i> Hari Keagamaan .....                 | 47      |
| Gambar 4.2.3.1 Tampilan <i>About</i> .....                               | 54      |
| Gambar 4.2.3.2 Tampilan <i>About Cont.</i> .....                         | 54      |
| Gambar 4.4.1 Tampilan Aplikasi di <i>Google Play</i> .....               | 56      |
| Gambar 4.4.2 Tampilan Aplikasi di <i>Google Play Cont.</i> .....         | 56      |

## DAFTAR TABEL

|   | Halaman |
|---|---------|
| Tabel 2.5.2.1 Notasi <i>Use Case Diagram</i> .....  | 21      |
| Tabel 2.5.2.2 Notasi <i>Activity Diagram</i> (Meildy, 2015) .....                                     | 23      |
| Tabel 2.5.2.3 Notasi <i>Sequence Diagram</i> (Meildy, 2015) .....                                     | 25      |
| Tabel 2.8.3.1 Preferensi Jawaban Skala Likert .....   | 29      |
| Tabel 3.6.1.1 Perencanaan Pengujian Menggunakan Metode EP<br>( <i>Equivalent Partitioning</i> ) ..... | 42      |
| Tabel 3.6.2.1 Perencanaan Pengujian <i>Skala Likert</i> .....   | 44      |
| Tabel 4.3.1.1 Hasil Pengujian Menggunakan Metode EP<br>( <i>Equivalent Partitioning</i> ) .....       | 49      |
| Tabel 4.3.2.1 Hasil Pengujian Non-Fungsional .....  | 52      |
| Tabel 4.3.2.2 Kategori Penilaian Skala Likert .....   | 52      |
| Tabel 4.3.2.3 Persentase Penilaian pada Pengujian Non-Fungsional .....                                | 55      |

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Terdapat 6 (enam) agama yang diakui di Indonesia, salah satunya adalah agama Buddha. Berdasarkan data Bimas Buddha Kementerian Agama tahun 2016, Bandar Lampung merupakan kota yang memiliki penduduk beragama Buddha paling banyak se-Provinsi Lampung dengan jumlah penganut 35.788 jiwa (<https://lampung.kemenag.go.id/artikel/36168/data-keagamaan-kementerian-agama-provinsi-lampung>, 2017). Setiap agama memiliki hari keagamaan masing-masing, tak terkecuali Agama Buddha.

Agama Buddha memiliki banyak hari keagamaan dan untuk melihat hari-hari keagamaan tersebut, umat Buddha menggunakan kombinasi dari dua kalender, yakni *Solar Calendar* dan *Lunisolar/Chinese Calendar*. Solar calendar dimulai dari *December Solstice* dan mengikuti 24 *jiéqì*. Sedangkan, *Chinese Calendar* dimulai dari Tahun Baru China atau Tahun Baru Imlek. (Aslaksen, 2010)

Tidak banyak orang yang hafal hari-hari keagamaan tersebut. Menurut hasil questioner yang telah penulis lakukan, dari 100 orang yang beragama Buddha, hanya 3 (tiga) orang yang hafal hari-hari keagamaan umat Budha. Kebanyakan penganut agama Buddha menyimpannya dalam bentuk *file* di komputer, *flash disk*, atau dalam bentuk *print out*. Jika suatu ketika, *file* atau *print out* tersebut hilang, maka mereka akan bertanya kepada orang lain yang hafal atau meminta *file* atau *print out* tersebut dari seseorang yang masih memilikinya.

Sekarang ini, Android sudah sangat populer, khususnya di pasaran *smartphone*. Menurut Bhardwaj, dkk, Android merupakan OS (*Operating System*) untuk *mobile phone* yang berbasis *Linux Kernel* dan dikembangkan oleh Google dan kemudian dikembangkan oleh “*Open Handset Alliance*”. Pengembangan aplikasinya berbasis pemrograman Java, karenanya memungkinkan pengembang untuk membuat program menggunakan bahasa pemrograman Java (Katti dan Suhas, 2012). Program yang dibuat dapat mengontrol *mobile phone* melalui *Google-enabled java libraries*. *Google-enabled java libraries* merupakan *platform* yang penting dalam pengembangan aplikasi mobile yang menggunakan *software stack* yang telah disediakan dalam *Google Android SDK*. Android mobile OS menyediakan lingkungan yang fleksibel untuk mengembangkan *Android Mobile Application* karena pengembang tidak hanya dapat menggunakan *Android Java Library*, tetapi juga memungkinkan untuk menggunakan *Normal Java IDEs*. (Gu, dkk, 2014)

Berdasarkan latar belakang ini, maka akan dikembangkan sebuah aplikasi Kalender yang tidak hanya mencatat seluruh hari-hari keagamaan umat Buddha dalam 1 (satu) tahun tetapi juga dapat mengingatkan pemeluk agama Buddha jika ada hari keagamaan pada hari ini.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang dari pembuatan aplikasi ini, maka rumusan masalah penelitian ini adalah bagaimana merancang dan membangun aplikasi “Kalender Buddhis” yang dapat mengkonversi *Chinese Calendar* menjadi *Solar Calendar* dan memperlihatkan hari-hari keagamaan umat Buddha.

## **1.3 Tujuan**

1. Membuat aplikasi yang memberitahu *user*, *Chinese Calendar* pada *Solar Calendar*,
2. Membuat aplikasi yang memperlihatkan hari-hari keagamaan pada 1 (satu) tahun,
3. Membuat aplikasi yang dapat memudahkan *user* untuk melaksanakan ibadah pada hari-hari keagamaan.

## **1.4 Manfaat**

1. Memperlihatkan *Chinese Calendar* pada *Solar Calendar*,
2. Umat Buddha bisa melihat hari-hari keagamaan yang ada dalam 1 (satu) tahun,

3. Mengingatkan user jika hari ini ada hari keagamaan.

### **1.5 Batasan Masalah**

1. Menampilkan hasil konversi menggunakan pengaturan konversi yang telah

dibuat oleh Vincent Cheung,

2. Hari-hari keagamaan umat Buddha aliran Mahayana, Buddhayana, dan

Maitreya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Kalender**

Kalender merupakan produk yang dihasilkan dari pengembangan astronomi dan merupakan hasil dari mengobservasi fenomena alam (Guo, dkk, 2014). Kalender digunakan dalam kehidupan manusia sehari-hari. Ketika kita membuat rencana kegiatan atau jadwal belajar, kita membutuhkan kalender. Ketika kita ingin merayakan hari khusus, kita juga membutuhkan kalender. Di sisi lain, festival tidak dapat dikonfirmasi tanpa bantuan kalender; petani tidak dapat bekerja tanpa bantuan kalender untuk melihat musim untuk bercocok tanam; ahli sejarah dan arkeolog pada saat melakukan penelitian juga membutuhkan kalender. Bukanlah hal yang sulit untuk menemukan bahwa kita tidak dapat jauh dari kalender dari awal kita hidup hingga meninggal dunia.

Dalam konsep ilmu pengetahuan, kalender dibagi menjadi Kalender Aritmatik dan Kalender Astronomi.

##### **1. Kalender Aritmatik**

Kalender Aritmatik dikalkulasi mengikuti sebuah mode yang tetap yang berarti sekumpulan hari dan bulan dengan menggunakan perbedaan rumus matematis,

tidak terpengaruh faktor luar. Keuntungan dari Kalender Aritmatik yakni mudah bekerja ketika ada hari tertentu. Kekurangan dari Kalender Aritmatik yakni kalender jenis ini tidak terlalu akurat.

## 2. Kalender Astronomi

Komputasi dari Kalender Astronomi mempertimbangkan fenomena alam dan perubahan lingkungan seperti perubahan musim dan cuaca yang menjadi faktor utama dalam pembuatan kalender. Jadi, kalender jenis ini selalu berubah. Keuntungan dari Kalender Astronomi yakni lebih akurat dengan kegiatan-kegiatan astronomi, tetapi kalender lebih diversifikasi. (Guo, dkk, 2014)

Kalender Astronomi sendiri dibagi menjadi tiga tipe: *Solar Calendar*, *Lunar Calendar*, dan *Lunisolar Calendar*.

### a. *Solar Calendar*

Kalender ini merupakan kalender yang disusun berdasarkan hubungan antara bumi dan matahari. *Solar Calendar* mengambil “Tahun Tropis atau Tahun Matahari” sebagai satu tahun. Satu tahun matahari sama dengan 365.2422 hari (365 hari 5 jam 48 menit 46 detik), jika diakumulasi 4 kali, akan sama dengan satu hari, tiap empat tahun ditambah satu hari pada bulan Februari. (Aslaksen, 2010)

### b. *Lunar Calendar*

Kalender ini merupakan kalender yang dibuat berdasarkan pergerakan bulan, disebut juga “Bulan Lunar”. Bulan Lunar berisikan 29 hari 12 jam 44 menit 3 detik. Lebih mudahnya, bulan besar memiliki 30 hari dan bulan kecil memiliki 29 hari. 12 bulan lunar sama dengan 354 hari, dan tahun lengkap memiliki total 355

hari, lebih sedikit 11 hari dibandingkan dengan tahun matahari. Masyarakat Islam menggunakan kalender ini untuk menentukan hari-hari keagamaan mereka. (Breon dan Igor, 2005)

### c. *Lunisolar Calendar*

Kalender ini juga merupakan kalender yang dibuat berdasarkan bulan lunar, 12 bulan sama dengan 1 tahun (354.3667 hari, hampir 355 hari). Kalender ini tidak hanya menggunakan Lunar tapi juga mementingkan Matahari, sehingga bisa dibidang bahwa kalender ini merupakan perpaduan dari *Solar Calendar* dan *Lunar Calendar*. *Lunisolar Calendar* meletakkan bulan lengkap tiap 3 tahun, sehingga panjangnya sama dengan tahun matahari (Guo, dkk, 2014). Kalender ini merupakan dasar yang digunakan dalam *Chinese Calendar*. (Fan, 2016)

## **2.2 Kalender Buddha**

Kalender Buddha merupakan salah satu kalender yang menggunakan sistem penanggalan *Lunisolar*. Sistem penanggalan Buddha merupakan salah satu sistem penanggalan kuno di dunia. Tidak ada yang tahu kapan persisnya sistem penanggalan ini diciptakan dan kapan mulai dipakai. Namun demikian, para ahli sejarah melalui penelitian mereka terhadap data-data sejarah yang ada menyimpulkan, bahwa sistem penanggalan pada dasarnya menginduk pada versi asli sistem penanggalan “Surya Siddhanta” yang mana masih digunakan hingga abad ke-3 Masehi. Surya Siddhanta sendiri, baik versi asli maupun versi perkembangannya merupakan rujukan yang dipakai oleh berbagai sistem penanggalan umat Hindu. Sehingga dapat disimpulkan bahwa sistem penanggalan Buddha dan sistem penganggalan Hindu berinduk pada sistem penanggalan yang

sama, yaitu sistem penanggalan Surya Siddhanta.

Sistem penanggalan Buddha berbasis Surya Siddhanta tidak dipakai oleh seluruh umat Buddha di seluruh dunia. Sistem penanggalan ini hanya digunakan oleh umat Buddha yang beraliran Theravada. Umat Buddha yang beraliran Mahayana biasanya menyesuaikan dengan sistem penanggalan yang telah ada pada daerah tersebut, sebagaimana yang terjadi di negara China, dimana hari raya umat Buddha pada kalender China tidak bersamaan dengan kalender Buddha lainnya.

Perbedaan sistem penanggalan Theravada dan Mahayana tidak menyebabkan sistem penanggalan keduanya berbeda sama sekali. Bagaimanapun, masih ada beberapa poin penting yang harus tetap dijaga dalam perhitungan kalender Buddha, baik Theravada maupun Mahayana. Poin-poin tersebut antara lain:

1. Perhitungan tanggal harus sesuai dengan sejarah kehidupan Sang Buddha. Sehingga, apabila kalender dimulai dari tahun wafatnya Buddha (paranibbana/paranirvana), maka tahun kelahiran Buddha berada pada tahun (-) 80 tahun Buddha. Dengan berpatokan hal tersebut, kejadian-kejadian yang terjadi dalam masa kehidupan Sang Buddha dapat diketahui dengan tepat;
2. Penanggalan peristiwa-peristiwa penting lainnya dalam sejarah Agama Buddha;
3. Perhitungan tanggal peribadahan (*uposatha*), dan hari-hari suci seperti Waisak ataupun awal dan akhir Vassa.

(Izzuddin, 2010)

Pada kalender Buddha, secara resmi terdapat 12 bulan yang nama-namanya berbeda antara satu tempat dengan tempat lainnya. Berikut ini merupakan contoh nama-nama bulan pada kalender Buddha (secara urut) dalam bahasa:

#### Sanskerta

- |               |               |
|---------------|---------------|
| 1. Caitra     | 7. Asvina     |
| 2. Vaisakha   | 8. Karttika   |
| 3. Jyaistha   | 9. Margasirsa |
| 4. Asadha     | 10. Pausa     |
| 5. Sravana    | 11. Magha     |
| 6. Bhadrapada | 12. Phalguna  |

#### Pali

- |               |               |
|---------------|---------------|
| 1. Citta      | 7. Assayuja   |
| 2. Viskaha    | 8. Katthika   |
| 3. Jettha     | 9. Magasira   |
| 4. Asalha     | 10. Phussa    |
| 5. Savana     | 11. Magha     |
| 6. Potthapada | 12. Phaggunna |

([http://www.buddhanet.net/cal\\_mnth.htm](http://www.buddhanet.net/cal_mnth.htm))

Semua penanggalan Buddha, baik yang awal bulannya terjadi pada saat bulan baru maupun yang awal bulannya terjadi pada saat bulan purnama, membagi setiap bulannya menjadi 2 (dua) bagian, biasanya disebut dwiminggu. Dwiminggu pertama berjumlah 15 hari, dan dwiminggu kedua berjumlah 14 atau 15 hari. Pada tiap bulannya terdapat hari untuk beribadah (*uposatha*). Dalam satu bulan

biasanya terdapat 4 (empat) kali uposatha, yaitu ketika bulan baru, bulan purnama, serta hari kedelapan pada tiap dwiminggu (ketika bulan terlihat  $\frac{1}{4}$  dan  $\frac{3}{4}$  bagian).

(Izzuddin, 2010)

Kalender Buddha seperti yang sudah dijelaskan sebelumnya, merupakan kalender lunisolar. Oleh sebab itu, dalam perhitungannya terdapat ‘Interkalasi Bulan (penambahan bulan)’ tiap beberapa tahun sekali. Kalender lunisolar pada umumnya mengalami 7 (tujuh) kali interkalasi bulan dalam kurun waktu 19 tahun. Dalam kalender Buddha, hal ini dikenal dengan nama ‘Adhikmasa’. Selain terdapat penambahan bulan, pada sistem penanggalan Buddha terdapat pula penambahan hari (adhikavara). Biasanya, dalam kurun waktu 57 tahun, terdapat 11 (sebelas) kali penambahan hari. Penambahan hari tersebut terjadi berdasarkan perhitungan yang rumit.

### **2.3 Definisi Buddha, Buddhis, dan Buddhisme**

Buddha, Buddhis, dan Buddhisme mungkin terlihat mirip, akan tetapi mereka memiliki makna yang berbeda. Menurut KBBI (Kamus Besar Bahasa Indonesia) ‘Buddha’ berarti agama yang diajarkan oleh Sidharta Gautama. ‘Buddhis’ berarti penganut Buddhisme. ‘Buddhisme’ berarti ajaran yang dikembangkan oleh Sidharta Gautama yang mengajarkan bahwa kesengsaraan adalah bagian kehidupan yang tak terpisahkan, dan orang dapat membebaskan diri dari kesengsaraan dengan menyucikan mental dan moral diri pribadi.

## 2.4 Android

Android merupakan *software platform* dan sistem operasi untuk *mobile device* yang dasarnya dari *Linux kernel* dan dikembangkan oleh Google dan kemudian oleh *Open Handset Alliance*. Android mengizinkan pengembang untuk membuat kodingan menggunakan bahasa Java, pengontrolan alat melalui *Google-developed Java Libraries*. (Bhardwaj, dkk, 2013)

### 2.4.1 Arsitektur Android

Android merupakan *software stack* untuk *mobile device* yang termasuk di dalamnya sistem operasi, *middleware*, dan *key applications*. *Android SDK* menyediakan *tools* dan *API* yang dibutuhkan untuk memulai mengembangkan aplikasi pada Android platform menggunakan bahasa pemrograman Java. Android berbasis pada Linux 2.6. Pelayanan sistem seperti keamanan, manajemen memori, manajemen proses semuanya dikontrol oleh Linux.



Gambar 2.4.1.1 Arsitektur Android

### **A. Application Layer**

*Application Layer* merupakan layer paling atas pada arsitektur Android. Beberapa aplikasi sudah terinstal di seluruh device, seperti: *SMS client application*, *Dialer*, *Web Browser*, dan *Contact Manager*. Pengembang dapat membuat aplikasinya sendiri dan dapat mengganti aplikasi yang sudah ada dengan aplikasi yang dibuatnya.

### **B. Application Framework**

Layer Application Framework menyediakan banyak pelayanan tingkat tinggi atau *major APIs* untuk aplikasi dalam bentuk *Java classes*. Dengan menyediakan *open development platform*, Android memberikan para pengembang kemampuan untuk membuat aplikasi yang sangat bermutu dan inovatif. Para pengembang aplikasi diizinkan untuk menggunakan pelayanan ini dalam aplikasi mereka dengan bebas. Pengembang memiliki akses penuh ke *framework APIs* yang sama yang digunakan oleh *core application*. Blok penting dalam *Application Framework*, yakni:

- **Activity Manager** yang manage life cycle application dan menyediakan *common navigation backstack*;
- **Content Providers** yang digunakan untuk manajemen *data sharing* antar aplikasi, manajemen bagaimana mengakses data dari aplikasi lain;
- **Telephony Manager** yang digunakan untuk manajemen seluruh *voice call* yang berhubungan dengan fungsionalitas;
- **Location manager** yang digunakan untuk manajemen lokasi menggunakan *GPS* atau *cell tower*;

- **Resource Manager** yang memajemen berbagai macam resource yang digunakan dalam aplikasi, dan memberikan akses kepada *non-code resources* seperti *localized strings*, *graphic*, dan *layout files*.

### C. Native Libraries

Di bagian Atas Linux Kernel Layer, terdapat *Native Libraries* dari Android. Layer ini mendukung *application framework* dan *Linux Kernel*. Seluruh *library* dibuat menggunakan bahasa pemrograman C atau C++. *Library - library* ini dipanggil melalui *java interface* sehingga nantinya dapat dimanfaatkan oleh komponen – komponen yang berbeda dalam *Android System*. Beberapa native libraries yang penting, yakni:

- **Surface Manager**: Digunakan untuk memajemen tampilan *device*. *Surface manager* digunakan untuk menyusun jendela pada layar;
- **SQLite**: Merupakan database yang digunakan pada Android untuk *data storage*. *SQLite* merupakan *relational database* dan dapat digunakan oleh semua aplikasi;
- **WebKit**: Merupakan *browser engine* yang digunakan untuk menampilkan isi HTML;
- **Media Framework**: Penyedia *playback* dan *recording* dari berbagai macam format *audio*, *video*, dan *picture* (contoh: MP3, AAC, AMR, JPG, MPEG4, H.264, dan PNG);
- **OpenGL / ES**: Digunakan untuk memberikan grafik 2D atau 3D pada layar;
- **libc**: Berisi library C yang berhubungan dengan sistem.

#### **D. Android Runtime**

*Android Runtime* terdiri dari *Java Core Library* dan *Dalvik Virtual Machine*. *Java Core Library* menyediakan *library* dengan banyak fungsi. *Dalvik Virtual Machine* merupakan sebuah tipe *Java Virtual Machine* yang digunakan untuk menjalankan aplikasi pada *Android Device*. Dalvik VM memungkinkan setiap aplikasi Android untuk berjalan pada prosesnya masing-masing, dengan instansi sendiri dari *Dalvik virtual machine*. Dalvik VM memungkinkan untuk banyak instance dari Virtual Machine untuk dibuat secara simultan menyediakan keamanan, isolasi, *memory management*, dan, *threading support*. *Dalvik VM* menjalankan *file* dengan format *.dex* yang dibuat dari file dengan format *.class* menggunakan *dx tool*. *dx tool* sudah ada dalam *Android SDK*. DVM dioptimalkan untuk *low processing power* dan *low memory environment*. (Chinetha dkk, 2015)

#### **E. Linux Kernel**

Layanan *Kernel System* yang disediakan oleh *Android inner nuclear layer* didasarkan pada *Linux 2.6 kernel*. Operasi seperti *internal storage*, *process management*, *internet protocol*, *bottom-drive*, dan layanan inti lainnya semua didasarkan pada *Linux Kernel*. (Holla, dan Katti, 2012), (Chinetha, dkk, 2015), dan (Gu, dkk, 2014)

#### **2.4.2 Android Versions**

Android mengalami perbaruan setiap hari. Perbaruan ini pada dasar sistem operasi berfokus pada berpaikan bug juga menambahkan fitur baru untuk menyediakan environment yang lebih nyaman. Umumnya tiap versi baru dari Android dikembangkan menggunakan kode nama berdasarkan nama 'desert'.

| Version   | Name                  | Release Date   |
|-----------|-----------------------|----------------|
| 1.5       | Cupcake               | April 2009     |
| 1.6       | Donut                 | September 2009 |
| 2.0-2.1   | Éclair                | October 2009   |
| 2.2-2.2.3 | Froyo                 | May 2010       |
| 2.3-2.3.7 | Gingerbread           | December 2010  |
| 3.0-3.2.6 | Honeycomb             | February 2011  |
| 4.0-4.0.4 | Ice Cream<br>Sandwich | October 2011   |
| 4.1-4.3.1 | Jelly Bean            | June 2012      |
| 4.4-4.4.4 | KitKat                | October 2013   |
| 5.0-5.0.2 | Lollipop              | November 2014  |

**Gambar 2.4.2.1 Sejarah Perilisan Android (Gilski, Stefanski, 2015)**

Versi lama dari Android, yakni *Cupcake* (Versi 1.5) dan *Donut* (Versi 1.6). Versi Android yang dirilis setelahnya sampai yang terbaru sekarang ini adalah:

#### **2.4.2.1 Éclair (Versi 2.0/2.1)**

Merupakan Android jenis ke-3 yang berbasis Linux kernel 2.6.32. *Messaging* diperbaharui dengan fungsi pencarian, memungkinkan untuk mencari menggunakan keseluruhan baik SMS maupun MMS. Juga ada fungsi *auto-delete* yang akan menghapus pesan lama setelah mencapai *inbox-limit*. UI (*User Interface*) juga sudah diperbaiki dan sudah mengenalkan HTML5 dan *Exchange ActiveSync 2.5 support*.

#### **2.4.2.2 Froyo (Versi 2.2)**

Merupakan Android jenis ke-4 yang berbasis Linux kernel 2.6.35. Versi ini sudah lebih maju dibanding versi sebelumnya. RAM pada versi ini dibebaskan untuk menjalankan aplikasi, hal ini membuat stabilitas dan kecepatan meningkat begitu juga pengoptimalan memori dan kinerja dari OS. Untuk pertama kalinya, *froyo* dibangun menggunakan JIT (*Just In Time*) *compilation* untuk membuat kinerja sistem lebih cepat dari sebelumnya. Kapabilitas tampilan dapat mendukung *screen* dengan *pixel densities* sampai dengan 320 ppi (*pixel-per-inch*). Langkah besar

buat *platform* adalah USB dan WIFI *hotspot tethering*. Hal ini membuat *user* dapat terkoneksi dengan internet secara langsung dari telpon, perubahan tambahan yakni penggunaan opsional PIN *password* daripada *touch signature*.

#### **2.4.2.3 Gingerbread (Versi 2.3)**

Merupakan Android jenis ke-5 yang sudah *user friendly* dan telah digunakna oleh banyak pengguna hingga sekarang. Fitur-fitur seperti kecepatan, *soft keyboard*, *copy/paste*, *power management*, dan UI telah diperbaiki dan ditingkatkan. Hal ini membuat perbedaan besar pada NFC (*Near Field Communication*) yang membantu untuk mengidentifikasi komunikasi jarak pendek antar *device*.

#### **2.4.2.4 Honeycomb (Versi 3.0)**

Android versi ini didesain untuk *device* dengan layar besar seperti tablet. Pada *Honeycomb* terdapat banyak elemen UI baru untuk tablet. Contohnya *System bar* baru pada bagian bawah layar untuk menggantikan *Status bar* pada telpon, *Notification bar* dan *Action bar* yang elegan yang terdapat pada *home screen* untuk menjalankan aplikasi bersama-sama yang membuatnya lebih praktis untuk pengguna. *Honeycomb* merupakan versi pertama dari *platform design* yang berjalan baik pada *single* maupun *multi-core processor architecture*. Versi ini memberikan *hardware-accelerated OpenGL renderer* yang baru yang memberikan *performance boost* ke banyak operasi grafik biasa untuk aplikasi yang berjalan pada *Android framework*.

#### **2.4.2.5 Ice Cream Sandwich (Versi 4.0)**

Merupakan Android versi ke-7 yang berbasis Linux kernel 3.0.1. *Ice Cream Sandwich* merupakan gabungan kemampuan dari *Gingerbread* dan *Honeycomb*

yang berpaut menjadi satu kesatuan utuh. Versi Android ini mengenalkan sejumlah fitur baru, termasuk *refreshed home screen*, *NFC support*, dan kemampuan untuk menge-“beam” isi ke pengguna lain yang menggunakan teknologi yang sama, *web browser* yang sudah diperbarui, *contact manager* yang baru dengan *social network integration*, kemampuan untuk mengakses kamera dan mengontrol *music playback* dari *lock screen*, *visual voicemail support*, *Face unlock*, kemampuan untuk memonitor dan membatasi penggunaan data ponsel, dan pengembangan *internal* lainnya.

#### **2.4.2.6 Jellybean (Versi 4.1)**

Versi yang dirilis berikutnya adalah *Jellybean* yang berbasis Linux Kernel 3.0.31. *Jellybean* membawa *multi-user accounts*, *actionable notifications*, *lock screen widgets*, *quick-settings* pada *notification bar*, *Android camera*, dan *Google talk* yang bekerja seperti *Siri-like functionality* yang mengubah perintahnya kekosongan. Beberapa fitur ditingkatkan dari versi sebelumnya, seperti *Equalise music*, *Turn notification off*, *Rearrange home screen*, *Fling to remove*, *Access App info*, *Volume control*, *Talkback*, *Offline speech-to-text*, dan *Reboot to safe mode*.

#### **2.4.2.7 Kitkat (Versi 4.4)**

Versi yang dirilis berikutnya adalah *Kitkat* yang memiliki banyak perkembangan, beberapa diantaranya ada yang bekerja di balik layar yang meningkatkan pengalaman pengguna Android. Fitur yang paling terlihat didesain untuk bekerja pada *device* dengan penggunaan RAM sesedikit mungkin yakni 512 MB, mendukung *cloud storage solution* seperti *Google Drive* yang dibangun ke dalam *operating system*, *printing on the go* yang memungkinkan mencetak dokumen

melalui WiFi atau *Bluetooth*.

#### **2.4.2.8 Lollipop (Versi 5.0)**

Versi yang paling baru dirilis adalah *Lollipop*. Pada versi Android ini, Dalvik diganti dengan ART dengan AOT (*Ahead-Of-Time*) compilation dan mendukung device 64-bit. Memiliki *OpenGL ES 3.1 support*, peningkatan daya baterai, *recent activities screen* dengan *task* daripada *application*, dan *audio input/output* melalui USB. (Chinetha dkk, 2015) dan (Bhardwaj dkk, 2013)

#### **2.4.3 Android SDK**

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, android *member* anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone* atau *Smartphone*. (Developers, 2015)

#### **2.4.4 API (*Application Programming Interface*)**

API (*Application Programming Interface*) bukan hanya satu *set class* dan *method* atau fungsi dan *signature* yang sederhana. API yang bertujuan utama untuk mengatasi “*clueless*” dalam membangun *software* yang berukuran besar, berawal

dari sesuatu yang sederhana sampai ke yang kompleks dan merupakan perilaku komponen yang sulit dipahami. (Halim, 2011)

## **2.5 Metodologi Pembuatan Aplikasi**

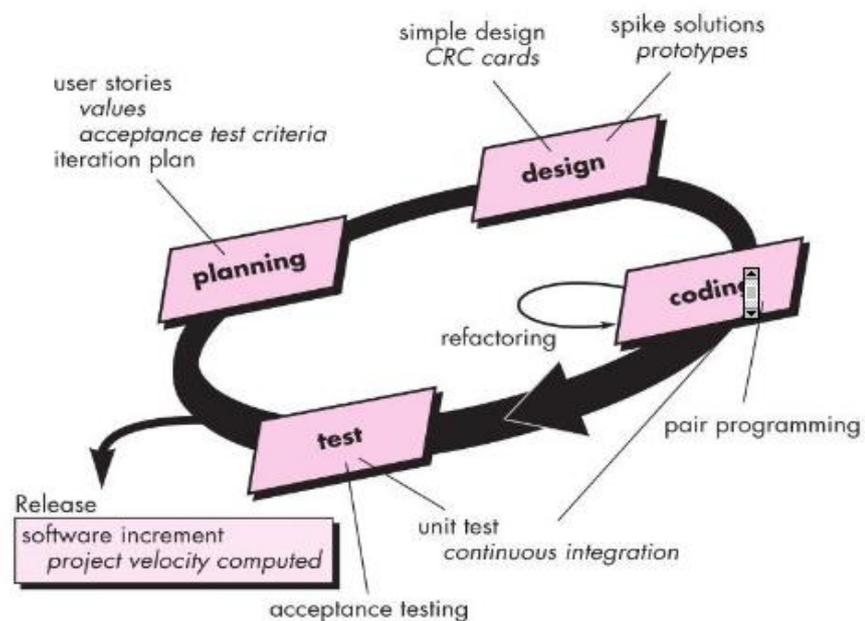
### **2.5.1 *Agile Methods Model Extreme Programming***

Pada dekade 90-an diperkenalkan metode baru yang dikenal dengan nama *Agile Methods*. Metode ini sangat revolusioner perubahannya jika dibandingkan dengan berbagai metode sebelumnya. *Agile Methods* dikembangkan karena pada metode tradisional terdapat banyak hal yang membuat proses pengembangan tidak dapat berhasil dengan baik sesuai tuntutan *user*. Menurut Widodo dan Subekti, saat ini *Agile Methods* sudah cukup banyak berkembang. Macam – macam *Agile Methods* yang berkembang diantaranya adalah:

1. *eXtreme Programing(XP)*
2. *Scrum Methodology*
3. *Crystal Family*
4. *Dynamic Systems Development Method (DSDM)*
5. *Adaptive Software Development (ASD)*
6. *Feature Driven Development (FDD)*

Salah satu model yang umum digunakan dalam *Agile Methods* adalah *eXtreme Programing (XP)*. Model ini merupakan metode pengembangan perangkat lunak yang ringan dan dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. XP merupakan *Agile Methods* yang paling banyak digunakan dan menjadi sebuah pendekatan yang sangat terkenal. Sasaran XP adalah tim yang dibentuk berukuran

antara kecil sampai sedang saja, tidak perlu menggunakan sebuah tim yang besar. Hal ini dimaksudkan untuk menghadapi *requirements* yang tidak jelas maupun terjadinya perubahan-perubahan *requirements* yang sangat cepat. Menurut Pressman (2001), terdapat 4 (empat) tahapan pada pengembangan perangkat lunak yang menggunakan XP yaitu *planning*, *design*, *coding*, dan *testing*. Tahapan-tahapan yang ada pada *extreme programming* dapat dilihat pada Gambar 2.3.1.1



**Gambar 2.5.1.1 Tahapan *eXtreme Programming* (Pressman, 2001)**

## 2.5.2 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah sebuah standar bahasa pemodelan yang memungkinkan untuk menspesifikasi, memvisualisasi, membangun, dan mendokumentasikan sebuah sistem perangkat lunak. Tujuan dari pemodelan ini

adalah untuk memodelkan sistem perangkat lunak dari segi pembangunan, produksi, kualitas, pengurangan biaya, dan juga waktu.

UML (*Unified Modeling Language*) merupakan standard yang relative terbuka yang dikontrol oleh OMG (*Object Management Group*), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar COBRA COBRA (*Common Object Request Broker Architecture*).

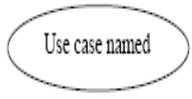
UML dideskripsikan oleh beberapa diagram, yaitu sebagai berikut:

### 1. *Use Case Diagram*

*Use Case Diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (user), sehingga pembuatan use case diagram lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah use case diagram merepresentasikan sebuah interaksi antara aktor dengan sistem. (Fowler, 2004)

**Tabel 2.5.2.1 Notasi *Use Case Diagram***

| SIMBOL  | NAMA  | KETERANGAN  |
|---|-------|---|
|  | Aktor | Aktor adalah pengguna sistem. Aktor tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan <i>output</i> , maka |

|   |                 |   |
|---|-----------------|---|
|   |                 | aplikasi tersebut juga bisa dianggap sebagai aktor.   |
|  | <b>Use Case</b> | <i>Use case</i> digambarkan sebagai lingkaran elips dengan nama ' <i>use case</i> ' dituliskan didalam elips tersebut.  |
|  | <b>Asosiasi</b> | Asosiasi digunakan untuk menghubungkan aktor dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara Aktor dengan <i>use case</i> . |

Komponen – komponen pada *use case diagram*:

#### 1. Aktor

Pada dasarnya actor bukanlah bagian dari use case diagram, namun untuk dapat terciptanya suatu *use case* diagram diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*.

#### 2. Use Case

Gambaran fungsionalitas dari suatu sistem, sehingga penggunasistem paham dan mengerti kegunaan sistem yang akan dibangun. Ada beberapa relasi yang terdapat pada *use case diagram*:

- a. *Association*, menghubungkan link antar element;

- b. *Generalization*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya;
- c. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya;
- d. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi yang mungkin terjadi pada *use case diagram*:

- a. *<<include>>*, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya;
- b. *<<extends>>*, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan;
- c. *<<communicates>>*, merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara aktor dan *use case*

## 2. Activity Diagram

Menggambaran rangkaian aliran aktivita, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat digunakan untuk aktivitas lainnya. (Fowler, 2004)

**Tabel 2.5.2.2 Notasi Activity Diagram (Meildy, 2014)**

| Simbol  | Keterangan |
|---|------------|
|  | Titik Awal |

|   |  |
|---|--|
|    | Titik Akhir  |
|    | <i>Rake</i> ; Menunjukkan adanya dekomposisi   |
|    | Tanda Waktu  |
|    | Tanda penerimaan   |
|    | Aliran akhir ( <i>Flow Final</i> )   |
|    | Tanda pengiriman   |
|    | <i>Activity</i>  |
|  | Pilihan untuk mengambil keputusan  |
|  | <i>Fork</i> ; Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |

Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case*. (Fowler, 2004)

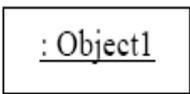
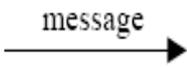
### 3. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara

objek yang terjadi pada titik tertentu dalam eksekusi sistem. (Fowler, 2014)

Berikut ini adalah notasi *sequence diagram*:

**Tabel 2.5.2.3 Notasi *Sequence Diagram* (Meildy, 2014)**

| Simbol  | Nama            | Keterangan   |
|---|-----------------|--|
|    | <b>Object</b>   | <i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma |
|   | <b>Actor</b>    | <i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>Actor</i> sama dengan simbol pada <i>Actor Use Case Diagram</i> .                                 |
|  | <b>Lifeline</b> | <i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah obyek.  |
|  | <b>Action</b>   | <i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.                                      |
|  | <b>Message</b>  | <i>Message</i> , digambarkan dengan anak panah <i>horizontal</i> antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> .  |

## 2.6 SQLite

SQLite merupakan salah satu *embedded database* yang populer pada *open source system*. Database ini sangat luas diaplikasikan pada manajemen data dari *embedded environment* seperti *mobile device*, *industrial control*, *information appliance*, dll. Kelebihan dari database ini adalah stabil, tahan uji, cepat, efisiensi tinggi, *portability*, dll. Android memberikan dukungan kepada SQLite database. Aplikasi dapat membuat *private database* yang diidentifikasi oleh nama. Database tersebut nantinya akan disimpan di `‘/data/data/<app package>/databases’` *directory* dengan nama yang diberikan oleh pembuatnya sebagai nama dari *file* tersebut. (Hay, 2012)(Bi, 2009)

## 2.7 Penelitian Sejenis

### 2.7.1 Android Lunar View

Penelitian ini dilakukan oleh Vincent Cheung pada tahun 2016. Penelitian yang beliau lakukan yakni untuk menampilkan hasil konversi dari *Chinese Calendar* ke *Solar Calendar* pada *Solar Calendar*. Hasil penelitian ini yakni aplikasi Chinese Lunar Calendar yang dapat mengkonversi *Chinese Calendar* ke *Solar Calendar* dari tahun 1900 sampai dengan tahun 2100.

Pada aplikasi ini, terdapat library widget baru yang terdiri dari beberapa program java yang nantinya akan digunakan untuk mengkonversi *Chinese Calendar* ke *Solar Calendar*. Ketika program ini dijalankan, maka *Solar Calendar* yang berisi hari raya di China dan hasil konversi *Chinese Calendar* ke *Solar Calendar* akan ditampilkan. Pada tampilan awal ini, juga terdapat tombol yang akan

mengarahkan *user* ke tanggal pada hari ini. Pada bagian atas tombol, user dapat melihat tanggal hari ini dalam tanggal *Solar* dan *Chinese*. (github.com/Coolerfall/Android-LunarView, 2017)

## 2.8 Metode Pengujian

### 2.8.1 *Black Box Testing*

*Black Box Testing* atau pengujian *Black Box* adalah suatu metode yang digunakan untuk menguji perangkat lunak dari segi spesifikasi fungsionalitas tanpa melakukan pengujian desain dan kode program. Pengujian *Black Box* dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan (*input*) dan keluaran (*output*) sudah sesuai dengan yang dibutuhkan. (Sukamto dan Shalahudin, 2011)

*Black Box Testing* juga merupakan pendekatan komplementer dari tehnik *White Box Testing* karena *Black Box Testing* mampu menangkap kesalahan yang lebih luas. *Black Box Testing* berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapatkan serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional program. (Yakub, 2012)

Tujuan dari pengujian *Black Box* adalah untuk menemukan:

1. Fungsi yang hilang atau tidak benar;
2. Kesalahan Interface;
3. Kesalahan atau *error* pada struktur data atau kses eksternal database;

4. Kesalahan atau *error* pada kinerja;
5. Batasan dari suatu data.

(Gries dan Fred, 2005)

### **2.8.2 *Equivalent Partitioning* (EP)**

*Equivalent Partitioning* (EP) merupakan suatu cara penurunan kasus uji. Partisi tersebut bergantung pada penemuan partisi himpunan data masukan (input) dan keluaran (output) serta melatih program dengan nilai-nilai dari partisi tersebut. Seringkali nilai yang paling mungkin menghasilkan uji yang berhasil merupakan nilai pada batas partisi. (Sommerville, 2003)

EP dapat diidentifikasi dengan menggunakan spesifikasi program atau dokumentasi pengguna dan dengan penguji menggunakan pengalamannya untuk meramalkan kelas input mana yang mungkin mendeteksi kesalahan (*error*) (Sommerville, 2003).

### **2.8.3 Skala Likert**

Metode penjumlahan rating merupakan metode penskalaan pernyataan sikap yang menggunakan distribusi respon sebagai dasar penentuan nilai skala. Nilai skala setiap pernyataan ditentukan oleh distribusi respon setuju dan tidak setuju dari responden yang bertindak sebagai kelompok uji coba. (Azwar, 2011)

Skala Likert merupakan skala yang memiliki lima tingkat preferensi jawaban dengan pilihan jawaban. (Azwar, 2011)

**Tabel 2.8.3.1 Preferensi Jawaban Skala Likert**

| <b>Angka</b> | <b>Keterangan</b>     |
|--------------|-----------------------|
| 1            | Sangat tidak setuju   |
| 2            | Tidak setuju          |
| 3            | Ragu-ragu atau netral |
| 4            | Setuju                |
| 5            | Sangat setuju         |

Untuk memperoleh presentase penilaian dari setiap kategori pertanyaan, digunakan skala likert yang diperoleh dengan rumus aritmatika sebagai berikut.

(Azwar, 2011)

$$P = \frac{Xi}{n \times N} \times 100\%$$

P = Presentase pertanyaan

Xi = Nilai kuantitatif total

n = Jumlah responden

N = Nilai item pernyataan terbaik

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Waktu dan Tempat Penelitian**

Penelitian ini dilakukan di Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang berada di Jln. Soemantri Brojonegoro no. 1 Gedung Meneng, Bandar Lampung pada semester genap tahun ajaran 2016/2017.

#### **3.2 Alat Pendukung Penelitian**

Penelitian ini dilakukan dengan menggunakan alat pendukung, antara lain:

##### *1. Hardware*

- *Notebook* dengan spesifikasi:

- *Processor* : Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz

2.40 GHz

- *Installed Memory(RAM)* : 4.00 GB

- *System Type* : 64-bit *Operating System*, x64-based *processor*

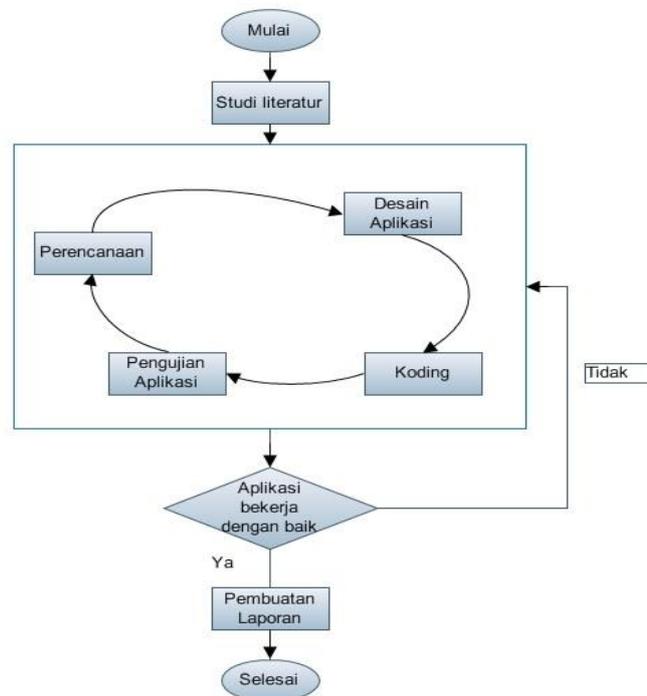
## 2. Software

- Sistem Operasi Windows 8.1
- Android Studio
- Notepad ++
- Adobe Photoshop CS 6
- Browser SQLite Database

## 3.3 Tahapan Penelitian

Penelitian ini dilakukan dengan mengikuti tahapan penelitian seperti pada

Gambar 3.3.1 berikut ini.



**Gambar 3.3.1 Alur Pembuatan Sistem**

Keterangan:

#### 1. Studi Literatur

Pada tahap ini, dilakukan pencarian teori dan hasil penelitian terkait pembuatan sistem yang akan dilakukan untuk dijadikan landasan penelitian.

#### 2. Perencanaan

Pada tahapan ini, dilakukan pengumpulan kebutuhan *user*. Hal ini dilakukan agar pengembang mengetahui *output* dan fitur-fitur dari sistem yang akan dibuat.

#### 3. Desain Sistem

Desain sistem dari sistem ini digambarkan dengan model UML berupa *Use Case Diagram*, *Activity Diagram*, dan Alur Diagram. Pembuatan desain pada XP tetap mengedepankan prinsip KIS (*Keep It Simple*). Desain dibuat untuk memudahkan pengembang dalam membangun sistem ini.

#### 4. Koding

Dalam pembangunan program ini digunakan bahasa pemrograman Java dan xml.

#### 5. Pengujian Sistem

Pengujian sistem yang sudah dibuat apakah sudah memenuhi kebutuhan dan *output* yang diinginkan *user*.

#### 6. Pembuatan Laporan

Tahap akhir penelitian dimana proses pembuatan hingga akhirnya tercipta suatu sistem dituangkan dalam suatu bentuk karya tulis ilmiah.

### **3.4 Desain Sistem**

#### **3.4.1 Analisis Kebutuhan Sistem**

Kebutuhan dasar dari sistem ini adalah:

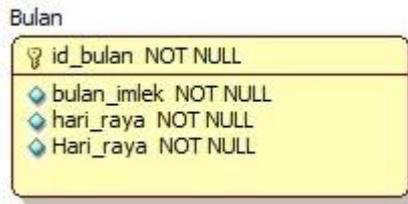
1. Sistem dapat mengkonversi *Chinese Calendar* menjadi *Solar Calendar*
2. Sistem dapat menampilkan *Chinese Calendar* yang sudah dikonversi ke *Solar Calendar* beserta hari raya pada hari ini
3. Sistem dapat menampilkan daftar hari keagamaan agama Buddha

#### **3.4.2 Perancangan Sistem**

Perancangan sistem merupakan tahap selanjutnya setelah analisis kebutuhan sistem untuk mendapatkan gambaran yang jelas tentang apa yang dikerjakan pada analisis sistem agar dapat membuat rancangan sistem yang baik. Perancangan sistem ini akan menentukan bagaimana sistem yang dibuat akan bekerja nantinya dalam upaya pelaksanaan tugas sesuai dengan yang ditentukan. Tahap ini termasuk komponen-komponen *hardware* dan *software* suatu sistem agar setelah instalasi akan memenuhi rancangan sistem yang telah dibuat.

#### **3.4.3 ERD (*Entity Relationship Diagram*)**

ERD merupakan representasi data pada *database*. Pada ERD, terdapat entity dan relationship antar entitas. ERD pada Gambar 3.4.3.1 berikut ini menggambarkan entitas pada *database* yang digunakan pada aplikasi Kalender Buddhis.



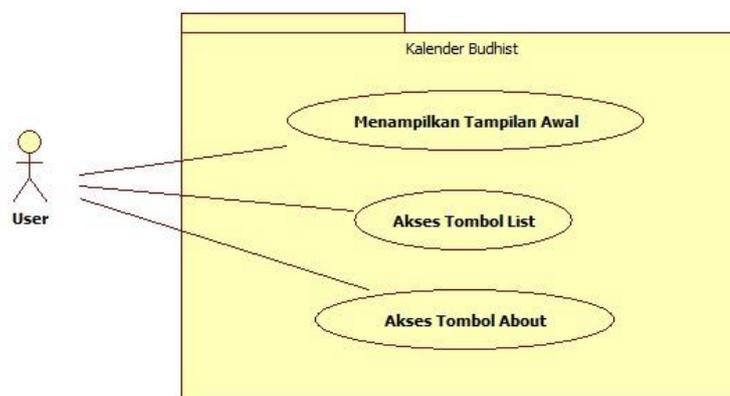
**Gambar 3.4.3.1 ERD (*Entity Relationship Diagram*)**

### 3.4.4 Perancangan UML (*Unified Modeling Language*)

Pemodelan adalah tahap merancang *software* sebelum melakukan koding. Pada penelitian ini, perancangan sistem dilakukan dengan memodelkan permasalahan dalam bentuk diagram-diagram UML sebagai berikut:

#### 1. *Use Case Diagram*

*Use Case Diagram* berikut ini menggambarkan bagaimana *user* menggunakan sistem yang dititik beratkan pada fungsionalitas sistem. Aplikasi ini akan menampilkan *Chinese Calendar* yang sudah dikonversi ke *Solar Calendar* kepada user untuk tampilan awalnya. Pada tampilan awal ini, terdapat 2 (dua) tombol fungsi yakni tombol *List* dan tombol *About*. *Use Case Diagram* aplikasi Kalender Buddhis dapat dilihat pada Gambar 3.4.4.1.



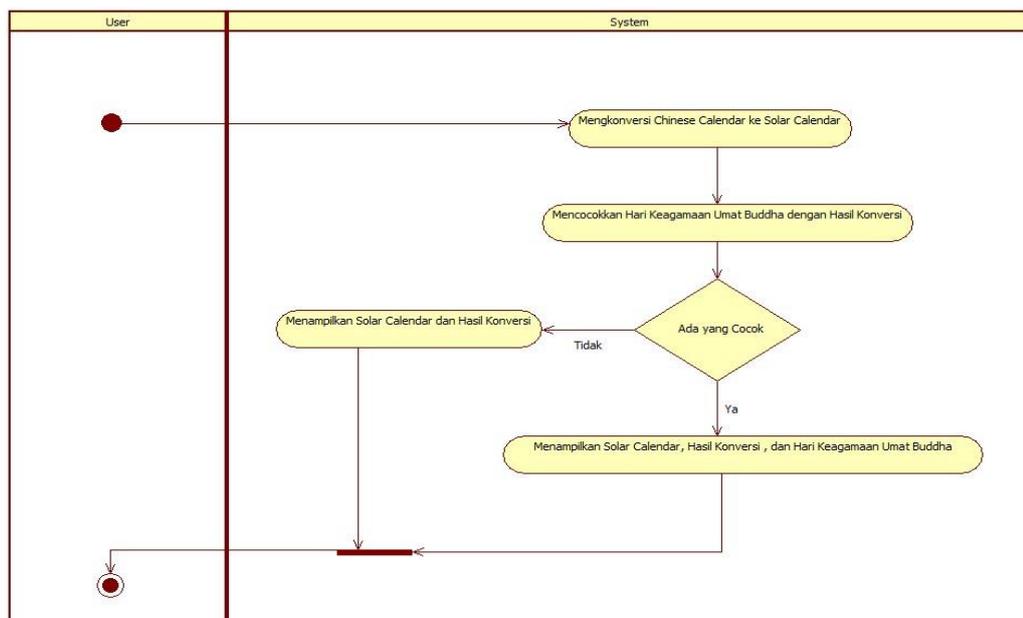
**Gambar 3.4.4.1 *Use Case Diagram***

## 2. Activity Diagram

*Activity Diagram* menggambarkan *workflow* (alur kerja) dari tiap *use case* yang ada pada *use case diagram*. Ada 3 (tiga) *activity diagram* pada aplikasi Kalender Buddhis, yaitu:

### a. *Activity Diagram* menampilkan Tampilan Awal

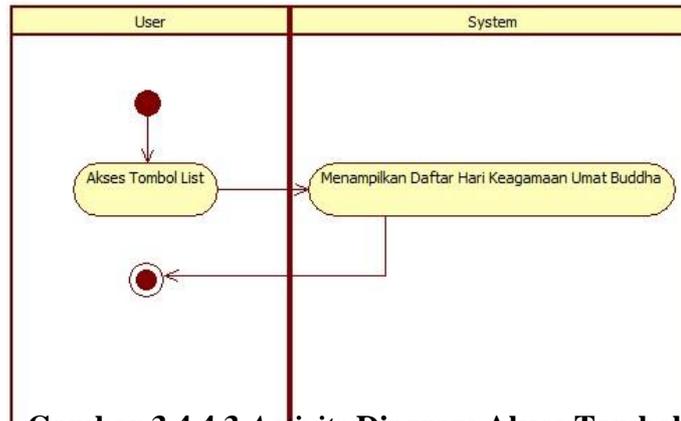
Ketika *user* membuka aplikasi Kalender Buddhis, sistem akan mulai mengkonversi *Chinese Calendar* ke *Solar Calendar*. Hasil dari konversi tersebut akan dicocokkan dengan daftar Hari Keagamaan Umat Buddha yang ada di *database*. Jika hasil konversi sama dengan data yang ada, maka akan ditampilkan *Solar Calendar*, hasil konversi, dan Hari Keagamaan Umat Buddha sebagai tampilan awalnya. Jika hasil konversi tidak sama dengan data yang ada, maka akan ditampilkan *Solar Calendar* dan hasil konversi sebagai tampilan awalnya. *Activity Diagram* untuk menampilkan Tampilan Awal dapat dilihat pada Gambar 3.4.4.2.



**Gambar 3.4.4.2 Activity Diagram Menampilkan Tampilan Awal**

b. *Activity Diagram* akses tombol *List*

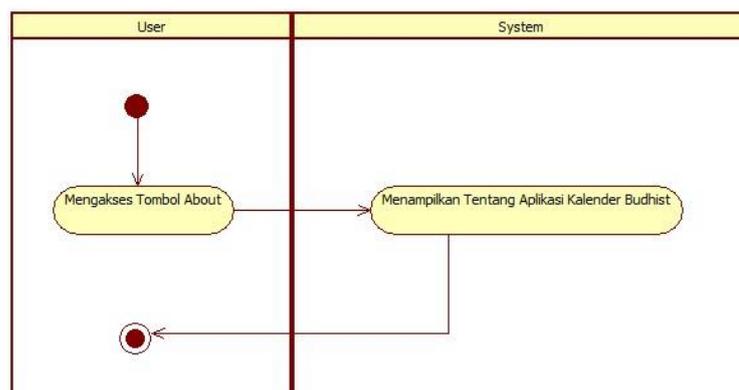
Ketika tombol *List* pada tampilan awal diklik, sistem akan menampilkan daftar Hari Keagamaan Umat Buddha. *Activity Diagram* akses tombol *List* dapat dilihat pada Gambar 3.4.4.3.



**Gambar 3.4.4.3 Activity Diagram Akses Tombol *List***

c. *Activity Diagram* akses tombol *About*

Ketika tombol *About* pada tampilan awal diklik, sistem akan menampilkan Tentang Aplikasi Kalender Buddhis. *Activity Diagram* akses tombol *About* dapat dilihat pada Gambar 3.4.4.4.



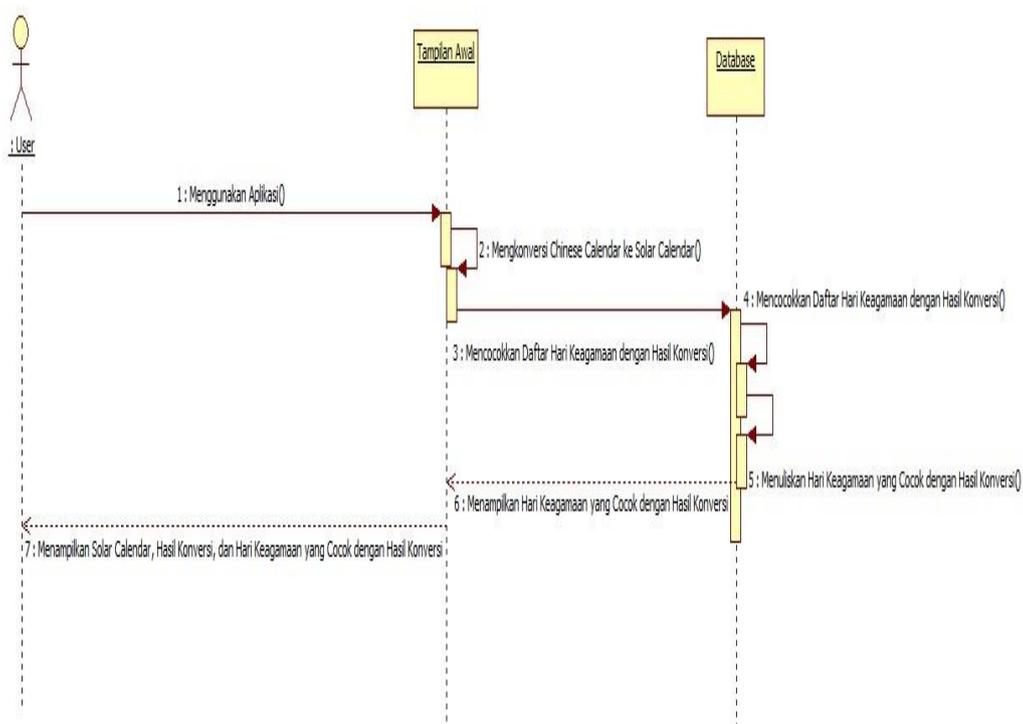
**Gambar 3.4.4.4 Activity Diagram Akses Tombol *About***

### 3. Sequence Diagram

*Sequence Diagram* berfungsi untuk menampilkan bagaimana interaksi objek dalam suatu aktivitas. Ada 3 (tiga) aktivitas pada aplikasi Kalender Buddhis, yaitu:

#### a. *Sequence Diagram* menampilkan Tampilan Awal

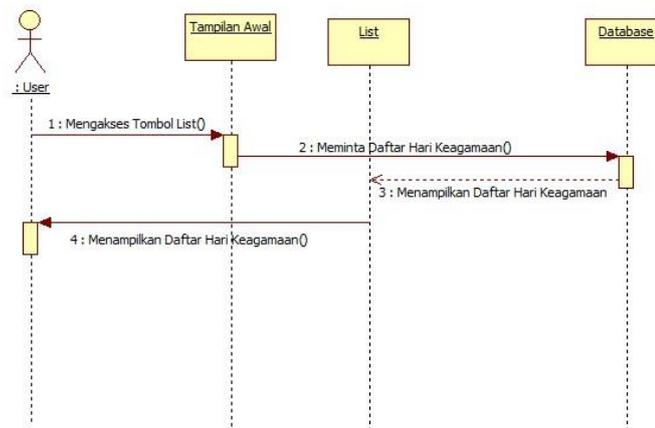
Ketika *user* membuka aplikasi Kalender Buddhis, sistem akan menampilkan *Solar Calendar*. Selanjutnya, sistem akan mengkonversi *Chinese Calendar* ke *Solar Calendar*. Ketika proses konversi selesai, sistem akan mencocokkan hasil konversi dengan daftar Hari Keagamaan Umat Buddha yang ada di *database*. Jika ada yang sama maka akan ditampilkan kepada *user*. *Sequence Diagram* menampilkan Tampilan Awal dapat dilihat pada Gambar 3.4.4.5.



**Gambar 3.4.4.5 Sequence Diagram Menampilkan Tampilan Awal**

b. *Sequence Diagram* akses tombol *List*

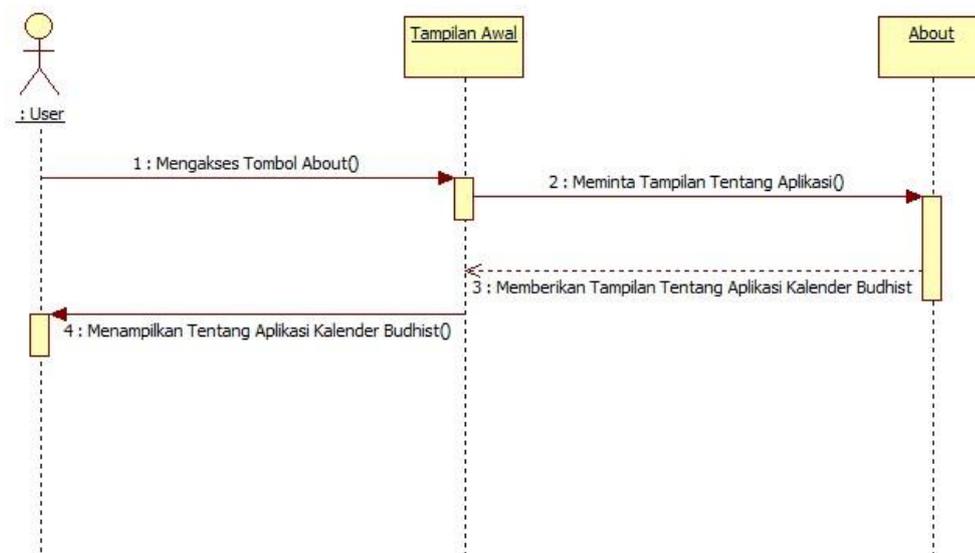
Ketika *user* sudah berada pada tampilan awal, terdapat 2 (dua) buah tombol yakni *List* dan *About*. Saat *user* mengklik tombol *List*, sistem akan mengambil seluruh data Hari Keagamaan Umat Buddha yang ada di *database* dan menampilkannya kepada *user*. *Sequence Diagram* akses tombol *List* dapat dilihat pada Gambar 3.4.4.6.



**Gambar 3.4.4.6** *Sequence Diagram* Akses Tombol *List*

c. *Sequence Diagram* akses tombol *About*

Ketika *user* sudah berada pada tampilan awal, terdapat 2 (dua) buah tombol yakni *List* dan *About*. Saat *user* mengklik tombol *About*, sistem akan menampilkan halaman yang berisi informasi tentang aplikasi Kalender Buddhis. *Sequence Diagram* akses tombol *About* dapat dilihat pada Gambar 3.4.4.7



**Gambar 3.4.4.7 Sequence Diagram Akses Tombol About**

### 3.4.5 Perancangan Antarmuka

Perancangan Antarmuka merupakan proses pengembangan *Interface* (tampilan) yang akan ditampilkan kepada *user*. Dalam aplikasi Kalender Buddhis terdapat beberapa *layout*, yakni:

#### 1. *Layout* Tampilan Awal

Ketika *user* menggunakan aplikasi Kalender Buddhis, sistem akan menampilkan *Solar Calendar*, *Chinese Calendar* yang sudah dikonversi ke *Solar Calendar*, dan hari keagamaan (jika ada) sebagai tampilan awal. Pada tampilan awal ini terdapat 2 (dua) buah tombol yakni tombol *List* dan tombol *About*. Tombol *List* akan mengarahkan *user* ke *Layout List*, dan tombol *About* akan mengarahkan *user* ke *Layout About*. *Layout* tampilan awal dapat dilihat pada Gambar 3.4.5.1.



**Gambar 3.4.5.1 *Layout Tampilan Awal***

## *2. Layout List*

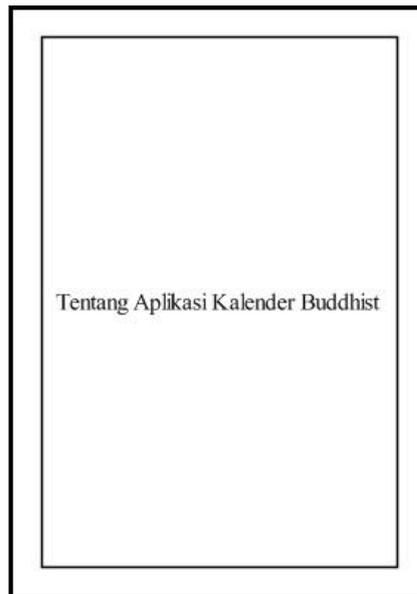
Ketika *user* mengklik tombol *List*, aplikasi Kalender Buddhis akan menampilkan daftar hari-hari keagamaan umat Buddha. *Layout List* dapat dilihat pada Gambar 3.4.5.2.



**Gambar 3.4.5.2 *Layout List***

### 3. *Layout About*

Ketika *user* mengklik tombol *About*, aplikasi Kalender Buddhis akan menampilkan Tentang Aplikasi Kalendaer Buddhis. *Layout About* dapat dilihat pada Gambar 3.4.5.3.



**Gambar 3.4.5.3 *Layout About***

### 3.5 Data Hari Keagamaan

Data hari keagamaan didapatkan dengan cara berkunjung ke vihara-vihara dan kuil-kuil yang beraliran Buddhayana, Mahayana, dan Maitreya yang ada di Bandar Lampung dan bertanya kepada pengurusnya. Setelah data didapatkan, seluruh data tersebut dijadikan satu dan diurutkan sesuai tanggal menurut *Chinese Calendar*-nya. Data keseluruhan hari keagamaan yang akan ditampilkan pada *Layout List*, merupakan gabungan keseluruhan hari keagamaan dari agama Buddha yang beraliran Buddhayana, Mahayana, dan Maitreya dengan hanya mengutamakan kecocokan tanggal pada *Chinese Calendar*-nya

### 3.6 Pengujian Sistem

Pengujian Sistem merupakan salah satu cara atau metode untuk menguji perangkat lunak dan data untuk menemukan kemungkinan adanya kesalahan pada perangkat lunak tersebut.

#### 3.6.1 Pengujian Fungsional

Perencanaan kasus uji coba untuk pengujian fungsional adalah *Black Box* dengan metode *Equivalent Partitioning* (EP). Pengujian ini dilakukan dengan membagi *domain* masukan ke dalam kelas-kelas sehingga *test case* untuk pengujian perangkat lunak dapat diperoleh. Perencanaan pengujian EP Kalender Buddhis, dapat dilihat pada Tabel 3.6.1.1.

**Tabel 3.6.1.1 Perencanaan Pengujian Menggunakan Metode EP (*Equivalent Partitioning*)**

| No. | Kelas Uji     | Daftar Pengujian                                       | Skenario Uji  | Relosiasi yang Diharapkan                                  |
|-----|---------------|--|---|--|
| 1.  | Versi Android | Pengujian Kompatibilitas aplikasi dengan versi Android | Pengujian pada Android versi 4.1 ( <i>Jelly Bean</i> )  | Kompatibel dengan Android versi 4.1 ( <i>Jelly Bean</i> )  |
|     |               |  | Pengujian pada android versi 4.4 ( <i>Kitkat</i> )      | Kompatibel dengan Android versi 4.4 ( <i>Kitkat</i> )      |
|     |               |  | Pengujian pada Android versi 5.0 ( <i>Lollipop</i> )    | Kompatibel dengan Android versi 5.0 ( <i>Lollipop</i> )    |
|     |               |  | Pengujian pada Android versi 6.0 ( <i>Marshmallow</i> ) | Kompatibel dengan Android versi 6.0 ( <i>Marshmallow</i> ) |

|    |                             |  |   |   |
|----|-----------------------------|--|---|---|
| 2. | Resolusi dan Densitas Layar | Pengujian Resolusi dan Densitas Layar pada Android | Pengujian pada Android dengan resolusi 4 inch           | Tampilan terlihat baik pada Android dengan resolusi 4.5 inch  |
|    |                             |  | Pengujian pada Android dengan resolusi 5 inch           | Tampilan terlihat baik pada Android dengan resolusi 5 inch  |
|    |                             |  | Pengujian pada Android dengan resolusi 5.5 inch         | Tampilan terlihat baik pada Android dengan resolusi 5.5 inch  |
|    |                             |  | Pengujian pada Android dengan resolusi 6 inch           | Tampilan terlihat baik pada Android dengan resolusi 6 inch  |
| 3. | <i>User Interface</i>       | Pengujian pada <i>Layout Tampilan Awal</i>         | Tampilan <i>Default</i>                                 | Menampilkan <i>Solar Calendar</i> , hasil konversi <i>Chinese Calendar</i> ke <i>Solar Calendar</i> , dan Hari Raya Umat Buddha yang sesuai dengan hasil konversi tersebut (jika ada) |
|    |                             |  | Klik tombol <i>List</i>                                 | Menampilkan isi dari <i>Layout List</i>   |
|    |                             |  | Klik tombol <i>About</i>                                | Menampilkan isi dari <i>Layout About</i>  |
|    |                             | Pengujian pada <i>Layout List</i>                  | Menampilkan daftar Hari Keagamaan Umat Buddha           |   |
|    |                             | Pengujian pada <i>Layout About</i>                 | Menampilkan informasi tentang aplikasi Kalender Buddhis |   |

### 3.6.2 Pengujian Non-Fungsional

Pengujian non-fungsional berfokus pada penilaian aspek kemudahan dalam menggunakan dan pemahaman aplikasi. Perencanaan untuk pengujian non-fungsional dilakukan dengan menggunakan *Skala Likert*. Perencanaan tersebut dapat dilihat pada Tabel 3.6.2.1.

**Tabel 3.6.2.1 Perencanaan Pengujian *Skala Likert***

| No | Kriteria Penilaian                                | Kategori Penilaian |        |         |         |         |
|----|---|--------------------|--------|---------|---------|---------|
|    |   | 5<br>SB            | 4<br>B | 3<br>CB | 2<br>KB | 1<br>TB |
| 1. | Kesesuaian warna <i>background</i> dengan tulisan |                    |        |         |         |         |
| 2. | Kesesuaian warna <i>background</i> dengan icon    |                    |        |         |         |         |
| 3. | Fungsi yang disediakan oleh aplikasi              |                    |        |         |         |         |
| 4. | Kemudahan dalam penggunaan aplikasi               |                    |        |         |         |         |
| 5. | Kemudahan dalam memahami informasi yang diberikan |                    |        |         |         |         |
| 6. | Respon aplikasi saat digunakan                    |                    |        |         |         |         |

Keterangan:

SB = Sangat Baik; B = Baik; CB = Cukup Baik; KB = Kurang Baik;

TB=Tidak Baik

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Kesimpulan yang dapat diambil dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Aplikasi Kalender Buddhis berbasis Android telah berhasil dibuat
2. Berdasarkan hasil pengujian fungsional dapat disimpulkan bahwa semua fungsi pada aplikasi Kalender Buddhis dapat berjalan dengan baik
3. Berdasarkan hasil pengujian non-fungsional dapat disimpulkan bahwa aplikasi Kalender Buddhis yang telah dibuat termasuk ke dalam kategori “Sangat Baik” dengan hasil rata-rata yang diperoleh yakni 85,11%

#### **5.2 Saran**

Saran untuk pengembangan aplikasi Kalender Buddhis, yakni untuk penambahan fungsi alarm untuk memberitahu *user* jika ada hari raya pada hari tersebut.

## DAFTAR PUSTAKA

- Aslaksen, Helmer. 2010. *The Mathematics of Chinese Calendar*. National University of Singapore: Singapore.
- Android Developers. 2015. *Android Developers*. [Online]. Tersedia: <http://developer.android.com/index.html>. Diakses pada tanggal 15 Agustus 2016.
- Azwar, S. 2011. *Sikap dan Perilaku Dalam: Sikap Manusia, Teori, dan Pengukurannya*. 2<sup>nd</sup> ed. Yogyakarta: Pustaka Pelajar.
- Bhardwaj, Saurabh, dkk. 2013. *Android Operating System*. Vol. 1, Iss. 1, pp 147 – 150. *International Journal of Engineering Technology & Management Research*.
- Bi, Chunyue. 2009. *Research and Application of SQLite Embedded Database Technology*. Vol. 8, Iss. 1, pp. 83 – 92. *WSEAS Transactions of Computers*.
- Brebon, Juliet, dan Igor Mitrophanon. 2005. *The Moon Year: A Record of Customs and Festivals*. pp. 1 – 9. New York: Kegan Paul.
- Chinetha, K., dkk. 2015. *An Evolution of Android Operating System and Its Version*. Vol. 2, pp. 30 – 33. *International Journal of Engineering and Applied Sciences (IJEAS)*.

- Fan, Dorothy. *Three Types of Calendar In The World*. pp. 1 – 6. *International Interpretations of Chinese Calendar*. Diakses pada tanggal 5 September 2016.
- Fowler, Martin. 2004. *UML Distilled Panduan Singkat Bahasa Pemodelan Objek Standar Edisi ketiga*. Yogyakarta: Andi Publishing.
- Gilski, Przemyslaw, dan J. Stefanski. 2015. *Android OS: A Review*. Vol. 4, No. 1, pp. 116 – 120. *TEM Journal*.
- Gu, Lei, dkk. 2014. *Research and Development of Mobile Application for Android Platform*. Vol. 9, No. 4, pp. 187 – 198. *International Journal of Multimedia and Ubiquitous Engineering*.
- Guo, Jing, dkk. 2014. *The History of Calendar*. RUC 3<sup>rd</sup> Semester Project.
- Gries, David, dan Fred B. Schneider. 2005. *An Integrated Approach to Software Engineering Third Edition*. Pankaj Jalote. Indian Institute of Technology: India.
- H., Nazruddin Safaat. 2014. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android Revisi kedua*. Bandung: Informatika.
- Halim, J. I., et al. 2011 *Framework Pemetaan Data Berbasis Peta dengan Menggunakan Google Maps API* (Skripsi). Universitas Bina Nusantara: Jakarta.
- Hay, Roe. 2012. *Android SQLite Journal Information Disclosure*. pp. 1 – 2. *IBM Application Security Research Group*.

<http://everything2.com/title/Buddhist+Calendar>. Diakses pada tanggal 18 Juni 2016.

<https://github.com/Coolerfall/Android-LunarView>. Diakses pada tanggal 3 Mei 2017.

<https://lampung.kemenag.go.id/artikel/36168/data-keagamaan-kementerian-agama-provinsi-lampung>. Diakses pada tanggal 25 Agustus 2017.

[http://www.Buddhanet.net/cal\\_mnth.htm](http://www.Buddhanet.net/cal_mnth.htm). Diakses pada tanggal 18 Juni 2016.

<http://www.hko.gov.hk/gts/time/conversion.htm>. Diakses tanggal 25 Juli 2017.

Izzuddin, Ahmad. 2010. Sistem Penanggalan Buddha. Pp. 4 – 6. Institut Agama Islam Negeri Walisongo: Semarang.

Katti, M.M., dan Suhas Holla. 2012. *Android Based Mobile Application Development and Its Security*. Vol. 3 Iss. 3, pp 486 – 490. *International Journal of Computer Trends and Technology*.

Meildy, Bayu. 2014. Daftar Simbol. [Online]. Tersedia: <http://elib.unikom.ac.id/downoad.php?id=83238>. Diakses pada tanggal 22 Agustus 2016.

Pressman, R.S. 2001. *Software Engineering A Practitioner's Approach Fifth Edition*. The McGraw-Hill Companies, Inc: New York.

Sommerville, Ian. 2003. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.

Sukanto, Rosa Ariani, dan M. Salahudin. 2011. Modul pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Modula.

Tim Penyusun kamus Pusat Pembinaan dan Pengembangan Bahasa. 1989. Kamus Besar bahasa Indonesia. Jakarta: Balai Pustaka.

Yakub. 2012. Pengantar Sistem Informasi. Yogyakarta: Graha Ilmu.