

BAB III

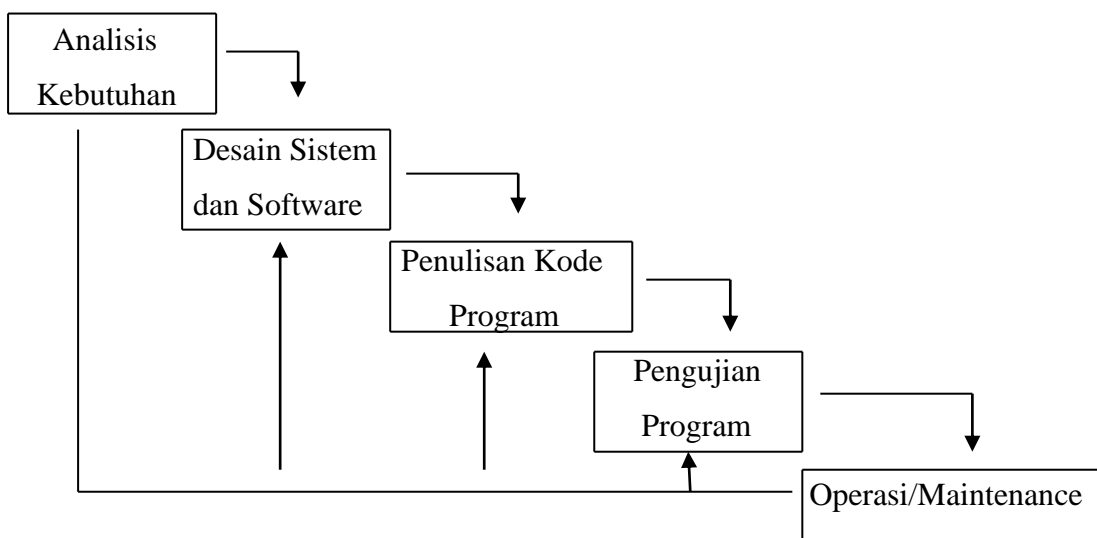
PERANCANGAN APLIKASI SISTEM INFORMASI AKADEMIK

3.1. METODE PENGEMBANGAN SISTEM

Langkah penyelesaian masalah tugas akhir ini sesuai dengan tahapan pengembangan perangkat lunak menggunakan model proses atau paradigma *waterfall*. Sebagai paradigma kehidupan klasik, *waterfall model* memiliki tempat penting dalam rekayasa perangkat lunak bahkan paradigma ini merupakan paradigma rekayasa perangkat lunak yang paling luas dipakai dan yang paling tua. Disamping itu, tahapan pada *waterfall model* mengambil kegiatan dasar yang digunakan dalam semua pengembangan perangkat lunak, sehingga dapat lebih mudah untuk dipahami terlebih bila hanya digunakan dalam mengembangkan perangkat lunak yang tidak begitu besar dan kompleks.

Waterfall model merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, dan evolusi dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi, pengujian unit, integrasi sistem, pengujian sistem, operasi dan pemeliharaan.

Tahap-tahap utama dari *waterfall model* pada gambar 3.1 merupakan pemetaan kegiatan-kegiatan pengembangan dasar, yaitu:^[4]



Gambar 3.1 *Waterfall Model*

1. Analisa Kebutuhan

Proses mengumpulkan informasi kebutuhan sistem/perangkat lunak melalui konsultasi dengan *user/system*. Proses ini mendefinisikan secara rinci mengenai fungsi-fungsi, atasan dan tujuan dari perangkat lunak sebagai spesifikasi sistem yang akan dibuat.

2. Desain Sistem dan *Software*

Proses perancangan sistem ini difokuskan pada 4 atribut yaitu struktur data, arsitektur perangkat lunak, representasi antarmuka dan detail (algoritma) prosedural. Struktur data adalah representasi dari hubungan logis antara elemen-elemen data individual. Perancangan sistem yang digunakan pada tugas akhir ini adalah menggunakan diagram *Unified Modeling Language* (UML).

3. Penulisan Kode Program (*Coding*)

Pada tahap ini merupakan proses menerjemahkan perancangan kedalam bahasa pemrograman yang dikenali oleh komputer, sehingga *user* dapat memahami sistem yang sedang dibangun.

4. Pengujian Program

Unit program/program individual diintegrasikan menjadi sebuah kesatuan sistem dan kemudian dilakukan pengujian. Dengan kata lain, pengujian ini ditujukan untuk menguji keterhubungan dari tiap-tiap fungsi perangkat lunak untuk mencari kesalahan yang mungkin terjadi serta menjamin bahwa persyaratan sistem telah terpenuhi. Setelah pengujian sistem selesai dilakukan, perangkat lunak dikirim ke pelanggan/*user*.

5. Operasi dan Pemeliharaan

Tahap ini biasanya memerlukan waktu yang paling lama. Sistem diterapkan (di-*install*) dan dipakai. Pemeliharaan mencakup koreksi dari beberapa kesalahan yang tidak diketemukan pada tahapan sebelumnya, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem.

3.2. ANALISIS KEBUTUHAN SISTEM

Analisis kebutuhan merupakan tahap yang dilakukan dalam menentukan kebutuhan sistem yang akan dibuat. Kebutuhan sistem yang diperlukan dalam penyusunan Tugas Akhir ini adalah:

3.2.1. *System Requirement*

System requirement adalah komponen berupa *hardware* maupun *software* yang digunakan untuk menjalankan suatu program atau aplikasi. *System requirement* yang digunakan dalam penyusunan Tugas Akhir ini adalah:

1. Laptop

Laptop digunakan sebagai media untuk merancang dan membuat aplikasi sistem informasi akademik. Kebutuhan minimal yang harus dimiliki oleh laptop untuk membuat aplikasi sistem informasi akademik ini tercantum pada tabel 3.1.

Tabel 3.1 Spesifikasi Laptop Dalam Pembuatan Aplikasi

	Spesifikasi Minimum	Spesifikasi yang Direkomendasikan
Jenis Processor	Pentium IV 2,4 GHz	Lebih baik/lebih cepat dibandingkan Pentium IV 2,4 GHz
Sistem Operasi	Windows XP (32 bit)	Windows 7 (32 bit)
RAM (<i>Random Access Memory</i>)	512 MB	1 GB atau lebih, hal ini ditujukan agar tidak membutuhkan waktu yang terlalu lama saat menjalankan emulator
<i>Hard Disk Space</i>	1 GB	2 GB atau lebih

Selain spesifikasi minimum *hardware* yang dibutuhkan oleh laptop, untuk membuat aplikasi sistem informasi akademik ini diperlukan beberapa perangkat lunak (*Software*) yang harus terinstall pada laptop yang digunakan. Beberapa perangkat lunak (*Software*) yang diperlukan adalah:

1. *Windows 7 Profesional* sebagai sistem operasi pada komputer.
 2. Paket JDK (*Java Developer Kit*) 7 sebagai perangkat bantu untuk menulis kode sumber, menguji dan men-*debug* program yang dibuat.
 3. Paket SDK (*Software Development Kit*) yaitu untuk mengembangkan aplikasi pada *platform* android dengan menggunakan bahasa pemrograman java.
 4. Paket IDE Eclipse Indigo sebagai bahasa pemrograman android pada aplikasi.
 5. *Plugin* ADT 20.0.0 merupakan seperangkat alat-alat pengembangan aplikasi untuk android.
2. *Handphone*

Handphone berbasis Android yang digunakan untuk melakukan pengujian terhadap aplikasi yang dibuat sehingga dapat dibuktikan apakah aplikasi yang telah dibuat berhasil atau tidak pada saat dijalankan. Kebutuhan minimal yang harus dimiliki untuk menjalankan aplikasi ini adalah:

- a. Sistem Operasi dapat berjalan pada sistem operasi Android versi 4.0. (*Ice Cream Sandwich*).
- b. *Space memory internal* minimal 1 MB.
- c. *Storage/ruang penyimpanan* minimal Mini atau *Micro SD*.
- d. Resolusi layar minimal 240 x 320 *pixels*.

3.3. PERANCANGAN SISTEM APLIKASI

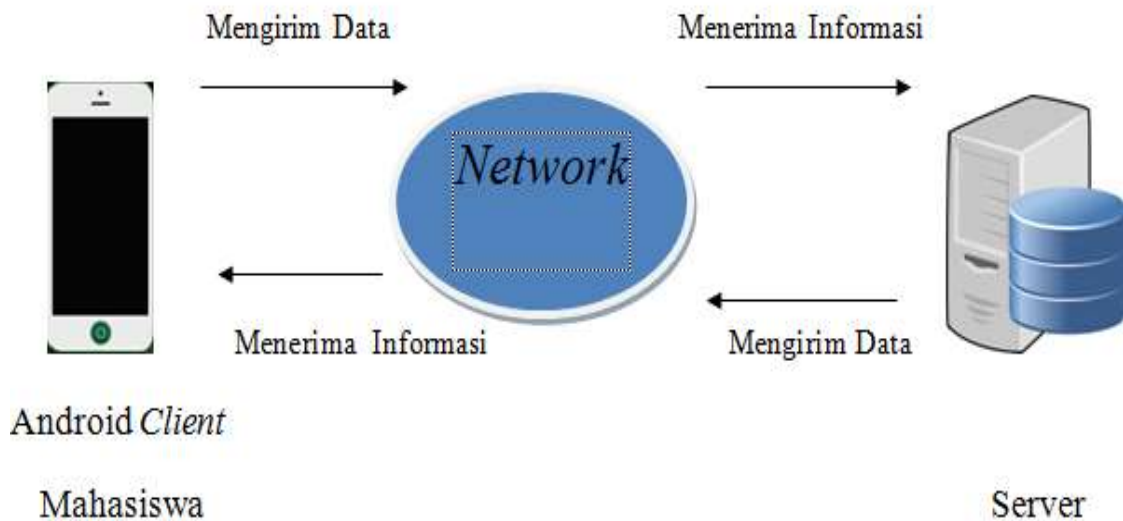
Perancangan dilakukan untuk menggambarkan, merencanakan, dan membuat sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Perancangan ini merupakan hasil transformasi dari analisa kedalam perancangan yang nantinya akan diimplementasikan.

Hal penting yang menjadi perhatian pada perancangan adalah bahwa rancangan yang dibuat diharapkan dapat digunakan dengan mudah oleh semua *user*. Yang dimaksud semua *user* adalah bahwa tidak hanya seorang ahli saja yang dapat menggunakan aplikasi ini, namun orang awam pun dapat menggunakannya. Selain

itu beberapa hal yang harus diperhatikan antara lain adalah kinerja program yang baik dalam mengoperasikan aplikasi yang dibuat.

3.3.1. Gambaran Umum Sistem

Perancangan pada Sistem Informasi Akademik berbasis Android ini nantinya tidak hanya merancang aplikasi yang akan dibangun tetapi mampu memahami bagaimana komunikasi data antara *server* berperan sebagai *admin* yang terhubung kedalam *database* dan berfungsi sebagai *server* dan perangkat keras *handphone* Android yang berfungsi sebagai *client*. Pada gambar 3.2 menunjukkan arsitektur sistem *client-server*.



Gambar 3.2 Model Arsitektur *Client-Server* Pada Sistem Aplikasi

Secara umum *server* menyediakan layanan yang dibutuhkan dan diminta oleh *client*. Hubungan yang terjadi antara *client* dan *server* adalah satu *server* melayani banyak *client*, dimana *client* selalu memulai interaksi dengan meminta layanan sedangkan *server* menunggu permintaan *client* dan mengirimkan kembali hasil permintaan *client*. Dalam perangkat keras dan perangkat lunak yang digunakan oleh masing-masing *client* tidak harus sama dengan yang digunakan oleh *server*, namun masih dapat saling terkoneksi antara satu dan yang lain.

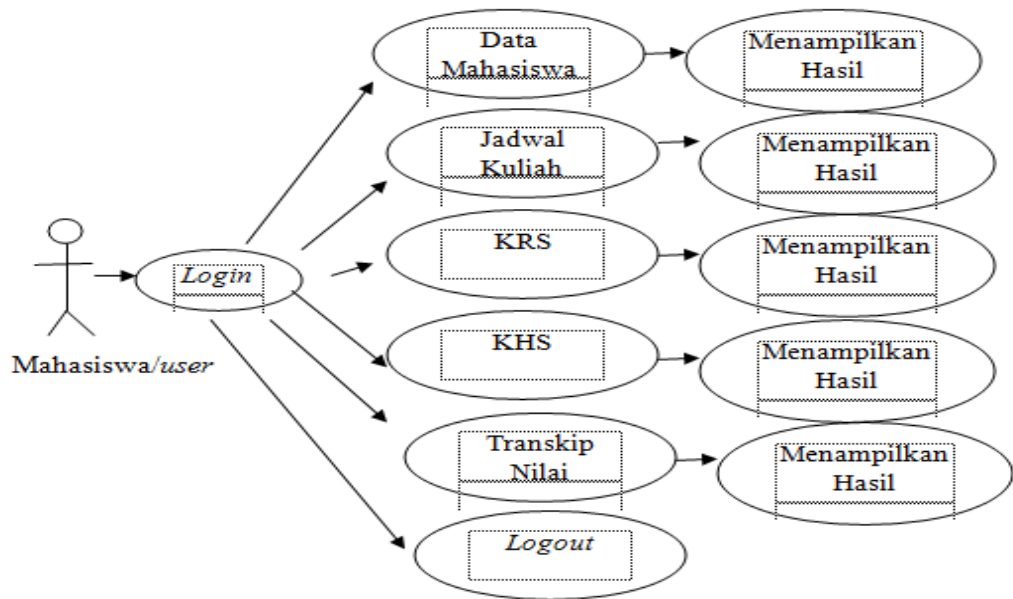
Pengguna sistem informasi yang berperan sebagai *client* ini nantinya adalah mahasiswa, sistem ini digunakan ketika mahasiswa sedang tidak berada dikampus untuk mengakses informasi akademik melalui terminal, sehingga mahasiswa tetap dapat memantau informasi akademik tanpa harus mengakses terminal dikampus. Pertama kali, *user* mengakses informasi akademik ini dengan menggunakan *handphone* berbasis android. Pada halaman *login user* diharuskan untuk mengetikkan *username* dan *password* sesuai dengan NIM mahasiswa secara *default* yang sudah terdaftar pada *database server*.

Setelah berhasil *login*, maka sistem akan menampilkan menu utama pada aplikasi android, dimana *user* dapat mengakses menu yang dibutuhkan dengan menekan pada salah satu menu yang tersedia. Langkah terakhir yang akan dijalankan adalah menampilkan informasi yang telah diproses oleh sistem kepada mahasiswa. Pada proses ini terjadi komunikasi data antara *user* berperan sebagai *client*, dimana *client* mengirimkan data ke *server* dan *server* akan mengirimkan kembali respon permintaan ke *client* sehingga *client* dapat mengakses permintaan tersebut.

3.3.2. Perancangan Diagram *Unified Modelling Language* (UML)

Pada pemrograman yang dibuat pada tugas akhir ini merupakan jenis pemrograman berorientasi objek (OOP), maka bahasa pemodelan yang digunakan adalah *Unified Modelling Language* (UML). UML merupakan standar bahasa pemodelan dalam pemrograman perangkat lunak (*software*) pada OOP yang digunakan untuk kebutuhan pemodelan virtual menggunakan diagram dan teks-teks sebagai pendukungnya. Penulis menggunakan 3 jenis diagram alir yaitu diagram *use case diagram*, *activity diagram*, dan *sequence diagram*. Berikut ilustrasi bentuk diagram alir yang penulis gunakan pada penyusunan Tugas Akhir ini yaitu:

1. Use Case Diagram



Gambar 3.3 Use Case Diagram

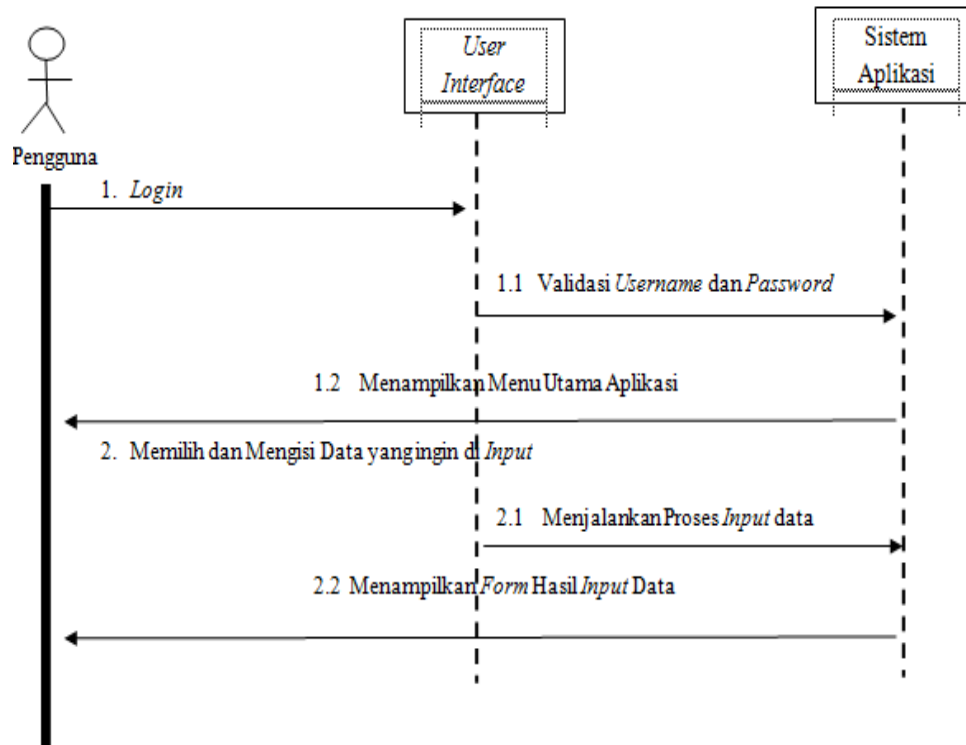
Diagram *use case* adalah sebuah pemodelan visual yang menggambarkan kelakuan dari sistem/aplikasi yang akan dirancang. Diagram ini menggambarkan sebuah interaksi/hubungan antar 1 atau lebih *client* dengan sistem yang akan dirancang. Diagram *use case* digunakan untuk mengetahui fungsi-fungsi yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Dari diagram *use case* pada gambar 3.2 dapat diketahui bahwa aplikasi sistem informasi akademik yang dirancang dapat dijabarkan melalui beberapa interaksi yang dilakukan mahasiswa/*user*. Interaksi awal yang dilakukan adalah mahasiswa/*user* masuk/*login* ke aplikasi dengan memasukkan *username* dan *password*. Setelah berhasil *login* sistem akan menampilkan menu utama dari sistem informasi akademik. Kemudian *user* memilih menu yang akan di *input* nantinya. Lalu setelah masuk menu yang dipilih maka sistem menampilkan submenu selanjutnya dan *user* mengisi data yang sesuai dengan data masing-masing *user* yang sudah terdaftar didalam *database server*. Setelah itu jika mahasiswa/*user* sudah mengisi

data dengan *valid* dan ingin keluar maka *user* kembali ke menu utama dengan memilih *logout* dan keluar.

2. Sequence Diagram

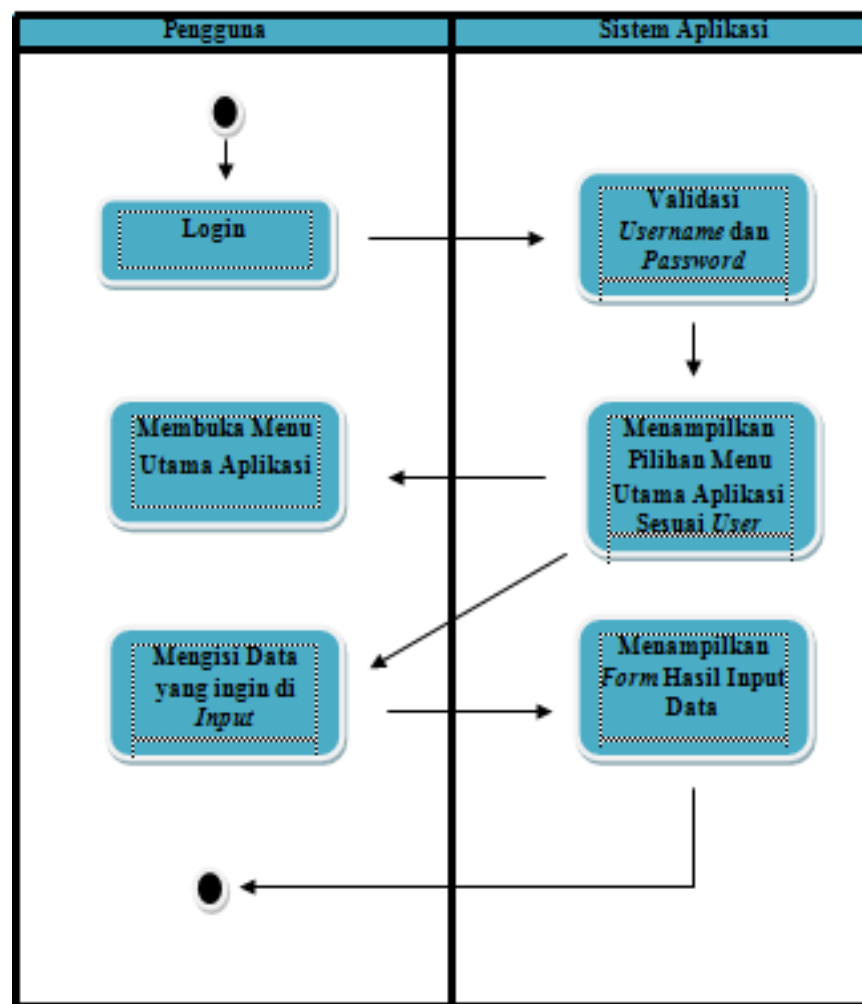
Diagram *sequence* merupakan diagram yang menggambarkan kelakuan objek yang dijabarkan pada diagram *use case* dengan mendeskripsikan disertai pesan yang dikirimkan dan diterima antar objek. Maka dari itu, untuk menggambarkan sebuah diagram *sequence* harus diketahui terlebih dahulu objek-objek yang terlibat dalam sebuah diagram *use case* beserta metode-metode yang digunakan. Pada *sequence* diagram ini *user* melakukan pemasukan data *login* yang langsung terintegrasi dengan *database* pada data *user* dan data *user* akan melakukan pengecekan *username* dan *password*. Namun meskipun mengalami alur proses yang sama antara pemasukan data *user*, mengenai data *user* hanya dapat dilakukan oleh *admin*. Setelah *user* memasukkan data yang sesuai dengan kebutuhan maka sistem akan memproses dan menyimpannya kedalam *database*.



Gambar 3.4 Sequence Diagram

3. Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan berbagai aliran aktivitas yang terjadi didalam sistem atau aplikasi. *Activity Diagram* ini merupakan rancangan rangkaian proses yang akan terjadi ketika *user* memilih untuk melakukan *login*. *User* lalu dapat memasukkan *username* dan *password*, kemudian sistem akan memvalidasi *username* dan *password* yang dimasukkan dan apabila *input*-an sesuai dengan data yang terdapat pada *database*, maka sistem akan menampilkan pilihan menu utama pada aplikasi. Selanjutnya *user* dapat membuka menu utama aplikasi dan mengisi data yang ingin di-*input* sesuai dengan kebutuhan *user* lalu sistem akan melakukan pengecekan dan menampilkan *form* hasil *input*-an sesuai dengan data yang sudah tersimpan didalam *database*.



Gambar 3.5 Activity Diagram

3.3.3. Perancangan *User Interface* (UI)

1. Rancangan *Form Login* Aplikasi

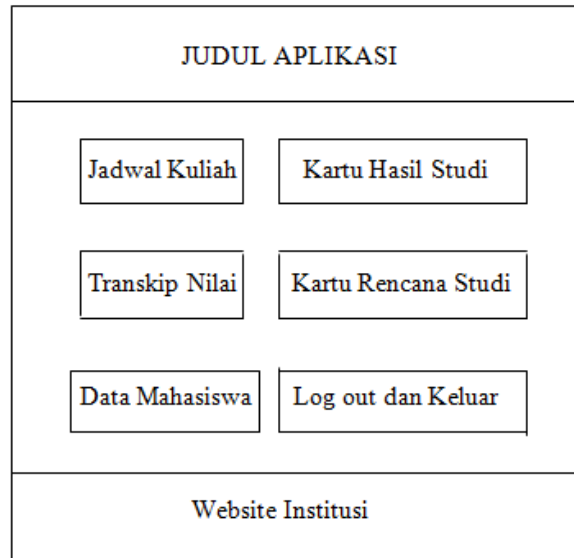
Pada rancangan *form login* aplikasi seperti pada gambar 3.6 menampilkan saat *user/pengguna* menekan *icon* aplikasi yang ada di *handphone*, setelah itu *pengguna* masuk ke tampilan *login* aplikasi dan sistem akan memprosesnya. Dalam rancangan *form* ini memuat nama aplikasi, logo institusi STT Telematika Telkom, dan *website* institusi. Untuk mengakses aplikasi, *user/pengguna* terlebih dahulu memasukkan/Meng-*input username* dan *password* sesuai dengan NIM mahasiswa secara *default*. Jika *user/pengguna* terjadi kesalahan dalam memasukkan *username* dan *password*, maka sistem akan memberikan peringatan dan kembali ke *login* aplikasi.

```
graph TD
    subgraph Form_Login [Form Login]
        subgraph Header
            Title[JUDUL APLIKASI]
        end
        subgraph Content
            Logo[LOGO INSTITUSI]
            Username[Username]
            Password[Password]
            Login[Masuk Sistem]
            Cancel[Batal]
        end
        subgraph Footer
            Website[Website Institusi]
        end
    end
```

Gambar 3.6 Rancangan *Form Login* Aplikasi

2. Rancangan *Form* menu utama

Dalam rancangan *form* menu utama aplikasi pada gambar 3.7 adalah *form* menu utama yang ditampilkan saat *user/pengguna* berhasil *login*. *Form* ini memuat pilihan menu yaitu data mahasiswa, jadwal kuliah, kartu rencana studi (KRS), kartu hasil studi (KHS), transkrip nilai dan logout.

Gambar 3.7 Rancangan *form* Menu Utama

3. Rancangan *Form* Jadwal Kuliah

Pada rancangan *form* Jadwal kuliah yang ditunjukkan pada gambar 3.8 adalah menu informasi jadwal perkuliahan berguna untuk membantu *user* melihat jadwal perkuliahan. Menu ini memuat kode mata kuliah, nama mata kuliah, dosen, hari, jam, dan ruang.

JUDUL APLIKASI					
Jadwal Perkuliahan					
Kode MK	Nama MK	Dosen	Hari	Jam	Ruang

Website Institusi

Gambar 3.8 Rancangan *Form* Jadwal Kuliah

4. Rancangan *Form* Kartu Hasil Studi (KHS)

Pada rancangan *form* kartu hasil studi (KHS) pada gambar 3.9 menampilkan jendela input kartu hasil studi (KHS) dimana menu ini memuat tahun akademik dan semester dari *user*.

JUDUL APLIKASI	
Kartu Hasil Studi (KHS)	
Tahun Akademik	<input type="text"/>
Semester	<input type="text"/>
<input type="button" value="Tampilkan KHS"/>	
Website Institusi	

Gambar 3.9 Rancangan *Form* Kartu Hasil Studi (KHS)5. Rancangan *Form* Kartu Rencana Studi (KRS)

Rancangan *form* kartu rencana studi (KRS) ini menampilkan jendela input kartu rencana studi (KRS) dimana menu ini memuat kode mata kuliah, mata kuliah, dosen dan SKS seperti pada gambar 3.10 berikut:

JUDUL APLIKASI			
KRS yang diambil			
Kode MK	Mata Kuliah	Dosen	SKS
Website Institusi			

Gambar 3.10 Rancangan *Form* Kartu Rencana Studi (KRS)6. Rancangan *Form* Transkrip Nilai

Rancangan *Form* Transkrip Nilai ini menampilkan jendela transkrip nilai perkuliahan dari *user* yang ditunjukkan pada gambar 3.11 berikut:

JUDUL APLIKASI			
Transkrip Nilai Kuliah			
Kode MK	Nama MK	SKS	Nilai
Website Institusi			

Gambar 3.11 Rancangan *Form* Transkrip Nilai

7. Rancangan *Form* Data Mahasiswa

Rancangan *Form* Data Mahasiswa pada gambar 3.12 menampilkan jendela data mahasiswa. Sistem secara otomatis akan menampilkan data mahasiswa sesuai *user* yang sudah terdaftar dalam sistem.

JUDUL APLIKASI
Data Mahasiswa
Website Institusi

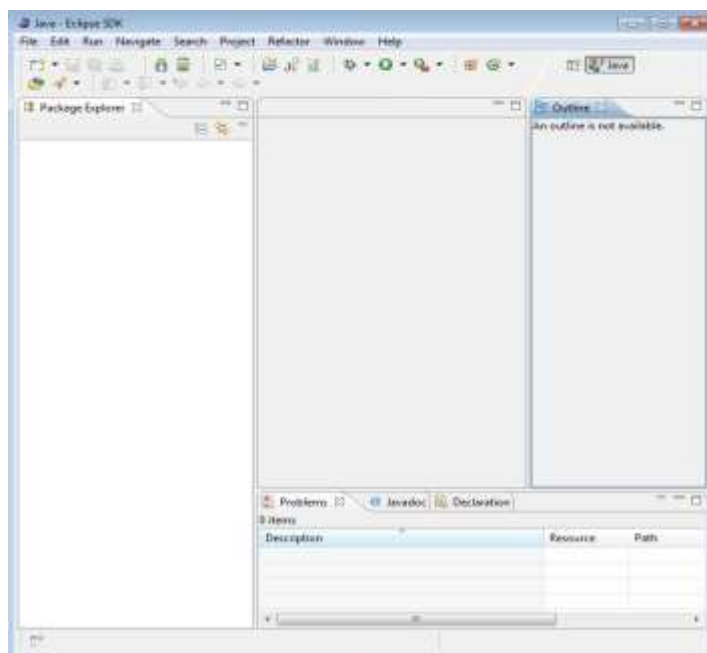
Gambar 3.12 Rancangan *Form* Data Mahasiswa

3.4. Penulisan Kode Program (*Coding*)



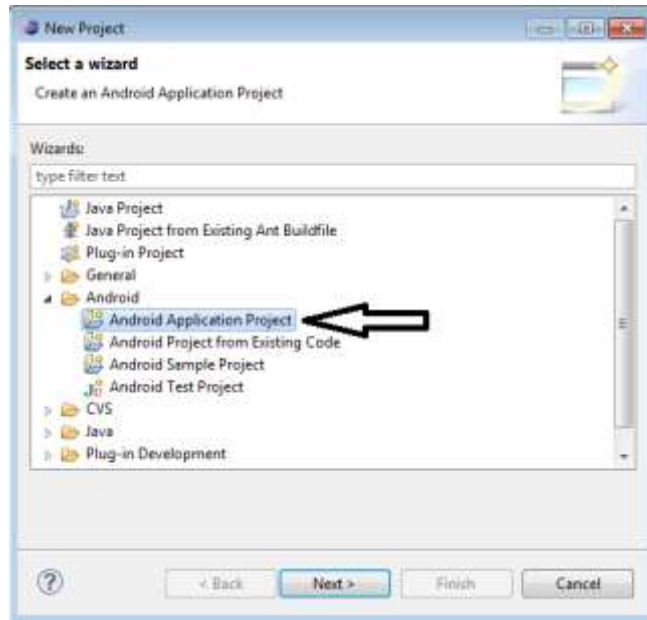
Gambar 3.13 Eclipse Indigo

Proses penulisan kode program (*coding*) ini dilakukan dengan tujuan agar perancangan aplikasi yang sebelumnya telah dilakukan dapat diterjemahkan kedalam bahasa pemrograman yang dipahami oleh komputer sehingga aplikasi tersebut dapat dibangun dan pengguna dapat memahami aplikasi yang telah dibangun tersebut. Penyusunan tugas akhir ini menggunakan bahasa pemrograman Java sebagai basis dalam membangun aplikasi pada sistem operasi Android. Selain itu aplikasi yang digunakan yaitu Eclipse Indigo yang berperan sebagai IDE (*Integrated Development Environment*) yang terlihat pada gambar 3.14 berikut:



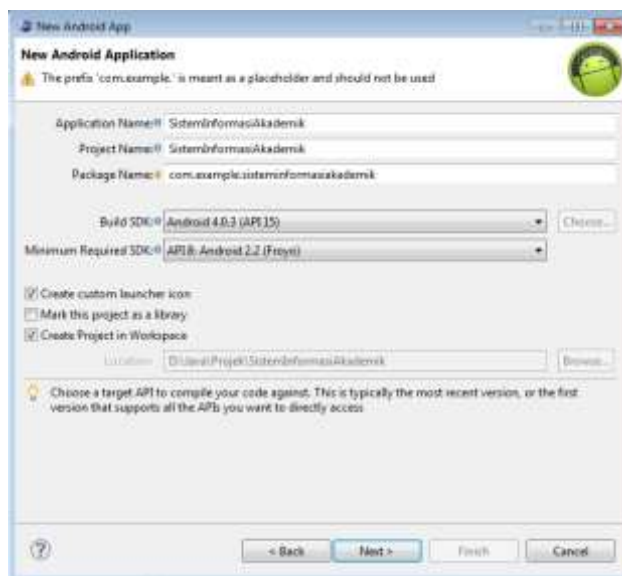
Gambar 3.14 Tampilan Awal Eclipse Indigo

Untuk membuat sebuah *project* Android baru, hal yang perlu dilakukan adalah membuat menu *file* dan memilih *new project*. Kemudian akan tampil jendela *new project* seperti berikut ini. Untuk membuat aplikasi Android maka digunakan tipe *Android Project* seperti pada gambar 3.15 berikut:



Gambar 3.15 Tampilan Pemilihan *Project*

Pada gambar menunjukkan proses pemberian nama *project* yang telah dipilih. Setelah memberikan nama *project*, hal yang selanjutnya dilakukan adalah menentukan direktori penyimpanan *project* tersebut seperti pada gambar 3.16 berikut:



Gambar 3.16 Tampilan Penamaan *Project*

Pada gambar merupakan *project* yang akan dibuat nantinya dan terbagi menjadi 4 kelas yang harus ada, diantaranya 2 kelas untuk “.java” dan 2 kelas untuk “.xml”. Berikut adalah penjelasan dari kelas-kelas tersebut yaitu:

- a. Hello.java adalah kelas yang menjalankan proses atau fungsi dari aplikasi (*Activity*).
- b. R.java berfungsi sebagai index yang menghubungkan semua *resource* pada aplikasi yang dikembangkan.
- c. Main.xml adalah kelas yang mengatur GUI dari aplikasi android yang digabungkan.
- d. String.xml berfungsi untuk menyimpan string objek agar nilai string tersebut nantinya dapat digunakan oleh komponen-komponen pada IDE Eclipse.

Berikut ini adalah *coding* yang digunakan untuk membangun aplikasi Sistem Informasi Akademik:

1. *Coding* pada String.xml

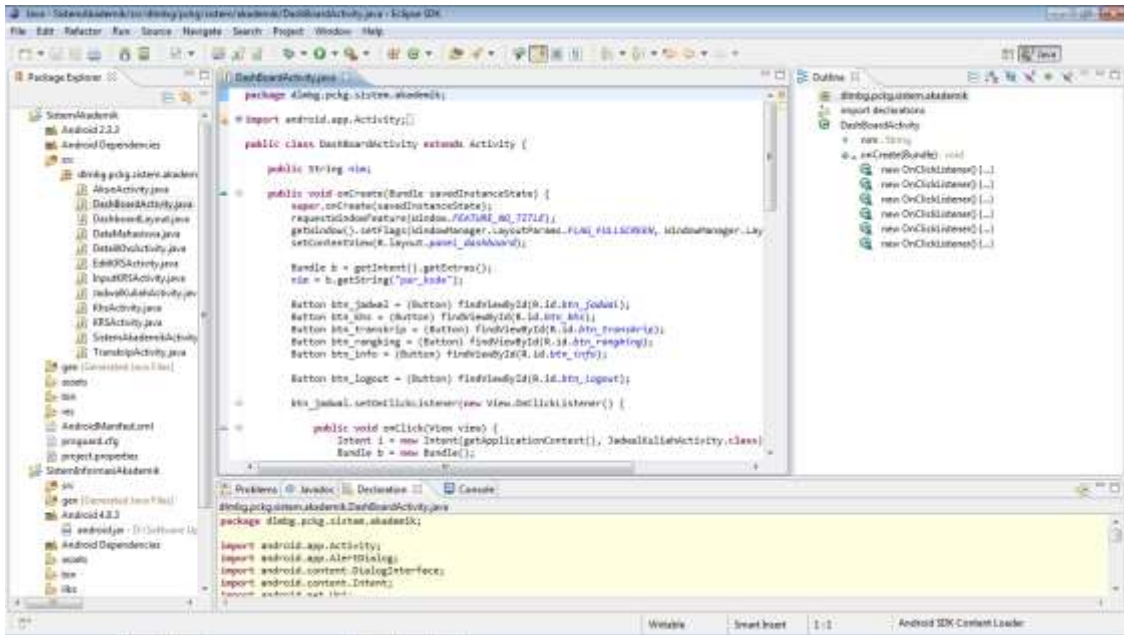
String.xml berfungsi menyimpan *string* objek agar nilai dari *string* objek tersebut nantinya dapat digunakan oleh komponen-komponen pada android Eclipse (contoh: *TextView*, *EditView*, *LinearLayout*, dll). Biasanya digunakan untuk memberikan label aplikasi, keterangan aplikasi dan mengubah warna latar belakang aplikasi. *String.xml* ini terletak pada *folder res > value*.

2. *Coding* pada main.xml

Kelas ini adalah kelas yang mengatur *Graphical User Interface* (GUI) dari aplikasi android yang dikembangkan. GUI dari android dapat dirancang dengan 2 cara yaitu dengan melakukan *coding* memakai bahas XML atau dengan cara *drag-and-drop*. Didalam main.xml ini terdapat berbagai macam *layout*, *widget* dan sebagainya yang diperlukan dalam pembangunan antarmuka aplikasi.

a. *Login*

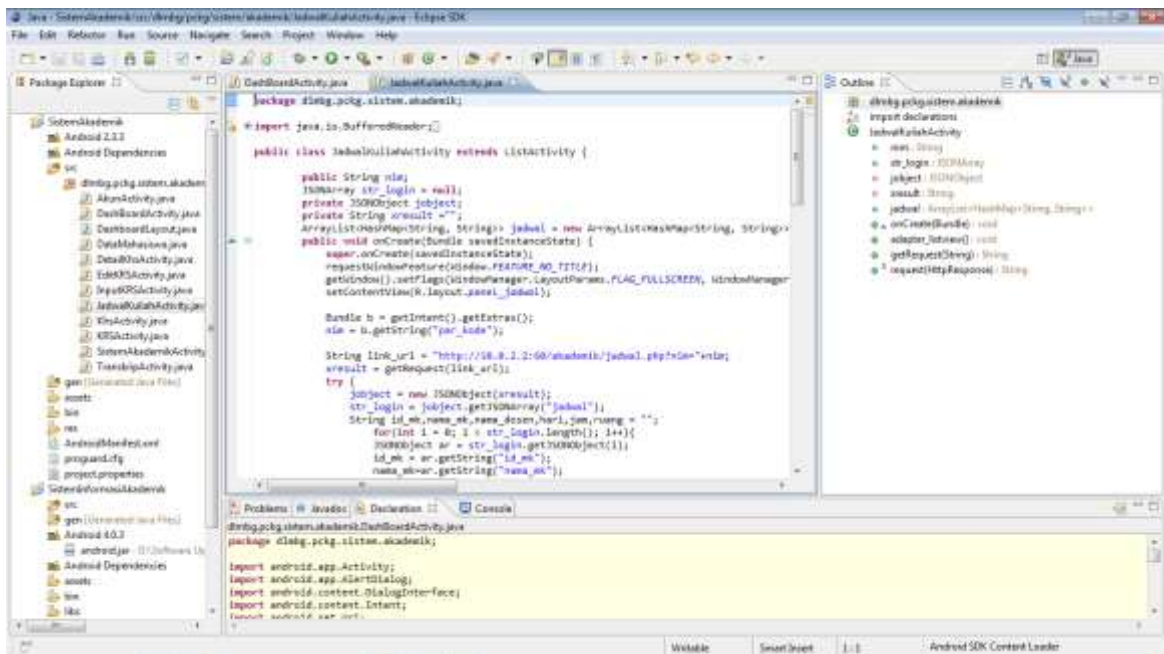
Pada tampilan *login* aplikasi ini merupakan tampilan utama dalam aplikasi sistem informasi akademik, yang dibuat pada *folder res > layout > login_layout.xml*. Tampilan *layout* ini memuat judul aplikasi, gambar logo institusi. Untuk membuat logo institusi pada *layout* ini gambar logo terlebih dahulu disimpan didalam folder *drawable*. Dalam tampilan *layout*



Gambar 3.18 Tampilan Source Code Menu Utama.java

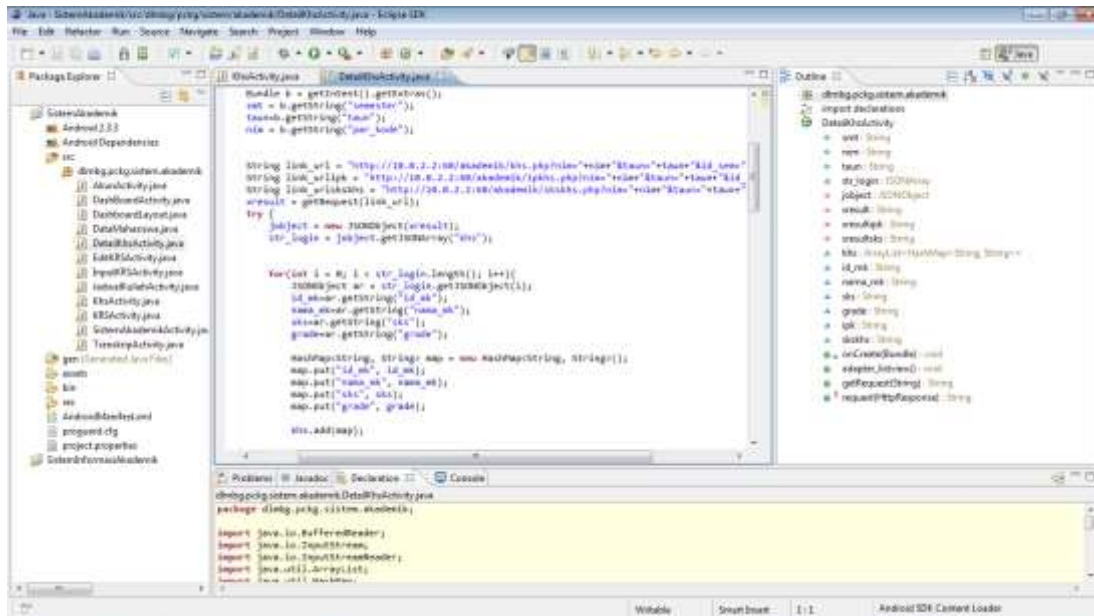
c. Jadwal Perkuliahan

Menu *layout* jadwal kuliah seperti pada gambar 3.19 dibuat pada *folder res > layout > jadwal_layout.xml*. Menu *layout* ini merupakan menu informasi penjadwalan kuliah yang di proses oleh sistem dan menampilkannya di menu *layout*.



Gambar 3.19 Tampilan Source Code Jadwal Kuliah.java

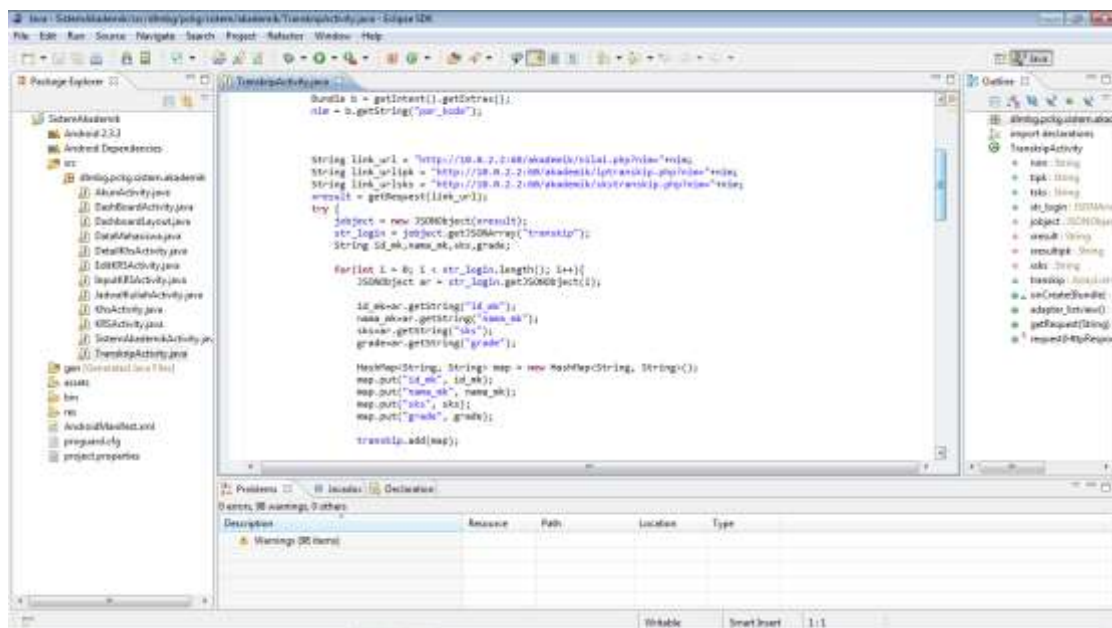
Setelah *user* selesai meng-*input* maka sistem akan menampilkan hasil menu *layout* Kartu Hasil Studi (KHS) yang ditunjukkan pada gambar 3.22 berikut:



Gambar 3.22 Tampilan *Source Code* Hasil Kartu Hasil Studi (KHS).java

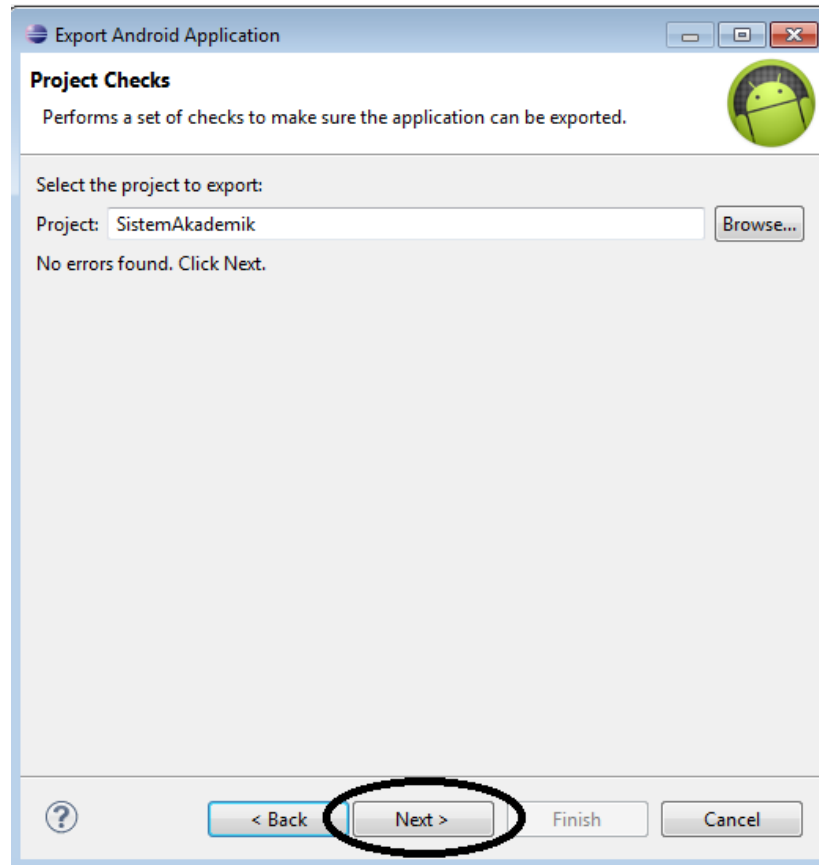
f. Transkrip Nilai

Pada menu transkrip nilai ini menampilkan hasil dari nilai *user* itu sendiri. Pada menu *layout* transkrip nilai dibuat pada *folder* res > layout > transkrip_layout.xml yang ditunjukkan pada gambar 3.23 berikut:



Gambar 3.23 Tampilan *Source Code* Transkrip Nilai.java

3. Memastikan bahwa tidak ada kesalahan pada aplikasi yang diekspor pada jendela *Project Check*. Jika tidak ada kesalahan pada aplikasi tersebut maka akan muncul peringatan *No Errors Found. Click Next*. Selanjutnya klik *browser* untuk mencari *file* yang akan dilakukan *Export*. Jika *file* sudah dipilih maka dilanjutkan dengan klik *Next* seperti pada gambar 3.27 berikut:



Gambar 3.27 Jendela *Project Check*

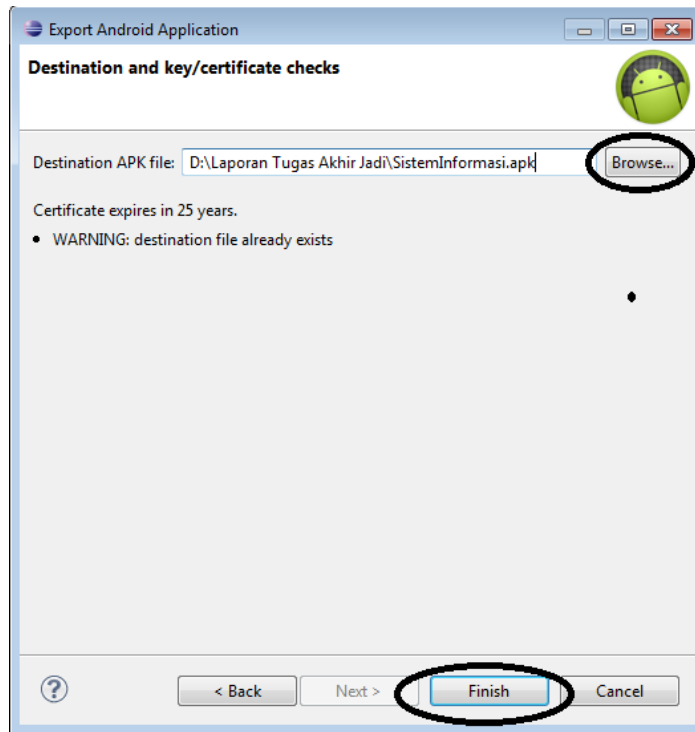
4. Setelah itu akan muncul jendela *Keystore Selection*. *Keystore* ini merupakan prosedur standar yang dianjurkan pada Google sebagai perizinan Android untuk mendistribusikan *file* berekstensi *.apk*. Selanjutnya pilih *Create New Keystore* pilihan dimana jika ingin membuat tempat penyimpanan baru untuk *Keystore*. Lalu klik *Browse* untuk menentukan tempat penyimpanan *keystore*. Beri nama *file* dengan cara *SistemAkademik.keystore*. kemudian berikan *password* dan *confirm* setelah itu klik *Next* seperti pada gambar 3.28 berikut:

Gambar 3.28 Jendela *keystore Selection*

5. Pada gambar 3.29 adalah langkah selanjutnya yang akan dilakukan yaitu pada jendela *Key Creation* dengan mengisi kotak kosong yang tampil pada layar. Adapun kotak kosong yang wajib diisi yaitu *alias*, *Password*, *Confirm*, *Validity (years)* dan *First and Last Name*. Setelah itu mengisi kotak kosong *Alias* dengan nama *file* aplikasi, *Password* dan *Confirm*, *Validity (years)* untuk *expired* jangka waktu, dan terakhir isi *First and Last Name*, jika sudah klik *Next*.

Gambar 3.29 Jendela *Key Creation*

- Setelah itu muncul jendela *Destination and key/certificate check*, lalu klik *browser* untuk menentukan letak *file .apk* kemudian beri nama *file apk*, klik *save* dan dilanjutkan dengan *finish* seperti pada gambar 3.30 berikut:



Gambar 3.30 Jendela *Destination and Key/Certificate Check*

- Jika *file .apk* sudah tersimpan pada direktori yang sudah ditentukan, maka hal terakhir yang dilakukan adalah mentransfer *file* tersebut kedalam *smartphone* dan melakukan penginstalan.