

BAB 2

LANDASAN TEORI

2.1 Monitoring

Monitoring adalah sebuah proses pengumpulan dan pengawasan sebuah informasi tentang sebuah kegiatan atau proses.[9]. Tujuan dari monitoring adalah memastikan bahwa seluruh proses berjalan dengan baik sesuai dengan rencana yang akan dilakukan. Seseorang yang melakukan monitoring memiliki tanggung jawab untuk melakukan pengawasan supaya prosesnya berjalan sesuai yang dikehendaki.

2.2 Anak Dibawah Umur

Anak dibawah umur adalah seorang yang belum masuk masa dewasa. Menurut Undang-undang No 1 tahun 1974 usia dewasa yaitu 18 tahun. Sehingga definisi anak dibawah umur di Indonesia bisa diartikan adalah seseorang yang belum mencapai 18 tahun. Jika dilihat dari pendidikan 18 tahun itu sudah memasuki masa Sekolah Menengah Atas, sehingga anak dibawah umur adalah anak-anak yang masih belum mencapai pendidikan Sekolah Menengah Atas.

2.3 Pola Asuh

Pola asuh terdiri dari dua suku kata yaitu pola dan asuh. Pola adalah model dan asuh diartikan menjaga, merawat, dan mendidik anak[10]. Pola asuh orang tua dibedakan menjadi tiga yaitu pola asuh otoriter, pola asuh permissif, dan pola asuh demokratis. Pola asuh otoriter yaitu pola asuh dimana orang tua yang mendidik anak dengan peraturan yang ketat, tidak adanya kesempatan mengemukakan pendapat, dan anak menerapkan peraturan yang dibuat secara mutlak.[11]. Pola asuh demokratis merupakan pola asuh yang merupakan pola asuh yang mementingkan kepentingan anak, tetapi tidak ragu untuk mengendalikan mereka, anak diberi kebebasan untuk melakukan suatu tindakan tetapi dengan aturan-aturan yang ditetapkan[12]. Pola permissif adalah pola asuh yang membebaskan anak untuk melakukan kegiatan tanpa diawasi maupun dikontrol sama sekali[13].

2.4 Android

Android adalah sebuah sistem operasi sumber terbuka yang berbasis linux yang bisa digunakan untuk berbagai macam perangkat seperti ponsel, table, televise, speaker, peralatan olahraga dan aksesoris. Android juga adalah platform terbuka (Open Source) bagi para pengembang yang akan menciptakan sebuah aplikasi yang menggunakan sistem operasi android.[14]

2.4.1 Android Package Manager

Package Manager adalah sebuah kelas yang terdapat pada Android SDK yang memiliki tanggung jawab untuk mengelola seluruh informasi tentang sebuah aplikasi yang terpasang di perangkat android[15]. Pada kelas ini menyediakan informasi mengenai aplikasi, komponen, versi dll. Implementasinya adalah sebagai berikut:

PackageInfo adalah sebuah class yang menyimpan informasi-informasi mengenai

```
PackageManager packageManager = getPackageManager();
try {
    PackageInfo packageInfo = packageManager.getPackageInfo("com.example.app", 0);
    String appName = packageInfo.applicationInfo.loadLabel(packageManager).toString();
    String versionName = packageInfo.versionName;
    int versionCode = packageInfo.versionCode;

} catch (PackageManager.NameNotFoundException e) {
    e.printStackTrace();
}
```

Gambar 2.1 Implementasi PackageManager

aplikasi yang terinstall di dalam smartphone, pada implementasi diatas PackageInfo digunakan untuk mendapatkan nama aplikas, nama versi, dan kode versinya.

2.4.2 Android Usage Stats Manager (Screen Time)

Usage Stats Manager adalah sebuah class yang disediakan oleh Android, untuk mendapatkan data mengenai penggunaan perangkat, data penggunaan digabungkan kedalam interval waktu hari, minggu dan tahun. Sebelum menggunakan usage Sberikut adalah atribut pada class Usage Stats Manager yang dapat mendapatkan informasi tersebut

```
//Waktu Harian
public static final int INTERVAL_DAILY

//Waktu Mingguan
public static final int INTERVAL_WEEKLY

//Waktu Bulanan
public static final int INTERVAL_MONTHLY

//Waktu Tahunan
public static final int INTERVAL_YEARLY
```

Gambar 2.2 Atribut UsageStatsManager

2.4.3 Android Web View

Class WebView adalah class yang memungkinkan aplikasi untuk menampilkan konten dari sebuah situs website. Web view diperkenalkan sejak android 4.4, Untuk mengakses sebuah WebView pada smartphone tersebut harus mendukung System Web View. Implementasi untuk menggunakan WebView adalah sebagai berikut:

```
<WebView
    android:id="@+id/webview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
/>
```

Gambar 2.3 Implementasi Pada XML (Layout)

Untuk menggunakan class webview kita harus implementasikan terlebih dahulu di dalam Layout XML, dengan cara seperti gambar 2.3

```
WebView myWebView = (WebView) findViewById(R.id.webview);
myWebView.loadUrl("http://www.example.com");
```

Gambar 2.4 Implementasi Pada Java

Setelah itu konfigurasi di dalam class Java, untuk memanggil Layout XML tersebut, dan dengan method `loadUrl()` layout tersebut akan menampilkan situs website yang sesuai.

2.5 Java

Java adalah sebuah bahasa pemrograman yang dapat dijalankan di berbagai macam perangkat. Bahasa ini ditemukan oleh tim dari Sun Microsystem yang di pimpin oleh seseorang yang bernama James Gosling. Bahasa Java banyak mengadopsi sintaksis yang terdapat pada bahasa C dan C++. Aplikasi yang dibuat menggunakan bahasa pemrograman java akan dikompilasi menjadi bytecode dan dapat dijalankan di berbagai perangkat yang memiliki Java Virtual Machine. Pada saat ini Java termasuk bahasa yang populer digunakan oleh seorang pengembang aplikasi, karena Java mendukung pembuatan aplikasi web, mobil, dan desktop. [16].

2.6 Android Studio

Android studio adalah sebuah Integrated Development Environment (IDE) untuk pengembangan aplikasi android. IDE ini muncul tahun 2018 bersamaan ketika google memperkenalkan Android Jetpack kepada komunitas pengembang. Android jetpack dirancang untuk membuat aplikasi android dengan cepat, mudah, dan andal. [17]

2.7 Firebase Cloud Messaging (FCM)

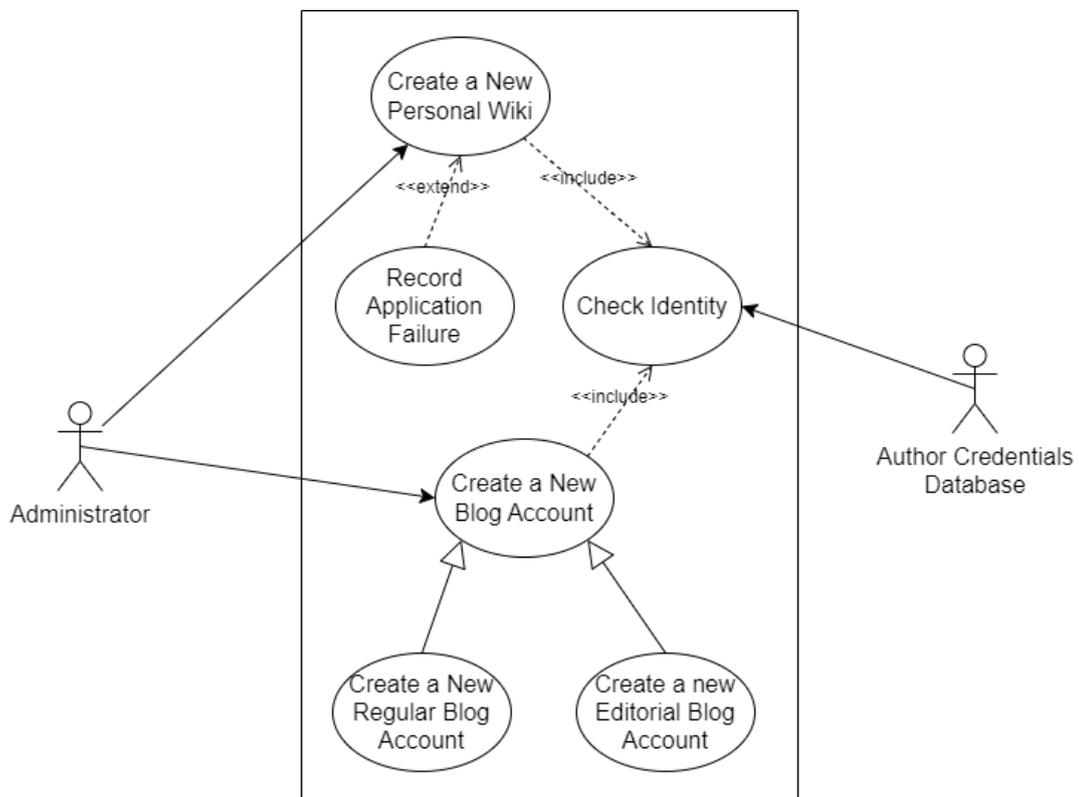
Firebase Cloud Messaging (FCM) adalah sebuah layanan untuk berkitim suatu pesan yang disediakan oleh Google secara open source. FCM menyediakan fitur untuk melakukan push notification, yaitu fungsi untuk memberikan notifikasi yang muncul pada layar smartphone. [18] FCM dapat digunakan untuk pesan baru, pembaruan data, atau pemberitahuan penting yang berkaitan dengan aplikasi. Fitur push notification akan mengirimkan notifikasi secara real time. Untuk menggunakannya pengguna hanya mengkonfigurasi dan mendapatkan acces key berupa token.

2.8 MySQL

MySQL adalah sistem manajemen database yang ditulis menggunakan bahasa SQL. SQL (Structured Query Language) adalah sebuah bahasa script yang digunakan untuk pengolahan database. MySQL juga bersifat open source sehingga seorang yang ingin menggunakan RDBMS ini bisa menggunakannya secara gratis. Pada saat ini MySQL ini menjadi RDBMS paling banyak digunakan.[19].

2.9 Use Case Diagram

Use case diagram adalah sebuah diagram yang menggambarkan fungsi-fungsi yang disediakan oleh sebuah sistem yang nantinya akan digunakan oleh pengguna[20]. Berikut adalah implementasi use case diagram pada beberapa contoh kasus:



Gambar 2.5 Contoh Kasus Use Case Diagram

Pada kasus ini terdapat 2 aktor yaitu administrator dan Author Credentials Database, aktor administrator dapat berinteraksi dengan sistem melalui proses “Create a New Personal Wiki”, dan Create a New Blog Account. Pada proses Create a New Personal Wiki terdapat hubungan “include” ke proses Check Identity, proses

Check Identity bisa dilakukan oleh Author Credentials Database, sehingga proses Create a New Personal Wiki harus menyertakan proses Check Identity dalam prosesnya. Selain itu proses tersebut memiliki extend terhadap Record Application Failure dimana ketika proses Create a New Personal Wiki dilakukan bisa saja menyertakan proses Record Application Failure, dimana proses ini digunakan untuk mencatat jika terjadi kesalahan, sehingga proses tersebut menggunakan “extend” sehingga bisa saja tidak dilakukan. Selanjutnya terdapat proses Create a New Blog Account yang memiliki hubungan include terhadap Check Identity, dimana proses ini diharuskan menyertakan proses Check Identity didalam prosesnya. Selain itu pada proses tersebut memiliki spesialisasi yaitu generalisasi dimana pada proses Create a New Blog Account terdapat 2 pilihan yaitu membuat blog regular atau membuat blog editorial, pada 2 proses tersebut harus terdapat seluruh proses dari Create a New Blog Account, tetapi ada beberapa proses yang memiliki perbedaan diantaranya adalah apakah seorang administrator akan membuat blog Regular atau blog Editorial.

2.9.1 Deskripsi Use Case

Deskripsi use case adalah sebuah informasi yang menjelaskan secara lebih rinci tentang bagaimana sebuah use case digunakan dan interaksi antara aktor pada sistem. Sistem ini melengkapi dekripsi yang sudah dijelaskan pada use case diagram[20]. Berikut adalah tabel penjelasan dari Deskripsi use case:

Tabel 2.1 Deskripsi Use Case

Deskripsi Use Case	Keterangan
Related Requirement	Indikasi mengenai persyaratan yang terpenuhi oleh use case
Goal Context	Kontribusi use case dalam mencapai tujuan sistem.
Precondition	Syarat yang harus terpenuhi sebelum use case ini terjadi

Successful End Condition	Kondisi sistem setelah use case dilaksanakan dengan sukses
Failed End Condition	Kondisi sistem jika use case gagal dilaksanakan
Primary Actor	Aktor-aktor yang memicu dan langsung menerima informasi dari use case tersebut
Trigger	Peristiwa yang dipicu oleh aktor dan menyebabkan use case terjadi
Main Flow	Mendekripsikan setiap langkah-langkah eksekusi normal pada suatu use case
Extension	Mendekripsikan langkah-langkah alternatif selain yang dijelaskan pada Main Flow

Contoh implementasi dari deskripsi usecase adalah sebagai berikut:

Tabel 2.2 Implementasi Deskripsi Use Case

Deskripsi Use Case	Keterangan	
Use Case Name	Create a New Personal Wiki	
Related Requirement	Requirement A.2	
Goal Context	A new or Existing author request a new blog account from the administrator	
Precondition	The author has appropriate proof of identity	
Successful End Condition	A new personal Wiki is created for the author	
Failed End Condition	The application for a new personal Wiki is rejected	
Primary Actor	Administrator	
Trigger	The Administrator asks the CMS to create a new personal Wiki.	
Included Cases	Check Identity	
Main Flow	Step	Action
	1	The Administrator asks the system to

		create a new personal Wiki.
	2	The Administrator enters the author's details
	3 Include: Check Identity	The author's details are checked
	4	The new personal Wiki is created
	5	A summary of the new personal Wiki's details are emailed to the Authhor

2.10 Activity Diagram

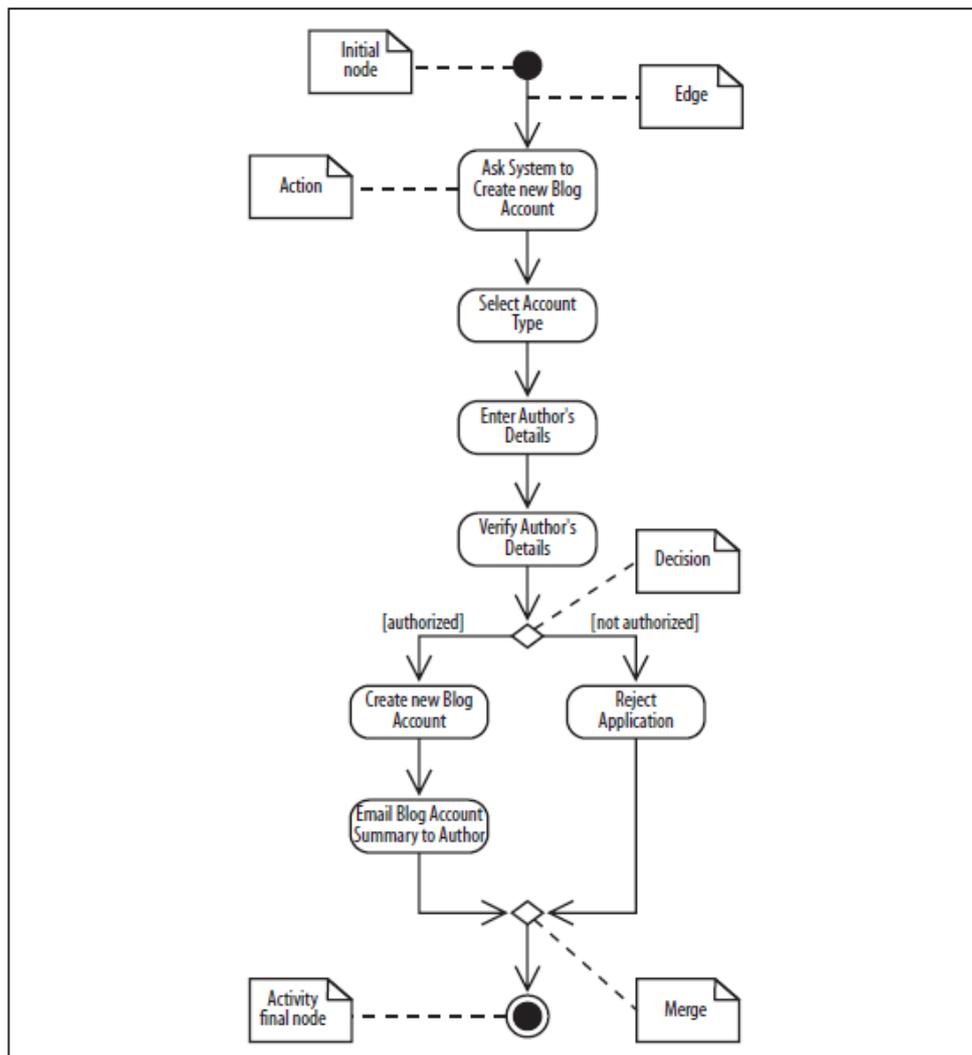
Activity diagram adalah sebuah diagram yang memvisualisasikan langkah-langkah dalam sebuah proses, diagram ini memungkinkan menentukan bagaimana sistem mencapai tujuannya[20]. Proses tersebut mengacu terhadap tabel deskripsi use case yang sudah didefinisikan sebelumnya. Contoh implementasi deskripsi use case:

Tabel 2.3 Implementasi Deskripsi Use Case

Deskripsi Use Case	Keterangan	
Use Case Name	Create a New Blog Account	
Related Requirement	Requirement A.1	
Goal Context	A new or existing author requests a new blog account from the Administrator.	
Precondition	The system is limited to recognized authors, and so the author needs to have appropriate proof of identity.	
Successful End Condition	A new blog account is created for the author.	
Failed End Condition	The application for a new blog account is rejected.	
Primary Actor	Administrator	
Secondary Actor	Author Credentials Database.	
Trigger	The Administrator asks the Content Management System to create a new blog account.	
Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects an account type.
	3	The Administrator enters the author's details.
	4	The author's details are verified using the Author Credentials Database.

	5	The new blog account is created.
	6	A summary of the new blog account's details are emailed to the author.
Extensions	Step	Branching Action
	1	The Author Credentials Database does not verify the author's details
	2	The author's new blog account application is rejected.

Deskripsi use case tersebut digunakan untuk dijadikan acuan dalam pembuatan activity diagram. Berikut Implementasi Activity diagramnya:

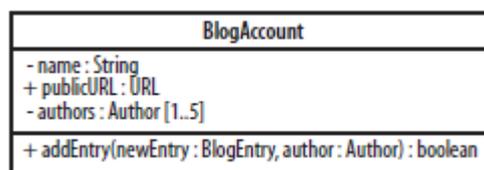


Gambar 2.6 Implementasi Activity Diagram Create a New Blog Account

Pertama-tama yang dilakukan adalah pendefinisian initial node, selanjutnya aktivitas yang dilakukan adalah “Ask System to Create a New Blog Account” setelah proses selesai maka akan dilakukan aktivitas selanjutnya, tetapi dalam aktivitas “Verify Authors Detail” terdapat kondisi branch yang dilambangkan dengan belah ketupat dimana jika proses tersebut “Authorized” maka akan masuk ke aktivitas “Create new Blog Account”, sedangkan jika “Not Authorized” akan masuk ke aktivitas “Reject Application”. Lalu pada diagram aktivitas tersebut terdapat simbol merge dimana menandakan bahwa terdapat akhir dari perilaku bersyarat. Pada implementasi tersebut terdapat 2 akhir dari perilaku bersyarat yang pertama Aktivitas “Email Blog Account Summary to Author” dan “Reject Application” kedua aktivitas tersebut digabungkan dan menghasilkan final node (akhir dari aktivitas).

2.11 Class Diagram

Class diagram digunakan untuk menggambarkan dengan jelas struktur dan dekripsi sebuah class seperti atribut, method dan hubungan dari setiap objeknya. Class diagram biasanya didefinisikan dalam sebuah persegi panjang yang dibagi menjadi 3 bagian yaitu nama class, atribut, dan method. Selain itu pada class diagram terdapat hubungan antar class diantaranya dependency, asosisasi, agregasi, komposisi, dan inheritance[20].



Gambar 2.7 Class Diagram Blog Account

Class diagram tersebut bernama “Blog Account” yang memiliki atribut name yang bertipe data string, publicURL yang bertipe data URL, dan Authors yang bertipe data Author. Selain itu ada method di dalam class BlogAccount dimana method tersebut bernama addEntry yang menerima parameter newEntry yang bertipe data BlogEntry dan author yang bertipe data Author dari method tersebut mereturnkan Boolean. Selain terdapat method, atribut dan tipe data, pada class

diagram juga memiliki hubungan antar class yaitu dependency, asosiasi, agregasi, komposisi, inheritance.

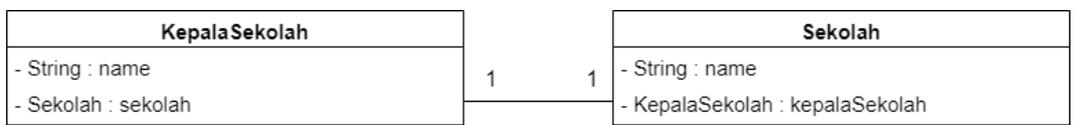
1. Dependency



Gambar 2.8 Class Diagram (Dependency)

Class diagram diatas memiliki hubungan dependency dimana ClassB memiliki method dengan parameter yang menerima Object A. Sehingga ClassB memiliki ketergantungan kepada ClassA.

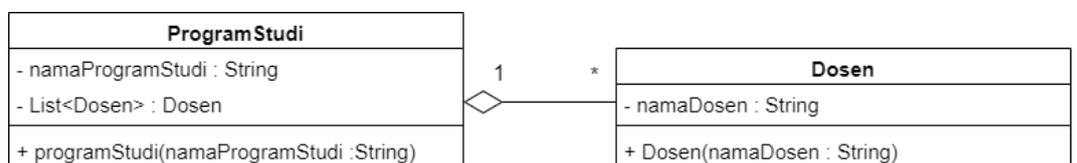
2. Asosiasi



Gambar 2.9 Class Diagram (Asosiasi)

Class KepalaSekolah, dan Kelas Sekolah jika memiliki hubungan asosiasi, akan melakukan pertukaran data dari KepalaSekolah kepada Sekolah, dan Sekolah kepada KepalaSekolah, sehingga untuk melakukan pertukaran tersebut harus diinstance keduanya

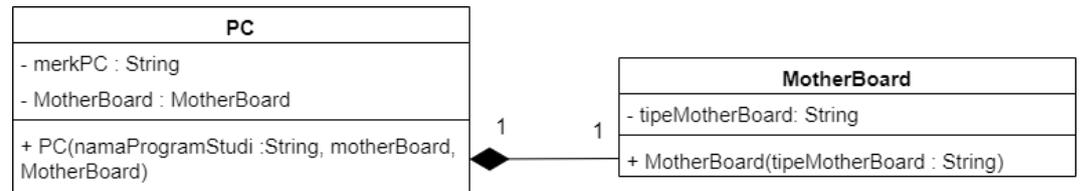
3. Agregasi



Gambar 2.10 Class Diagram (Agregasi)

Class ProgramStudi dan class Dosen bisa menggunakan relasi Agregasi dimana setiap menginstance Objek dari class Program Studi bisa menambahkan List<Dosen> tanpa harus melakukan pertukaran, atau tanpa harus menginisiasi object Dosen.

4. Komposisi

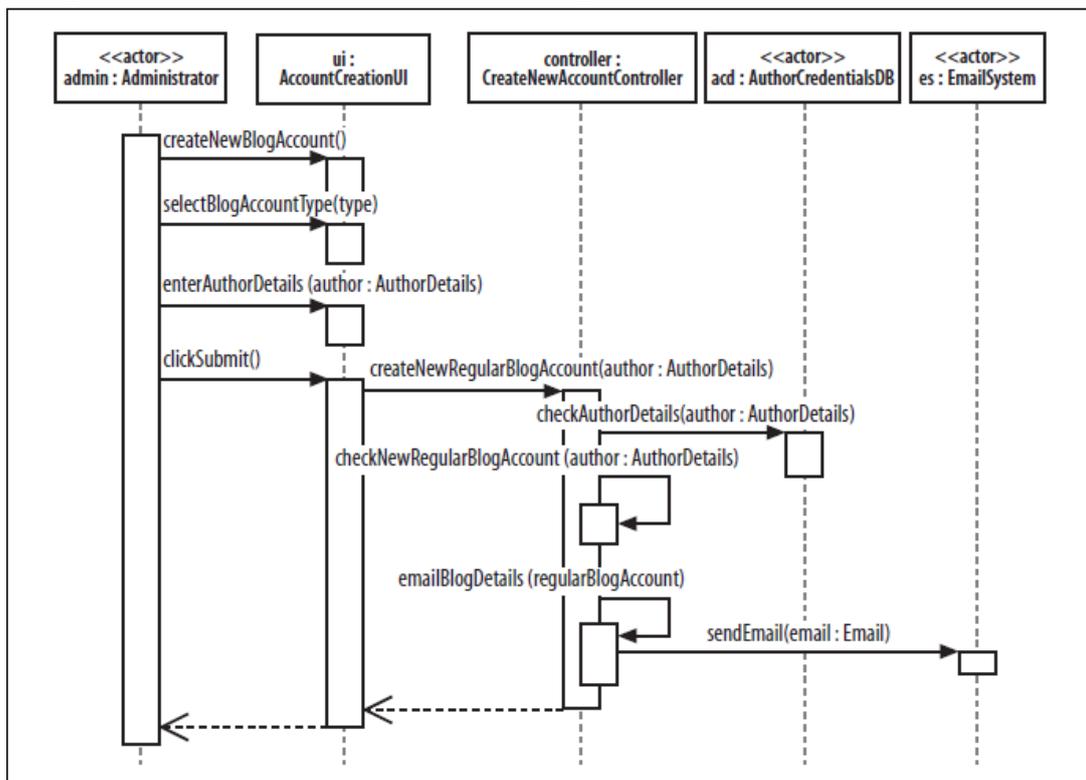


Gambar 2.11 Class Diagram (Komposisi)

Class PC dengan MotherBoard bisa menggunakan relasi komposisi dimana objek PC harus menginsansiasi terlebih dahulu kelas MotherBoard, sehingga jika class PC tidak ada, maka class motherboard ikut tidak ada.

2.12 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan urutan interaksi antara aktor dan class-class yang terdapat pada sistem, seta menjelaskan interaksi apa yang akan dipicu ketika sebuah penggunaan tertentu dieksekusi[20]. Contoh implementasi simbol-simbol tersebut sebagai berikut:



Gambar 2.12 Implementasi Sequence Diagram

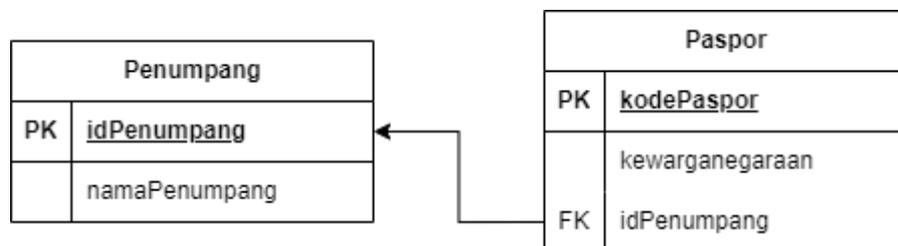
Pada gambar 2.8 terdapat sequence diagram untuk proses pembuatan akun blog, pertama-tama yang dilakukan pada proses ini yaitu Administrator yang berperan sebagai aktor, meminta sistem untuk membuat baru akun blog dengan method `createNewBlogAccount()`, Lalu Administrator memilih akun regular dengan method `selectBlogAccount(type)`, Lalu Administrator memasukkan detail penulis dengan method `enterAuthorsDetails (AuthorDetails author)`, dan sistem akan memverifikasi Author Credentials, Setelah dicek dan berhasil, maka proses pembuatan akun blog berhasil. Dan akan diberitahukan bahwa pembuatan blog sudah berhasil melalui email melalui method `sendEmail(Email email)`.

2.13 Skema Relasi

Skema relasi adalah sebuah skema yang mengorganisir data dalam bentuk tabel, dimana tabel-tabel tersebut adalah entitas-entitas yang dapat saling terhubung melalui atribut-atributnya.[21]

1. Relasi One to One

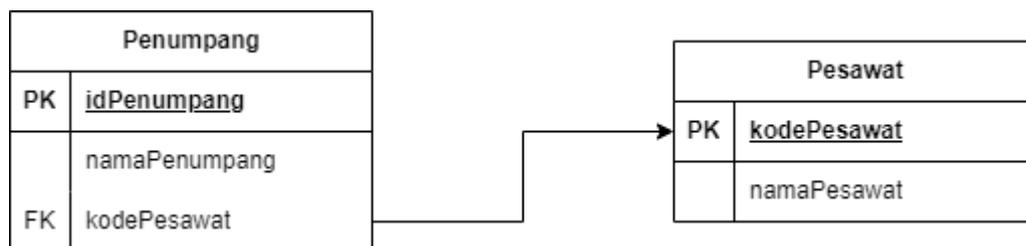
Ketika salah satu entitas dari hubungan memiliki partisipasi penuh, dan sisi lainnya memiliki partisipasi parsial maka simpan kunci asing / utama dari entitas (partisipasi penuh) ke dalam entitas (partisipasi parsial). Berikut contoh relasi One to One



Gambar 2.13 Relasi One to One

2. Relasi Many to One

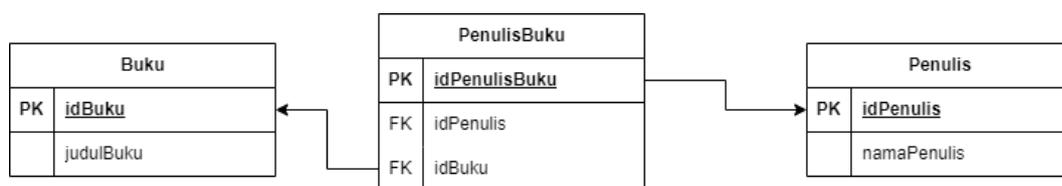
Ketika entitas N (Many) memiliki partisipasi penuh maka sertakan kunci asing yang didapat dari hubungan dengan sisi entitas 1 (One).



Gambar 2.14 Relasi Many to One

3. Relasi Many to Many.

Ketika kedua entitas adalah N (Many) maka buat tabel baru yang terdapat dua kunci asing dari kedua entitas tersebut.



Gambar 2.15 Relasi Many to Many

2.14 Skala Likert

Skala likert adalah skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Skala likert masuk ke dalam skala ordinal,[22] karena kategori respon pada skala likert mempunyai tingkatan tetapi jarak diantara kategori tidak dapat dianggap sama sehingga setiap tingkatan tidak dapat dimasukkan ke dalam bentuk persamaan matematika. Pada skala likert terdapat 5 alternatif jawaban yaitu sangat setuju, setuju, netral, tidak setuju, dan sangat tidak setuju.[23] Data ordinal yang didapatkan dari skala likert dianalisis menggunakan modus.[24]