BAB 2

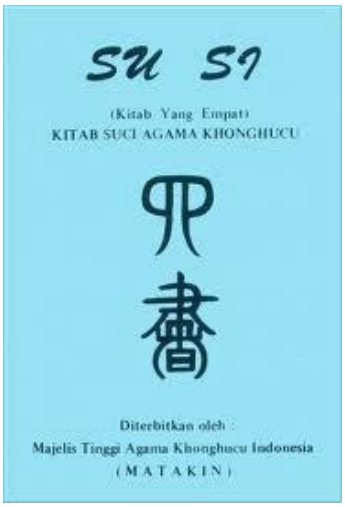
LANDASAN TEORI

2.1 Khonghucu

Agama Khonghucu atau Khong Kau (*Khong Jiao*), istilah aslinya disebut Ji Kau (*Ru Jiao*) yang artinya agama dari orang-orang yang lembut hati, yang terbimbing atau terpelajar [1]. Ji Kau lebih dikenal dengan sebutan Khonghucu karena merujuk istilah yang digunakan sarjana barat yang menerjemahkan Ji Kau dengan Konfusianisme (*Confucianism*). Agama Khonghucu bukan suatu ajaran yang diciptakan Nabi Khongcu melainkan agama yang telah diturunkan *Tian*, Tuhan Yang Maha Esa melalui para Nabi dan Rajasuci purba ribuan tahun sebelum lahir Nabi Khongcu, lalu kemudian disempurnakan oleh Nabi Khongcu [2].

2.1.1 Kitab Sishu

Kitab Sishu (Kitab Yang Empat) adalah salah satu kitab suci agama Khonghucu. Kitab Sishu terdiri dari empat bagian yakni: Ajaran Besar (*Da Xue*), Tengah Sempurna (*Zhong Yong*), Sabda Suci (*Lun Yu*) dan Bingcu (*Meng Zi*) [1]. Gambar 2.1 menunjukkan Kitab Sishu dalam terjemahan bahasa Indonesia:



Gambar 2.1 Kitab sishu terjemahan bahasa Indonesia

Kitab Ajaran Besar berisi tuntunan pembinaan diri, Kitab Tengah Sempurna berisi Ajaran Keimanan, Kitab Sabda Suci berisi kumpulan ajaran-ajaran Nabi dan percakapan Nabi dengan murid-muridnya, Kitab Bingcu berisi ajaran Bingcu yang menjelaskan dan menerangkan ajaran Nabi Khonghucu [2].

2.1.2 Tempat Ibadah

Umat Khonghucu biasa melaksanakan kegiatan peribadahan keluarga di Rumah Abu Leluhur atau Altar meja sembayang serta melaksanakan peribadahan bersama di Khongcu Bio (*Kong Miao*), Lithang, dan Kelenteng [1]. Gambar 2.2 menunjukkan contoh tempat ibadah umat Khonghucu:

Gambar 2.2 Contoh tempat ibadah umat Khonghucu

2.1.3 Nyanyian Pujian

Kegiatan peribadahan dan upacara persembahyangan umat Khonghucu biasa diiringi dengan nyanyian pujian (lagu rohani) dalam pelaksanaannya. Nyanyian pujian ini biasa diciptakan oleh umat ataupun rohaniawan.

2.2 Bot

Bot merupakan program komputer yang dirancang untuk memberikan respon sesuai dengan permintaan yang diberikan [7]. Bot dapat memberikan respon berdasarkan permintaan berupa teks, suara, gambar, dan permintaan lainnya berdasarkan pengetahuan yang sudah dimiliki sebelumnya. Sedangkan *Chatbot* adalah program komputer yang diprogram untuk dapat berkomunikasi dengan manusia menggunakan bahasa manusia dan dapat menstimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara audio maupun teks [13].

2.2.1 Messaging Bot

Messaging bot atau bot pada pesan instan adalah bot yang sederhana memiliki antarmuka textual yang memungkinkan pengguna untuk dapat mengakses informasi, menggunakan layanan atau menyediakan hiburan melalui aplikasi pesan instan secara online. Layanan pesan instan dan media sosial menunjukkan tingkat penggunaan dan pertumbuhan pengguna yang luar biasa dibandingkan aplikasi yang sering digunakan lainnya [6].

Mulai dari tahun 2014 banyak layanan pesan instan yang memperkenalkan dukungan terhadap pengembangan bot. Hal ini didukung dengan fakta bahwa pengguna pesan instan menggunakan aplikasi pesan instan beberapa kali sehari dan mereka lebih mudah memahami alur penggunaan antarmuka aplikasi pesan instan tersebut. Dibandingkan dengan mencoba membuat aplikasi baru, bot menawarkan layanan yang memudahkan pengembang untuk menarik perhatian penggunanya melalui layanan pesan instan [6].

2.2.2 Tujuan Membuat Bot

Bot banyak dibuat untuk berbagai macam tujuan diantaranya yakni: Melakukan pencarian informasi wisata, promosi produk, automasi informasi layanan kampus, informasi jadwal film, monitoring keamanan web dan jaringan [7] [8] [9].

2.2.3 Metode Membuat Bot

Banyak pendekatan teknik dan metode yang dapat digunakan dalam membuat bot diantaranya yakni: menggunakan kesamaan teks (*text similarity*), menggunakan aturan yang sudah ditentukan (*rule based*) atau melalui penelusuran menu (*menu based*). Dengan menggunakan pendekatan kesamaan teks, teks masukan dibandingkan dengan informasi yang sudah tersedia sebelumnya pada database dan diukur kemiripannya menggunakan metode tertentu sehingga diperoleh hasil teks keluaran yang relevan dengan teks masukan yang diberikan. Dengan menggunakan aturan yang sudah ditetapkan, bot memberikan keluaran sesuai dengan aturan yang sebelumnya sudah ditetapkan oleh pengembang bot. Sedangkan dengan menggunakan penelusuran menu, pengguna bot menelusuri menu untuk mengakses informasi yang sudah tersedia sebelumnya pada bot.

2.3 Basis Data

Basis data (*database*) adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat. Selain berisi data, database juga berisi metadata. *Metadata* adalah data yang menjelaskan tentang struktur dari data itu sendiri [14].

2.3.1 Table

Table adalah suatu entitas yang tersusun atas kolom dan baris. Dalam dunia database, kolom disebut sebagai *field* dan baris disebut sebagai *record*. Dalam model relasional, sebuah database akan tersusun atas beberapa tabel yang saling berelasi atau memiliki keterkaitan satu sama lain [14].

2.3.2 Constraint

Constraint adalah suatu aturan atau batasan yang mendefinisikan nilai atau data yang dapat disimpan di dalam database, baik melalui operasi INSERT, UPDATE maupun DELETE [14]. Dalam SQL standar ANSI, constraint diebadakan menjadi empat yaitu: *Primary Key*, *Foreign Key, Unique* dan *Check*.

2.3.2.1 Primary Key

Primary Key adalah suatu aturan yang berguna untuk memastikan bahwa setiap baris data di dalam suatu tabel bersifat unik (berbeda antara baris yang satu dengan yang lainnya). *Primary Key* diterapkan pada kolom-kolom yang akan djadikan sebagai pembeda [14].

2.3.2.2 Foreign Key

Foreign Key berguna untuk mendefinisikan kolom-kolom pada suatu tabel yang nilainya mengacu ke tabel lain. Dengan kata lain, kolom-kolom yang didefinisikan sebagai foreign key nilainya harus diambil dari nilai kolom pada tabel lain. Kolom pada tabel lain yang nilainya akan diacu harus berupa kolom *primary key* atau *unique* [14].

2.3.2.3 Unique

Unique pada dasarnya sama dengan primary key, yaitu untuk memastikan bahwa setiap baris data yang terdapat dalam suatu tabel bersifat unik (tidak sama). Perbedaannya pada unique key boleh terdapat data yang bernilai NULL [14].

2.3.3 Index

Indeks adalah suatu objek *database* yang berfungsi untuk mempercepat proses pengambilan, pengurutan maupun pencarian data pada suatu tabel di dalam *database*. Data pada tabel yang sudah diindeks akan diurutkan berdasarkan kolom indeks. Dengan demikian proses pencarian data dapat lebih cepat dilakukan [14].

Indeks harus berasosiasi dengan suatu kolom dalam sebuah tabel. Dengan kata lain, suatu tabel dapat diindeks berdasarkan kolom tertentu yang telah ditentukan sebelumnya. *Constraint primary key* secara otomatis dianggap sebagai indeks [14].

2.4 Structured Query Language

Structured Query Language (SQL) adalah bahasa atau kumpulan perintah standar yang digunakan untuk berkomunikasi dengan database. Perintah dalam SQL diklasifikasikan menjadi tiga bagian besar, yaitu: *Data Definition Language* (DDL), *Data Manipulation Language* (DML) dan *Data Control Language* (DCL).

2.4.1 Data Definition Language

Data Definition Language (DDL) adalah kumpulan perintah SQL yang berkaitan dengan pembuatan, perubahan, dan penghapusan *database* maupun objek-objek yang terdapat di dalam *database*, seperti tabel, indeks, prosedur/fungsi, *trigger*, dan sebagainya [14].

Perintah SQL yang termasuk kategori DDL adalah:

1. CREATE, berfungsi untuk membuat *database* dan objek-objek di dalam *database*.
2. ALTER, berfungsi untuk mengubah *database* dan objek-objek di dalam *database*.
3. DROP, berfungsi untuk menghapus *database* dan objek-objek di dalam *database*.

2.4.2 Data Manipulation Language

Data Manipulation Language (DML) adalah kumpulan perintah SQL yang berkaitan dengan data atau isi dari suatu tabel. Dengan perintah-perintah di dalam DML, kita dapat memanipulasi (menambah, mengubah, dan menghapus) data yang terdapat pada suatu tabel secara mudah [14].

Perintah-perintah yang termasuk ke dalam DML adalah:

1. INSERT, berfungsi untuk menambah atau memasukkan data baru ke dalam tabel.
2. UPDATE, berfungsi mengubah data dalam tabel dengan nilai baru.
3. DELETE, berfungsi untuk menghapus data dari suatu tabel.

2.4.3 Data Control Language

Data Control Language (DCL) adalah kumpulan perintah SQL yang digunakan untuk mengontrol data. DCL digunakan untuk menyimpan atau membatalkan transaksi, manajemen user dan hak akses [14].

Perintah-perintah yang termasuk ke dalam DCL diantaranya:

1. GRANT, digunakan untuk memberikan hak akses kepada user tertentu.
2. REVOKE, digunakan untuk mencabut salah satu atau beberapa hak akses dari user tertentu.
3. COMMIT, berfungsi untuk menyimpan perubahan-perubahan yang dilakukan terhadap database melalui perintah INSERT, UPDATE dan DELETE secara permanen.
4. ROLLBACK, membatalkan transaksi atau perubahan-perubahan yang telah dilakukan ke dalam database melalui perintah INSERT, UPDATE atau DELETE.

2.5 Application Programming Interface

Application Programming Interface (API) adalah antarmuka yang digunakan untuk dapat berkomunikasi dengan aplikasi lain di luar sistem. API berguna untuk meningkatkan skalabilitas aplikasi dan kemudahan untuk dapat digunakan antar pengguna dan perangkat. REST (*Representational State Transfer*) adalah gaya arsitektur yang dibuat untuk membantu membuat dan mengatur sistem terdistribusi [15]. Keunggulan menggunakan arsitektur REST:

1. Performa : Gaya komunikasi yang terdapat pada REST adalah mengutamakan efisiensi dan kemudahan. Hal ini dilakukan dengan cara memperbolehkan peningkatan performansi pada sistem yang menggunakan REST.
2. Skalabilitas : Dengan kemudahan yang terdapat pada REST memungkinkan untuk mengatur hal ini.

3. Kemudahan Antarmuka : Antarmuka yang sederhana membuat interaksi antara sistem lebih mudah yang menjadi keunggulan REST.
4. Kemudahan Modifikasi : Dengan menggunakan REST memungkinkan untuk mengubah setiap komponen satu-persatu.
5. Portabilitas : REST merupakan teknologi yang memungkinkan perbedaan bahasa, yang berarti REST dapat diimplementasikan dan dikonsumsi dari berbagai macam jenis teknologi.

2.6 Javascript Object Notation (JSON)

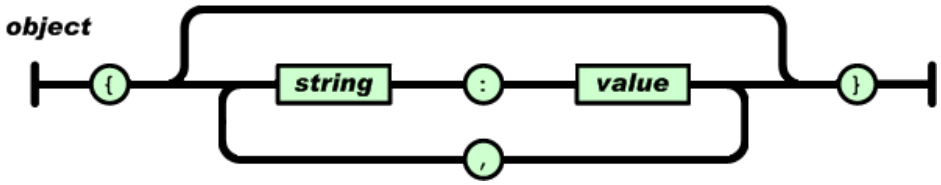
JSON (Javascript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman Javascript, Standar ECMA-262 Edisi ke-3 – Desember 1999 [16].

JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh bahasa pemrograman keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python, dll. Oleh karena sifat-sifat tersebut, JSON ideal digunakan sebagai bahasa pertukaran-data. Struktur-struktur data ini disebut sebagai struktur data universal. JSON terbentuk dari dua struktur yakni:

1. Kumpulan pasangan nama dan nilai. Dalam beberapa bahasa pemrograman, bentuk ini biasa dikenal dengan *object, record, struct, dictionary, hash table, keyed list* atau *associative array*.
2. Kumpulan list berurutan dari suatu nilai. Dalam beberapa bahasa pemrograman, bentuk ini dikenal dengan *array, vector, list* atau *sequence*.

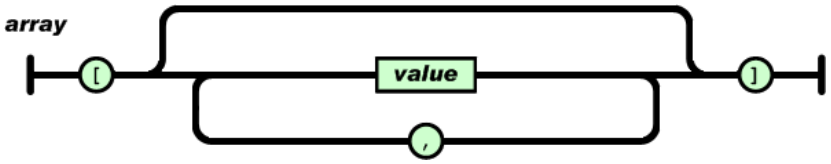
Secara umum semua bahasa pemprograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan.

Object adalah pasangan nilai dan nama yang tidak berurutan. *Object* dimulai dengan tanda kurung kurawal buka dan diakhiri dengan tanda kurung kurawal tutup. Setiap nama diikuti dengan tanda titik dua dan setiap pasangan nama dan nilai dipisahkan dengan tanda koma [16]. Gambar 2.3 menunjukkan format object pada json:



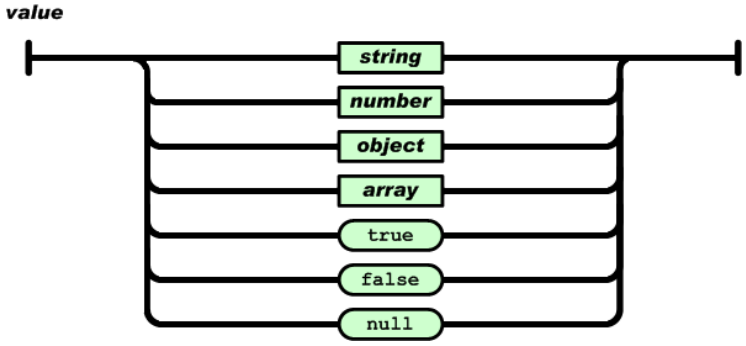
Gambar 2.3 Format object pada json

Array adalah kumpulan nilai yang terurut. *Array* dimulai dengan tanda kurung siku buka dan diakhiri dengan tanda kurung siku tutup. Setiap elemen didalam array dipisahkan dengan tanda koma. Gambar 2.4 menunjukkan format array pada json:



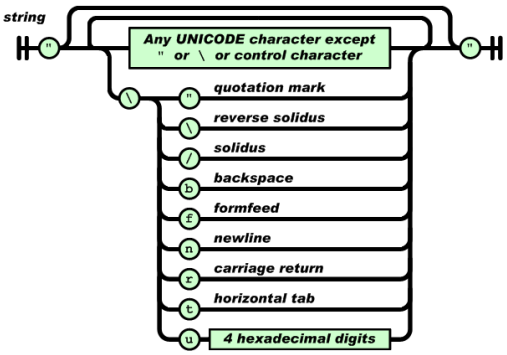
Gambar 2.4 Format array pada json

Value pada json bisa berupa string dalam tanda petik dua, angka, boolean (true/false), null, kumpulan *object* atau kumpulan *array*. Struktur tersebut bisa tersusun secara bersarang. Gambar 2.5 menunjukkan format value pada json:



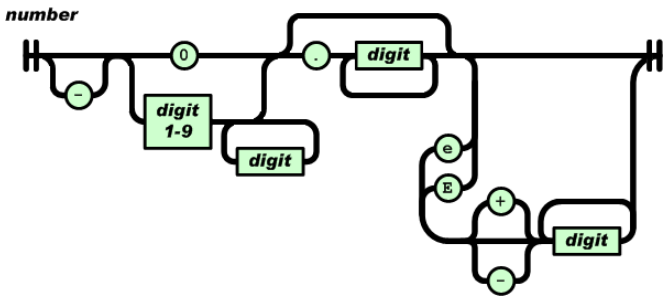
Gambar 2.5 Format value pada json

String adalah kumpulan nol atau lebih karakter unicode yang diapit oleh tanda kutip dua dan dapat menggunakan tanda backslash. Karakter direpresentasikan sebagai karakter string tunggal. Gambar 2.6 menunjukkan format string pada json:



Gambar 2.6 Format string pada json

Number pada json mirip dengan number pada bahasa pemrograman C atau Java. Perbedaannya pada json tidak terdapat bilangan oktal dan heksadesimal. Gambar 2.7 menunjukkan format number pada json:



Gambar 2.7 Format number pada json

2.7 Pengujian *Black Box*

Pengujian adalah proses untuk menemukan *error* pada perangkat lunak sebelum dikirim kepada pengguna. Pengujian perangkat lunak adalah kegiatan yang ditujukan untuk mengevaluasi atribut atau kemampuan program dan memastikan bahwa itu memenuhi hasil yang dicari, atau suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (*under test*), Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya [17].

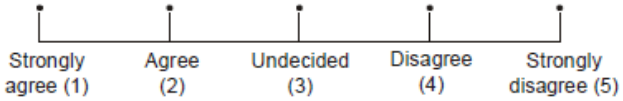
1. Prinsip Pengujian
    a) Semua pengujian harus dapat dirunut sampai kepada spesifikasi kebutuhan perangkat lunak.
    b) Pengujian harus dimulai dari lingkup yang kecil ke lingkup yang besar
    c) Pengujian yang mendalam tidak mungkin dilakukan karena tidak mungkin mengeksekusi semua jalur permutasi
    d) Supaya efektif (memiliki probabilitas yang tinggi dalam menemukan kesalahan), Pengujian harus dilakukan oleh pihak lain yang independen
    e) Pengujian harus direncanakan jauh sebelum dilakukan
2. Kualitas Pengujian yang baik
    a) Mencakup semua kemungkinan skenario pengoperasian perangkat lunak.
    b) Mencakup sebanyak mungkin jalur yang dibentuk dari struktur program.
    c) Tidak terlalu sederhana dan tidak terlalu rumit.

Pengujian *Black Box* adalah pengujian yang dilakukan untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinsikan. Metode *Black Box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya sesuai dengan persyaratan fungsional untuk suatu program [17].

2.8 Skala Likert

Skala likert adalah cara melakukan analisis terhadap pandangan responden dalam melihat suatu persoalan dengan cara menilai nilai terendah dan tertinggi yang diperoleh. Responden diminta untuk memberikan penilaian berdasarkan persetujuan atau ketidaksetujuannya terhadap setiap pernyataan yang diberikan. Setiap respon diberikan nilai berupa skot angka yang mengindikasikan kesukaan atau ketidaksukaannya dan seluruh skor dihitung untuk melihat sikap responden. Dengan kata lain, skor rata-rata merepresentasikan posisi responden dalam menyikapi isu yang diberikan [18].

Dalam skala likert, responden diminta untuk memberikan pendapat terhadap setiap pernyataan yang diberikan dalam bentuk derajat persetujuan yang biasanya terdiri dari lima level, contoh: sangat setuju, setuju, biasa saja, tidak setuju, sangat tidak setuju [18]. Gambar 2.8 menunjukkan contoh skala likert:



Gambar 2.8 Contoh skala likert

Setiap skor biasanya dinyatakan dalam angka yang dapat diukur. Misalnya saja pada skala likert dengan lima pilihan maka:

a. Sangat tidak setuju (1 poin)
b. Tidak setuju (2 poin)
c. Biasa saja (3 poin)
d. Setuju (4 poin)
e. Sangat tidak setuju (5 poin)

Jika terdapat 30 pertanyaan yang diajukan kepada responden maka untuk mengukur pandangan responden dapat diketahui dengan cara:

a. 30 x 5 = 150 (Responden sepenuhnya setuju)
b. 30 x 4 = 120 (Responden setuju)
c. 30 x 3 = 90 (Responden menunjukkan sikap netral)
d. 30 x 2 = 60 (Responden tidak setuju)
e. 30 x 1 = 30 (Responden sangat tidak setuju)

Dengan demikian dapat diketahui posisi, pandangan dan pendapat responden terhadap pernyataan yang diberikan.

Langkah membuat skala likert:

1. Peneliti mengumpulkan seluruh pernyataan yang relevan terhadap penelitian yang sedang dilakukan. Setiap pernyataan mengekspresikan kesetujuan dan ketidaksetujuan dalam berbagai sudut pandang.
2. Melakukan pengujian terhadap beberapa subjek. Dengan kata lain terhadap orang-orang tertentu (kelompok kecil) untuk mengetahui respon yang diberikan.
3. Respon yang menyatakan kesetujuan dapat dilihat dari skor tertinggi yang digunakan (misalnya skor 5) dan respon yang menyatakan ketidaksetujuan dapat dilihat dari skor terendah yang ada (misalnya skor 1)
4. Jumlahkan seluruh total skor yang diperoleh dari seluruh pernyataan yang ada.
5. Menentukan hasil pendapat responden terhadap pernyataan yang diberikan.

2.9 Object Oriented Analysis and Design

Konsep pemrograman berorientasi objek sudah ada lebih dari 30 tahun lamanya. Pada awal tahun 1990, metode yang diperkenalkan oleh Grady Booch dan James Rumbaugh menjadi cukup populer. Booch dan Rumbaugh mengkombinasikan metode yang dikenal dengan *Unified Method* (UM), metode ini yang kemudan dikenal dengan nama *Unified Modelling Language* (UML). Metode berorientasi objek adalah metode yang memandang sesuatu sebagai bagian dari objek yang ada di dunia nyata [19].

2.10 Unified Modeling Language

UML (*Unified Modeling Language*) adalah bahasa standar yang digunakan dalam memodelkan software dan pengembangan sistem [20] Di dalam UML dikenal Model Kruchten yang membagi masing-masing diagram yang terdapat pada UML sesuai dengan pengelompokannya.

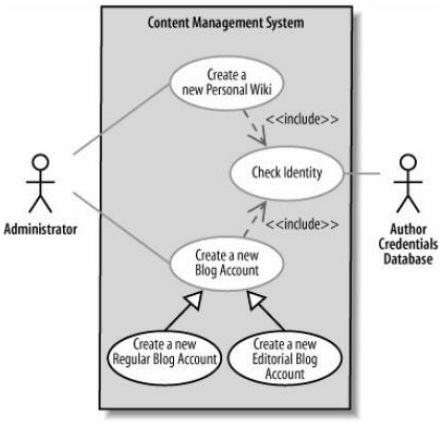
Model Kruchten 4+1

Model Kruchten digunakan untuk membagi masing-masing diagram yang terdapat pada UML sesuai dengan masing-masing pengelompokannya.

1. Use Case View
    a. Use Case Diagram
2. Process View
    a. Activity Diagram
    b. Sequence Diagram
    c. Communication Diagram
    d. Timing Diagram
3. Logical View
    a. Class Diagram
    b. Object Diagram
    c. State Machine Diagram
    d. Interaction Diagram
4. Physical View
    a. Deployment Diagram
5. Development View
    a. Package Diagram
    b. Component Diagram

2.10.1 Use Case Diagram

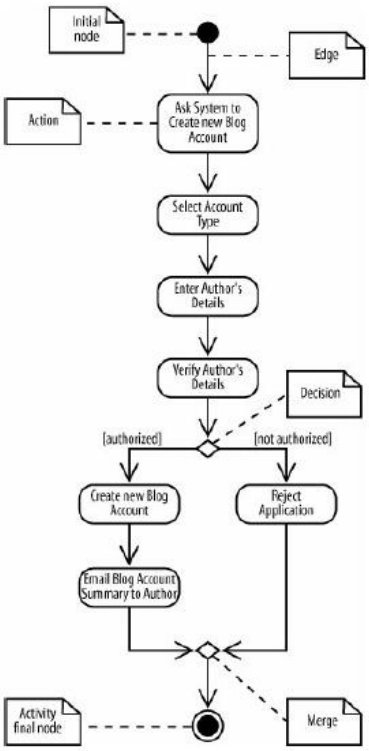
Use case diagram digunakan untuk memodelkan proses bisnis dari perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh aktor. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke use case. Gambar 2.9 menunjukkan contoh *use case diagram*:



Gambar 2.9 Contoh *use case diagram*

2.10.2 Activity Diagram

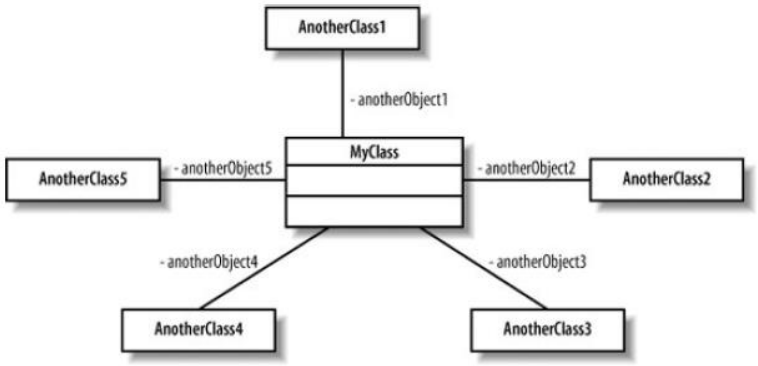
Activity Diagram digunakan untuk memodelkan proses-proses yang terjadi dalam setiap use case yang terdapat pada use case diagram. *Activity Diagram* juga digunakan untuk memperjelas langkah-langkah yang dilakukan pada use case. Gambar 2.10 menunjukkan contoh *activity diagram*:



Gambar 2.10 Contoh *activity diagram*

2.10.3 Class Diagram

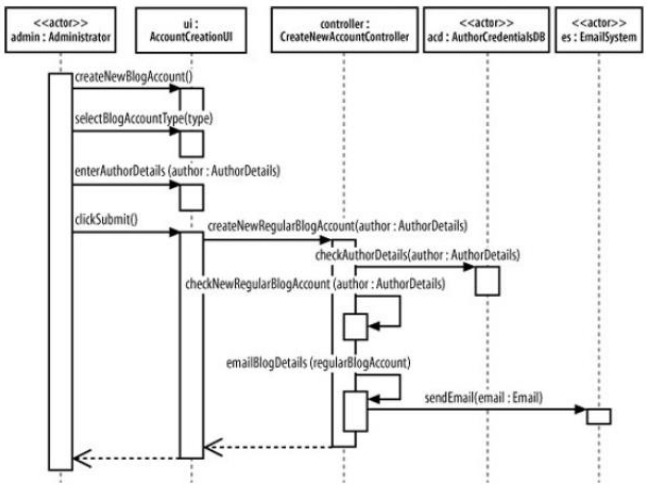
Class diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Komponen yang terdapat pada class diagram diantaranya: *Class*, *Property*, dan *Method*. Gambar 2.11 menunjukkan contoh class diagram:



Gambar 2.11 Contoh *class diagram*

2.10.4 Sequence Diagram

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Gambar 2.12 menunjukkan contoh *sequene diagram*:



Gambar 2.12 Contoh *sequence diagram*

2.11 PHP

PHP merupakan secara umum dikenal sebagai bahasa pemrograman script yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML dikenal juga sebagai bahasa pemrograman server side. PHP/FI merupakan nama awal dari PHP. PHP – *Personal Home Page*, FI adalah *Form Interface*. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima input melalui form yang ditampilkan dalam browser web. *Software* ini disebarkan dan dilisensikan sebagai perangkat lunak Open Source [21].

Integrasi PHP dengan server web dilakukan dengan teknin CGI, FastCGI, dan modul server web. Teknik CGI dan FastCGI memisahkan antara server web dan PHP; sedangkan modul server web menjadi PHP sebagai bagian dari server web. Kini, PHP adalah kependekan dari PHP: *HyperText Preprocessor* (rekursif, mengikut gaya penamaan di *nix), merupakan bahasa utama *script server-side* yang disisipkan pada HTML yan dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi desktop [21].

2.12 Lumen

Lumen adalah *framework* pengembangan perangkat lunak yang dibangun dalam bahasa pemrograman PHP. Lumen merupakan *framework* minimalis yang menggunakan komponen yang sama yang terdapat pada Laravel [22]. Laravel adalah framework pengembangan perangkat lunak yang dibuat untuk memenuhi kebutuhan pengembangan aplikasi web secara menyeluruh mulai dari antarmuka hingga alur logika aplikasi. Gambar 2.13 menunjukkan kode program menggunakan Lumen:

```php
<?php

/**
 * Reimagine what you expect...
 */
$app->get('/', function() {
    return ['version' => '5.3']
});

/**
 * From your micro-framework...
 */
$app->post('framework/{id}', function($framework) {

    $this->dispatch(new Energy($framework));
});


$app->get('api/users/{id}', function($id) {
```

Gambar 2.13 Contoh kode program dengan lumen framework

2.13 MySQL

MySQL merupakan software RDBMS (atau *server database*) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*) dan dapat melakukan suatu proses secara sinkron tau berbarengan (*multi-threaded*) [14]. MySQL menyediakan kapabilitas atau kemampuan untuk mendefinisikan prosedur dan fungsi sebagaimana layaknya bahasa pemrograman. Dalam dunia *database*, prosedur yang disimpan didalam *database* sering dinamakan sebagai *stored procedure* dan fungsi dinamakan *stored function* [14].

Prosedur dan fungsi merupakan objek *database* yang berisi runtunan statemen atau perintah yang dibuat untuk memenuhi kebutuhan-kebutuhan khusus tertentu. Sekali dibuat, prosedur dan fungsi dapat digunakan secara berulang. Hal ini, tentu akan mempersingkat penulisan kode atau perintah yang digunakan untuk mengirim permintaan ke *server database*. Sama halnya seperti tabel atau *view*, prosedur dan fungsi juga berasosiasi dengan *database* tertentu. Artinya, jika kita menghapus suatu *database* maka perosedur dan fungsi yang terdapat didalam *database* tersebut juga akan ikut dihapus [14]. Contoh query MySQL:

```
SELECT *
FROM user
WHERE gender = 'M' AND name NOT LIKE '%asep%';
```

2.14 Fulltext Search

Fulltext Search adalah fitur yang dapat digunakan untuk melakukan pencarian pada database MySQL berdasarkan kata kunci dengan cara yang lebih spesifik dibandingkan pencarian dengan *query* LIKE [7]. Fulltext Search pada engine database InnoDB baru tersedia di MySQL versi 5.6 yang rilis pada tahun 2013.

Fulltext Search hanya dapat diterapkan pada engine database MyISAM dan InnoDB serta pada tipe data CHAR, VARCHAR, atau TEXT. Secara umum terdapat dua metode pencarian yang dapat digunakan dalam melakukan pencarian Fulltext pada database MySQL:

1. Metode *Natural Language*

   Metode *Natural Language Fulltext Search* adalah metode pencarian pada database MySQL yang dilakukan dengan pendekatan bahasa natural.
2. Metode *Boolean*

   Metode *Boolean Fulltext Search* adalah metode pencarian pada database MySQL yang menerapkan metode yang sama dengan *Natural Language Fulltext Search*, namun pada metode ini dapat diterapkan operator dalam melakukan pencarian [23].

2.14.1 Operator MySQL *Fulltext Search Boolean Mode*

Tabel 2.1 menunjukkan operator-operator yang dapat digunakan untuk melakukan pencarian pada database mysql menggunakan metode fulltext search boolean mode.

Tabel 2.1 Operator *fulltext search boolean mode*

Operator	Deskripsi
(Tanpa Operator)	Tanpa operator merupakan konfigurasi default pada MATCH() ... AGAINST()
+	Operator tambah menunjukkan bahwa kata ini harus ada di setiap baris yang ingin dicocokkan.
-	Operator minus menunjukkan bahwa kata ini tidak boleh ada di salah satu baris yang ingin dicocokkan.
@distance	Operator ini bekerja pada tabel InnoDB saja. Operator ini menguji apakah dua kata atau lebih semuanya

	dimulai dalam jarak yang ditentukan satu sama lain, diukur dengan kata-kata. Tentukan kata-kata pencarian dalam string kutipan ganda sebelum operator @distance, misalnya MATCH (col1) AGAINST ("'word1 word2 word3" @ 8' IN BOOLEAN MODE)
< >	Kedua operator ini digunakan untuk mengubah kontribusi kata ke nilai relevansi yang ditugaskan berturut-turut. Operator > meningkatkan kontribusi dan < operator menurunkannya.
~	Operator negasi, menyebabkan kontribusi kata tersebut ke relevansi baris menjadi negatif. Ini berguna untuk menandai kata-kata "*noise*". Baris yang berisi kata tersebut dinilai lebih rendah dari yang lain, namun tidak dikecualikan sama sekali, seperti halnya dengan operator *minus* (-).
*	Operator * berfungsi sebagai operator pemotongan (atau wildcard). Kata cocok jika mereka dimulai dengan kata yang didahului operator *.
"…"	Operator tanda petik dua yaitu untuk mencari kata atau frasa yang benarbenar sama dengan masukan.

2.14.2 Perhitungan Relevansi *Fulltext Search* InnoDB

Fitur pencarian menggunakan metode *fulltext search* pada engine InnoDB dimodelkan berdasarkan *Fulltext Search Engine Sphinx*. Algoritma yang digunakan berdasarkan algoritma BM25 dan TF-IDF. Hal ini membuat perhitungan relevansi pada engine InnoDB dengan MyISAM bisa saja berbeda [23]. InnoDB menggunakan sistem pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*) untuk melakukan perankingan dokumen berdasarkan relevansi query terhadap dokumen. Pembobotan dilakukan berdasarkan seberapa sering sebuah kata muncul didalam dokumen dibandingkan dengan seberapa banyak kata tersebut berada didalam koleksi dokumen. Dengan kata lain, semakin banyak kata tersebut muncul dalam dokumen dan semakin sedikit sebaran kata tersebut berada didalam koleksi dokumen maka dokumen tersebut dikatakn semakin relevan dengan kata yang dicari [23].

Term Frequency (TF) adalah banyaknya kata muncul di dalam dokumen. Sedangkan *Inverse Document Frequency* (IDF) adalah hasil perhitungan pada persamaan (1):

$$\${IDF\} = log10\left(\frac{\${total\ records\}}{\${matching\ records\}}\right) \qquad (1)$$

Saat dokumen memiliki kata yang sama berakali-kali maka nilai IDF dikalikan dengan nilai TF, sehingga formulasi relevansi diketahui pada persamaan (2):

$$\${rank\} = \${TF\} * \${IDF\} * \${IDF\} \qquad (2)$$

Saat terdapat banyak kata yang dicari menggunakan metode *fulltext search* maka untuk setiap kata yang dicari diberlakukan perhitungan menggunakan persamaan (2) sehingga untuk melakukan pencarian banyak kata yakni hasil penjumlahan dari persamaan (2). Persamaan 3 menunjukkan pencarian dengan banyak kata:

$$\${rank\} = \sum_{i=1}^{n} (\${TF\} * \${IDF\} * \${IDF\}) \qquad (3)$$

Contoh query pencarian pada database MySQL dengan menggunakan *Fulltext Search Boolean Mode*:

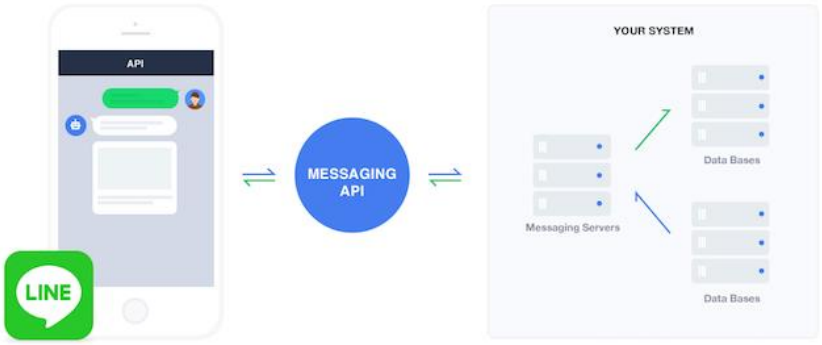
```
SELECT *, MATCH(name) AGAINST('asep' IN BOOLEAN MODE) relevancy
FROM user WHERE MATCH(name) AGAINST('asep' IN BOOLEAN MODE) > 0;
```

2.15 Line

Line adalah aplikasi pengirim pesan instan yang dapat digunakan pada berbagai macam platform seperti smartphone, tablet, dan komputer. Line dapat digunakan melalui jaringan internet dan penggunanya dapat melakukan aktivitas mengirim pesan teks, gambar, video dan suara. Line memiliki beberapa fitur yang cukup digemari salah satunya yakni fitur bot yang dapat dibuat dan digunakan pada aplikasi Line untuk berbagai macam kebutuhan seperti diantaranya pencarian informasi, sosialisasi, pemasaran produk, bermain game, dsb. Line merupakan aplikasi pesan instan yang banyak digunakan oleh masyarakat di Indonesia. Berdasarkan data pada tahun 2018 jumlah pengguna Line di Indonesia tercatat sebanyak 90 juta pengguna yang 80% diantaranya didominasi oleh anak muda [10].

2.15.1 Line Messaging API

Line Messaging API adalah API yang dikembangkan untuk membangun bot pada platform Line. API ini bekerja dengan cara menerima dan mengirimkan data antara server bot dengan server pada platform Line. Saat pengguna mengirimkan pesan kepada bot, bot akan memeriksa alamat server bot (*webhook*) dan kemudian meneruskan pesan tersebut. Server bot kemudian memproses dan mengirimkan kembali respon terhadap pengirim pesan. Pengiriman dan penerimaan pesan dilakukan melalui protokol HTTPS dalam format JSON [24]. Gambar 2.14 menunjukkan arsitektur line messaging api:



Gambar 2.14 Arsitektur line messaging api

2.16 Google API

Google API adalah kumpulan api yang dikembangkan oleh Google yang memungkinkan komunikasi dengan layanan Google untuk diintegrasikan dengan layanan lain. Beberapa contoh layanan yang terdapat pada Google API diantaranya: Google map, place, earth, geocoding dan youtube [25]. Untuk dapat menggunakan layanan google, pengembang harus memiliki akses terhadap layanan tersebut. Cara mendapatkan akses yakni dengan mendaftar pada layanan Google Console untuk kemudian mendaftarkan layanan-layanan yang akan digunakan.

2.16.1 Google Distance Matrix API

Google Distance Matrix API adalah salah satu API yang disediakan oleh Google untuk dapat digunakan oleh pengembang produk untuk membuat produknya. Distance Matrix API adalah layanan yang menyediakan jarak dan waktu perjalanan antara lokasi asal dengan lokasi tujuan berdasarkan rekomendasi rute antara titik awal dan akhir [26]. Gambar 2.15 menunjukkan contoh respon Google Distance Matrix API:


```
{
   "destination_addresses" : [ "New York, NY, USA" ],
   "origin_addresses" : [ "Washington, DC, USA" ],
   "rows" : [
      {
         "elements" : [
            {
               "distance" : {
                  "text" : "225 mi",
                  "value" : 361715
               },
               "duration" : {
                  "text" : "3 hours 49 mins",
                  "value" : 13725
               },
               "status" : "OK"
            }
         ]
      }
   ],
   "status" : "OK"
}
```


Gambar 2.15 Contoh respon google distance matrix api