

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Tinjauan pustaka merupakan penelitian-penelitian yang berkaitan dengan permasalahan yang telah diuraikan. Penelitian tersebut adalah sebagai berikut:

Penelitian yang dilakukan oleh Findra Kartika Sari Dewi (2016). "Rancang Bangun Aplikasi Pengingat Kegiatan Akademik Berbasis Android". Jurusan Teknik Informatika, Program Strata S1, Universitas Atma Jaya Yogyakarta. Sistem yang dihasilkan berbasis android dan *website* dengan menggunakan bahasa pemrograman Java dan PHP kemudian untuk DBMS yang digunakan adalah MySQL. Aplikasi *mobile* dibuat untuk *broadcast* pesan atau informasi kepada mahasiswa yang mengambil kelasnya, melihat jadwal kuliah dan melihat kalender agenda perkuliahan.

Penelitian yang dilakukan oleh Mohd Siddik (2018). "Teknologi Push Notifikasi Berbasis Android Untuk Informasi Perkuliahan".Jurusan Sistem Informasi, STMIK Royal. Aplikasi yang dibuat berbasis android Untuk Informasi Perkuliahan menggunakan *Firebase Cloud Massaging, Google Cloud Messaging Service*.

Penelitian yang dilakukan oleh Ommi Alfina (2019). "Sistem Informasi Mobile Assistant Mahasiswa Jurusan Sistem Informasi Fakultas Komputer Universitas Potensi Utama Berbasis Android". Yang bertujuan untuk pelaksanaan kegiatan sesuai dengan jadwal yang telah ditentukan, pelaksanaan perkuliahan tepat pada waktunya,

mahasiswa dapat mengetahui waktu untuk perkuliahan dan dapat mengikutinya, mahasiswa dapat mengikuti ujian dan tepat pada waktunya.

Penelitian yang dilakukan oleh Nopi Ramsari, Achmad Rifaldi (2018). “Rancang Bangun Aplikasi Penjadwalan Kegiatan Akademik Disertai Sistem Reminder Berbasis Responsiv Web Design”. Yaitu aplikasi penjadwalan kegiatan akademik dengan sistem *reminder* bisa menjadi alat untuk mempermudah proses pencatatan dan pengingat kegiatan akademik FIKI Universitas Nurtanio Bandung sehingga dapat meningkatkan keefektifitasan kegiatan akademik Fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung.

Penelitian yang dilakukan oleh Irma Salamah, Lindawati, Herlina Fitri Handayan (2019).”Rancang Bangun Aplikasi Infirmasi Perkuliahan Berbasis Android”.Berbasis android yang memberikan informasi perkuliahan dilengkapi dengan fitur notifikasi sebagai pengingat bagi mahasiswa. Aplikasi ini berguna bagi mahasiswa karena sangat efektif dan efisien dalam pengaksesan infromasi perkuliahan sehingga mahasiswa dapat mendapatkan informasi up to date dimana saja dengan lebih mudah selama terhubung dengan koneksi internet

Adapun perbedaan penelitian – penelitian sebelumnya dengan penelitian sekarang bisa dilihat pada Tabel 2.1.

Tabel 2. 1 Perbedaan Penelitian

No	Nama Peneliti	Judul Penelitian	Software	Hasil Penelitian
1	Findra Kartika Sari Dewi (2016)	Rancang Bangun Aplikasi Pengingat Kegiatan	JAVA MySQL PHP	Aplikasi <i>mobile</i> dibuat untuk <i>broadcast</i> pesan atau informasi kepada mahasiswa yang mengambil kelasnya, melihat jadwal kuliah dan melihat

		Akademik Berbasis Mobile		kalender agenda perkuliahan.
2	Mohd Siddik (2018)	Teknologi Push Notifikasi Berbasis Android Untuk Informasi Perkuliahan	FCM GCM	Aplikasi yang dibuat berbasis android yang menggunakan teknologi push notification sebagai pesan informasi yang akan terkirim kepada pengguna.
3	Omni Alfina (2019)	Sistem Informasi Mobile Assistant Mahasiswa Jurusan Sistem Informasi Fakultas Komputer Universitas Potensi Utama Berbasis Android	PHP HTML JAVA XML ANDROID STUDIO	pelaksanaan kegiatan sesuai dengan jadwal yang telah ditentukan, pelaksanaan perkuliahan tepat pada waktunya, mahasiswa dapat mengetahui waktu untuk perkuliahan dan dapat mengikutinya, mahasiswa dapat mengikuti ujian dan tepat pada waktunya.
4	Nopi Ramsari, Achmad Rifaldi (2018)	Rancang Bangun Aplikasi Penjadwalan Kegiatan Akademik Disertai Sistem Reminder Berbasis Responsiv Web Design	Reminder Sms Email API RWD Ruby on Rails Postgresql Postfix	aplikasi penjadwalan kegiatan akademik dengan sistem reminder bisa menjadi alat untuk mempermudah proses pencatatan dan pengingat kegiatan akademik FIKI Universitas Nurtanio Bandung sehingga dapat meningkatkan keefektifitasan kegiatan akademik Fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung.
5	Irma Salamah, Lindawati, Herlina Fitri Handayani (2019)	Rancang Bangun Aplikasi Perkuliahan Berbasis Android	RUP	Aplikasi Go-Campus merupakan aplikasi berbasis Android yang memberikan informasi perkuliahan dilengkapi dengan fitur notifikasi sebagai pengingat bagi mahasiswa. Aplikasi ini berguna bagi mahasiswa karena sangat efektif dan efisien dalam pengaksesan informasi perkuliahan sehingga mahasiswa dapat mendapatkan informasi up to date dimana saja dengan lebih mudah selama terhubung dengan koneksi internet
6	Suryatus Sholihin (2020)	Rancang Bangun Sistem Notifikasi Kalender Akademik Rogram Studi Sistem Informasi Berbasis Android	PHP HTML JAVA ANDROID STUDIO MySQL	Sebuah sistem notifikasi kalender akademik yang diharapkan mempermudah Bagian Akademik dalam menyampaikan informasi kegiatan Prodi Sistem Informasi kepada mahasiswa dan sebaliknya mahasiswa dapat menerima informasi kegiatan tersebut dengan cepat dan mudah.

2.2. Landasan Teori

2.2.1. Teori Konsep yang Diterapkan

a. Sistem

Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu.

b. Notifikasi

Menurut *website* kamus besar bahasa indonesia yang memiliki domain www.kbbi.com notifikasi adalah pemberitahuan atau penawaran barang dan sebagainya.

c. Kalender Akademik

Kalender akademik adalah pengaturan waktu untuk kegiatan pembelajaran peserta didik selama satu tahun ajaran yang mencakup permulaan tahun pelajaran, minggu efektif belajar, waktu pembelajaran efektif dan hari libur.

2.2.2. Perangkat Lunak yang Digunakan

a. Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu – Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor code IntelliJ dan alat

pengembang yang berdaya guna, Android Studio menawarkan lebih banyak fitur. Gunanya untuk meningkatkan produktivitas kamu saat membuat aplikasi Android, misalnya:

- Sistem versi berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- Alat pengujian dan kerangka kerja yang ekstensif
- Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- Dukungan C++ dan NDK
- Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

b. PHP (*Hypertext Preprocessor*)

Hypertext Preprocessor (PHP) yaitu bahasa pemrograman web server side yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer *server*. Hal ini berbeda

dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada web *browser(client)*. Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang *powerfull* dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla dan lainnya (Anhar, 2010).

c. JavaScript

Javascript diperkenalkan pertama kali oleh Netscape pada tahun Pada awalnya bahasa ini dinamakan LiveScript yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru buru dan tidak ada pesan kesalahan yang di tampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara Netscape dan Sun (pengembang bahasa pemrograman Java) pada masa itu, maka Netscape memberikan nama JavaScript kepada bahasa tersebut pada tanggal 4 desember Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai Jscript di browser Internet Explorer

d. MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web, desktop maupun *mobile* yang menggunakan *database* sebagai sumber dan pengelolaan datanya. *MySQL* menggunakan SQL sebagai bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan (S. Rosa A, 2013).

e. REST API

Representational State Transfer (REST) adalah arsitektur perangkat lunak yang digunakan untuk pendistribusian sistem layanan web, sedangkan REST API adalah aplikasi yang berfungsi untuk melakukan komunikasi dengan layanan web menggunakan HTTP untuk melakukan perintah *GET*, *PUT*, *POST*, dan *DELETE* data. Layanan web (*web services*) sendiri adalah *server web* yang dibuat khusus untuk mendukung kebutuhan situs atau aplikasi lainnya sehingga bisa secara langsung menerima dan menanggapi permintaan klien menggunakan API yang bisa mengekspos satu set data dan memfasilitasi antar program komputer yang memungkinkan untuk bertukar informasi (Masse, 2011).

2.2.3. Basis Data

a. Pengertian Basis Data

Basis data terdiri atas 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul.

Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, atau kombinasinya (Fathansyah, 2015)

b. Kunci Relasi Basis Data

Kunci relasi merupakan satu atau gabungan atribut yang bersifat unik yang dapat digunakan untuk mengidentifikasi atau membedakan setiap *record* dalam relasi. Kunci relasi diperlukan dalam rangka pengaksesan data dari dalam relasi atau untuk menyusun kerelasian antar relasi. Berdasarkan macamnya kunci relasi terdiri atas (Edhy Sutanta, 2012).

1. Kunci kandidat (*Candidate key/CK*)

Kunci kandidat (*candidate key/CK*) adalah satu atau gabungan minim alat ribut yang bersifat unik yang dapat digunakan untuk mengidentifikasi atau membedakan setiap *record* dalam relasi (Edhy Sutanta, 2012).

2. Kunci primer (*Primary key/PK*)

Kunci primer (*primarykey/ PK*) adalah bagian atau salah satu dari *candidate key (CK)* yang dipilih atau digunakan sebagai kunci utama untuk mengidentifikasi atau membedakan setiap *record* dalam relasi. Dalam setiap relasi harus mempunyai PK dan jumlahnya hanya satu buah. PK haru unik dan tidak boleh *null* (Edhy Sutanta, 2012).

3. Kunci alternatif (*Alternate key /AK*)

Kunci alternatif (*alternate key/AK*) adalah bagian dari CK yang tidak dipilih atau digunakan sebagai PK.

4. Kunci penghubung (*Foreignkey /FK*)

Kunci penghubung (*foreign key/FK*) atau sering pula disebut sebagai kunci tamu atau kunci asing adalah satu atau gabungan sembarang atribut yang menjadi PK dalam relasi lain yang mempunyai hubungan secara logik. Dalam basisdata (*database*) kerelasian antar relasi yang mengacu atau mereferensi pada relasi lain disebut sebagai relasi anak, sedangkan relasi yang menjadi acuan atau referensi disebut sebagai relasi induk (Edhy Sutanta, 2012).

c. Kerelasian Antar Tabel

Dalam model data relasional (*relational database model/RDBM*) istilah kerelasian (*relationship*) dibedakan dengan istilah relasi (*relational*). Relasi menyatakan sebuah tabel dalam basis data sedangkan kerelasian menyatakan hubungan antar relasi dalam basis data (Edhy Sutanta, 2012).

Jenis-jenis kerelasian antar relasi pada basisdata (*database*) antara lain sebagai berikut:

1. Satu kesatu / 1-ke-1 (*one to one / 1-to-1*)

Jenis kereliasian 1-ke-1 terjadi jika setiap nilai pada suatu relasi hanya mengimplikasikan sebuah nilai pada relasi lain yang direlasikan secara logik (Edhy Sutanta, 2012).



Gambar 2. 1 Simbol Relasi *One to One*

2. Satu banyak / 1-ke-n (*one to many / 1-to-n*)

Jenis kereliasian 1-ke-n terjadi jika setiap nilai pada suatu relasi mengimplikasikan banyak (lebih dari satu) nilai pada relasi lain yang direlasikan secara logik (Edhy Sutanta, 2012).



Gambar 2. 2 Simbol Relasi *One to Many*

3. Banyak kesatu / n-ke-1 (*many to one / n-to-1*)

Jenis kereliasian-ke1 terjadi jika banyak (lebih dari satu) nilai pada suatu relasi mengimplikasikan hanya satu nilai pada relasi lain yang direlasikan secara logik (Edhy Sutanta, 2012).



Gambar 2. 3 Simbol Relasi *Many to One*

4. Banyak banyak / n-ke-n (*many to many / n-to-n*)

Jenis kerelasian-ke-n terjadi jika banyak (lebih dari satu) nilai pada suatu relasi mengimplikasikan banyak (lebih dari satu) nilai pada relasi lain yang direlasikan secara logik (Edhy Sutanta, 2012).



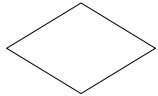

Gambar 2. 4 Simbol Relasi *Many to Many*

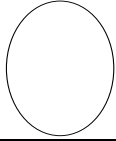

2.2.4. Entity Relationship Diagram (ERD)

ERD (*Entity Relationship Diagram*) merupakan suatu model data yang dikembangkan berdasarkan obyek. ERD digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pemakai secara logis (Edhy Sutanta, 2012).

Simbol-simbol yang sering digunakan ERD seperti yang diperlihatkan pada tabel 2.2

Tabel 2. 2 Simbol *Entity Relationship Diagram* (ERD)

Nama	Simbol	Keterangan
<i>Relasi</i>		Menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda dan derajat relasi menunjukkan jumlah maksimum entitas yang dapat berelasi.
<i>Entitas</i>		Merupakan individu yang mewakili sesuatu yang nyata (<i>eksistensinya</i>) dan dapat dibedakan dari suatu yang lain.






<i>Atribut</i>		Merupakan bagian dari entitas yang mendeskripsikan karakteristik (<i>properti</i>) dari entitas tersebut.
<i>Asosiasi</i>		Penghubung antar relasi dan entitas di mana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian.

2.2.5. Unified Modeling Language (UML)

a. Use Case Diagram

Bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Tabel 2.2 menjelaskan simbol-simbol dari *use case* :

Tabel 2. 3 Simbol Diagram *Use Case*

Simbol	Deskripsi
<i>Use case</i>  nama use case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
Aktor/ <i>actor</i>  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi/ <i>extend</i> <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walaupun tanpa <i>usecase</i> tambahan.
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

Simbol	Deskripsi
<p><i>Include</i> <<include>>></p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat menjalankan <i>use case</i> ini.

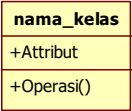





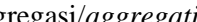
Sumber : (Yuni Sugiarti, 2013)

b. Diagram Kelas (*Class Diagram*)

Diagram kelas bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi serta relasi-relasi.

Tabel 2.3 menjelaskan simbol-simbol diagram kelas:

Tabel 2. 4 Simbol Diagram Kelas


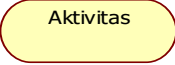
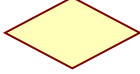



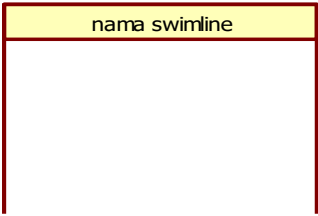
Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Antarmuka/<i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi/<i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum – khusus).
<p>Ketergantungan</p> 	Ketergantungan antar kelas.
<p>Agregasi/<i>aggregation</i></p> 	Relasi antar kelas dengan makna semua-bagian (<i>whole – part</i>).

Sumber : (Yuni Sugiarti, 2013)

2.2.6. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem dan bersifat dinamis. Tabel 2.4 menjelaskan simbol-simbol diagram aktivitas :

Tabel 2. 5 Simbol Diagram Aktivitas

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
<i>Transition</i> 	Menambah transisi dari suatu aktivitas ke aktivitas yang lain.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimline</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : (Yuni Sugiarti, 2013)