

BAB II DASAR TEORI

2.1 Tinjauan Pustaka

Perancangan dan implementasi yang berhubungan dengan penelitian ini adalah “Rancang Bangun Aplikasi Penjadwalan Kegiatan Akademik Disertai Sistem *Reminder* Berbasis *Responsive Web Design*”. Sistem informasi ini digunakan sebagai aplikasi penjadwalan dengan sistem *reminder* untuk kegiatan akademik Fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung berbasis *Responsive Web Design* menggunakan *framework Ruby on Rails*. Teknik *reminder* yang digunakan dalam pembuatan aplikasi ini menggunakan teknik *EmailBlast*, *SMBlash* dan *Notification*. Aplikasi yang digunakan untuk mengirimkan notifikasi adalah aplikasi *third-party* seperti Twilio, NusaSMS, dan Onesignal [6].

Penelitian lainnya yaitu yang berjudul “Rancang Bangun Aplikasi *Mobile* Penjadwalan Perkuliahan dengan *Firebase* dengan *Realtime Notification*”. Metode pengembangan sistem pada penelitian ini menggunakan metode *Rapid Application Development* (RAD). Pada penelitian ini, peneliti menerapkan *Firebase Cloud Messaging* (FCM) yang akan mengirimkan pemberitahuan kepada pengguna. Pengguna dapat melihat isi pemberitahuan yang di kirim oleh dosen pada menu lihat notifikasi [7].

Penelitian lainnya yaitu “Rancang Bangun Aplikasi Informasi Perkuliahan Berbasis Android”. Hasil penelitian ini berupa aplikasi *Go-Campus* yang berisi bebrapa menu seperti informasi jadwal kuliah, jadwal ujian akhir semester, jadwal *incidental*, kalender akademik dan pengumuman terbaru yang dilengkapi dengan *push notification* sebagai pengingat bagi mahasiswa. Metode pengembangan perangkat lunak yang digunakan yaitu metode *Rational Unified Process* (RUP) dengan menggunakan *use case driven* dan dilakukan secara *iterative* [8].

Penelitian lainnya yaitu “Rancang Bangun Aplikasi Jadwal Kegiatan Akademik Berbasis Android (Studi Kasus: Program Studi Teknik Informatika Fakultas Teknik Universitas Tanjungpura)”. Aplikasi yang dirancang pada penelitian ini merupakan aplikasi Jadwal Kegiatan Akademik Berbasis Android pada Program Studi Teknik Informatika Fakultas Teknik Universitas Tanjungpura. Aplikasi ini dibangun untuk memfasilitasi mahasiswa dan dosen untuk mengakses jadwal kegiatan akademik. Aplikasi ini dilengkapi dengan fitur *push*

notification yang akan mengirimkan pesan kepada pengguna jika jadwal baru tersedia sehingga pengguna dapat lebih mudah mengakses informasi jadwal akademik yang baru [9].

Berdasarkan penelitian-penelitian yang sudah ada terkait dengan pembuatan sistem informasi kalender, maka akan dilakukan penelitian yang membahas tentang kalender akademik Politeknik Negeri Cilacap berbasis *website* untuk menyampaikan informasi kegiatan akademik Politeknik Negeri Cilacap melalui notifikasi telegram. Beberapa hal yang membedakan antara penelitian yang akan dilakukan dengan penelitian-penelitian sebelumnya diantaranya adalah dari segi desain aplikasi, fungsi yang ada didalam aplikasi, dari segi pengingat dimana penelitian ini menggunakan notifikasi telegram sebagai pengingat kegiatan yang ada di dalam kalender akademik Politeknik Negeri Cilacap. Penelitian ini juga membahas hal yang sama yaitu tentang penjadwalan kegiatan akademik.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi merupakan proses pengumpulan, penyimpanan, analisis sebuah informasi dengan tujuan tertentu. Sistem informasi yang terdiri dari data (*input*) dan menghasilkan laporan (*output*) sehingga diterima oleh sistem lainnya serta kegiatan strategi dalam suatu organisasi dalam melakukan tindakan atau keputusan [10].

2.2.2 Kalender Akademik

Kalender akademik adalah istilah umum dalam dunia pendidikan yang merujuk kepada jadwal kegiatan tahunan suatu lembaga pendidikan yang memuat semua hal terkait dengan proses belajar-mengajar, penerimaan peserta didik, dan kelulusan [11].

2.2.3 Bot Telegram

Telegram bot *Application Programming Interface* (API) adalah sebuah teknologi *open source* yang disediakan oleh Telegram *Messenger* LLP untuk membangun aplikasi bot telegram bagi para pengembang. Bot API ini merupakan *interface* berbasis HTTP untuk menghubungkan bot yang dikembangkan oleh para pengembang dengan sistem telegram [12]. Kelebihan dari telegram ini adalah adanya landasan untuk menggunakan *Application Programming Interface* (API)

untuk masyarakat luas. Salah satu API yang disediakan adalah fitur bot. Bot telegram merupakan sebuah akun khusus yang tidak memerlukan nomer telepon. Akun ini berfungsi sebagai *interface* untuk menjalankan *code* yang sudah dibangun. Untuk keamanan data, *server* perantara pada telegram akan menangani semua enkripsi dan komunikasi dengan bot API. Sehingga para pengembang tidak perlu mengetahui bagaimana protokol enkripsi MTProto pada *server* telegram bekerja [13].

2.2.4 Situs Web (*Website*)

Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia, selama terkoneksi dengan jaringan internet [14].

2.2.5 Basis Data

Basis data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu. Dalam basis data ada beberapa istilah dasar yang perlu diketahui [15].

1. **Entitas**, biasanya berupa objek yang berwujud, seperti: orang rumah, mobil dan lainnya. *Entitas* terbagi menjadi dua yaitu *entitas* kuat dan *entitas* lemah.
2. **Atribut** adalah karakteristik yang melekat yang menjelaskan secara detail *entitas*.
3. **Record** adalah isian data yang saling terhubung.
4. **File** merupakan kumpulan dari *record* yang menggambarkan *entitas*.
5. **Key** bagian dari *record* yang dipergunakan untuk proses akses dan menemukan *record* yang dimaksud. Ada beberapa jenis *key* dalam database yaitu:
 - a. **Primary key** bersifat unik dan *field* yang mengidentifikasi sebuah *record*.
 - b. **Foreign key** (kunci tamu) merupakan *field* yang bukan kunci dan biasanya disebut sebagai kunci tamu yang merupakan kunci pada *file* yang lain.
 - c. **Candidate key** merupakan atribut yang mengidentifikasi secara unik sebuah *entitas*. Syarat sebuah *candidate key* adalah *unique identifier* dan *no redundancy*.
 - d. **Composite key** adalah sebuah kunci yang terdiri dari 2 *atribut* atau lebih yang mengidentifikasi suatu *entitas*.

2.2.6 *Structured Query Language (SQL)*

SQL merupakan kependekan dari “*Structured Query Language*”. SQL merupakan suatu Bahasa permintaan yang terstruktur. SQL adalah Bahasa permintaan yang melekat pada suatu *database* atau SMBD tertentu, berikut beberapa perintah dalam SQL [16]:

1. *Data Definition Language (DDL)*

Data Definition Language (DDL) merupakan sub bahasa SQL yang digunakan untuk membangun kerangka *database*. Ada tiga perintah dalam DDL, yaitu:

- a. **CREATE**: Perintah ini digunakan untuk membuat, termasuk membuat *database* baru, tabel baru, *view* baru, dan kolom.
- b. **ALTER**: Perintah ini digunakan untuk mengubah struktur table yang telah dibuat.
- c. **DROP**: Perintah ini digunakan untuk menghapus *database* dan tabel.

2. *Data Manipulation Language (DML)*

Data Manipulation Language (DML) adalah sub bahasa *SQL* yang digunakan untuk memanipulasi data dalam *database* yang telah terbuat. Perintah yang digunakan, diantaranya:

- a. **INSERT**: Perintah ini digunakan untuk menyisipkan atau memasukkan data baru ke dalam tabel. Penggunaannya setelah *database* dan tabel selesai dibuat.
- b. **SELECT**: Perintah ini digunakan untuk mengambil data atau menampilkan data dari tabel atau beberapa tabel dalam satu relasi.
- c. **UPDATE**: Perintah ini digunakan untuk memperbarui data lama menjadi data terkini.
- d. **DELETE**: Perintah ini digunakan untuk menghapus data dari tabel.

2.2.7 *Rekayasa Perangkat Lunak*

Jawadekar (2004) menyatakan bahwa perangkat lunak adalah sekelompok intruksi untuk mendapatkan masukan dan untuk memanipulasinya hingga menghasilkan keluaran yang diharapkan dalam bentuk fungsi dan kinerja seperti ditentukan oleh pemakai

perangkat lunak[17]. Kriteria yang dapat digunakan sebagai acuan dalam merekayasa perangkat lunak [18]:

1. Dapat terus dirawat dan dipelihara (*maintainability*).
2. Dapat mengikuti keinginan pengguna (*robust*).
3. Efektif dan efisien dalam menggunakan energi dan penggunaannya.
4. Dapat memenuhi kebutuhan yang diinginkan (*usability*).

2.2.8 *Hypertext Preprocessor (PHP)*

PHP (*Hypertext Preprocessor*) yaitu Bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegasi dengan HTML dan berada pada *server (server side HTML embedded scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Keunggulan PHP sebagai salah satu bahasa pemrograman berbasis *website* [19]:

1. Gratis. PHP dapat diunduh dan dipergunakan secara gratis. PHP dikembangkan oleh komunitas *open source* dan akan selalu didistribusikan secara gratis.
2. PHP berlisensi GNU *General Public License (GPL)*.
3. Performa handal. PHP sangatlah efisien dengan spesifikasi *server* yang tidak mahal, dapat melayani jutaan akses per hari.
4. Dukungan basisdata. PHP hampir mendukung semua perangkat basisdata.
5. Pustaka bawaan. PHP dibangun secara khusus untuk aplikasi berbasis *web*. Oleh karena itu, disediakan banyak pustaka (*library*) bawaan berkaitan dengan *web* yang dapat langsung digunakan.
6. *Cross Platfrom*. PHP dapat dijalankan dengan baik pada hampir semua sistem operasi.
7. Mudah dipelajari. Perintah-perintah PHP sangat mudah dipelajari, karena sebagai besar perintah PHP diadopsi dari bahasa pemrograman populer seperti bahasa C++, java, perl.

2.2.9 *Unified Modeling Language (UML)*

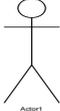
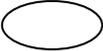
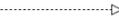
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga

merupakan alat untuk mendukung pengembangan sistem [20]. Adapun jenis diagram UML diantaranya seperti:

1. *Use case diagram*

Use case diagram, yaitu diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan *actor*. Diagram ini hanya menggambarkan secara *global*. Karena *use case diagram* hanya menggambarkan sistem secara *global*, maka elemen-elemen yang digunakan pun sangat sedikit, berikut elemen-elemen yang digunakan pada *use case diagram* [20].

Tabel 2. 1 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Elemen yang menjadi pemicu sistem. Aktor bisa merupakan orang, mesin ataupun sistem lain yang berinteraksi dengan <i>use case</i> .
2.		<i>System</i>	<i>System</i> merupakan batasan-batasan proses yang sudah kita deskripsikan dalam sebuah sistem.
3.		<i>Use case</i>	Potongan proses yang merupakan bagian dari sistem.
4.		<i>Association</i>	menggambarkan interaksi antara <i>use case</i> dengan aktor.
5.		<i>Dependency</i>	Menggambarkan relasi antara dua <i>use case</i> .
6.		<i>Generalization</i>	Menggambarkan pewarisan antara dua aktor atau <i>use case</i>

			dimana salah satu aktor atau <i>use case</i> mewarisi <i>properties</i> ke aktor atau <i>use case</i> yang satunya.
--	--	--	---

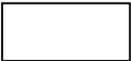
2. **Flowchart**

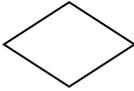
Flowchart adalah bagian alir yang menggambarkan tentang urutan Langkah jalannya suatu program dalam sebuah bagan dengan simbol-simbol bagan yang sudah ditentukan. Ketentuan menuliskan *flowchart* adalah sebagai berikut[20]:

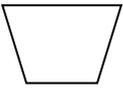
1. *Flowchart* dituliskan dari atas ke bawah.
2. Jika tidak cukup dan dituliskan ke samping maka *flowchart* dituliskan dari kiri ke kanan. Tiap-tiap simbol harus memberikan/diberikan keterangan yang jelas.
3. Untuk symbol terminal/terminator, keterangan yang bisa dituliskan di dalamnya adalah [Mulai|Selesai|Start|End] → atau yang menjelaskan tentang *state* awal dan *state* akhir.
4. Untuk simbol proses terdapat *assignment* atau operator aritmatika.
5. Untuk simbol keputusan boleh terdapat operator perbandingan.
6. Untuk penggunaan konektor dalam satu halaman menggunakan simbol konektor dengan bentuk lingkaran, dan untuk konektor dari satu simbol yang lain dengan simbol yang berbentuk segi lima.

Simbol-simbol dalam kelas diagram dapat dilihat dalam tabel berikut [21]:

Tabel 2. 2 Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1.		Proses	Merupakan simbol yang digunakan untuk memberikan nilai tertentu, apakah berupa rumus, perhitungan <i>counter</i> atau hanya pemberian nilai tertentu terhadap suatu <i>variable</i> .

2.		<i>Input / Output</i>	Merupakan simbol yang digunakan untuk memasukan nilai dan untuk menampilkan karakter/nilai (isi) dari suatu <i>variable</i> .
3.		Alur proses	Merupakan simbol yang digunakan untuk menentukan aliran dari sebuah <i>flowchart</i> program.
4.		<i>Predefine process</i>	Merupakan simbol yang penggunaannya seperti <i>link</i> atau menu.
5.		<i>Connector/p</i> enghubung	Konektor dalam satu halaman merupakan penghubung dari simbol yang satu ke simbol yang lain, tanpa harus menuliskan arus yang panjang.
6.		<i>Manual input</i>	Input yang dimasukkan secara manual dari <i>keyboard</i> .
7.		<i>Decision /</i> simbol keputusan	Digunakan untuk menentukan pilihan suatu kondisi (ya atau tidak). Ciri simbol ini dibandingkan dengan simbol-simbol <i>flowchart</i> program yang lain adalah simbol keputusan ini minimal keluaran arusnya 2, jadi jika hanya satu keluaran maka penulisan simbol ini adalah salah.

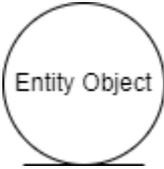
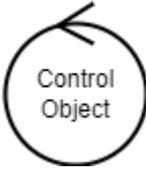
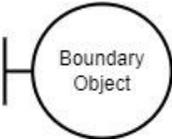
8.		<i>Document Symbol</i>	Simbol yang menunjukkan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> yang perlu cetak diatas kertas.
9.		<i>Manual Operation Symbol</i>	Simbol yang digunakan untuk menunjukkan kegiatan/proses yang tidak dilakukan oleh komputer.

3. *Sequence Diagram*

Sequence diagram adalah salah satu dari diagram-diagram yang ada pada UML, *sequence diagram* ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukan rangkaian pesan yang terkirim antara *object* juga interaksi antara *object*. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem [20].

Tabel 2. 3 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Gambaran orang yang melakukan interaksi dengan sistem.

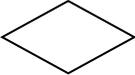
2.		<i>Entity Class</i>	Gambaran hubungan melakukan kegiatan.
3.		<i>Control Class</i>	Penghubung antara <i>boundary</i> dengan <i>database</i> .
4.		<i>Boundary Class</i>	Gambaran halaman atau tampilan.
5.		<i>A Focus of Control & Active Line</i>	Gambaran tempat memulai dan mengakhiri pesan.
6.		<i>A Message</i>	Gambaran alur mengirim pesan.

2.2.10 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan *tools* yang digunakan untuk memodelkan struktur data dengan menggambarkan *entitas* dan hubungan antara *entitas* (*relationship*) secara abstrak (konseptual). Fungsi utama ERD yaitu [22]:

1. Sebagai alat untuk memodelkan hasil analisis data.
2. Sebagai alat untuk memodelkan data konseptual.
3. Sebagai alat untuk memodelkan objek-objek dalam suatu sistem (dasar dari objek diagram/*class diagram*).

Tabel 2. 4 Simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Nama	Keterangan
1.		<i>Entity</i>	<i>Entity</i> adalah suatu objek yang utuh dan <i>independent</i> terhadap suatu objek lain dalam lingkup masalah yang dituju (memiliki fungsi relevan terhadap sistem).
2.		<i>Relationship</i>	Hubungan antara <i>entity</i> dalam sebuah ERD disebut <i>relationship</i> .
3.		<i>Atribute</i>	<i>Atribut</i> adalah karakteristik umum yang dimiliki oleh semua <i>instance</i> dalam suatu <i>entity</i> tertentu.

-HALAMAN INI SENGAJA DIKOSONGKAN-