

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu dalam pada penelitian ini penulis mengambil dua penelitian yaitu:

Penelitian pertama di lakukan oleh (Simarmata and Harahap, 2018) dengan study kasus “**Sistem Penjadwalan Iklan Menggunakan Metode Priority Schedulling pada PT. Kidung Indah Selaras Suara (Radio Kiss FM) untuk Efektivitas dan Efisiensi Produksi Siaran**” pada penelitian ini memiliki masalah yaitu jumlah iklan yang harus disiarkan pada waktu yang terbatas cukup banyak, maka kelalaian dari salah satu bagian akan berakibat pada efisiensi waktu dan biaya, padahal waktu adalah komponen yang sangat berarti dalam memperoleh pendapatan. Selain itu, dalam manajemen untuk proses laporan, misalnya laporan tentang iklan yang disiarkan, harus dilakukan cross check antara bagian monitoring dan bagian produksi, ini merupakan proses yang memakan waktu lama jika dilakukan secara manual. Untuk itu, proses penjadwalan iklan yang digunakan adalah dengan algoritma Priority Schedulling. Algoritma ini, dipilih karena dengan melakukan penjadwalan penyiaran iklan berdasarkan pada prioritas dapat mengurangi kerugian biaya yang ditimbulkan karena kesalahan penjadwalan iklan. Hal ini, sesuai dengan keunggulan algoritma penjadwalan berprioritas dinamis yakni memenuhi kebijaksanaan yang ingin mencapai kondisi maksimum dari suatu kriteria yang ditetapkan.

Penelitian kedua di lakukan oleh (Nardialis, Permana and Salisah, 2017) dengan study kasus “**Sistem Penjadwalan Otomatis Tempat Ceramah Mubaligh Pada Bulan Ramadhan**” pada penelitian ini memiliki masalah yaitu jadwal bentrok pada saat penyusunan jadwal disebabkan oleh empat hal, yaitu: (1) pengelolaan jadwal yang masih manual; (2) harus mempertimbangkan enam kriteria yang telah dijelaskan sebelumnya; (3) banyaknya jumlah mubaligh/ghah dan masjid/musalah; dan (4) adanya masjid/musalah yang melakukan permintaan hari libur ceramah yang berbeda-beda pada bulan Ramadhan. Penelitian ini

menggunakan metode priority scheduling dan constraint satisfaction problem sebagai pendekatan penyusunan jadwal tempat ceramah mubaligh pada bulan Ramadhan. Pendekatan tersebut digunakan karena pendekatan tersebut telah sukses menyelesaikan berbagai kasus penjadwalan, seperti: (1) penjadwalan tempat khutbah jum'at mubaligh; (2) penjadwalan iklan; (3) penjadwalan perwalian; (4) penjadwalan jaga staff laboratorium [5]; dan (5) penjadwalan mata kuliah.

Dari kedua penelitian diatas dapat dijelaskan kelebihan dan kekurangan dari masing-masing penelitian adalah sebagai berikut :

Tabel 2.1 Kelebihan dan Kekurangan Penelitian Terdahulu

No	Judul	Kelebihan	Kekurangan
1	Sistem Penjadwalan Iklan Menggunakan Metode Priority Scheduling pada PT. Kidung Indah Selaras Suara (Radio Kiss FM) untuk Efektivitas dan Efisiensi Produksi Siaran	Dengan penggunaan metode priority scheduling sistem penjadwalan iklan dapat diatur secara mudah berdasarkan tingkat kepentingannya.	Sistem yang dibuat masih memerlukan beberapa bagian dalam mengelola penjadwalan dan juga pengiklan tidak terlibat.
2	Sistem Penjadwalan Otomatis Tempat Ceramah Mubaligh Pada Bulan Ramadhan	Sistem penjadwal mubaligh menjadi lebih mudah karena sistem secara otomatis menempatkan jadwal agar tidak ada jadwal yang tumpang tindih.	Sistem hanya tersedia hanya dalam bentuk web sehingga perlu dikases terlebih dahulu dan tidak dapat memberikan notifikasinya.

2.2 Android Studio

Menurut (Firly, 2018) menjelaskan bahwa *Android studio* merupakan *integrated development environment (IDE)* atau dalam artian lain adalah sebuah lingkungan pengembangan terintegrasi resmi yang memang merancang khusus untuk pengembangan sistem operasi *Google Android*. Aplikasi ini dibangun di atas sebuah perangkat lunak yang dinamakan *IntelliJ IDEA* milik *JetBrains*. Bisa juga dibilang bahwa *android studio* merupakan pengganti dari *Eclipse android development tool* atau *ADT* sebagai *IDE* utama dalam pengembangan aplikasi *android* yang asli.

Android studio diluncurkan pada tanggal 16 mei 2013 dalam *konferensi google I/O* yang pada saat itu masih dalam tahap pratinjau akses *versi 0.1* sebagai perintis. Hingga pada akhirnya *versi stabil 3.0* yang *liris* pada pertengahan bulan oktober 2017 dan menjadi *software* terlaris dikalangan *developer* muda. Aplikasi ini dapat digunakan diberbagai sistem operasi yaitu *windows,linux* dan *macOS*.

Aplikasi ini menawarkan berbagai fitur canggih yang akan meningkatkan kemampuan *produktivitas* dalam proses pengembangan aplikasi. Berikut ini adalah beberapa hal yang akhirnya banyak mengundang *developer* untuk melirik *android studio* sebagai *software* pengembang :

- a. Dukungan dari C++, *NDK* dan sekarang *kotlin*
- b. Perkembangan yang *up to date*
- c. Sistem berbasis *Gradle* yang dinilai fleksibel
- d. Lingkungan yang mencakup seluruh perangkat *android*
- e. *Emulator* yang cepat dan kaya akan fitur
- f. Alat pengujian dan kerangka yang juga *ekstensif*
- g. *Instant Run*
- h. Dukungan *google cloud platform*

2.3 Sejarah Android

Perjalanan Android dimulai sejak Oktober 2003 ketika 4 orang pakar IT, Andi Rubin, Rich Miner, Nick Sears dan Chris White mendirikan *Android.Inc*, di

California US. Visi Android untuk mewujudkan mobile device yang lebih peka dan mengerti pemiliknya, kemudian menarik raksasa dunia maya Google. Google kemudian mengakuisisi Android pada Agustus 2005. OS Android dibangun berbasis platform Linux yang bersifat open source, senada dengan Linux, Android juga bersifat Open Source. Dengan nama besar Google dan konsep open source pada OS Android, tidak membutuhkan waktu lama bagi android untuk bersaing dan menyisihkan Mobile OS lainnya seperti Symbian, Windos Mobile, Blackberry dan iOS. Kini siapa yang tak kenal Android yang telah menjelma menjadi penguasa Operating System bagi Smartphone.

2.4 Priority Scheduling

Priority Scheduling merupakan algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi. Setiap proses memiliki prioritasnya masing-masing (Simarmata and Harahap, 2018).

Prioritas suatu proses dapat ditentukan melalui beberapa karakteristik antara lain:

1. *Time limit*.
2. *Memory requirement*.
3. Akses file.
4. Perbandingan antara *burst M/K* dengan *CPU burst*.
5. Tingkat kepentingan proses.

Priority scheduling juga dapat dijalankan secara *preemptive* maupun *non-preemptive*. Pada *preemptive*, jika ada suatu proses yang baru datang memiliki prioritas yang lebih tinggi daripada proses yang sedang dijalankan, maka proses yang sedang berjalan tersebut dihentikan, lalu CPU dialihkan untuk proses yang baru datang tersebut. Sementara itu, pada *non-preemptive*, proses yang baru datang tidak dapat mengganggu proses yang sedang berjalan, tetapi hanya diletakkan di depan *queue*. Contoh algoritma priority scheduling :

Tabel 2.2 Contoh Algoritma Priority Scheduling

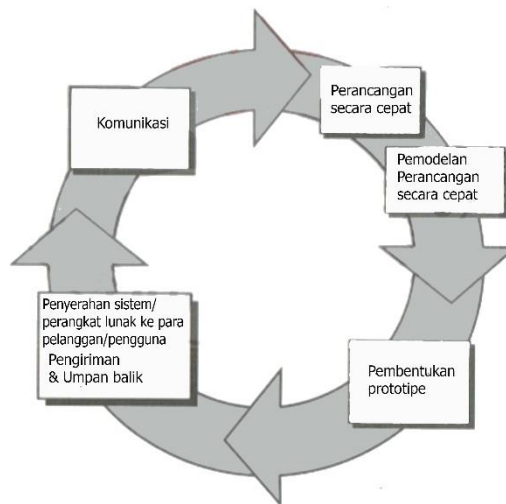
Process	Duration	Priority	Arival Time
P1	6	4	0
P2	8	1	0
P3	7	3	0
P4	3	2	0

Pada tabel 2.1 diatas merupakan contoh dalam penentuan priority scheduling dimana setiap proses akan dilihat priority nya yang akan di jadikan acuan dalam menentukan mana proses yang akan didahulukan berdasarkan prioritas utama terlebih dahulu.

Kelemahan pada *priority scheduling* adalah dapat terjadinya *indefinite blocking (starvation)*. Suatu proses dengan prioritas yang rendah memiliki kemungkinan untuk tidak dieksekusi jika terdapat proses lain yang memiliki prioritas lebih tinggi darinya. (Simarmata and Harahap, 2018) Solusi dari permasalahan ini adalah *aging*, yaitu meningkatkan prioritas dari setiap proses yang menunggu dalam *queue* secara bertahap.

2.5 Metode Pengembangan Perangkat Lunak

Menurut (Pressman, 2010) menjelaskan meskipun pembuatan *prototype* dapat di gunakan sebagai model proses yang berdiri sendiri, pembuatan protoipe lebih umum digunakan sebagai teknik yang dapat diimplementasikan di dalam konteks setiap model proses perangkat lunak, dan paradigma pembuatan *prototype* seringkali membantu tim pengembang perangkat lunak dan para *stakeholder* untuk memahami lebih baik apa yang akan dikembangkan saat spesifikasi kebutuhan belum jelas.



Gambar 2.1 Tahapan Pengembangan Aplikasi

Berikut ini adalah penjelasan dari gambar 2.1

1. Komunikasi

Tahap komunikasi ini adalah tahapan komunikasi antara *developer* dan pelanggan mengenai tujuan pembuatan *software*, mengidentifikasi apakah kebutuhan diketahui.

2. Perancangan secara cepat

Tahap perancangan secara cepat ini adalah tahapan perancangan cepat setelah terjalin komunikasi.

3. Pemodelan perancangan secara cepat

Tahap pemodelan perancangan secara cepat ini adalah tahapan segera membuat model, dan pemodelan cepat fokus pada gambaran dari segi *software* apakah *visible* menurut pelanggan.

4. Pembentukan prototipe

Tahap pembentukan prototipe ini adalah tahapan pemodelan cepat menuntun pada pembuatan dari prototipe.

5. Penyerahan sistem

Tahap penyerahan sistem / perangkat lunak ke para pelanggan / pengguna pengiriman dan umpan balik ahapan ini adalah prototipe yang dikirimkan kemudian dievaluasi oleh pelanggan, umpan balik digunakan untuk menyaring kebutuhan untuk program.

2.6 Unified Modeling Language (UML)


Menurut (Mulyani, 2016) *Unified Modeling Language (UML)* adalah sebuah Teknik pengembangan sistem yang menggunakan Bahasa garfish sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. Menurut Mulyani (2016) tipe-tipe diagram UML adalah sebagai berikut :




2.6.1 Use Case Diagram

Use case diagram adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mepresentasikan sebuah interaksi antara actor dan sistem.

Dalam *use case* diagram terdapat istilah seperti aktor, *use case* dan *case relationship*

Tabel 2.3 Komponen *Use Case Diagram*




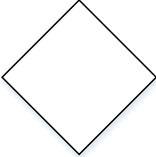
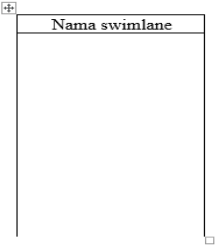
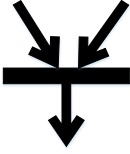
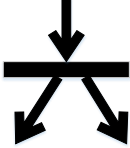
Simbol	Nama elemen	Keterangan
	Aktor	Merupakan kesatuan eksternal yang berinteraksi dengan sistem

Simbol	Nama elemen	Keterangan
	Use Case	Rangkaian / uraian sekelompok yang saling terkait dan membentuk sistem
	Asosiasi	Penghubung antar elemen (aktor/use case) didalam sistem.
	Generalisasi	Sebuah elemen yang dihasilkan dari pewarisan elemen lain.
<<extend>>	Extend	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tamahan itu.
<<include>>	Include	<i>Use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

2.6.2 Activity Diagram

Activity Diagram yang disediakan oleh UML melengkapi *use case* yang telah dibuat sebelumnya memberikan representasi grafis dari aliran-aliran interaksi di dalam suatu skenario yang sifatnya spesifik. Mirip dengan diagram alir, suatu diagram aktifitas menggunakan sebuah kotan yang berisi lengkung untuk menggambarkan fungsi tertentu yang ada dalam suatu sistem yang akan dikembangkan, sementara itu tanda panah menggambarkan aliran didalam sistem dan seterusnya.

Tabel 2.4 Komponen *Activity Diagram*

Simbol	<i>Keterangan</i>
	<i>Activity</i> : Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial Node</i> : Bagaimana objek dibentuk atau diawali
	<i>Activity Final Node</i> : Bagaimana objek dibentuk dan diakhiri.
	<i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
	<i>Swimlane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	<i>Join</i> : Digunakan untuk menunjukkan kegiatan yang digabungkan.
	<i>Fork</i> : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel


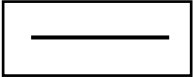


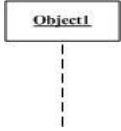
2.6.3 Sequence Diagram

Sequence diagram di UML terutama digunakan untuk memodelkan interaksi antara aktor dan objek dalam sistem dan interaksi antara obyek itu sendiri. UML memiliki sintaks yang kaya untuk sequence diagram, yang memungkinkan berbagai jenis interaksi yang dimodelkan. Sesuai namanya,

sequence diagram menunjukkan urutan interaksi yang terjadi antara use case. Sequence diagram memiliki dua buah karakteristik yaitu :

1. Setiap objek memiliki lifeline yang digambarkan dengan garis putus-putus vertikal dan garis ini menunjukkan daur hidup dari sebuah objek.
2. Terdapat fokus kontrol yang digambarkan dengan sebuah persegi panjang yang tipis dan tinggi. Fokus kontrol ini menunjukkan periode waktu selama sebuah objek melakukan sebuah event.

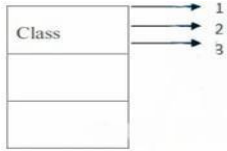

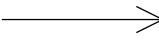
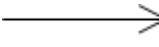
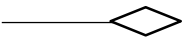
Tabel 2.5 Komponen *Sequence Diagram*

Simbol	Nama Elemen	Keterangan
	<i>Life Line</i>	Objek entitas, antar muka yang saling berinteraksi
	<i>Activation</i>	Menggambarkan hubungan antar objek dengan <i>message</i>
	<i>Message (call)</i>	Menggambarkan alur message yang merupakan kejadian objek pengirim <i>life line</i> ke objek penerima <i>life line</i>
	<i>Message (return)</i>	Menggambarkan alur pengambilan <i>message</i> ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.
	<i>Object</i>	Object adalah instance dari sebuah class yang dituliskan tersusun secara horizontal diikuti lifeline

2.6.4 Class Diagram

Class Diagram adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan saling berhubungan secara diagram ini menggambarkan alur struktur statis dari sebuah sistem. Karena itu *Class Diagram* merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML.

Tabel 2.6 Komponen *Class Diagram*

Simbol	Nama Elemen	Keterangan
	<i>Class</i>	Simbol untuk membangun sebuah pemrograman dengan objek. Terdiri 3 bagian, bagian atas adalah nama kelas, bagian tengah adalah atribut dan bagian bawah adalah metode dari kelas tersebut.
	Generalisasi	Simbol yang menandakan adanya generalisasi dari kelas input untuk menghasilkan data yang dibutuhkan
	Asosiasi berarah / directed association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Kebergantungan /dependency	Relasi antar kelas dengan makna ketergantungan antar kelas
	Agresi /aggregation	Relasi antar kelas dengan makna semua – bagian (<i>whole-part</i>)

2.7 Database

Menurut (Kadir, 2009) menguraikan Sebuah Basis Data adalah sebuah struktur yang umumnya dikategorikan dalam hal Sebuah Basis Data flat dan sebuah Basis Data relasional. Basis data relasional lebih disukai karena lebih masuk akal dibandingkan Basis data flat. Ada tabel-tabel yang menyimpan data. Setiap tabel terdiri dari kolom dan baris. Sebuah kolom mendefinisikan jenis informasi apa yang

akan disimpan. Diperlukan kolom khusus untuk setiap jenis informasi yang ingin di simpan.

2.8 HTML5

HTML kode-kode tag yang mengintruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Menurut (Sibero, 2013) “Hypertext Markup Language HTML adalah bahasa yang digunakan pada dokumen web sebagai bahasa untuk pertukaran dokumen web mobile”. Sedangkan menurut (Bekti, 2015) “HTML5 merupakan bahasa pemograman yang bebas yang tidak dimiliki oleh siapapun, pengembangan dilakukan oleh banyak orang di berbagai Negara dan biasanya dikatakan sebagai sebuah bahasa yang dikembangkan bersama-sama secara global”.

2.9 JQuery

Jquery sebuah library yang dibangun dengan menggunakan javascript untuk menyederhanakan perintah-perintah umum, jquery jauh lebih populer karena kemampuannya untuk menjalankan perintah pada peramban lama.

Menurut “JQuery adalah salah satu javascript framework terbaik saat ini”. Selain itu, menurut Bekti (2015) “jQuery merupakan suatu library JavaScript yang memungkinkan anda untuk membuat program web pada suatu halaman web tanpa harus menambahkan event atau pun property pada halaman web tersebut”. Jadi dapat disimpulkan bahwa JQuery ialah sebuah framework yang digunakan untuk membuat halaman pada program web.