

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Chusna (2017) dengan judul rancang bangun sistem penjadwalan guru mengajar berbasis *web*, menyimpulkan bahwa untuk menyusun jadwal yang baik, penjadwal perlu menjaga agar jadwalnya sedemikian rupa sehingga tidak ada konflik atau kesamaan jam antara satu guru dengan yang lain. Perbedaan penelitian ini dengan penelitian Chusna yaitu penelitian Chusna membuat sistem penjadwalan guru mengajar, sedangkan penelitian ini untuk pergantian guru sementara mengajar.

Berdasarkan penelitian yang dilakukan oleh Sunoto (2019) dengan judul perancangan aplikasi penjadwalan mengajar guru di SMK Adi Luhur 2 Jakarta, menyimpulkan bahwa sistem pengaturan jadwal mengajar guru dibuat dengan menggunakan bahasa pemrograman java dan *database* MySQL. Pembuatan dilakukan melalui beberapa tahapan, yaitu tahap pengembangan sistem, rancangan awal, rancangan sistem, dan analisis sistem. Perbedaan penelitian ini dengan penelitian Sunoto yaitu penelitian Sunoto membuat sistem penjadwalan guru mengajar, sedangkan penelitian ini untuk pergantian guru sementara mengajar.

Berdasarkan penelitian yang dilakukan oleh Multazam (2017) dengan sistem informasi jadwal mengajar berbasis *web* pada TPQ Al Ikhlas Ampenan, menyimpulkan bahwa sistem diharapkan yang dapat memberikan kemudahan bagi admin dalam mengelola data jadwal mengajar. Dengan sistem informasi ini, santri

dan guru dapat melihat jadwal mengajar melalui *website*. Perbedaan penelitian ini dengan penelitian Multazam yaitu penelitian Multazam membuat sistem informasi jadwal mengajar, sedangkan penelitian ini untuk pergantian guru sementara mengajar.

Berdasarkan penelitian yang dilakukan oleh Mandala, dkk (2015) dengan aplikasi penentuan jadwal mengajar otomatis untuk guru di SMP Negeri 6 Bogor, menyimpulkan bahwa dengan dibangunnya aplikasi penentuan jadwal mengajar berbasis *web* untuk membantu dalam penentuan jadwal mengajar agar tidak terjadinya jadwal yang saling bertubrukan. Perbedaan penelitian ini dengan penelitian Mandala, dkk yaitu penelitian Mandala, dkk membuat aplikasi penentuan jadwal mengajar otomatis untuk guru sedangkan penelitian ini untuk pergantian guru sementara mengajar.

Berdasarkan penelitian yang dilakukan oleh Sari, dkk (2019) dengan sistem informasi pengolahan data guru mengajar berbasis *web*, menyimpulkan bahwa penggunaan komputer dengan aplikasi akan sangat membantu dalam pengolahan data, baik dari segi waktu ketepatan dan hasil yang baik, karena *user* dapat berinteraksi langsung dengan data yang akan diolah. Perbedaan penelitian ini dengan penelitian Sari, dkk yaitu penelitian Sari, dkk membuat sistem informasi pengolahan data guru mengajar, sedangkan penelitian ini untuk pergantian guru sementara mengajar.

II.2. Landasan Teori

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang diambil dari beberapa jurnal:

II.2.1. Perancangan

Perancangan adalah proses penerapan berbagai teknik dan prinsip untuk mendefinisikan sebuah peralatan, satu proses atau satu sistem secara detail yang membolehkan dilakukan realisasi fisik. Tujuan dari perancangan adalah untuk memenuhi kebutuhan pemakai sistem (*user*), memberikan gambaran yang jelas bagi pemrogram komputer dan ahli teknik lain yang relevan dan menghasilkan rancangan yang lengkap dalam pengembangan atau pembuatan sistem. (Adiyanti, dkk, 2021: 11).

Perancangan merupakan dasar utama dalam membuat aplikasi, dengan tujuan memberikan gambaran lengkap dengan jelas kepada programmer tentang aplikasi yang akan dibuat. perancangan atau rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. (Rivanthio, 2020: 188).

II.2.2. Aplikasi

Aplikasi merupakan suatu program komputer yang dapat digunakan pada berbagai perangkat elektronik, terutama pada komputer/laptop atau telepon seluler yang berbasis *android*. (Efendi, dkk, 2020: 341).

Komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data. (Nugraha, dkk, 2020: 138).

II.2.3. Guru

Guru adalah sebutan untuk jabatan, posisi, dan profesi bagi seseorang yang mengabdikan dirinya pada bidang pendidikan melalui interaksi edukatif yang teratur, formal, dan sistematis. Dalam UU R.I. Nomor 14 Tahun 2005 tentang guru dan dosen pada bab I pasal 1 dinyatakan bahwa: Guru merupakan pendidik profesional dengan tugas utama yaitu mendidik, mengajar, membimbing, mengarahkan, melatih, menilai, dan mengevaluasi peserta didik pada pendidikan anak usia dini, pendidikan dasar, dan pendidikan menengah. (Sari, dkk, 2019: 530).

Guru merupakan sosok yang mempunyai peran penting dalam proses pembelajaran untuk meningkatkan perkembangan anak sesuai dengan kemampuannya. Guru merupakan fasilitator dalam pembelajaran untuk menunjang perkembangan anak, apalagi pada masa pandemi Covid-19 guru dituntut harus mampu menunjukkan kompetensi guru dalam membimbing anak. Pada masa pandemi ini sesuai dengan instruksi Menteri Pendidikan dan Kebudayaan sekolah harus mampu melakukan pembelajaran secara daring/*online*, oleh karenanya baik guru maupun peserta didik harus mampu melakukan pembelajaran daring/*online*. Guru harus tetap melaksanakan tugasnya dalam mengajar dan mendidik anak meskipun tanpa harus bertatap muka langsung

dengan peserta didik. Maka sangat diperlukan peran guru dalam menunjang proses pembelajaran secara daring/*online* agar pada masa pandemi Covid-19 proses belajar anak tidak menjadi terbengkalai dan mereka tetap bisa belajar dengan senang tanpa ada rasa beban dalam proses belajarnya. (Sukitman, dkk, 2020:91).

II.2.4. Ganti Sementara

Menurut Kamus Besar Bahasa Indonesia (KBBI), arti kata ganti adalah sesuatu yang menjadi penukar yang tidak ada atau hilang, seperti sulih, pampas. Contoh: Santan ini dapat dipakai akan ganti susu. Arti lainnya dari ganti adalah orang yang menggantikan pekerjaan, jabatan, dan sebagainya. Jadi dapat disimpulkan ganti sementara adalah menggantikan pekerjaan untuk sementara waktu hingga waktu yang telah ditentukan. (kbbi.lektur.id).

II.2.5. SMP Wiraswasta Batang Kuis

Sekolah Menengah Pertama (SMP) Wiraswasta Batang Kuis merupakan lembaga pendidikan yang dikembangkan sesuai konteks Manajemen Berbasis Sekolah (MBS) dengan penerapan Kurikulum Tingkat Satuan Pendidikan (KTSP) dan Kurikulum 2013. SMP Wiraswasta Batang Kuis saat ini dipimpin oleh Elfida SE terus melakukan pembenahan dan penataan guna memberikan layanan pendidikan yang berkualitas bagi masyarakat. Pembenahan dan penataan tersebut mencakup penyempurnaan sarana dan prasarana. (Fadillah, dkk, 2018: 29).

II.2.6. *Web*

Web adalah sebuah sistem yang berisi beragam informasi baik berupa teks, gambar, audio maupun video yang dapat diakses melalui *web browser*. *Web* merupakan kumpulan informasi baik yang bersifat statis maupun dinamis yang terdiri dari halaman yang dibuat. (Oktasari dan Kurniadi, 2019: 151).

Web merupakan salah satu sumber daya internet yang berkembang pesat. Pendistribusian informasi *Web* dilakukan melalui pendekatan *hyperlink*, yang memungkinkan suatu teks, gambar, ataupun objek yang lain menjadi acuan untuk membuka halaman-halaman yang lain. Melalui pendekatan ini, seseorang dapat memperoleh informasi dengan beranjak dari satu halaman ke halaman lain. Pada awalnya aplikasi *Web* dibangun hanya dengan menggunakan bahasa yang disebut HTML. Pada perkembangan berikutnya, sejumlah *script* dan objek yang dikembangkan untuk memperluas kemampuan HTML. Pada saat ini *script* seperti itu antara lain: PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah *applet* (Java). (Mahayadi, dkk, 2017:22).

II.2.7. *Android*

Android merupakan sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam perangkat. Pemrograman *android* membutuhkan *android SDK* (*Software Development Kit*) dan juga *Java Development Kit*. Setelah itu dibutuhkan *Java IDE* yang digunakan untuk menuliskan *coding android*. Salah satu *tools* yang

dapat digunakan dalam pemrograman *android* adalah *Eclipse Helios* karena *Eclipse* sangat mendukung proses penulisan *code android* dan menjadikan proses *development android* menjadi lebih mudah. (Efendi, dkk, 2020: 344).

Dalam perjalanannya *android* telah merilis banyak versi tentunya dengan berbagai macam fitur dan perbaikan di tiap versi yang baru. Berikut versi-versi *android* yang telah dirilis hingga saat ini:

1. *Android 1.1*
2. *Android 1.5 (Cupcake)*
3. *Android 1.6 (Donut)*
4. *Android 2.0-2.1 (Eclair)*
5. *Android 2.2.3 (Froyo)*
6. *Android 2.3-2.3.7 (Gingerbread)*
7. *Android 3.0-3.2.6 (Honeycomb)*
8. *Android 4.0-4.0.4 (Ice Cream Sandwich)*
9. *Android 4.1-4.3.1 (Jelly Bean)*
10. *Android 4.4 (KitKat)*
11. *Android 5.0 (Lollipop)*
12. *Android 6.0 (Marshmallow)*
13. *Android 7.0 (Nougat)*
14. *Android 8.0 (Oreo)*
15. *Android 9.0 (Pie)*. (Nugraha, dkk, 2020: 138).

II.2.8. *Hyper Text Markup Language (HTML)*

Hypertext Markup Language (HTML) memungkinkan bagi *user* untuk membuat dan menyusun paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk membuat halaman *web* dan aplikasi. HTML bukanlah bahasa pemrograman yang berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti Microsoft Word. (Sugijanto, dkk, 2020: 2).

Definisi HTML merupakan bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui *tag-tag* yang ditulis dalam HTML. (Rahmasari, 2019: 415).

II.2.9. *Hypertext Preprocessor (PHP)*

PHP merupakan singkatan dari “*Hypertext Preprocessor*”. PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrograman C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan *web* menjadi lebih dinamis, dengan PHP kita bisa menampilkan atau menjalankan beberapa *file* dalam 1 *file* dengan cara di *include* dan *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle. (Rahmasari, 2019: 415).

Sejarah PHP pada awalnya merupakan singkatan dari *Personal Home*

Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada tahun itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script*/program tersebut akan dijalankan/diproses oleh *server*. (Sugijanto, dkk, 2020: 2).

II.2.10. Java

Java adalah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan 1990-an. Menurut definisi Sun, java adalah nama untuk seperangkat teknologi yang digunakan untuk membuat dan menjalankan perangkat lunak pada komputer mandiri atau dalam lingkungan jaringan. keunggulan java dari bahasa pemrograman lain adalah dapat berjalan diberbagai sistem operasi sehingga dikenal sebagai bahasa pemrograman *multiplatform*, bersifat pemrograman berorientasi objek (PBO), memiliki *library* yang lengkap. (Hasmia, dkk, 2022).

Berdasarkan definisi java dapat disimpulkan bahwa Java merupakan suatu bahasa pemrograman yang dapat dijalankan di berbagai sistem operasi sehingga dikenal sebagai bahasa pemrograman *multiplatform*.

II.2.11. Database

Basis data atau *database* adalah kumpulan data terstruktur. Agar dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam *database* komputer, dibutuhkan sistem manajemen basis data (*database management*

system). Dalam pengembangan perangkat lunak tradisional yang memanfaatkan pemrosesan file, setiap kelompok pengguna menyimpan arsip-arsipnya sendiri untuk menangani aplikasi pengolahan datanya masing-masing. Hal ini mengakibatkan adanya kerangkapan data atau disebut dengan *redundancy*. (Arfianto dan Nugrahanti, 2018 : 175).

II.2.12. MYSQL

MySQL adalah *database open source* terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, MySQL telah menjadi pilihan *database* terdepan untuk aplikasi berbasis *web*, yang digunakan oleh properti *web* profil tinggi termasuk Facebook, Twitter, YouTube, Yahoo! dan banyak lagi. Kata “SQL” dari “MySQL” adalah singkatan dari “*Structured Query Language*”. SQL adalah bahasa standar yang paling umum digunakan untuk mengakses database. Bergantung pada lingkungan pemrograman yang kita pakai, kita bisa memasukkan SQL secara langsung (misalnya, untuk *men-generate* laporan), memasukkan pernyataan SQL ke dalam kode yang ditulis dalam bahasa lain, atau menggunakan API khusus yang dapat menyembunyikan sintaks SQL. (Arfianto dan Nugrahanti, 2018 : 175).

II.2.13. Unified Modeling Language (UML)

UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti

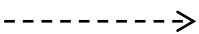
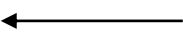
sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019: 39).

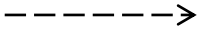
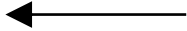
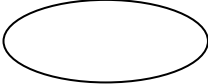
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem *use case* diagram dapat digambarkan dengan sumber-sumber pada Tabel II.1.

Tabel II.1. *Use Case Diagram*

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang berinteraksi dengan <i>use case</i> .
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

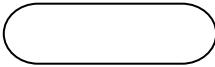
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber: Andikos, 2019: 39)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

Tabel II.2. Simbol *Activity Diagram*

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan.
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

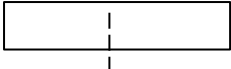
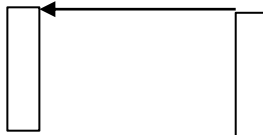
(Sumber: Andikos, 2019: 39)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang

digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

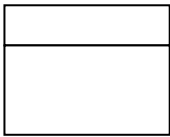
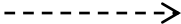
(Sumber: Andikos, 2019: 39)

4. Diagram Kelas (*Class Diagram*)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *Class Diagram*

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek

		yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Assocation</i>	menghubungkan antara objek satu dengan objek lainnya

(Sumber: Andikos, 2019: 39)