

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian yang pernah dilakukan terkait dengan penelitian yang dilakukan, dapat dilihat pada tabel 2.1:

Tabel 2. 1 Tinjauan Pustaka

No. 1	Akbar and Susafa'ati (2019)
Judul	Penerapan Metode Waterfall Untuk Perancangan Sistem Informasi Kehadiran Siswa Berbasis Web dan SMS GateWay Studi Kasus SMA IP Yakin Jakarta
Jurnal	JTIS
Volume dan Halaman	Volume 2 Nomor 3, Hal 89-94
Tahun	2019
Penulis	Akbar and Susafa'ati
Identifikasi Masalah	Masih menggunakan sistem absensi secara manual, berupa pencatatan di kertas, yaitu menggunakan buku absensi kehadiran siswa sehingga sering terjadinya kesalahan dalam proses rekap absensi siswa sehingga memperlambat proses pembuatan laporan evaluasi pembelaaran atau rapor untuk para siswa. Selain
Metode / Tools	<i>Waterfall</i> dan UML
Hasil Penelitian	Hasil penelitian yaitu aplikasi kehadiran siswa ini mempermudah guru untuk melakukan proses perhitungan dan pembuatan laporan absensi siswa dan untuk para orang tua aplikasi ini dapat digunakan untuk mengawasi atau memonitoring kehadiran siswa selama di sekolah dan juga proses perancangan menggunakan metode waterfall ini membantu dalam proses mengurutkan perkerjaaan yang harus diselesaikan akan tetapi dalam imlementasinya membutuhkan waktu yang cukup lama.
No. 2	Napitu et al (2020)
Judul	Perancangan Sistem Absensi Berbasis Web pada Program Studi PTI UNIMUDA Sorong
Jurnal	Jurnal PETISI
Volume dan Halaman	Vol. 1, No. 2 Hal 7-12
Tahun	2020
Penulis	Napitu et al
Identifikasi Masalah	Masih dilakukan secara manual
Metode / Tools	<i>Waterfall</i> dan UML
Hasil Penelitian	Menghasilkan produk berupa sistem absensi berbasis web pada Prodi PTI FKIP UNIMUDA Sorong.
No. 3	Zebua and Pristiwanto (2017)

Judul	Pembangunan Web Mobile Absensi Mahasiswa Pada Platform Android Yang Terintegrasi Dengan Website Utama Sistem Informasi Akademik (SIA) STMIK Budi Darma
Jurnal	Jurnal Teknik Informatika Unika
Volume dan Halaman	Volume 02 Nomor 01, Hal 100-107
Tahun	2017
Penulis	Zebua and Pristiwanto
Identifikasi Masalah	Kesalahan penginputan kehadiran mahasiswa ke dalam sistem yang disajikan pada Daftar Nilai Ujian Akhir sangat berpengaruh pada syarat keikutsertaan mahasiswa pada ujian akhir semester maupun pada penetapan nilai ujian akhir mahasiswa pada Kartu Hasil Studi (KHS) oleh dosen pengampu matakuliah, sehingga hingga saat ini complain nilai akhir ujian semester oleh mahasiswa sering diterima oleh program studi maupun dosen pengampu matakuliah.
Metode / Tools	Metode <i>Object Oriented Analysis and Design</i> dan UML
Hasil Penelitian	Aplikasi Web mobile absensi kehadiran mahasiswa sangat optimal dalam meningkatkan validitas pengolahan data absensi kehadiran mahasiswa serta menghemat waktu dan tenaga petugas dalam melakukan rekapitulasi ulang seperti yang masih dilakukan selama ini oleh pihak prodi STMIK Budidarma Medan
No. 4	Octa and Amrullah (2019)
Judul	Penerapan Sistem Informasi Presensi Mahasiswa Menggunakan Fingerprint Berbasis Web
Jurnal	Journal of Computer Science and Applied Informatics
Volume dan Halaman	Vol. 1, No. 1, Agustus 2019, Halaman 7-10
Tahun	2019
Penulis	Octa and Amrullah
Identifikasi Masalah	Masih dicatat kedalam buku absensi
Metode / Tools	SDLC dan UML
Hasil Penelitian	Sistem ini dapat dikembangkan lagi untuk keperluan selanjutnya dapat ditambahkan data berupa jadwal kuliah dan data absen dosen maupun staff karyawan yang ada di Fakultas Teknologi Informasi.
No. 5	Wibowo (2018)
Judul	Analisis Dan Perancangan Sistem Informasi Presensi Siswa Berbasis Android Pada Sd Dan Smp Kanaan Global School Jambi
Jurnal	Jurnal Manajemen Sistem Informasi
Volume dan Halaman	Vol.3, No.2, Hal 1093-1105
Tahun	2018
Penulis	Wibowo
Identifikasi Masalah	SD dan SMP Kanaan Global School Jambi saat ini menjalankan prosesnya dengan cara siswa/i melakukan

	presensi menggunakan sidik jari yang telah terdaftar pada fingerprint. Namun ketika ada siswa/i yang terlambat maka ada salah satu guru yang berdiri di sebelah fingerprint untuk mencatat siswa/i yang terlambat tersebut di daftar hadir siswa. Selain
Metode / Tools	<i>Prototype</i>
Hasil Penelitian	Hasil penelitian yaitu sistem yang dapat memberikan kemudahan dalam proses rekapitulasi data presensi siswa/i serta memberikan informasi presensi untuk mengetahui jumlah/tingkat kehadiran, sakit, izin, alpa dan keterlambatan siswa/i di sekolah secara realtime, lengkap, dan faktual

Berdasarkan tinjauan pustaka diatas maka perbedaan antara penelitian terdahulu dengan penelitian yang dilakukan adalah :

1. Sistem yang dibangun menambahkan menu *chating whatsapp*
2. Metode pengembangan sistem menggunakan *prototype*
3. Metode perancangan menggunakan UML
4. Aplikasi yang digunakan adalah *Sublime Text*
5. Laporan yang dihasilkan adalah laporan presensi dan akan menampilkan grafik presensi siswa.

2.2. Rancang Bangun

Rancang Bangun adalah tahapan dalam menggambarkan atau pembuatan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada (Tohari, 2017).

2.3. Pengertian Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan (Tohari, 2017)

Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware dan software yang saling berinteraksi sebagai kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama (Maniah and Hamidin, 2017)

Dari beberapa kutipan di atas maka penulis dapat menyimpulkan bahwa sistem informasi adalah sistem di dalam suatu instansi atau organisasi perusahaan yang mempertemukan kebutuhan pengolahan transaksi harian dan memberikan laporan-laporan atau informasi yang dibutuhkan.

2.4. Pengertian Informasi

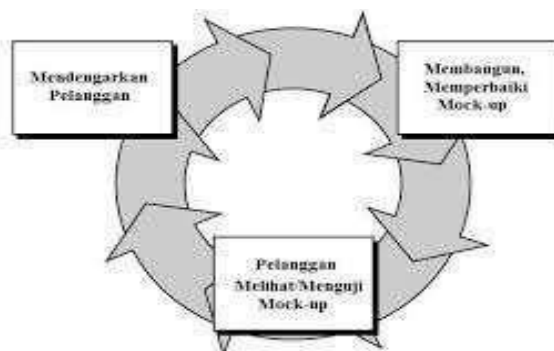
Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi (Kristanto, 2018).

2.5. Sistem Informasi

Sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi (Kristanto, 2018)

2.6. Metode *Prototype*

Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa and Shalahudin, 2018)



Gambar 2. 1 Ilustrasi model prototipe.

(Sumber : Rosa A.S, M. Shalahudin)

Menurut (Rosa & Shalahuddin, 2018) terdapat tahapan dalam proses *prototipe* yaitu:

1. Mendengarkan Pelanggan

Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan pelanggan sebagai pengguna sistem perangkat lunak untuk menganalisis serta mengembangkan kebutuhan pengguna. Pada tahapan ini pelanggan dan pengembang bersama sama mendefinisikan format seluruh perangkat lunak, semua kebutuhan sistem yang akan dibangun, dan garis besar sistem yang dibuat. Dalam tahapan ini dilakukan dengan cara pengumpulan data (wawancara, observasi, dan dokumentasi) selanjutnya dilakukan analisis

kebutuhan sistem yang akan dibangun seperti analisis fungsional dan analisis non fungsional.

2. Merancang dan Membuat Prototipe

Pada tahap ini, dilakukan perancangan dan pembuatan prototipe sistem yang disesuaikan dengan kebutuhan pengguna. Pengguna membuat rancangan sementara yang berfokus dalam menggambarkan sistem untuk disajikan kepada pelanggan dengan cara membuat gambaran alur sistem menggunakan UML, penggambaran rancangan aplikasi yang akan dibuat, setelah itu tahapan membuat sistem menggunakan bahasa pemograman PHP dan MySQL sebagai *database*.

3. Uji Coba

Pada tahap ini, dilakukan pengujian *prototipe* sistem oleh pengguna kemudian dilakukan evaluasi sesuai dengan kekurangan-kekurangan dari kebutuhan pelanggan. Jika sistem sudah sesuai dengan prototipe, maka sistem akan diselesaikan sepenuhnya. Namun, jika masih tidak sesuai kembali ke tahap pertama. Dalam tahapan ini pengujian sistem dapat dilakukan menggunakan teknik pengujian sistem yaitu metode pengujian *black box* atau ISO 25010 yang bertujuan untuk melakukan evaluasi apakah sistem layak digunakan atau tidak sesuai dengan kebutuhan pengguna.

2.7. Definisi *Unified Modelling Language* (UML)

Unified Modeling Language (UML) adalah bahasa standar untuk menulis perangkat lunak dalam bentuk gambar. UML dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sebuah



sistem perangkat lunak (Rosa & Shalahuddin, 2018). Beberapa jenis diagram *UML* antara lain sebagai berikut:





2.7.1. Use Case Diagram

Use case diagram membantu anda menentukan fungsi dan fitur dari perangkat lunak. Dalam diagram ini, gambar yang menyerupai boneka kayu mewakili aktor yang berhubungan dengan kategori dari pengguna. Di dalam diagram *use case*. Para aktor terhubung oleh garis ke *use case* yang mereka kerjakan (Rosa & Shalahuddin, 2018).

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.2 di bawah ini:

Tabel 2. 2 Simbol Diagram *Use Case*

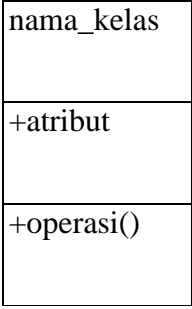




Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan 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 <i>actor</i>
Asosiasi/ <i>association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau



Simbol	Deskripsi
	<i>use case</i> memiliki interaksi dengan <i>actor</i>
Ekstensi/ <i>extend</i> << <i>extend</i> >> 	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> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
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.
Menggunakan/ <i>Include/uses</i> << <i>include</i> >> 	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

2.7.2. Class Diagram

Unsur-unsur utama dari diagram kelas adalah kotak, yang merupakan ikon yang digunakan untuk mewakili kelas dan *interface*. Setiap kotak dibagi menjadi bagian-bagian horisontal. Bagian atas berisi nama kelas. Bagian tengah berisi daftar atribut kelas (Rosa & Shalahuddin, 2018). Dan bagian bawah merupakan *operation* dari kelas tersebut. Simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.3.

Tabel 2. 3 Simbol *Class Diagram*


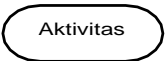


Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka/<i>Interface</i></p>  <p>nama_<i>interface</i></p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi/<i>asociation</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <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 disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)

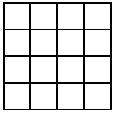


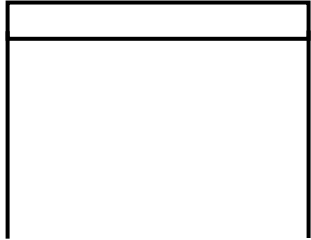
Simbol	Deskripsi
Kebergantungan/ <i>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2.7.3. Activity Diagram

Sebuah diagram *activity* menggambarkan perilaku dinamis dari sistem atau bagian dari sistem melalui aliran kontrol antara tindakan yang sistem lakukan. Hal ini mirip dengan sebuah *flowchart* kecuali bahwa suatu diagram *activity* dapat menunjukkan arus bersamaan (Rosa & Shalahuddin, 2018). Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2. 4 Simbol *Activity Diagram*

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

Simbol	Deskripsi
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.8. Definisi Database

Database adalah sekumpulan informasi yang diatur agar mudah dicari. Dalam arti umum basis data adalah sekumpulan data yang diproses dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan tepat, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi (Sheldon and Moes, 2017).

2.9. Xampp

XAMPP merupakan paket program web lengkap yang dapat Anda gunakan untuk mempelajari pemrograman web khususnya PHP dan MySQL (Sheldon and Moes, 2017).

Keunggulan Xampp adalah performa tinggi, stabil, banyak fitur, mudah dikonfigurasi, hanya menggunakan sedikit resource di server, tidak bergantung pada thread untuk melayani klien.

Kekurangan Xampp adalah tidak support IPV6, update / patch terbaru sudah lama, Fast-CGI-nya tidak berfungsi maksimal, penggunaanya tidak sebanyak aktivitas paket program lainnya.

2.10. MySQL

MySQL adalah server database relasional yang menawarkan berbagai mekanisme untuk memproses data yang dikenal sebagai mesin penyimpanan. *MySQL* berfungsi untuk mengolah *database* menggunakan bahasa *SQL*. *MySQL* bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/support dengan *database MySQL* (Sutanta, 2018)

2.11. Pengujian ISO 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang software engineering (Iqbal, 2018). Product quality ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

1. Kualitas dalam model penggunaan
2. Model kualitas produk
3. Data model kualitas

Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

- 1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk Merupakan tingkatan dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi spesifik tertentu dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna.
- 2) *Reliability* Merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik dalam hal ini perangkat lunak dapat beroperasi dan siap ketika dibutuhkan untuk digunakan dan juga dapat bertahan pada tingkat kemampuan tertentu terhadap kegagalan, kesalahan serta perangkat lunak kembali pada tingkat tertentu dalam mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak.
- 3) *Performance efficiency* Merupakan tingkatan dimana perangkat lunak dapat memberikan kinerja terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu dalam hal ini *performance efficiency* dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan

aksi dari sebuah fungsi dan perangkat lunak dapat menggunakan sejumlah sumber daya ketika melakukan aksi dari sebuah fungsi.

- 4) *Usability* Perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik pengguna ketika digunakan dalam hal ini perangkat lunak mudah dipelajari oleh pengguna, perangkat lunak dapat digunakan dan dioperasikan oleh pengguna.
- 5) *Security* Merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman atau keganjalan dalam hal ini perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna dan merupakan dari kelengkapan, ketepatan dari sejumlah *asset* yang telah dijaga sehingga aksi atau tindakan yang dilakukan telah terbukti dan hal tersebut tidak dapat ditolak.
- 6) *Compability* Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.
- 7) *Maintainability* Merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan, kebutuhan dan fungsionalitas yang spesifik. Selain itu perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak untuk mengidentifikasi bagian yang dapat dimodifikasi.
- 8) *Transferability*. Merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan satu ke lingkungan yang lain dalam hal

ini perangkat lunak dapat beradaptasi dengan cepat pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.

2.12. Skala Pengukuran

Skala pengukuran yang digunakan adalah skala Likert, skala yang didasarkan pada penjumlahan sikap responden dalam merespon pernyataan berkaitan indikator-indikator suatu konsep atau variable yang sedang diukur (Sugiyono, 2017). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). Skala Likert dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Skala Likert

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat pada Gambar 2.2.

$$\text{Persentase} = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\%$$

Gambar 2. 2 Persentase Kelayakan

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
$\geq 90\%$ -100%	Sangat Baik
$\geq 70\%$ - <90%	Baik
$\geq 70\%$ - <80%	Cukup Baik
$\geq 60\%$ - <70%	Kurang Baik
0%-60%	Tidak Baik