

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka dari penelitian sebelumnya digunakan untuk mendukung penelitian. Seperti yang digunakan dari daftar literature yang ada pada Tabel 2.1:

Tabel 2.1 Daftar Literature

No	Nama	Tahun	Judul
1.	Moonbeom Kim, Jongho Lee dan Jeongyeup Paek.	2018	Neutralizing BLE Beacon-Based Electronic Attendance System Using Signal Imitation Attack
2.	Siti Nur Azizah	2019	Implementasi Sistem Absensi Pegawai Menggunakan Mac Address Smartphone Dengan Sensor <i>Bluetooth</i> Berbasis Mikrokontroler Arduino
3.	A.Komarudin, Hariyadi Singgih, Siswoko	2019	Kajian Rancang-Bangun Alat Fingerprint Presensi Security Menggunakan Basis Komunikasi <i>Bluetooth</i>
4.	Nesrine A. Azim , Aisha Mohamed Hussein , Hanan Ibrahim Shawky, dan Heba Ahmed Joudah	2020	<i>Bluetooth</i> Smart based Automated Attendance System
5.	I Gusti Ngurah Alit Indrawan, dan Made Agung Raharja	2020	Implementasi Sistem Absensi Mahasiswa Menggunakan <i>Bluetooth</i> Pada Program Studi Teknik Informatika FMIPA Universitas Udayana

2.1.1 Tinjauan Literature 1

1. Masalah

Permasalahan yang terdapat dari penelitian ini yaitu sistem absensi masih manual dengan cara memanggil nama mahasiswa satu persatu karena hal itu cenderung menghabiskan banyak waktu.

2. Metode

Metode yang digunakan dalam penelitian ini yaitu *BLE beacon-based electronic attendance system* (BEAS).

3. Kesimpulan

Membangun beberapa perangkat *beacon* BLE (PBB) dan *beacon signal forwarder* (BSF) yang dapat diprogram yang dapat meniru, meneruskan, dan memutar ulang sinyal beacon yang sah di lokasi terpencil. Setelah percobaan dapat dikategorikan berbagai kerentanan yang dimiliki sistem kehadiran, mempresentasikan kemungkinan tindakan balasan untuk masalah tersebut, dan membahas tindakan balasan tambahan yang dapat melucuti tindakan balasan tersebut.

2.1.2 Tinjauan literature 2

1. Permasalahan

Permasalahan yang terdapat dari penelitian ini yaitu sistem absensi masih memiliki banyak kelemahan.

2. Metode

Metode yang digunakan yaitu melihat kebutuhan sistem, *flowchart*, rangkaian elektronika, dan diagram blok.

3. Kesimpulan

Berdasarkan pengujian dan implementasi sistem absensi karyawan dengan menggunakan alamat *MAC Bluetooth* berbasis *mikrokontroler Arduino*, dapat disimpulkan bahwa perangkat komunikasi seluler dapat dihubungkan melalui koneksi *bluetooth* dan mengirimkan perintah ke modul perangkat keras, serta membangkitkan karyawan. aplikasi absensi.

2.1.3 Tinjauan Literature 3

1. Permasalahan

Permasalahannya yaitu sistem absensi masih menggunakan cara yang manual sehingga kurang optimal dalam penggunaannya.

2. Metode

Metode penelitian yang digunakan yaitu membuat sistem diagram, prinsip kerja sistem, Perancangan *Fingerprint SM-630*, Perancangan *RTC (Real Time Clock)*, Perancangan *Hardware LCD*, Perancangan Perangkat Lunak,

3. Kesimpulan

Kesimpulannya yaitu sistem identifikasi keamanan sidik jari komputer server yang menggunakan modul *Bluetooth HC-05* dapat menjangkau jarak 9 meter. Jika komputer / perangkat server tidak berada dalam jangkauan *Bluetooth*, koneksi *Bluetooth* akan diputuskan secara otomatis. Hasil pengolahan sidik jari ditampilkan pada aplikasi monitor serial *Arduino IDE*. Tunjukkan keberadaan petugas keamanan. Data yang diterima komputer server akan masuk ke serial monitor di software *Arduino IDE*. Selain itu, data komputer juga bisa dikirim ke hp *android* menggunakan aplikasi serial *bluetooth*, yang bisa di download di *playstore*.

2.1.4 Tinjauan literature 4

1. Permasalahan

Permasalahan dalam penelitian ini adalah metode absensi manual perlu diganti dengan sistem absensi otomatis.

2. Metode

Metode yang digunakan ialah RFID (*Radio Frequency Identification*).

3. Kesimpulan

Aplikasi absensi ini mengurangi waktu yang dihabiskan untuk absensi manual, menghilangkan kesalahan manusia, dan menyediakan informasi statistik yang diperlukan administrator untuk keputusan manajemen lebih lanjut.

2.1.5 Tinjauan Literature 5

1. Permasalahan

Permasalahan dalam penelitian ini yaitu Sistem absensi merupakan salah satu hal yang utama khususnya dalam suatu kehadiran. Penggunaan hardware pada berbagai aktivitas kerja yang di dukung oleh berbagai macam software sebagai penunjang kebutuhan user yang sangat menjanjikan terutama untuk pengabsensian.

2. Metode

Metode penelitian yang digunakan adalah *Android, Php, Mysql, Phpmyadmin, Bluetooth, Bluetooth Dongle, Use Case, UML (Unified Modelling Language), Retrofit, Aplikasi*.

3. Kesimpulan

Kesimpulan penelitian adalah penulis memiliki pemahaman yang benar tentang situasi dunia kerja sehingga dapat mempersiapkan diri untuk menghadapi

persaingan dunia kerja di masa depan. Dengan berkembangnya sistem perencanaan pembelajaran maka penyebaran informasi terkait rencana pembelajaran menjadi lebih mudah, efisien dan efektif.

2.2 Teknologi

Secara universal, penafsiran teknologi ialah ilmu pengetahuan yang menekuni menimpa keahlian dalam menghasilkan perlengkapan, tata cara pengolahan, dan ekstraksi barang, dalam menolong menuntaskan bermacam kasus dan pula pekerjaan manusia tiap hari.

Secara etimologis, kata “ teknologi” ini berasal dari bahasa Yunani, yaitu “ *technologia*” yang mana kata tech ini berarti kemampuan dan logia berarti pengetahuan. Tadinya arti teknologi ini cuma terbatas pada benda- benda yang memiliki bentuk, misalnya mesin dan perlengkapan. Tetapi arti teknologi ini hadapi ekspansi dan tidak cuma terbatas pada barang berwujud saja tetapi pula barang yang tidak berwujud, misalnya tata cara, ilmu pengetahuan, aplikasi, dan lain- lain. Sehingga penafsiran teknologi ini yakni sesuatu metode, proses, perlengkapan, mesin, aktivitas maupun gagasan yang terbuat buat bisa memudahkan seluruh berbagai aktivitas manusia (Ibeng, 2020).

2.2.1 Klasifikasi Teknologi

Berikut ini klasifikasi teknologi menurut terbagi menjadi dua (Furqon, 2008) yaitu :

1. Klasifikasi Teknologi secara *universal* dipecah jadi 4 ialah:
 - a) Teknologi yang tercantum dalam perlengkapan
 - b) Teknologi yang tercantum dalam produk

- c) Teknologi yang tercantum dalam proses
 - d) Teknologi yang tercantum dalam penciptaan
2. Klasifikasi Teknologi bagi Khalil
- a) *New Technology*: teknologi yang baru diperkenalkan ataupun diimplementasikan(berakibat pada jalur sesuatu industri dalam memproduksi produk ataupun sediakan servis/ layanan).
 - b) *Emerging Technology*: teknologi yang bukan cuma dikomersilkan tetapi hendak sangat diperlukan buat 5 tahun kedepan.
 - c) *High Technology*: teknologi yang mengacu ke suatu yang maju ataupun teknologi yang rumit *Low Technology*: teknologi yang mengacu pada teknologi yang telah terdapat di sebagian besar segmen kehidupan sosial.
 - d) *Medium Technology*: teknologi yang mengacu pada teknologi yang mulai berusia.
 - e) *Appropriate Technology*(Teknologi yang cocok): teknologi yang digunakan buat melaksanakan perbandingan yang baik antara digunakannya teknologi ataupun sumber energi yang digunakan buat pemakaian yang optimal.

2.3 Bluetooth

Menurut (Diariono, 2009) *Bluetooth* merupakan suatu teknologi komunikasi *wireless* (tanpa kabel) yang diciptakan bukan hanya untuk mengambil alih ataupun melenyapkan pemakaian kabel di dalam melaksanakan pertukaran data, namun pula sanggup menawarkan fitur yang baik buat teknologi mobile *wireless*

dengan bayaran yang relatif rendah, *interoperability* yang menjanjikan, gampang dalam pengoperasian serta sanggup buat menyediakan berbagai berbagai layanan.

2.4 Sistem

Penafsiran sistem bagi (Romney dan Steinbart, 2015) Sistem merupakan rangkaian dari dua ataupun lebih komponen – komponen yang saling berhubungan, yang berhubungan buat menggapai sesuatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil menunjang sistem yang lebih besar.

2.5 Absensi

Absensi merupakan catatan kedatangan pegawai/ siswa, yang berisi jam datang, jam kembali, dan alibi/ penjelasan kedatangan pegawai. Absensi merupakan sesuatu metode buat mengenali sepanjang mana tingkatan kedisiplinan kedatangan pegawai dalam bekerja. (Putri dan Soeliman, 2017).

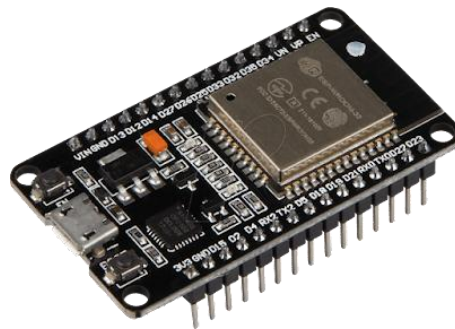
Di samping itu absensi ini bisa pula jadi data tentang gimana kedisiplinan karyawan yang bersangkutan. Berarti terdapatnya untuk pihak perusahaan buat mengenali kedatangan karyawannya, baik yang kerap tiba pas waktu ataupun yang tiba terlambat. Penggajian memegang peranan berarti dalam menantikan pendapatan seseorang karyawan. Sistem penggajian disetiap seluruh industri mempunyai konsep yang sama, dengan menghubungkan absensi, hingga tiap bulannya (Harumy, T.H.F., Julham Sitorus, 2018).

2.6 ESP32

ESP32 dibuat oleh Espressif Systems, ESP32 adalah sistem dengan biaya yang rendah, berdaya rendah pada seri chip (SoC) dengan Wi-Fi & kemampuan Bluetooth dua mode Keluarga ESP32 termasuk chip ESP32-D0WDQ6 (dan

ESP32-D0WD), ESP32-D2WD, ESP32-S0WD, dan sistem dalam paket (SiP) ESP32-PICO-D4. Pada intinya, ada mikroprosesor Tensilica Xtensa LX6 dual-core atau single-core dengan clock rate hingga 240 MHz. ESP32 sudah terintegrasi dengan built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. Didesain untuk perangkat seluler, perangkat elektronik yang dapat dipakai, dan aplikasi IoT, ESP32 juga bekerja dengan konsumsi daya sangat rendah melalui fitur hemat daya termasuk fine resolution clock gating, multiple power modes, and dynamic power scaling.

Module ESP32 merupakan penerus dari module ESP8266 yang cukup populer untuk Aplikasi IoT. Pada ESP32 terdapat inti CPU serta Wi-Fi yang lebih cepat, GPIO yang lebih, dan mendukung *Bluetooth Low Energy* (Prastyo, 2019). Modul ESP32 yang digunakan tampak seperti Gambar 2.1 dibawah ini.



Gambar 2.1 *Bluetooth Low Energy*

2.7 Arduino Ide

Arduino IDE (*Integrated Development Environment*) adalah software yang di gunakan untuk memprogram di arduino, dengan kata lain Arduino IDE sebagai media untuk memprogram board Arduino. Arduino IDE bisa di download secara

gratis di website resmi Arduino IDE. Arduino IDE ini berguna sebagai text editor untuk membuat, mengedit, dan juga mevalidasi kode program. bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “sketch” atau disebut juga source code arduino, dengan ekstensi file source code .ino (Software Info, 2017).

2.8 PHP

Pada awalnya PHP merupakan kependekan dari *Personal Home Page*. PHP pertama kali dibuat pada tahun 1995 oleh Rasmus Lerdorf. Pada waktu itu PHP bernama FI(*Form Interpreted*). Pada kala itu PHP merupakan sekumpulan script yang digunakan buat mencerna informasi form dari website. PHP merupakan bahasa *open source* yang bisa digunakan di bermacam mesin (*linux, unix, windows*) serta bisa dijalankan secara runtime lewat *console* dan pula pada PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman server side yang telah banyak digunakan pada dikala ini, paling utama buat pembuatan web dinamis. Buat hal- hal tertentu dalam pembuatan *website*, bahasa pemrograman PHP memanglah dibutuhkan, misalnya saja buat memproses informasi yang dikirimkan oleh wisatawan website. (Syahreza, 2015).

2.9 HTML

Hypertext Markup Language (HTML) adalah bahasa pemrograman yang digunakan untuk menampilkan situs web. HTML termasuk dalam bahasa pemograman gratis yang artinya tidak dimiliki oleh siapapun, perkembangannya dilakukan oleh banyak orang di banyak negara, dapat dikatakan sebagai bahasa yang dikembangkan bersama di seluruh dunia. Dokumen HTML adalah dokumen

teks yang dapat diedit oleh editor teks apa pun. Dan simpan dengan file berekstensi .html. Dokumen HTML memiliki beberapa elemen yang dikelilingi oleh tag teks, yang dimulai dengan simbol "<" dan diakhiri dengan simbol ">".(Sari and Abdilah, 2015).

2.10 Framework Laravel

Menurut Betha Sidik (2012) *framework* adalah "Kumpulan instruksi yang dikumpulkan dalam kelas dan fungsi, masing-masing fungsi memudahkan pengembang untuk memanggil mereka tanpa harus menulis sintaks program yang sama berulang kali, dan dapat menghemat waktu" .

Laravel adalah kerangka kerja PHP berbasis web dan sumber terbuka gratis yang dibuat oleh Taylor Otwell, dirancang untuk mengembangkan aplikasi web yang mengikuti Model-View-Controller (MVC) atau pola arsitektur. Beberapa fitur Laravel adalah pengembangan sistem modul yang dapat dikelola yang memperkenalkan berbagai cara mengakses database relasional, utilitas yang memfasilitasi penerapan aplikasi dan perawatan yang mudah.

Di laravel ada banyak fungsi untuk berinteraksi dengan database, seperti mengambil semua baris, mengambil baris menggunakan kata kunci utama, memfilter baris menggunakan klausa, melakukan penyisipan, melakukan pembaruan, dll. Oleh karena itu, kebutuhan untuk membangun sistem berbasis web dapat dengan mudah dipenuhi. Pada awal Maret 2015, Laravel telah menjadi salah satu framework PHP paling populer, bersama Symfony2, Nette, CodeIgniter, dan Yii2 (Firma Sahrul B, 2017).

2.11 Mysql

Unified Modelling Language (UML) merupakan suatu "bahasa" yg sudah jadi standar dalam industri buat visualisasi, merancang serta mendokumentasikan sistem piranti lunak. UML menawarkan suatu standar buat merancang model suatu sistem. Dengan memakai UML kita bisa membuat model buat seluruh tipe aplikasi piranti lunak, dimana aplikasi tersebut bisa berjalan pada piranti keras, sistem pembedahan serta jaringan apapun, dan ditulis dalam bahasa pemrograman apapun. (Dharwiyanti dan Wahono, 2003).

2.12 CSS

CSS adalah singkatan dari *Cascading Style Sheet*, yaitu bahasa berbasis teks yang digunakan untuk memformat tampilan halaman web yang dibuat dengan HTML, seperti dari segi warna, ukuran, dan posisi. Jika HTML adalah fondasi dan batu bata yang menyusun struktur rumah, maka CSS adalah plester dan cat yang menghiasinya. Dengan menggunakan CSS, ini memungkinkan kita memberikan efek khusus pada konten web secara lebih spesifik. Memungkinkan kami mengubah tampilan beberapa halaman web sekaligus dengan sedikit kode. (Mukhlisin, 2013).

Sehingga dapat disimpulkan bahwa CSS adalah tampilan suatu dokumen yang ditulis dalam bahasa markup / markup. Jika kita berdiskusi dalam lingkungan jaringan, dapat diartikan sebagai bahasa yang digunakan untuk mengatur tampilan / desain halaman HTML.

2.13 *Balsamiq Mockups*

Membuat web yang besar, seseorang desainer handal hendak memakai perlengkapan pembuat mockup buat menganalisa tata letak, desain serta guna. *User interface* ataupun yang dalam bahasa Indonesia diucap dengan antarmuka pengguna merupakan salah satu perihal berarti dalam membangun suatu aplikasi. Aplikasi yang baik senantiasa mempunyai *user interface* yang bukan cuma bagus, namun gampang dimengerti oleh *user*.

Salah satu tools buat membuat mockup yang sangat terkenal merupakan *Balsamiq Mockups*. Salah satu alasannya merupakan sebab *Balsamiq Mockups* berbasis cloud, diiringi aplikasi desktop yang membolehkan Kamu dengan kilat serta gampang membuat rancangan web. Aplikasi ini dapat digunakan buat sistem pembedahan *Windows, Mac OS, serta Linux*.

Balsamiq Mockups merupakan aplikasi yang sangat bermanfaat bila kamu merupakan seseorang designer paling utama di bidang *website*. *Mockups* umumnya di pakai oleh para designer buat mengawali sesuatu coretan- coretan, dengan kata lain kamu tidak butuh lagi memakai pensil buat merancang suatu. *Mockups* merupakan sebutan yang tidak asing untuk para designer, mockups sendiri ataupun yang kerap dimaksud *prototype* sangat dibutuhkan buat demonstrasi produk dini saat sebelum jadi produk jadi yang nantinya hendak dipasarkan ke publik. (Putu, 2016)

2.14 *Phpmyadmin*

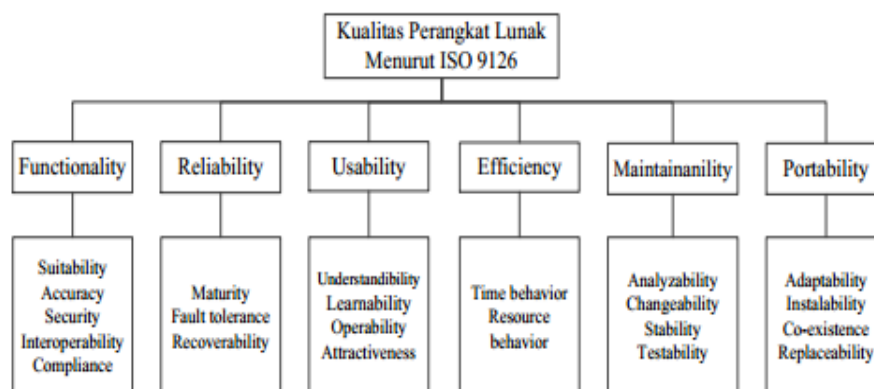
Menurut (Nugroho, 2017) *PhpMyAdmin* merupakan tools yang bisa digunakan dengan gampang buat manajemen *database MySQL* secara visual serta Server MySQL, sehingga kita tidak butuh lagi wajib menulis query SQL tiap

hendak melaksanakan perintah pembedahan *database*. *PhpMyAmin* juga salah satu aplikasi website yang bertabiat opensource.

2.15 ISO 9126

ISO 9126 adalah standar kualitas perangkat lunak yang diakui secara internasional. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik kualitas dan indikator terkait yang digunakan untuk mengevaluasi dan menentukan kualitas produk perangkat lunak. Selain itu, standar ISO juga harus dipenuhi dalam hal manajemen. Jika manajemen tidak memenuhi standar ISO, hasil kerjanya tidak akan dapat memperoleh sertifikat standar ISO. (Junyati, 2019).

Keuntungan utama ISO 9126 adalah karakteristiknya dapat diteruskan pada semua jenis perangkat lunak sambil memberikan terminologi yang konsisten. Ini juga mencakup fitur-fitur penting, seperti hierarkis; kriteria evaluasi dan definisinya sederhana. Semua atribut kualitas diklasifikasikan menurut hierarki dan subfitur, kualitas perangkat lunak menurut ISO 9126 pada Gambar 2.2.



Gambar 2.2 Model Kualitas Perangkat Lunak Model *ISO 9126*

Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut:

1. Fungsionalitas (*Functionality*). Perangkat lunak ini menyediakan fungsi sesuai dengan kebutuhan pengguna dan memenuhi kemampuan pengguna.
2. Keandalan (*Reliability*). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja/kinerja perangkat lunak tertentu (misalnya: akurasi, konsistensi, kesederhanaan, toleransi kesalahan).
3. Kegunaan (*Usability*). Kemampuan perangkat lunak untuk memahami, mempelajari, menggunakan, dan menarik pengguna.
4. Efisiensi (*Efficiency*). Kemampuan perangkat lunak untuk memberikan kinerja dan kemampuan yang sesuai dengan jumlah sumber daya yang digunakan pada saat itu (misalnya: efisiensi penyimpanan).
5. Pemeliharaan (*Maintainability*). Kemampuan perangkat lunak untuk dimodifikasi. Modifikasi termasuk mengoreksi, meningkatkan atau beradaptasi dengan perubahan lingkungan, persyaratan, dan spesifikasi fungsional (misalnya, konsistensi).
6. Portabilitas (*Portability*). Kemampuan perangkat lunak untuk berpindah dari satu lingkungan ke lingkungan lain atau kemampuan beradaptasi perangkat lunak saat digunakan di area tertentu (misalnya: dokumentasi diri, rutin). Setiap karakteristik

kualitas perangkat lunak model ISO 9126 dibagi menjadi beberapa subkarakteristik kualitas, seperti dibawah ini :

Tabel 2.2 *ISO 9126-Functionality*

Sub-Karakteristik	Deskripsi
Suitability	Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.
Accuracy	Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.
Security	Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup (hacker) maupun otorisasi dalam modifikasi data.
Interoperability	Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.
Compliance	Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.

Tabel 2.3 *ISO 9126-Reliability*

Sub-Karakteristik	Deskripsi
Maturity	Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam

	perangkat lunak.
Fault tolerance	Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak.
Recoverability	Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan.

Tabel 2.4 ISO 9126-*Usability*

Sub-Karakteristik	Deskripsi
Understandibility	Kemampuan perangkat lunak dalam kemudahan untuk dipahami.
Learnability	Kemampuan perangkat lunak dalam kemudahan untuk dipelajari.
Operability	Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan.
Attractiveness	Kemampuan perangkat lunak dalam menarik pengguna.

Tabel 2.5 ISO 9126-*Efficiency*

Sub-Karakteristik	Deskripsi
Time behavior	Kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya.

Resource behavior	Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan.
-------------------	---

Tabel 2.6 ISO 9126-*Maintainability*

Sub-Karakteristik	Deskripsi
Analyzability	Kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan.
Changeability	Kemampuan perangkat lunak untuk dimodifikasi tertentu.
Stability	Kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak.
Testability	Kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain.

Tabel 2.7 ISO 9126-*Portability*

Sub-Karakteristik	Deskripsi
Adaptability	Kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda.
Instalability	Kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda.
Coexistence	Kemampuan perangkat lunak untuk

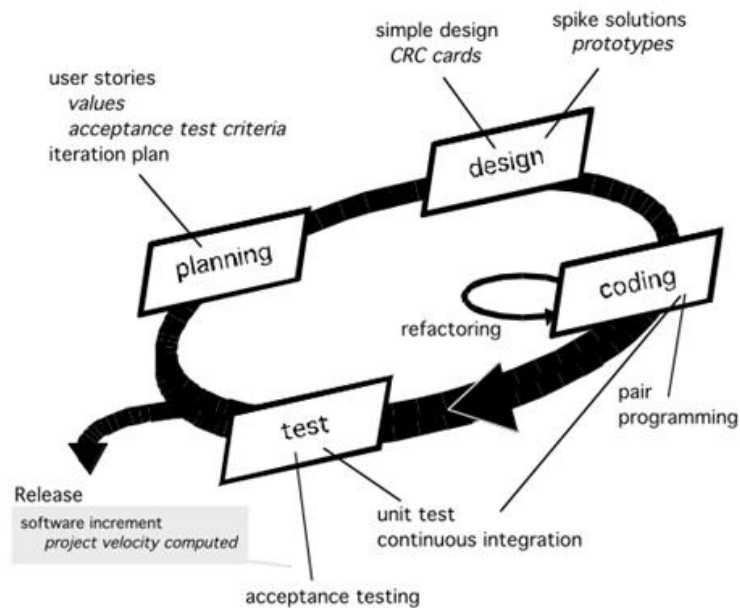
	berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya.
Replaceability	Kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya.

2.16 *Extreme Programming*

Menurut (Fatoni dan Dwi, 2016) Metode *Extreme Programming* biasanya disebut dengan metode XP. Metode ini ditemukan oleh ahli rekayasa perangkat lunak Kent Beck. Pemrograman ekstrim adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem, menjadikannya lebih efisien, adaptif dan fleksibel. Nilai dasar dari *Extreme Programming*:

- 1) *Communication*: Perhatikan komunikasi yang baik antara programmer dan user serta antar programmer.
- 2) *Keberanian*: Pengembang perangkat lunak harus selalu percaya diri, berani dan jujur saat melakukan pekerjaannya.
- 3) *Kesederhanaan*: Cukup lakukan semuanya.
- 4) *Umpan Balik*: Andalkan umpan balik, jadi dibutuhkan anggota tim yang berkualifikasi.
- 5) *Kualitas Kerja* : Proses kualitas pada akhirnya akan mempengaruhi kualitas perangkat lunak.

Sementara itu, menurut Ferdiana (Lubis, 2016), *Extreme Programming* (XP) disebut sebagai metode atau "metode teknis" untuk bagaimana tim teknis dapat secara efektif mengembangkan perangkat lunak melalui berbagai prinsip dan teknik pengembangan perangkat lunak praktis, seperti Gambar 2.3 dibawah ini.



Gambar 2.3 Metode *Extreme Programming*

XP merupakan dasar dari pekerjaan sehari-hari tim, dan metode program ekstrim (xp) harus menyelesaikan empat tahap yaitu:

- 1) perencanaan. Tahap ini merupakan tahap awal pengembangan sistem, pada tahap ini akan dilakukan beberapa kegiatan perencanaan yaitu mengidentifikasi masalah dan menganalisis kebutuhan untuk menentukan jadwal waktu pengembangan dan implementasi sistem.
- 2) Desain. Tahap perancangan selanjutnya adalah melakukan kegiatan pemodelan pada tahap ini, mulai dari pemodelan sistem, pemodelan arsitektur hingga pemodelan basis data. Pemodelan sistem dan arsitektur menggunakan diagram unified modeling language (UML),

sedangkan pemodelan *database* menggunakan diagram hubungan entitas (ERD).

- 3) Pengkodean (*encoding*). Tahap ini merupakan kegiatan mengimplementasikan pemodelan yang telah dibuat menjadi antarmuka pengguna dengan menggunakan bahasa pemrograman. Bahasa pemrograman yang digunakan adalah PHP dengan pendekatan terstruktur. Untuk sistem pengelolaan *database* menggunakan *software* MySQL.
- 4) Pengujian. Setelah tahap coding selesai dilakukan maka dilakukan tahap pengujian sistem untuk mengetahui *error* apa saja yang terjadi pada saat aplikasi berjalan, dan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Metode pengujian yang digunakan pada tahap ini adalah metode pengujian *black box*, dimana pengujian dilakukan dalam beberapa bentuk untuk menginput apakah sudah berjalan sesuai dengan fungsinya masing-masing.

2.17 Bagan Alir (*Flowchart*)


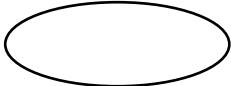


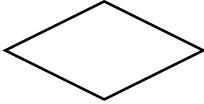
Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis. Flowchart merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan antara proses digambarkan dengan garis penghubung.




Dengan menggunakan flowchart akan memudahkan untuk melakukan pengecekan bagian-bagian yang terlupakan 4 dalam analisis masalah. Flowchart dapat juga merupakan penggambaran secara grafik dari langkah-langkah dan

urutan prosedur dari suatu program. Flowchart biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut (Tominanto dan Subinarto, 2018).

Pada dasarnya simbol-simbol dalam flowchart memiliki arti yang berbeda-beda. Berikut adalah simbol-simbol yang sering digunakan dalam proses pembuatan flowchart. simbol-simbol dalam Flow Chart dapat di lihat pada tabel 2.8 sebagai berikut :

Tabel 2.8 Simbol Bagan Alir (*flowchart*)

Simbol	Keterangan
<p>Garis Alir (Flow Line)</p> 	Arah aliran program
<p>Terminator</p> 	Permulaan/akhir program
<p>Process</p> 	Proses perhitungan/proses pengolahan data
<p>Input/Output Data</p> 	Proses input/output data,parameter, informasi
<p>Decision</p> 	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya

<p style="text-align: center;">Predefined Process (Sub Program)</p> 	<p>Permulaan sub program/proses menjalankan sub program</p>
<p style="text-align: center;">On Page Connector</p> 	<p>Penghubung bagian-bagian flowchart yang berada pada satu halaman</p>
<p style="text-align: center;">Off Page Connector</p> 	<p>Penghubung bagian-bagian flowchart yang berada pada halaman berbeda</p>

2.18 UML (*Unified Modelling Language*)

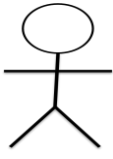
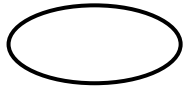

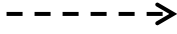

Unified Modelling Language (UML) merupakan suatu "bahasa" yang sudah jadi standar dalam industri buat visualisasi, merancang serta mendokumentasikan sistem piranti lunak. UML menawarkan suatu standar buat merancang model suatu sistem. Dengan memakai UML kita bisa membuat model buat seluruh tipe aplikasi piranti lunak, dimana aplikasi tersebut bisa berjalan pada piranti keras, sistem pembedahan serta jaringan apapun, dan ditulis dalam bahasa pemrograman apapun. (Dharwiyanti dan Wahono, 2003).

2.18.1 *Use Case Diagram*

Menurut (Sulistyorini, 2009) *Use case Diagram* ini bersifat statis. Diagram ini memperlihatkan himpunan *use case* serta aktor- aktor (suatu jenis khusus dari kelas). Diagram ini paling utama sangat penting buat mengorganisasi dan memodelkan sikap dari suatu sistem yang dibutuhkan dan diharapkan pengguna.

Didalam *use case* ini hendak dikenal fungsi- fungsi apa saja yang terletak pada sistem yang terbuat. Adapun simbol dari *use case* diagram pada Tabel 2.9.

Tabel 2.9 Simbol *Use Case Diagram*

No	Simbol	Keterangan
1.	Aktor / Actor 	Actor mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
2.	Use Case 	<i>Use case</i> gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun
3.	Asosiasi / Association 	Assosiasi menghubungkan actor dengan <i>use case</i> yang memiliki interaksi.
4.	Menggunakan / Uses / Include <<include>> 	Include garis yang menyatakan interaksi actor dengan <i>use case</i> yang harus terpenuhi.
5.	Generalisasi / Generalization 	Generalisasi sebuah elemen yang menjadi spesialisasi dari elemen yang lain.

6.	Ekstensi / Extend <<extend>> ----->	Extend Garis yang menyatakan interaksi actor dengan <i>use case</i> yang dapat dilakukan atau tidak (conditional).
----	--	--

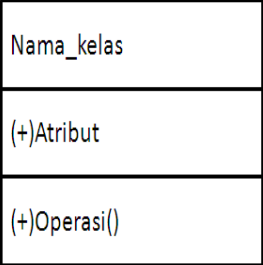

2.18.2 Class Diagram


Diagram kelas atau diagram kelas menggambarkan struktur sistem dari perspektif pendefinisian kelas-kelas yang akan digunakan untuk membangun sistem. Kelas memiliki apa yang disebut properti, metode, atau operasi. Komponen utama diagram kelas adalah:

1. Atribut: variabel yang termasuk dalam kelas
2. Operasi atau metode: fungsi yang dimiliki kelas

Diagram kelas menjelaskan jenis objek dalam sistem dan berbagai hubungan statis yang terkandung di dalamnya. mereka. Diagram kelas juga menunjukkan properti dan operasi kelas dan batasan yang terdapat dalam hubungan objek. Simbol dalam diagram kelas yang ada pada Tabel 2.10.

Tabel 2.10 Simbol class diagram

No.	Simbol	Keterangan
1.	Kelas / Class 	Kelas adalah sekelompok objek dengan berbagai properti dan operasi yang sama.
2.	Asosiasi / Association 	Asosiasi adalah hubungan antar kelas dengan pengertian umum.

3.	Asosiasi berarah / <i>Directed association</i> 	Asosiasi terarah adalah hubungan antara kelas dan kelas yang digunakan oleh kelas lain.
4.	Generalisasi / <i>Generalization</i> 	Generalisasi adalah hubungan antar kelas dan memiliki arti generalisasi-spesialisasi (umum-khusus).
5.	Kebergantungan / <i>Dependency</i> 	Ketergantungan adalah saling ketergantungan antar kelas.
6.	Agregasi / Aggregation 	Agregasi adalah hubungan antar kelas yang memiliki segala arti bagian.
7.	Antarmuka / Interface  Nama_interface	Antarmuka sama dengan konsep antarmuka dalam pemrograman berorientasi objek.

2.19 Analisis *PIECES*

Menurut (Sihotang, Sumarya dan Irwan, 2019) *PIECES* adalah program pembelajaran dan pengembangan praktik terbaik yang menyediakan cara untuk memahami dan meningkatkan perawatan individu dengan kebutuhan fisik dan kognitif yang kompleks serta perubahan perilaku. *PIECES* terus meningkatkan co-care melalui pengembangan sumber daya manusia. Dalam *PIECES* terdapat enam komponen yang dapat digunakan untuk mengevaluasi kepuasan pengguna sistem informasi, yaitu: *Performance, Information, Economic, Control, Efficiency, Service*.

Berikut kriteria yang harus ada menurut analisis kerja James Wetherbe (2012):

1. *Performance* (Kinerja)

Bagian ini menjelaskan situasi beban kerja saat ini yang diperlukan untuk menyelesaikan tugas dalam satuan jam kerja, hari kerja, atau bulan kerja.

2. *Information* (Informasi)

Menilai apakah prosedur yang ada dapat ditingkatkan untuk menjadikan kualitas informasi yang dihasilkan semakin baik. Informasi yang diberikan memang harus memiliki nilai manfaat. Ini dapat diukur dengan cara berikut:

- a) Keluaran (output): Sistem yang digunakan untuk menghasilkan keluaran.
- b) Input: Saat memasukkan data, buat pemrosesan informasi yang berguna nanti.

3. *Economy* (Ekonomi)

Biaya ekonomi tidak diketahui, biaya tidak dapat dilacak ke sumbernya, dan biaya yang terlalu tinggi. Secara umum manfaat penerapan sistem informasi selain yang disebutkan di bawah ini memiliki manfaat lain, semua orang jelas. Oleh karena itu, pada bagian ini akan diuraikan manfaat penerapan teknologi informasi atau sistem informasi pada saat mengimplementasikan proses bisnisnya.

4. *Control* (Pengendalian)

Pengendalian Bagian ini memperkenalkan situasi terkini mengenai penguasaan arus data dan informasi ketika keamanan atau pengontrolan terlihat lemah, sehingga data dan informasi mudah digunakan oleh pihak yang tidak berkepentingan. Begitu pula ketika pengamanan atau pengendalian arus data dan informasi terlalu ketat, sehingga sistem kewalahan dengan prosedur pengamanan atau pengendalian tersebut, hal tersebut juga akan mengganggu keamanan dan kenyamanan pengguna dan pengguna data dan informasi yang dihasilkan oleh perusahaan. sistem.

5. *Efficiency* (efisiensi)

Efisiensi Jika terlalu banyak data yang dimasukkan dan diproses, terlalu banyak informasi yang dihasilkan akan membuat sistem menjadi tidak efisien dalam hal penggunaan sumber daya. Sumber daya dapat berupa sumber daya prosesor, memori, ruang penyimpanan, listrik, personel, dll.

6. *Service* (layanan)

Layanan Bagian ini memperkenalkan status layanan saat ini yang disediakan oleh sistem operasi saat ini. Serangkaian kelemahan layanan data sistem diidentifikasi di bawah ini. Berikut adalah beberapa kelemahan layanan sistem yang teridentifikasi: Sistem menghasilkan produk yang tidak akurat Sistem menghasilkan produk yang tidak konsisten Sistem menghasilkan produk yang tidak dapat dipercaya Sistem tidak mudah dipelajari.