

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Berdasarkan penelitian menurut Afrilia Gita Risqi, Isnawaty, Subardin, dan Bambang Pramono (2022) dalam jurnal berjudul "Sistem *Monitoring* Lampu Penerangan Jalan Umum Berbasis *Internet of Things*", membuktikan bahwa sistem *monitoring* lampu jalan berbasis *IoT* dapat dibangun secara utuh mulai dari tahapan analisis kebutuhan perangkat keras dan lunak, perancangan sistem, hingga menjadi sebuah prototipe alat yang dapat diakses melalui *platform* berbasis *web* dan *android*.

Penelitian tersebut juga menunjukkan bahwa penerapan sistem ini mampu membantu petugas penerangan jalan umum (PJU) dalam mendeteksi kerusakan lampu tanpa harus melakukan pengecekan langsung ke lokasi. Dengan integrasi sensor *LDR* dan sensor arus dalam jaringan nirkabel (*Wireless Sensor Network*), sistem dapat mendeteksi kondisi menyala atau padam lampu secara *real-time*.

Hasil pengujian membuktikan bahwa sistem *monitoring* tersebut berfungsi dengan baik, mampu menampilkan kondisi lampu ketika dalam keadaan menyala maupun padam, serta dapat diakses melalui internet baik menggunakan perangkat *desktop* maupun *smartphone*. Penelitian ini menjadi referensi penting dalam pengembangan sistem serupa yang bertujuan untuk meningkatkan efisiensi dan efektivitas pengelolaan lampu jalan[4].

2.2 Landasan Teori

2.2.1 Lampu penerangan jalan

Lampu penerangan jalan merupakan bagian dari bangunan pelengkap jalan yang dapat diletakkan atau dipasangkan di kiri atau kanan jalan dan atau di tengah (di bagian medan jalan) yang digunakan sebagai penerangan jalan maupun lingkungan di sekitar jalan yang diperlukan termasuk persimpangan jalan (*intersection*), jalan layang (*interchange, overpass, fly over*), jembatan dan jalan di bawah tanah (*underpass, terowongan*).

Lampu penerangan yang dimaksud adalah suatu unit lengkap yang terdiri dari sumber cahaya (lampu/luminer), elemen-elemen optik (pemantul/reflector, pembias/refractor, penyebar/*diffuser*). Elemen-elemen elektrik (konektor ke sumber tenaga/*power supply*. dll.), struktur penopang yang terdiri dari lengan penopang, tiang penopang vertical dan pondasi tiang lampu[5].

2.2.2 Monitoring

Monitoring (pemantauan) adalah kegiatan untuk mengamati perkembangan pelaksanaan program atau proyek. Dengan *monitoring* dapat diketahui program atau proyek berjalan sesuai atau kurang sesuai dengan rencana[6].

2.2.3 Website

Menurut Wahidin Abidin (2015:19), *website* merupakan Kumpulan halaman *web* yang saling terhubung satu sama lain. Halaman

utama dari sebuah *website* disebut juga sebagai *home page*, sedangkan halaman-halaman lainnya disebut sebagai *web page*. Secara umum, *website* adalah sebuah situs yang dapat diakses dan dilihat oleh pengguna internet dari berbagai belahan dunia. Seiring dengan meningkatnya jumlah pengguna internet setiap harinya, *website* menjadi sebuah peluang pasar yang terus berkembang[7].

2.2.4 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman open-source yang umumnya digunakan untuk membangun aplikasi *web* dinamis dan interaktif. PHP dapat dijalankan pada server *web* dan dikombinasikan dengan HTML, CSS, dan JavaScript untuk membuat halaman *web* yang dinamis[8].

PHP dikembangkan dengan tujuan awal hanya untuk mencatat pengunjung pada *website* pribadi Rasmus Lerdorf. Pada rilisan keduanya ditambahkan *Form Interpreter*, sebuah tools untuk melakukan penerjemahan perintah SWL. Rilisan kedua disebut dengan PHP/FI. Sejak itu, PHP mulai diterima sebagai sebuah Bahasa pemrograman baru yang sangat diminati, terbukti dengan tercatatnya sekitar 50.000 situs diseluruh dunia yang telah menggunakan PHP pada tahun 1997[9].

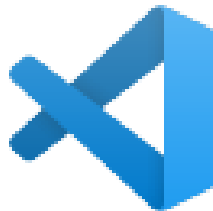


Gambar 2. 1 Logo PHP

2.2.5 Visual Studio Code

Visual Studio Code adalah kode editor yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux*, dan *macOS*. Ini termasuk dukungan untuk debugging, control *git* yang tertanam dan *GitHub*, penyorotan sintaksis, penyelesaian kode cerdas, snippet, dan *refactoring* kode. Kegunaannya pun dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan *keyboard*, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan.

Kelebihan lain dari *text editor* ini juga memiliki fitur yang lengkap dan juga gratis, menjadikannya unggul dalam segi kelengkapan fitur. Peren extension mark etplace menopang besar kelebihan *Visual Studio Code* yang membuat penggunaanya lebih leluasa menambahkan sejumlah fitur tertentu[10].



Gambar 2. 2 Logo Visual Studio Code

2.2.6 Database

Menurut Andaru (2018), basis data merupakan sekumpulan data yang disusun secara teratur di dalam komputer, sehingga dapat diakses dan dikelola melalui program komputer guna memperoleh informasi dari data tersebut.

Secara umum, *database* adalah tempat penyimpanan informasi yang tersusun secara sistematis dan memungkinkan pengguna untuk menelusuri data dengan bantuan perangkat lunak. Tujuan utama dari sistem basis data adalah memberikan pandangan abstrak terhadap data kepada pengguna, sehingga interaksi antara pengguna dan sistem menjadi lebih mudah. Sistem ini juga mampu menyajikan perspektif yang berbeda bagi masing-masing pihak, seperti pengguna, programmer, maupun administrator[11].

2.2.7 UML

Unified Modeling Language atau UML merupakan salah satu metode yang banyak digunakan dalam rekayasa perangkat lunak untuk menggambarkan alur kerja sistem, termasuk fungsionalitas, tujuan, serta mekanisme pengendaliannya. Dalam proses analisis dan perancangan sistem informasi, UML menggabungkan pendekatan pemrograman berorientasi objek dengan teknik pengembangan perangkat lunak. Pendekatan ini memandang sistem sebagai kumpulan objek yang memiliki data dan fungsi sendiri, serta mampu beroperasi secara mandiri dalam satu kesatuan sistem atau paket.

Sejak tahun 1997, UML telah diakui sebagai standar dalam teknik rekayasa perangkat lunak berbasis objek. Dalam konteks ini, UML bertindak sebagai bahasa pemodelan yang menyediakan kosakata serta aturan-aturan penyusunan simbol dan istilah teknis yang mempermudah komunikasi dalam pengembangan sistem. Sebagai bahasa pemodelan standar, UML tidak menetapkan secara spesifik apa saja model yang harus dibuat atau kapan harus dibuat, namun fokus pada cara membangun dan menggambarkan model tersebut.

UML bukan sekadar bahasa visual untuk pemrograman, tetapi juga memiliki kemampuan untuk diintegrasikan dengan berbagai bahasa pemrograman seperti *Java*, *C++*, *Visual Basic*, bahkan basis data berorientasi objek. Selain itu, UML juga mendukung proses dokumentasi yang mencakup berbagai aspek proyek, mulai dari kebutuhan sistem (*requirement*), arsitektur, desain, kode sumber, perencanaan proyek, pengujian, hingga prototipe.

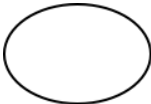



Untuk memahami UML secara menyeluruh, diperlukan pemahaman terhadap konsep dasar dari bahasa pemodelan, termasuk tiga elemen penting dalam UML yaitu: *building blocks* (komponen dasar penyusun model), aturan-aturan yang mengatur bagaimana komponen tersebut disusun, serta mekanisme umum yang mendukung fleksibilitas dalam membangun model[12].



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

1) *Use case*

Use case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* Diagram yaitu:

Tabel 2. 1 *Use Case* Diagram


No	Simbol	Keterangan
1		<i>Use case</i> menggambarkan fungsionalitas yang disediakan system sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
2		<i>Actor</i> adalah Abstraction dari orang atau, system yang lain yang mengaktifkan fungsi dari target system. Untuk mengidentifikasi actor harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target system. Orang atau system bisa muncul.
3		Asosiasi antara aktor dan <i>use case</i> digambarkan dengan garis tanpa panah untuk menunjukkan interaksi langsung, bukan aliran data.
4		Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem



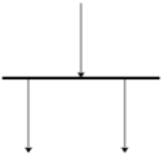
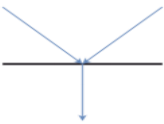
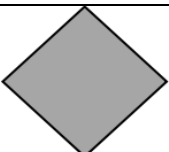
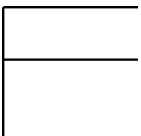
No	Simbol	Keterangan
5		<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
6		<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

2) Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *Activity* Diagram yaitu:

Tabel 2. 2 *Activity* Diagram

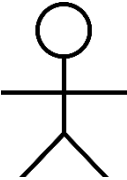
No	Simbol	Keterangan
1		<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas

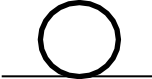




No	Simbol	Keterangan
2		<i>End Point</i> , akhir aktivitas.
3		<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis.
4		<i>Fork</i> digunakan untuk menunjukkan aktivitas paralel atau penggabungan dua aktivitas.
5		<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
6		Decision points menggambarkan pilihan untuk pengambilan keputusan, true atau false
7		<i>Swimlane points</i> pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa.

3) Sequence Diagram

Sequence Diagram merupakan representasi visual yang menunjukkan bagaimana interaksi antar objek berlangsung seiring waktu. Diagram ini berfungsi untuk menggambarkan urutan proses dalam suatu sistem. Beberapa simbol digunakan dalam *Sequence Diagram* untuk merepresentasikan elemen-elemennya. Simbol-simbol yang digunakan dalam *Sequence Diagram* antara lain:

Tabel 2. 3 *Sequence Diagram*

No	Simbol	Pengertian	Keterangan
1.		<i>Aktor</i>	Menggambarkan seseorang yang berinteraksi dengan sistem.

No	Simbol	Pengertian	Keterangan
2.		<i>Entity Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
3.		<i>Asynchronous Message</i>	Menggambarkan pengiriman pesan.
4.		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
5.		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem.
6.		<i>A Focus Of Control & A Life Line</i>	Menggambarkan penghubung antara boundary dengan tabel.

4) *JavaScript*

Javascript adalah bahasa yang berbentuk kumpulan skrip yang fungsinya digunakan untuk Menambahkan interaksi antara halaman *web* dengan pengunjung halaman *web*. *Javascript* dijalankan pada sisi klien yang akan memberikan kemampuan fitur-fitur tambahan halaman *web* yang lebih baik dibandingkan fitur-fitur yang terdapat pada *HTML*[12].

5) *Xampp*

Xampp adalah perangkat lunak berbasis *web* server yang bersifat *open source* atau bebas dipakai yang mendukung di berbagai sistem operasi, baik di *Windows*, *Linux*, atau *MacOS*. *Xampp* dipakai

sebagai *standalone* server atau biasa disebut dengan *localhost*. Hal tersebut memudahkan dalam proses pengeditan, desain, dan pengembangan aplikasi. Terdapat banyak juga manfaat dari penggunaan *Xampp* seperti mengkonfigurasi pengaturan *database* pada *PhpMyAdmin* hingga menjalankan *framework web PHP* dan bahasa pemrograman lainnya secara dinamis dan fleksibel melalui perangkat komputer[13].

2.2.8 MySQL

MySQL adalah salah satu dari sekian banyak *DBMS* seperti *Oracle*, *MSSQL*, *Postagre SQL*, dan lainnya. *MySQL* adalah sistem manajemen basis data yang menggunakan *SQL* untuk mengelola data. *MySQL* adalah *database open source*, yang artinya Anda dapat menggunakannya secara gratis. Pemrograman *PHP* juga sangat mendukung atau support dengan Basis Data *MySQL* [14].

Salah satu kelebihan utama dari *MySQL* adalah kemampuannya dalam mendukung operasi dasar seperti menambahkan data baru (*Create*), menampilkan data (*Read*), memperbarui informasi (*Update*), dan menghapus data (*Delete*) atau yang biasa dikenal dengan istilah operasi *CRUD*. Sebagai contoh, ketika ada barang baru yang masuk ke dalam sistem, pengguna dapat dengan mudah menambahkannya ke dalam basis data. Jika terjadi perubahan pada stok atau harga, data tersebut pun bisa langsung diperbarui. Bahkan, ketika suatu produk

sudah tidak tersedia, pengguna bisa menghapus datanya agar informasi tetap akurat dan terkini.

Selain fitur-fitur tersebut, *MySQL* juga menyediakan sistem keamanan yang cukup andal, termasuk autentikasi pengguna dan pengaturan hak akses. Fitur ini berfungsi untuk memastikan bahwa hanya pengguna tertentu yang dapat memodifikasi atau menghapus data, sehingga mencegah terjadinya kesalahan atau penyalahgunaan data. Tak hanya itu, *MySQL* juga mendukung integrasi dengan berbagai bahasa pemrograman seperti *PHP*, *Python*, maupun *JavaScript*, sehingga memudahkan pengembang dalam menciptakan aplikasi *web* yang dinamis dan responsif