BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada penelitian terdahulu bertujuan untuk penulis mendapatkan acuan dan perbandingan dalam penelitian. Referensi yang didapat oleh peneliti bertujuan untuk menemukan algoritma dan metode yang lebih efektif untuk di terapkan dalam penelitian penulis. Pada tabel 2.1 berisi tentang penelitian terdahulu berupa jurnal yang saling berhubungan pada penelitian yang dibuat peneliti.

Tabel 2.1 Hasil Penelitian Terdahulu

No	Nama (Tahun)	Judul	Hasil
1	(Rachmawati et al., 2019)	Pembangunan *Helpdesk Ticketing System* Berbasis Web (Studi Kasus: Universitas Yarsi)	Hasil pada penelitian ini yaitu pembuatan sistem *Helpdesk* yang berfungsi untuk perusahaan atau institusi. Lembaga memiliki tujuan dan program yang spesifik, menggunakan fasilitas yang sudah ada dan dimaksudkan untuk membuat staf atau karyawan lebih produktif. *Helpdesk ticketing system* yang berbasis internet akan membantu dalam menyelesaikan konflik dan mendapatkan infomasi dengan cepat, tepat, akurat dan lebih efisien. Pembaharuan sistem *Helpdesk* yang berbasis web. Hal ini akan menghasilkan pengaruh yang lebih besar dan lebih baik dari pada harus monitoring secara manual. Pengujian dilakukan dengan menggunakan metode yang berfokus pada kepuasan

			pengguna untuk mendapatkan hasil 87,5%.
2	(Liharja et al., 2022)	Rancang Bangun Sistem Informasi *Helpdesk* IT *Support* Berbasis Website	Hasil penelitian ini jaringan atau perbaikan komputer yang masih dilakukan secara manual membutuhkan waktu yang lebih banyak karena membutuhkan izin dari manajer. Hal ini bisa menimbulkan konflik dalam bentuk data dari reportase yang tidak dikumpulkan, rumit dalam melakukan penelusuran data dan kemungkinan kehilangan data. Selama bulan pertama, tidak ada salahnya untuk melaporkan semua hal yang *user* alami terkait dengan akses ke komputer dan jaringan. Dalam 1 bulan, ini tidak diklasifikasikan dengan baik dan itu sulit untuk membuat evaluasi dari gangguan ini. Karena itu, dibutuhkan sistem informasi *Helpdeks* untuk bagian IT *support* yang terfokus dan dapat diakses dari mana saja dengan menggunakan koneksi internet, sehingga *problem* yang dihadapi dapat diselesaikan dengan baik. Prosedur monitoring dan melaporkan *request* untuk masuk ke komputer serta gangguan yang terkait dengan komputer yang di komplain,

			dapat dilihat dengan baik dan dapat dievaluasi untuk menemukan solusi yang akurat.
3	(Bahrudin et al., 2019)	Penerapan *Helpdesk Ticketing System* Dalam Penanganan Keluhan Penggunaan Sistem Informasi Berbasis Web	Hasil Penelitian ini pada Unit Pelaksana Tugas Informasi Teknologi (UPT IT) merupakan jantung dari sistem informasi teknologi yang ada di kampus Universitas Islam Syekh-Yusuf. Mereka melaporkan kekurangan sistem informasi akademik seperti halnya dengan masalah finansial, mereka masih harus pergi ke tempat teknisi IT secara langsung. Pada saat *user* ingin melaporkan, mereka juga merasa bingung karena dia tidak tahu harus melaporkannya ke siapa ketika teknisi IT tidak sedang berada di tempat untuk melaporkan dengan baik. Untuk menyelesaikan konflik yang ada, dibutuhkan aplikasi *Helpdesk* untuk menjadi tempat *user* melaporkan gangguannya. Penelitian ini dibuat menggunakan metode *prototype*, aplikasi ini dikembangkan dengan menggunakan MySQL sebagai database dan aplikasi web sebagai *platform*. Hasil dari studi ini adalah menghasilkan sistem *Helpdesk* yang

			berfungsi untuk membantu teknisi IT dalam mengelola dan melaporkan masalah.
4	(Mauliana et al., 2020)	Pengembangan *E-Helpdesk Support* System Berbasis Web di PT Akur Pratama	Hasil penelitian ini pada PT Akur Pratama adalah perusahaan ritel yang mengandalkan dukungan TI untuk pemeliharaan sistem dan *Helpdeks*. Perusahaan telah mengidentifikasi beberapa masalah dengan kegiatan pemantauan yang perlu ditangani. Karyawan internal perusahaan kesulitan untuk mengajukan keluhan ketika muncul masalah teknis dalam sistem kerja mereka. Tidak ada biaya operasi yang tidak signifikan untuk pemeliharaan sistem dan durasi pemrosesan teknis dengan dukungan TI. Di penelitian ini digunakan metode penelitian pengembangan. Oleh karena itu, tujuan dari penelitian ini adalah untuk mengembangkan sistem dukungan layanan e-*Helpdeks* berbasis web yang memungkinkan meja layanan memantau pekerjaan dukungan TI dan keluhan dari staf internal PT Akuri. Hasil penelitian menunjukkan bahwa sistem yang dibangun mampu mempertimbangkan keluhan dari pelanggan pengguna atau karyawan

			internal perusahaan sehingga memudahkan dalam penyampaian keluhan. Sistem dukungan e-*Helpdesk* membantu manajemen menghemat biaya operasional dan mempercepat pemrosesan teknis dukungan TI. Dengan mengembangkan sistem ini, dimungkinkan untuk menyediakan meja layanan dengan cara memantau pekerjaan dukungan TI dan menghasilkan laporan di PT Akur Pratama.
5	(Rilvani, 2020)	Sistem Informasi *Helpdesk* Berbasis Web Menggunakan Metode *Prototype* Pada PT Denso Manufacturing Indonesia	Hasil penelitian ini pada perusahaan manufaktur saat ini dibutuhkan suatu sistem informasi yang mendukung semua proses internal perusahaan guna mempercepat proses bisnisnya. Dengan perintah kerja TI, pengguna masih menggunakan sumber daya kertas untuk membuat perintah kerja sehingga masalah yang muncul saat pengguna tidak segera teratasi oleh tim IT. PT Denso Manufacturing Indonesia memiliki sistem informasi *Helpdesk* yang membantu pengguna untuk menanyakan masalah yang terjadi pada perangkat keras, perangkat lunak, dan jaringan komputernya sehingga tim IT

			dapat menyelesaikan masalah dengan cepat dan sistem informasi *Helpdesk* ini juga membantu tim IT menyelesaikan umum masalah-masalah dapat diminimalkan dengan solusi yang sudah dicoba. Metode yang digunakan pada Sistem Informasi Helpdesk PT Denso Indonesia menggunakan metode *prototype* dimana metode ini sangat membantu tim IT untuk mengembangkan Sistem Informasi *Helpdesk*. Hasil dari penelitian ini adalah diluncurkannya sebuah sistem informasi bantuan yang dapat digunakan oleh seluruh karyawan PT Denso Manufacturing Indonesia yang menggunakan komputer dalam pekerjaannya, sehingga dapat dengan mudah menanyakan permasalahan yang terjadi pada komputernya. Dengan penambahan sistem informasi *Helpdesk* ini semua masalah dapat dilihat langsung melalui sistem ini, tidak perlu menghitungnya secara manual dan dapat membantu pekerjaan manajer TI untuk mengambil keputusan. Dengan menggunakan data yang diekspor, tim IT dapat menjelaskan masalah yang terkait dengan koneksi

			jaringan, kesalahan perangkat lunak, kesalahan perangkat keras, sehingga tim IT dapat memberi tahu tim dukungan PT Denso Manufacturing Indonesia sejauh mana masalah yang mereka hadapi.
6	(Putra, 2020)	*Analysis of Preemptive Shortest Job First (SJF) Algorithm in CPU Scheduling*	Penjadwalan adalah konsep kunci dalam desain sistem multitugas dan multiprosesor dalam desain sistem operasi *real-time* yang memindahkan prosesor antar proses. *Shortest Job First* (SJF) adalah algoritma terkenal dalam pemrosesan CPU. Algoritma SJF ada 2 macam yaitu algoritma SJF *preemptive* dan algoritma SJF *non-preemptive*. Algoritma SJF *preemptive* berarti proses dapat dihentikan ketika proses baru tiba. Saat menggunakan algoritma *non-preemptive*, algoritma akan selesai bahkan jika proses baru datang. Pengolahan efisien urutan pertama yang tersingkat berarti waktu tunggu dan pemrosesan rata-rata yang lebih singkat.
7	(Setyawatu & Bachtiar Maulachela, 2020)	Penerapan Algoritma Dynamic Priority Scheduling Pada Aplikasi Antrian Pencucian Mobil Berbasis Mobile (*Implementation of Dynamic Priority Scheduling Algorithm in*	Penelitian ini membahas tentang antrian merupakan suatu kondisi dimana jumlah penerima layanan lebih banyak dari jumlah penyedia layanan. Kondisi ini dapat menyebabkan penumpukan

		Mobile Car Queue Washing Applications)	penerima manfaat dan akhirnya terjadi kemacetan. Semua organisasi jasa yang berfokus pada layanan pelanggan dapat menghadapinya. Tak terkecuali bisnis cuci mobil yang proses bisnisnya berorientasi pada penyediaan jasa cuci mobil yang cepat dan berkualitas. Sayangnya, banyak keluhan pelanggan yang diterima di industri cuci mobil, terutama terkait kemacetan antrian dan ketidakjelasan informasi antrian. Oleh karena itu, tujuan dari penelitian ini adalah untuk membangun sebuah aplikasi antrian mobile car wash dengan algoritma penjadwalan prioritas dinamis tertanam yang bertindak sebagai pengelola antrian. Untuk mencapai tujuan tersebut, digunakan metodologi pengembangan perangkat lunak yang menggunakan metodologi penelitian sekuensial dan iteratif. Model ini terdiri dari empat fase: post design, prototyping, system development dan terakhir testing. Dua pendekatan yang digunakan dalam pengujian aplikasi, yaitu pengujian kode aplikasi, penerapan algoritma penjadwalan prioritas dinamis

			khusus, dan pengujian seluruh sistem operasi. Dari hasil pengujian diketahui bahwa pihak pengelola pencucian mobil berhasil memilah pesanan pelanggan sesuai dengan aturan prioritas yang telah ditetapkan yaitu. sesuai dengan jarak dan waktu pemesanan. Hasil pengujian kinerja sistem menunjukkan bahwa aplikasi berhasil menangani sejumlah besar kegagalan yang berasal dari kesalahan sistem dan kesalahan manusia.
8	(Manalu et al., 2022)	Implementasi Metode *Shortest-Job First* untuk Penjadwalan Penggunaan Laboratorium Fisika di SMA 1 Pegajahan	Pada penelitian ini yaitu terdapat Masalah antrian dapat terjadi dalam banyak kasus, termasuk saat menggunakan laboratorium fisika. Pada saat menggunakan laboratorium fisika di SMA Negeri 1 Pegajahan, timbul permasalahan antara lain dengan pengumpulan formulir pemesanan yang masih dilakukan secara manual yang sering menyebabkan penumpukan data, kehilangan data, bahkan kesalahan. Tujuan dari penelitian ini adalah untuk membuat timeline penggunaan laboratorium fisika untuk mengatasi masalah antrian. Pada penelitian ini

			menggunakan algoritma *Shortest Job First* (SJF), yaitu metode penjadwalan yang menentukan setiap proses dalam *ready queue* bekerja berdasarkan waktu burst terpendek, sehingga *latency* dari setiap proses menjadi singkat dan waktu tunggu rata-rata juga singkat. Berdasarkan permasalahan yang ada, dapat disimpulkan bahwa diperlukan suatu sistem perencanaan laboratorium fisika "*Shortest Work First*" yang mampu mengatur dan menetapkan jadwal operasional sesuai dengan durasi pekerjaan.
9	(Sundari et al., 2023)	Penerapan algoritma *shortest job first* (SJF) dan *Priority Scheduling* (PS) pada *Maintanance* mesin ATM	Pada penelitian ini, peneliti ingin mengusulkan suatu sistem penjadwalan dengan menggunakan dua algoritma yang dapat digunakan sebagai pembanding penjadwalan. Ada masalah pemrograman di dunia perbankan yang menjadi bagian penting dari kelancaran operasional bisnis perusahaan, khususnya di ruang ATM. Belum adanya sistem perencanaan untuk menjadwalkan perbaikan dan pembersihan dana ATM yang diperlukan untuk kelancaran operasional bisnis. Algoritma yang dapat diimplementasikan

			pada sistem antrian antara lain SJF (*Shortest Job First*) dan PS (*Priority Scheduling*). Aplikasi yang dilengkapi dengan sistem ini memungkinkan pengguna untuk mengetahui perawatan preventif untuk memudahkan operasi dan pemeliharaan perusahaan untuk menyelesaikan tugasnya.

2.2 Tinjauan Teoritis

Pada penelitian ini, peneliti mempunyai beberapa teori pendukung yang dijadikan acuan dalam membuat laporan penelitian.

2.2.1 API

Application Programming Interface (API) berbagi konsep kemampuan antarmuka pemrograman aplikasi, yang merupakan cara untuk mengakses aplikasi dan memanfaatkannya untuk digunakan oleh pihak lain, dan memungkinkan sistem untuk berkomunikasi satu sama lain meskipun berbeda platform (Muri et al., 2019). API juga dapat menyediakan mekanisme ekstensibilitas untuk memungkinkan pengguna memperluas fungsionalitas yang ada dengan cara yang berbeda dan pada tingkatan yang berbeda. Representasi fungsi yang ditargetkan yang dideklarasikan dalam API dimaksudkan untuk menyediakan serangkaian layanan khusus untuk tujuan tertentu. API karenanya bertindak sebagai perantara antara aplikasi yang berbeda, baik pada platform yang sama atau pada platform yan g berbeda.

2.2.2 PHP

PHP ataupun *Hypertext Preprocessor* ialah bahasa pemrograman yang digunakan secara luas buat penindakan, pembuatan, serta pengembangan suatu web website serta umumnya digunakan bertepatan dengan HTML (Oetomo & Mahargiono, 2020). Server web yang mendukung PHP dapat ditemukan di mana saja dari IIS (*Internet Information Services*) hingga Apache dengan konfigurasi yang relatif sederhana. Lebih mudah di sisi pengembangan karena ada banyak milis (*Mailing List)*

dan *Developer* yang bersedia membantu pengembangan. Karena banyaknya referensi, PHP adalah bahasa *scripting* yang paling mudah dipahami.

2.2.3 HTML

HTML atau *HyperText Markup Language* adalah bahasa markup halaman yang berisi perintah dalam format tertentu untuk menampilkan tata letak tertentu (Anamisa & Mufarroha, 2022). Untuk membuat website tidak cukup hanya menggunakan HTML saja, kita membutuhkan bantuan CSS, JavaScript dan PHP untuk membuat website yang dinamis. HTML memiliki beberapa elemen yang terdiri dari tag-tag yang memiliki fungsinya masing-masing. Seperti tag judul, paragraf, bentuk bangunan, tombol, daftar, membuat hyperlink atau tautan antar halaman web.

2.2.4 CSS

CSS atau *Cascading Style Sheets* adalah bahasa pemrograman yang digunakan untuk mendesain halaman web (Salamah, 2021). CSS menggunakan karakter yang dikenal dengan id dan class. CSS dapat mengubah *font*, ukuran *font*, dan format *font*. CSS juga dapat mengatur ukuran tata letak, lebar, tinggi, warna elemen, mengubah tampilan formulir, membuat halaman web yang responsif, dan masih banyak lagi yang dapat dilakukan CSS.

2.2.5 Laragon

Laragon adalah perangkat lunak bebas yang mendukung banyak sistem operasi dan berfungsi sebagai server hosting mandiri/lokal (Harianto et al., 2019). Laragon juga merupakan lingkungan pengembangan universal yang memiliki fungsi serupa dengan XAMPP, tetapi terdapat manfaat tambahan antara lain yaitu sangat cepat, mudah untuk digunakan, dan sangat ringan. Lingkungan Laragon berbasis Windows, dan mereka memiliki program untuk membantu pengembangan web modern seperti Django, MEAN, Laravel, Ruby on Rails, Spring Boot, dan Flask serta adapun kemampuan untuk menggunakan MongoDB, redis, Ruby, Node.js, MySQL, PHP, PostgreSQL, Java, Python, dan Memcached.

2.2.6 Bootstrap

Bootstrap adalah *library framework* CSS dengan komponen siap pakai seperti kelas. Karena itulah framework ini sangat berguna bagi para developer, khususnya

developer web front-end, karena mereka hanya memanggil satu class dan tidak perlu mengkodekan CSS dari awal (Sopyana, 2020). *Framework* ini banyak digunakan oleh *front-end* developer, namun tidak menghalangi untuk digunakan oleh *back-end* developer juga. Keuntungan dari Bootstrap adalah tidak hanya membuat tampilan statis, tetapi juga dapat membuat tampilan dan animasi dinamis dengan plugin JavaScript (Somya, 2018).

2.2.7 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web dimana database adalah sumber dan pengelola datanya (Harianto et al., 2019). MySQL juga bisa dibilang perangkat lunak *DataBase Management System* (DBMS), sistem manajemen basis data sumber terbuka yang dilisensikan dalam dua cara berbeda yaitu perangkat lunak gratis dan perangkat berbagi. Bagi para pengembang, MySQL sangat banyak sekali yang menggunakan *software* tersebut dalam membuat *database* dengan beberapa alasan, seperti berikut.

1) MySQL adalah sistem *database* cepat yang ideal untuk usaha kecil hingga menengah.
2) MySQL bisa dikatakan sistem manajemen basis data yang *open source* dan tidak dikenai biaya. Dengan demikian ada juga versi komersial yang telah disempurnakan dengan kemampuan khusus dan dukungan teknis yang disediakan oleh MySQL.
3) MySQL juga dapat memproses *database* dengan ukuran lebih dari 50 juta catatan.

2.2.8 *Visual Studio Code*

Visual Studio Code merupakan aplikasi editor teks yang ringan dan kuat juga tersedia untuk berbagai platform, ini berarti juga tersedia untuk versi Linux, Mac, dan Windows (Salamah, 2021). Aplikasi ini mendukung banyak bahasa pemrograman seperti JavaScript, TypeScript dan Node.js dan masih banyak lagi. Bahasa pemrograman lainnya yang dapat di program dengan bantuan plugin yang dapat diinstal di dalam aplikasi *Visual Studio Code* seperti C++, C, Python, Go, Java, dan lain-lain. Banyak sekali fitur yang disertakan dalam aplikasi Visual Studio Code termasuk *Intellisense, Git Integration, Debugging.* Aplikasi ini juga gratis dan *open*

source, ini memungkinkan pengguna aplikasi untuk melihat kode dan berkontribusi pada pengembangannya.

2.2.9 *Unified Modelling Language*

UML atau *Unified Modeling Language* adalah metode terbuka untuk mendefinisikan, memvisualisasikan, membuat, dan mendokumentasikan sistem perangkat lunak yang akan dikembangkan (Sari & Utami, 2021). Diagram UML digunakan sebagai alat yang memfasilitasi komunikasi antara beberapa aspek sistem yang berbeda melalui sejumlah elemen grafis yang dapat digabungkan menjadi diagram. Diagram UML memiliki berbagai macam diagram yang dapat mengakomodir berbagai perspektif pada suatu perangkat lunak yang akan dibangun. Diagram UML terdiri dari beberapa diagram diantaranya:

a) *Use Case Diagram*

Use Case Diagram adalah representasi dari perilaku yang dimaksudkan dari sistem informasi yang akan dibuat. Contoh menggambarkan skenario dimana satu atau lebih peserta berinteraksi dengan sistem yang akan dibuat. Kasus penggunaan digunakan untuk menentukan fungsi apa yang ada dalam sistem informasi dan siapa yang berhak memanfaatkannya.

b) *Activity Diagram*

Flowchart dan *Activity Diagram* proses menggambarkan proses atau alur kerja dari suatu sistem atau proses bisnis atau menu dalam perangkat lunak. *Activity Diagram* menggambarkan aktivitas sistem, bukan individu yang terlibat sehingga aktivitas yang dapat dilakukan oleh sistem dicantumkan.

c) Sequence Diagram

Sequence Diagram adalah diagram yang menggambarkan hubungan antar objek dan menunjukkan komunikasi antar objek tersebut. *Sequence Diagram* diinginkan untuk menggambarkan perilaku skenario dan menggambarkan interaksi antara entitas dan sistem, termasuk metode komunikasi yang digunakan selama interaksi. Semua pesan dicantumkan dalam urutan eksekusinya. *Sequence Diagram* menggambarkan perilaku objek dalam kasus penggunaan dengan menggambarkan masa pakai objek dan pesan yang dipertukarkan antar objek.

d) Entity Relationship Diagram

Entity Relationship Diagram merupakan model yang menerangkan hubungan antara data dan objek dalam basis data yang mempunyai hubungan antara satu sama lain.

2.2.10 *Priority Scheduling*

Priority Scheduling adalah metode penjadwalan yang memprioritaskan proses dengan pioritas tertinggi (Fachri et al., 2020). Penjadwalan preemptive apabila proses baru lebih penting dibandingkan proses yang sedang berjalan, maka proses yang sedang berjalan dibatalkan dan CPU dialihkan ke proses baru. Adapun perhitungan algoritma *Priority Scheduling* sebagai berikut:

Proses	Waktu Kedatangan	Prioritas
P1	10	3
P2	2	4
P3	5	3
P4	2	1
P5	7	2

P2	P5	P1	P3	P4

0 2 9 19 24 26

Waktu tunggu pada P1 yaitu 9, P2 adalah 0, P3 adalah 19, P4 adalah 24, dan P5 yaitu 2 sehingga rata-rata waktu tunggu adalah (9+0+19+24+2)/5 = 10,8 jam.

2.2.11 Metode *Prototype*

Prototype adalah aplikasi yang menjelaskan atau mendemonstrasikan beberapa aspek dari aplikasi yang berfungsi (Sari & Utami, 2021:286). Metode *prototype* bisa juga dibilang metode untuk pengembangan perangkat lunak yang melibatkan pembuatan program secara cepat dan bertahap yang dapat dievaluasi oleh pengguna. Metode *prototype* ini dibagi menjadi lima tahap yang masing-masing memiliki ciri pendekatan yang berbeda, yaitu.

1) Mulailah dengan mengumpulkan informasi dari klien yang akan dimasukkan ke dalam proses pengembangan.
2) Setelah itu, tim akan merancang sistem dan mengembangkan *prototype* perangkat lunak. Hasilnya kemudian diteruskan ke klien untuk evaluasi.

3) Jika timbul masalah, tim akan mendesain ulang sistem hingga benar-benar bermanfaat bagi pelanggan.
4) Apabila revisi sudah selesai dan telah disetujui, berarti perangkat lunak sudah siap diterjemahkan menjadi perangkat keras.
5) Setelah itu dilanjutkan dengan proses uji coba yang melibatkan beberapa kali revisi sebelum dapat dimanfaatkan.

2.2.12 *Shortest Job First* (SJF) *Scheduling*

Shortest Job First (SJF) adalah penggabungan di setiap proses yang panjang dari kerusakan CPU berikutnya. Proses yang panjang digunakan untuk menjadwalkan proses pada waktu terpendek. Pada algoritma ini, setiap proses dalam *ready queue* akan dieksekusi berdasarkan *burst time* terpendek. Hal ini mengakibatkan waktu tunggu yang cepat untuk setiap prosesnya, akibatnya waktu tunggu rata-ratanya menjadi cepat. Adapun contoh perhitungan algoritma *Shortest Job First* sebagai berikut:

Tabel 2.2 Perhitungan Algoritma *Shortest Job First*

Proses	Waktu Kedatangan	Waktu Penyelesaian
P1	2	6
P2	4	4
P3	6	8
P4	8	5

Pada tabel 2.2 merupakan contoh proses perhitungan algoritma *Shortest Job First*. Dalam sebuah sistem pemrosesan, terdapat empat proses yang akan diproses secara berurutan. Proses pertama P1 akan diproses terlebih dahulu karena tidak ada antrian sebelumnya. Kemudian, satu detik setelah itu, muncul proses kedua P2 dengan waktu penyelesaian 4 detik, sehingga CPU beralih untuk menyelesaikan proses P2. Pada detik kedua dan ketiga, muncul proses ketiga P3 dan keempat P4 secara berturut-turut. Namun, kedua proses tersebut membutuhkan waktu lebih lama untuk diselesaikan dibandingkan dengan P2, sehingga CPU tetap menyelesaikan P2 terlebih dahulu. Setelah menyelesaikan P2, CPU beralih ke P4 dengan waktu penyelesaian 5 detik. Terakhir, CPU akan menyelesaikan P1 dan P3 secara berurutan sampai kedua proses tersebut selesai

2.2.13 Codeigniter 4

Codeigniter adalah sebuah *framework* pengembangan aplikasi web menggunakan bahasa pemrograman PHP yang memiliki fleksibilitas tinggi dan dapat disesuaikan dengan kebutuhan pengembangan. *Framework* ini memiliki inti yang lengkap untuk memungkinkan pembuatan sistem yang berfungsi dengan baik dan tersedia struktur aplikasi yang baik, standar coding, praktik terbaik, pola desain, dan fungsi umum yang berguna. Peneliti menggunakan Codeigniter versi 4 untuk mengembangkan penelitian yang sedang dikembangkan.

2.2.14 Black-Box

Metode *black box* adalah metode pengujian yang diterapkan pada implementasi kode yang hanya menerima *input* dan *output* (Sari & Utami, 2021:284). Pengujian *black box* melakukan bentuk pengujian yang kurang memiliki pengetahuan tentang struktur internal sistem atau komponen yang diuji. Ini juga dapat dikenal sebagai pengujian perilaku, pengujian berbasis spesifikasi, pengujian *input* atau *output*, dan pengujian fungsional. Pada metode *black box* terdapat ciri-ciri, seperti berikut.

1. Pengujian *black box* berkaitan dengan persyaratan fungsional perangkat lunak, hal ini dicapai dengan mengikuti spesifikasi persyaratan perangkat lunak, dan
2. Pengujian *black box* bukanlah pengganti pengujian *white box*. Selain itu, ini merupakan pendekatan pelengkap untuk mengoreksi kesalahan dengan kelas metode *white box* yang berbeda.

Kegunaan pengujian *black box* itu sendiri dimaksudkan untuk memudahkan deteksi awal kesalahan atau kekurangan pada aplikasi yang sedang diuji, peneliti memeriksa fungsional fitur sistem, sebagai berikut.

Tabel 3.1 Fitur/halaman yang diuji dengan *Black-box*

No.	Fitur/Halaman
1.	Tampilan Halaman *Login*
2.	Tampilan Halaman *Home*
3.	Tampilan Halaman *Form Request* IT
4.	Tampilan Halaman *Request Form*
5.	Tampilan Halaman *Detail Request*
6.	Tampilan Halaman *Monitoring*

2.2.15 White-Box

White Box didasarkan pada penggunaan struktur kontrol desain program prosedural untuk membagi pengujian menjadi beberapa kasus pengujian untuk memeriksa detail desain (Sa`ad, 2020). Metode ini dinamai dengan demikian menurut *software*, di pandangan penguji kotak berwarna putih atau transparan yang terlihat jelas oleh pandangan penguji. Mereka semua berbagi konsep pengujian pengoperasian perangkat lunak yang benar dengan menggunakan struktur implementasi sebagai contoh. Tujuan dari pengujian *white box* yaitu untuk memastikan bahwa komponen berfungsi sebagaimana mestinya. Adapun kelebihan yang didapat oleh penguji dalam menggunakan metode *white box,* sebagai berikut.

1) Pengujian ini menggunakan sintaks 'IF' dan pengulangan. Prosedur berikut merupakan metode *white box*, nantinya pengujian ini akan mencari dan
2) mencari ketika proses perulangan telah berakhir.
3) Mendeteksi dan mengidentifikasi bahasa yang dianggap sensitif pada huruf besar-kecil.

Metode pengujian *white box* digunakan untuk mengevaluasi kode untuk setiap fitur sebagai berikut.

Tabel 3.2 Fitur/halaman yang diuji dengan *White-box*

No.	Fitur/Halaman
1.	Pengisian *form request* IT pada fitur *Helpdesk*
2.	Pemilihan kategori permintaan sesuai dengan kebutuhan
3.	Melihat detail pengajuan setelah mengirim *form request* IT
4.	Pemberian *approval* pengajuan permintaan
5.	Staf IT dapat mengambil projek yang telah tersedia
6.	Penugasan projek untuk Staf IT
7.	Validasi projek yang sudah diselesaikan