

BAB II

TINJAUAN PUSTAKA

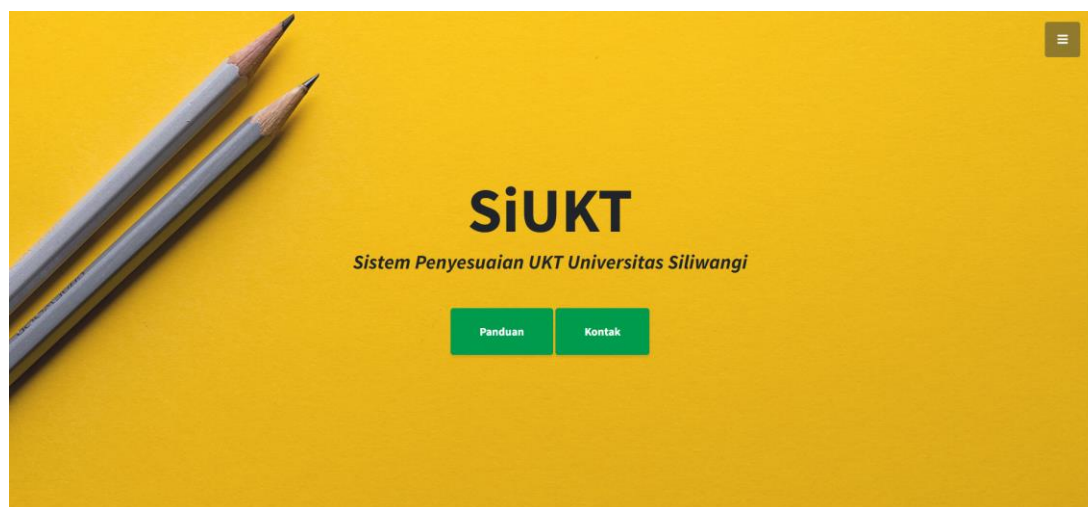
2.1 Landasan Teori

2.1.1 Aplikasi SiUKT

SiUKT adalah sistem informasi yang disediakan oleh Universitas Siliwangi, sistem ini memungkinkan Mahasiswa untuk melakukan pengajuan penyesuaian Uang Kuliah Tunggal. Sistem ini dikembangkan oleh UPA TIK Universitas Siliwangi. Tujuan SiUKT ini adalah untuk memberikan keringanan biaya kuliah bagi mahasiswa yang kurang mampu secara ekonomi. Ada beberapa fitur pada SiUKT, antara lain :

- Pengajuan permohonan penyesuaian UKT
- Verifikasi data permohonan penyesuaian UKT
- Pengambilan keputusan penyesuaian UKT
- Pemberitahuan hasil penyesuaian UKT

Adapun kelebihan dari sistem yaitu memudahkan mahasiswa untuk melakukan pengajuan penyesuaian UKT secara online (*Sistem Informasi Uang Kuliah Tunggal (SiUKT)*, n.d.). Berikut tampilan utama *website* SiUKT terdapat pada gambar 2.1



Gambar 2. 1 Beranda web SiUKT

2.1.2 *Software Testing*

Pengujian perangkat lunak atau *software testing* merupakan tahap yang biasanya dilakukan setelah implementasi didalam proses Pembangunan *software*(Arlinta Christy Barus et al., 2021). Pengujian perangkat lunak juga sangat dibutuhkan untuk dapat memastikan aplikasi yang sudah maupun sedang dibuat itu dapat berjalan sesuai dengan fungsionalitasnya. Pengembang maupun penguji *software* harus menyiapkan waktu khusus untuk melakukan pengujian pada program yang sudah maupun sedang dibuat agar kesalahan atau kekurangannya dapat dideteksi sejak awal dan dapat dikoreksi secepatnya (Desy Intan Permatasari et al., 2020).

2.1.3 *Automation Testing*

Automation testing atau Pengujian otomatis adalah proses menjalankan kasus uji tanpa intervensi manual. Pengujian ini dilakukan dengan merekam dan memutar kembali tindakan yang dilakukan oleh penguji. Hal ini memungkinkan penguji untuk mengulangi pekerjaan mereka di setiap siklus pengujian (Feng & Zhuang,

2007). penggunaan pengujian otomatis disini dapat mengurangi Upaya pengujian dan dapat mencapai pelaksanaan pengujian yang efisien.

2.1.4 Performance Testing

Performance Testing adalah suatu pengujian yang bertujuan untuk memastikan *software* akan bekerja dengan baik dibawah beban kerja yang diharapkan oleh pengembang. Tujuan utama dari pengujian ini adalah bukan untuk mencari *bug*, tapi untuk mengeliminasi *performance bottleneck*(Desy Intan Permatasari et al., 2020).

- Pengujian ini berfokus pada beberapa faktor, yaitu :
- *Speed* – untuk menentukan apakah aplikasi yang diuji merespon dengan cepat
- *Scalability* – menentukan apakah jumlah maksimal pengguna *load* dapat ditangani
- *Stability* – menentukan apakah aplikasi stabil dengan pengujian berbagai beban yang diuji.

2.1.5 Load Test

Load Testing merupakan salah satu metode dalam pengujian *performance*. Metode ini mengukur respon sistem dalam berbagai kondisi *load*. Pengujian ini berfungsi untuk mengevaluasi perilaku *software* saat diakses secara bersamaan oleh sejumlah pengguna(Desy Intan Permatasari et al., 2020).

Load Testing diperlukan untuk mensimulasikan akses aplikasi secara bersamaan. Metode simulasi ini lebih efektif dan tepat dibandingkan dengan pengujian yang melibatkan ribuan orang untuk mengakses aplikasi secara

bersamaan. Pengujian ini menerapkan skenario berdasarkan kondisi nyata, bertujuan agar aplikasi yang sedang atau telah dikembangkan dapat berfungsi optimal saat digunakan oleh para penggunanya nanti (Arlinta Christy Barus et al., 2021).

2.1.6 API (*Application Programming Interface*)

Application Programming Interface (API) dapat diartikan sebagai sebuah *tools* yang membuat beberapa sistem bisa saling terhubung (Galindra Wardhana et al., 2020). API juga dapat diartikan sebagai konsep fungsi antarmuka pemrograman aplikasi yang menjadi salah satu cara untuk suatu sistem atau aplikasi dapat diakses dan dimanfaatkan oleh sistem atau aplikasi lain dan bisa saling berhubungan antar sistem walaupun berbeda platform (Muri et al., 2019). *Application Programming Interface* umumnya disebut juga sebagai *Web Application Programming Interface Server* atau *Web service*.

2.1.7 API Testing

API *testing* merupakan salah satu bentuk pengujian perangkat lunak yang ditujukan untuk menguji fungsi, kehandalan, kinerja, dan keamanan API secara langsung. Pengujian ini dapat dianggap sebagai bagian dari *integration testing* untuk memastikan kesesuaian sistem yang diuji dengan standar yang berlaku. Proses API *testing* melibatkan penggunaan *software* untuk mengirim permintaan ke API dan mencatat respon sistem. Kode pengujian API akan mengirimkan permintaan spesifik ke API, lalu mencatat respon aktual sistem dan respon yang diharapkan oleh penguji. Dalam otomatisasi API *testing*, para penguji umumnya menggunakan *tools* khusus seperti Postman atau Jmeter Studio. Tujuan utama

otomasi adalah untuk mengotomatiskan pemeriksaan aplikasi guna memastikan bahwa aplikasi dikembangkan sesuai dengan kebutuhan yang telah ditetapkan (Arlinta Christy Barus et al., 2021).

2.1.8 *Baseline Response time*

Baseline merupakan penetapan suatu standar atau acuan. Dalam penelitian ini, acuan yang dimaksud adalah kinerja *web service* pada *web application* di Universitas Siliwangi. Setelah menetapkan *benchmark* kinerja *web service* Universitas Siliwangi, peneliti akan menggunakannya sebagai tolok ukur untuk mengevaluasi kelayakan suatu *web service* untuk diintegrasikan ke dalam *server* Universitas Siliwangi.

. Parameter *benchmark* yang ditetapkan oleh peneliti adalah *response time web service* pada *website* saat menerima permintaan dari pengguna. *Response time* didefinisikan sebagai durasi yang dibutuhkan sistem untuk merespons permintaan pengguna. Pengukuran *response time* dimulai saat pengguna mengirimkan permintaan dan berakhir ketika aplikasi mengindikasikan bahwa permintaan tersebut telah selesai diproses.

2.1.9 *Response time*

Response time merupakan waktu yang dibutuhkan sistem untuk menanggapi permintaan dari pengguna. *Response time* dimulai dari pengguna mengirim permintaan dan berakhir saat aplikasi menyatakan bahwa permintaan yang dikirim oleh pengguna telah selesai dilakukan.

2.1.10 Jmeter

Jmeter adalah sebuah alat pengujian beban yang berbasis desktop dan open source, dikembangkan oleh Apache, yang digunakan untuk menguji performa dan kinerja sistem saat menghadapi beban yang tinggi atau penggunaan yang besar secara bersamaan. Jmeter dirancang untuk menguji kemampuan sistem dalam menangani jumlah pengguna, transaksi, atau permintaan yang besar secara bersamaan dan untuk memastikan sistem dapat menangani beban yang tinggi tanpa mengalami overload atau kinerja yang menurun. Dengan menggunakan JMeter, pengembang dapat menguji sistem secara realistis dan memastikan bahwa sistem dapat berfungsi dengan baik di bawah beban yang tinggi dan berbagai situasi penggunaan yang berbeda-beda.

2.2 Penelitian Terkait

Terdapat beberapa penelitian terkait yang sudah dilakukan sebelumnya. Tabel 2.1 di bawah ini menjelaskan mengenai metode yang digunakan serta hasil dari penelitian terkait tersebut

Tabel 2. 1 Penelitian Terkait

No	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
1	Adaptive REST API Testing with Reinforcement Learning	(Kim et al., 2023)	Reinforcement Learning	Penelitian dilakukan untuk melakukan pengujian REST Api menggunakan pendekatan berbasis <i>Reinforcement Learning</i> untuk menilai efektivitas, efesiensi dan kemampuan deteksi, dan kesimpulan yang didapat yaitu menggunakan pendekatan tersebut lebih unggul dibandingkan dengan alat pengujian REST API terkini yang menghasilkan lebih banyak permintaan yang kurang valid dan dapat memicu kesalahan.
2	Automation Testing Api, Android, dan Website Menggunakan Serenity Bdd Pada Software Sistem Manajemen Rumah Sakit	(Saputra & Stefanie, 2023)	Serenity Bdd	Pada penelitian ini mengambil masalah dalam pengujian otomatis pada <i>software</i> atau aplikasi Hospital Management System menggunakan <i>serenity cucumber dan Jenkins. Serenity Framework</i> yang dimanfaatkan dalam dokumen ini menghasilkan pengujian pengujian otomatis kemampuannya lebih baik dan berhasil dari pada pengujian secara manual. Dalam hasil percobaan dapat ditarik kesimpulan bahwa automated testing menggunakan <i>serenity</i> lebih baik dari pada <i>manual testing</i> .

Tabel 2. 1 Penelitian Terkait Lanjutan

No	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
3	Pengujian Api Untuk Perbaikan Performansi Aplikasi Ditenun	(Arlinta Christy Barus et al., 2021)	Load testing	Pada penelitian ini dilakukannya pengujian API menggunakan metode <i>Load Test</i> , dimana 8 dari 9 API mampu memberikan permormasi yang baik dalam menerima <i>request</i> dan mengembalikan respon dimana meskipun API tersebut mengalami peningkatan <i>response time</i> yang mengindentifikasikan bahwa performa API mengalami penurunan berdasarkan jumlah <i>user</i> yang melakukan <i>request</i> .
4	Pengujian Performa API (Application Programming Interface) dengan Metode Load Testing	(Setiawan, Made, et al., 2022)	Load testing	Pada penelitian ini dilakukan pengujian menggunakan metode <i>load testing</i> pada beberapa <i>endpoint</i> API dengan variasi beban pada masing masing <i>endpoint</i> . Dan hasil pengujian menunjukan bahwa performa API mengalami penurunan seiring dengan bertambahnya beban <i>request</i> dari <i>user</i> , pada <i>endpoint</i> API dengan beban request tertinggi, respon time API meningkat dari yang awalnya 200ms menjadi 1000ms dan Pada <i>endpoint</i> API dengan beban <i>request</i> tertinggi, <i>throughput</i> API menurun dari 1000 request per detik menjadi 100 <i>request</i> per detik.

Tabel 2. 1 Penelitian Terkait Lanjutan

No	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
5	Implementasi metode pengujian equivalence partitioning pada pengembangan RESTful API Sistem Informasi Klinik Pratama UPN “Veteran” Yogyakarta.	(Iskhak & Rizkika, 2021)	equivalence partitioning	Pada penelitian ini dilakukan pengujian RESTful API Sistem Informasi Klinik UPN Veteran Yogyakarta menggunakan metode <i>black box</i> dengan teknik <i>equivalence partitioning</i> , dapat disimpulkan bahwa RESTful API telah berhasil dikembangkan dengan baik ditunjukkan dengan kasus - kasus uji yang telah dilakukan berhasil sesuai dengan yang diharapkan. Adapun untuk penelitian RESTful API selanjutnya penulis memberikan saran agar diterapkan teknik pengujian <i>black box testing</i> lainnya seperti <i>behavior testing</i> untuk menemukan kesalahan yang terjadi pada sistem yang dibuat dan membuat kualitas sistem menjadi lebih baik
6	Pengujian Stress Testing API Sistem Pelayanan dengan Apache JMeter	(Luh et al., 2021)	Stress testing	Penelitian dilakukan dengan melakukan pengujian <i>stress testing</i> API dengan menggunakan apache JMeter untuk lebih mengoptimalkan sistem dan memeriksa ketahanan dari sistem. Hasil yang didapatkan pada pengujian adalah pada bagian rata-rata waktu pengujian (<i>average</i>) mendapatkan nilai yang konstan, pada bagian <i>throughput</i> mendapatkan nilai yang meningkat sebanding dengan jumlah <i>request</i> sampelnya, dan pada bagian <i>deviation</i> mendapatkan nilai yang menurun sehingga ini membuktikan bahwa sistem mempunyai ketahanan yang lumayan baik dimana hasil grafik nilai <i>average</i> , <i>throughput</i> , dan <i>deviation</i> mendapat nilai yang baik.

Tabel 2. 1 Penelitian Terkait Lanjutan

No	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
7	Automated Functional Testing pada API menggunakan Keyword Driven Framework	(Yutia & Satrinia, 2021)	Keyword Driven Framework	Pada penelitian ini dilakukannya pengujian <i>functional</i> untuk memastikan fungsi API berjalan dengan baik dapat lebih mudah dilakukan dengan <i>automation testing</i> . Salah satu metode pada <i>automation testing</i> yaitu <i>keyword driven framework</i> dengan menggunakan <i>keyword</i> untuk mendeskripsikan langkah-langkah pengujian. Untuk hasil dari penelitian ini yaitu penerapan <i>Tools Robot Framework</i> dalam pengujian API telah berhasil dalam menghasilkan <i>report</i> pengujian dan test case yang dieksekusi berjalan lancar.
8	Pengujian Implementasi Rest Api Pada Website Sistem Pencarian Informasi Produk Fashion Di Shopee	(Umami & Ningrum, 2020)	Black box testing & white box testing	Pada Penelitian Ini Dilakukan Pengujian Rest Api Pada Website Sistem Pencarian Informasi Produk Fashion Di Shopee yang diperoleh kesimpulan bahwa REST API yang diterapkan pada penelitian ini berhasil mengintegrasikan <i>service</i> antar <i>website</i> sehingga dapat menampilkan data keseluruhan dari data yang telah diambil dari <i>web</i> Shopee. Penelitian ini menghasilkan kesesuaian antara <i>keyword</i> yang diinputkan dengan informasi yang ditampilkan dengan akurasi sebesar 92,8% untuk <i>keyword</i> “murah” dan akurasi sebesar 85,7%. Dengan semakin tingginya kesesuaian informasi yang ditampilkan sesuai dengan <i>keyword</i> yang diinputkan oleh <i>user</i> , maka semakin memudahkan <i>user</i> dalam menerima informasi dan melakukan transaksi di Shopee.

Tabel 2. 1 Penelitian Terkait Lanjutan

No	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
9	Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian	(Desy Intan Permatasari et al., 2020)	Load Testing	Pada penelitian ini dilakukan pengujian API menggunakan metode <i>load testing</i> yang bertujuan untuk membantu menentukan bagaimana <i>software</i> berperilaku Ketika <i>user</i> mengakses <i>software</i> secara bersamaan dan untuk <i>test case</i> samplingnya menggunakan <i>gorilla testing</i> dimana modul program disini berulang kali diuji untuk memastikan bahwa modul tersebut berfungsi dengan benar dan tidak ada <i>bug</i> , untuk hasil pengujian disini bahwa target pengujian <i>loading time</i> dan <i>process memory</i> sudah terpenuhi.
10	Automation Testing Tool Dalam Pengujian Aplikasi Belajar Tajwid Pada Platform Android	(Katarina & Windia Ambarsari, 2019)	Black box testing	Pada penelitian ini dilakukan pengujian aplikasi belajar tajwid menggunakan katalon, tujuannya yaitu untuk menganalisis efektivitas aplikasi yang sudah dibangun. Hasil dari penelitian didapatkan terjadi tiga kesalahan didalam aplikasi, diantaranya ketidaksesuai ukuran layar dalam mode <i>landscape</i> , terdapatnya kegagalan eksekusi dikarenakan Katalon tidak dapat mendeteksi objek yang tidak masuk dalam tangkap layar <i>Device View</i> , dan objek suara yang tidak berhenti jika sudah pindah halaman layar.

Tabel 2. 1 Penelitian Terkait Lanjutan

No	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
11	Automated Warship Software Testing System Based on LoadRunner Automation API	(Han et al., 2018)	LoadRunner	Pada penelitian ini dilakukan pengujian sistem perangkat lunak kapal perang otomatis dimana ada isu isu utama termasuk formalisasi protokol, pembuatan kasus uji otomatis, penggerak kasus uji otomatis, dan konversi antarmuka perangkat keras. Keuntungan hasil pengujian otomatis dibandingkan dengan pengujian manual sebelumnya sebagai berikut: - Memfasilitasi uji regresi, dimana disini menggunakan skrip pengujian yang sudah ada sehingga memudahkan pengujian regresi - Mengurangi waktu pengujian, dengan menggunakan skrip pengujian otomatis, kami dapat melakukan pengujian yang berbeda berdasarkan kasus pengujian yang sama, yang dapat digunakan Kembali sehingga meningkatkan efisiensi. - Dapat mensimulasikan beban kerja yang besar, mampu menjalankan pengujian yang sulit atau tidak mungkin secara manual Hal ini menunjukkan bahwa sistem pengujian perangkat lunak otomatis layak untuk perangkat lunak kapal perang. Namun, pengujian interaksi manusia-komputer belum dipertimbangkan dalam sistem pengujian perangkat lunak ini, dan jenis antarmuka perangkat keras yang didukung oleh sistem pengujian perangkat lunak ini tidak banyak, yang memberikan arahan untuk penelitian di masa depan.

Pada Tabel 2.2 memperlihatkan perbandingan keterbaruan penelitian yang dilakukan dengan penelitian terkait yang telah dilakukan sebelumnya.

Tabel 2. 2 Matriks Penelitian Terkait

No.	Judul Penelitian	Metode							
		Load Testing	Stress testing	Black box testing	White box testing	LoadRunner	Keyword driven	equivalence partitioning	Reinforcement Learning
1	Adaptive REST API Testing with Reinforcement Learning (Kim et al., 2023)								√
2	Automation Testing Api, Android, dan <i>Website</i> Menggunakan Serenity Bdd Pada Software Sistem Manajemen Rumah Sakit (Saputra & Stefanie, 2023)								√
3	Pengujian Performa API (Application Programming Interface) dengan Metode Load Testing (Setiawan, Made, et al., 2022)	√							

Tabel 2. 2 Matriks Penelitian Terkait Lanjutan

4	Pengujian Stress Testing API Sistem Pelayanan dengan Apache Jmeter (Luh et al., 2021)		√							
5	Pengujian Api <i>Website</i> Untuk Perbaikan Performansi Aplikasi Ditenun (Arlinta Christy Barus et al., 2021)	√								
6	Implementasi metode pengujian equivalence partitioning pada pengembangan RESTful API Sistem Informasi Klinik Pratama UPN “Veteran” Yogyakarta (Iskhak & Rizkika, 2021)							√		
7	Automated Functional Testing pada API menggunakan Keyword Driven Framework (Yutia & Satrinia, 2021)						√			
8	Pengujian Implementasi Rest Api Pada <i>Website</i> Sistem Pencarian Informasi Produk Fashion Di Shopee (Umami & Ningrum, 2020)			√	√					
9	Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian (Desy Intan Permatasari et al., 2020)	√								

Tabel 2.2 Matriks Penelitian Terkait Lanjutan

10	Automation Testing Tool Dalam Pengujian Aplikasi Belajar Tajwid Pada Platform Android (Katarina & Windia Ambarsari, 2019)			√						
11	Automated Warship Software Testing System Based on LoadRunner Automation API (Han et al., 2018)					√				