

BAB I. PENDAHULUAN

1.1 Latar Belakang

PT Microdata Indonesia merupakan perusahaan dengan penyedia jasa konsultasi dalam bidang teknologi informasi yang berfokus pada pengembangan produk TI (Teknologi Informasi). Produk yang dihasilkan ditargetkan menjadi solusi untuk kebutuhan berbagai segmen institusi pemerintah, universitas, industri *enterprise*, serta lembaga kesehatan dalam bidang teknologi. Berbekal pengalaman yang telah dimiliki sejak tahun 2010, microdata Indonesia mampu menarik banyak klien untuk mengembangkan teknologi informasi dari perusahaan besar sampai lembaga pemerintahan contohnya Kementerian Dalam Negeri (Deily, 2021).

Peraturan Menteri Dalam Negeri Republik Indonesia Nomor 1 Tahun 2021, Kementerian Dalam Negeri memiliki tanggung jawab dalam mengelola sistem dasar pengenaan pajak kendaraan dan bea (DPPKB) di Indonesia. Sistem dasar ini ditujukan untuk mempermudah dan memperjelas penghitungan jumlah pajak kendaraan yang wajib dibayarkan oleh pemilik kendaraan. Perhitungan jumlah pajak didasarkan atas berbagai faktor, seperti jenis kendaraan, kapasitas mesin, usia kendaraan, dan faktor lainnya yang relevan. Sistem yang baik harus melalui Pengukuran kualitas sebelum sebuah sistem akan digunakan oleh pengguna. Hal ini bertujuan untuk mengetahui kelemahan dari sistem agar data yang dihasilkan bisa sesuai dengan data yang dimasukkan setelah data di eksekusi (Hanifah, et.all, 2017)

Pengembangan sistem perhitungan dasar ini, pengujian merupakan tahap penting untuk memastikan keakuratan dan keandalan sistem. Pengujian otomatis dinilai lebih efektif dan mengurangi kemungkinan kesalahan manusia karena pengujian otomatis dirancang untuk mengikuti serangkaian langkah yang telah ditentukan secara ketat (Taufik et.all, 2023). Pengujian otomatis atau *automatic testing* merupakan kumpulan langkah yang digunakan untuk mengoperasikan sebuah program yang bertujuan menemukan *error*, *bug*, kesalahan pada program tersebut, prosedur pengoperasian program serta membandingkan perilaku pada

program dengan harapan menghasilkan produk bermutu tinggi (Priyaungga, et.all., 2020).

Salah satu Metode yang bisa digunakan dalam pengujian otomatis yakni *Equivalence Partitions* yang membagi domain *input* dari aplikasi ke dalam kelas-kelas data sehingga akan diperoleh *Test Case*. *Test Case* pada *Equivalence Partitions* dirancang berdasarkan evaluasi pada kelas *Equivalence Partitions* pada setiap kondisi *input* untuk menggambarkan *feedback* keadaan valid atau tidak valid. *Input* dapat berupa nilai *numeric*, *range* nilai, atau kumpulan nilai (Sasongko, dkk, 2017) (Arwaz, dkk., 2019). Metode ini adalah salah satu pendekatan yang umum digunakan dalam pengujian perangkat lunak. Pendekatan dilakukan dengan membagi kumpulan data masukan menjadi beberapa kelas kesetaraan, di mana setiap kelas dianggap memiliki karakteristik yang serupa dan dapat diuji dengan satu set data yang representatif (Siregar, 2020).

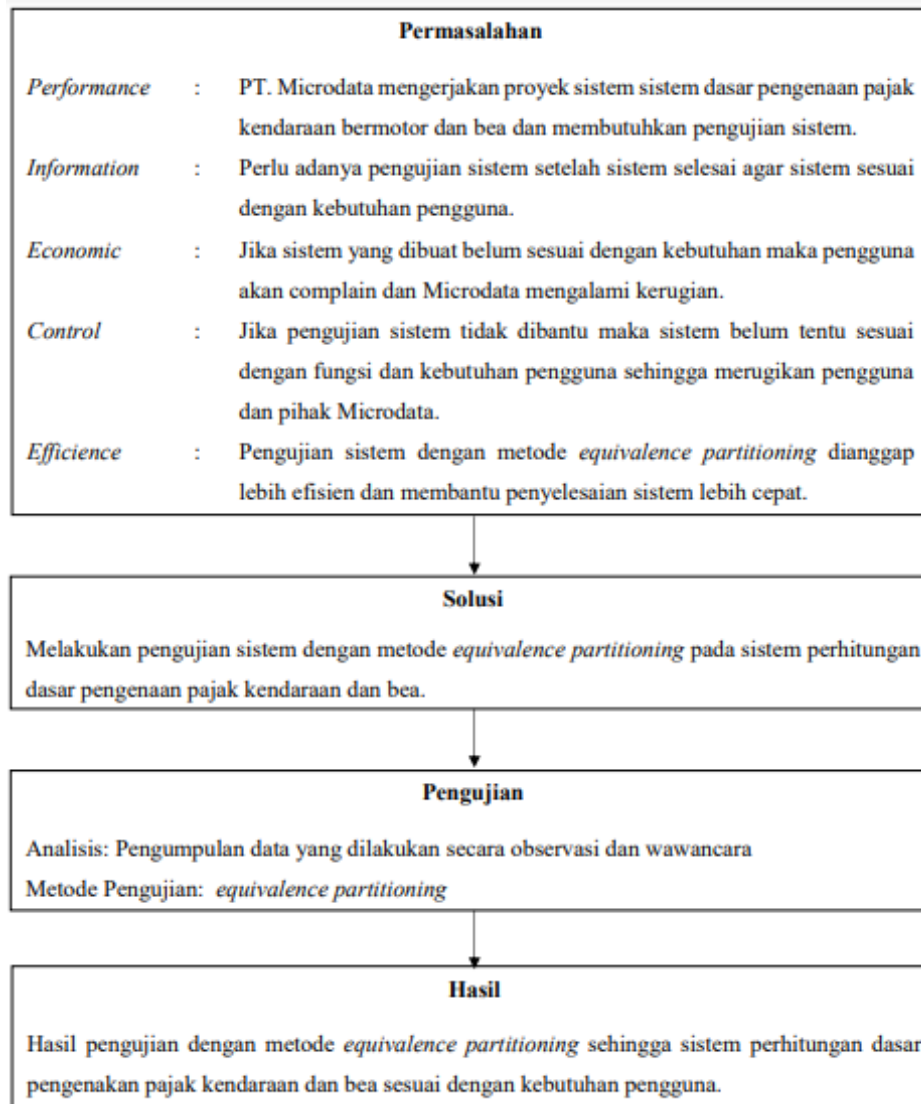
Melalui penerapan metode *equivalence partitioning* dalam pengujian otomatis, diharapkan sistem perhitungan dasar pengenaan pajak kendaraan dan bea dapat diuji secara menyeluruh dan efisien. Hal ini akan membantu mengidentifikasi dan memperbaiki kesalahan dalam sistem, sehingga dapat memastikan keakuratan perhitungan pajak dan menghindari kesalahan yang mungkin merugikan pemilik kendaraan dan instansi terkait.

1.2 Tujuan

Tujuan dilakukannya pembuatan tugas akhir ini untuk memastikan sistem yang dikembangkan memiliki tingkat akurasi yang tinggi dan dapat mengurangi kesalahan dalam perhitungan pengenaan pajak kendaraan bermotor dan bea balik nama.

1.3 Kerangka Pemikiran

Berdasarkan latar belakang masalah yang ada, dapat disimpulkan suatu kerangka pemikiran yang disajikan dalam Gambar 1.



Gambar 1. Kerangka Pemikiran

1.4 Manfaat

Manfaat yang dapat diperoleh dari pengerjaan Tugas Akhir Mahasiswa yaitu sebagai berikut:

1. Bagi pemilik membantu pengerjaan dalam hal melakukan pengujian sistem dengan metode *equivalence partitioning*.
2. Bagi pengguna dapat menggunakan sistem DPPKB secara optimal dan sesuai kebutuhan.
3. Diharapkan dapat menjadi bahan refrensi bagi pembaca.

BAB II. TINJAUAN PUSTAKA

2.1 *Equivalence Partitioning*

Blackbox testing adalah suatu metode yang digunakan untuk menguji aplikasi tanpa memerlukan pengetahuan tentang detail aplikasi, seperti *source code*. Dalam metode *Blackbox testing*, hanya dilakukan pemeriksaan terhadap hasil berdasarkan *input* yang diberikan (Sangsoko, et.all, 2021). Pengujian *Blackbox* yang menguraikan domain masukan dari program menjadi beberapa kelas data, dengan tujuan untuk mendapatkan *Test Case* disebut dengan *Equivalence partitioning* (Krismadi, et.all, 2019).

Teknik *equivalence partitioning* adalah metode yang digunakan untuk menggambarkan Jenis dan kondisi input yang dapat dikelompokkan menjadi beberapa kelas *ekuivalensi*, yakni yang valid dan yang tidak valid. Pengelompokan ini bertujuan agar jika terdapat kesalahan pada satu kasus uji dalam salah satu kelas, maka kesalahan tersebut juga akan terjadi pada kelas lainnya (Amalia, et.all, 2021).

Pengujian *software* secara menyeluruh dilaksanakan dengan menggunakan partisi setara, dimana penggunaan, manfaat, dan kesimpulan dari penggunaan *software* dievaluasi. Tujuan pengujian ini adalah untuk mengukur kualitas aplikasi dengan menganalisis dokumentasi pengujian *software* dan mendeteksi kesalahan dalam form (Hartono & sugiarti, 2022).

2.2 Pengujian Otomatis

Pengujian otomatis adalah jenis pengujian yang menggunakan bantuan tool testing atau menggunakan *script*. (Setiawan, 2023). Berbeda dengan pengujian manual yang dilakukan oleh manusia yang duduk di depan komputer dan secara hati-hati menjalankan pengujian secara *step by step*. *Software* pengujian otomatis memiliki *tool* untuk dapat merekam dan memutar kembali *test case* yang akan dikerjakan. Selain itu juga dapat mengubah data pada *test case* yang sudah ada atau yang baru dijalankan. (Sastra, 2022).

Pengujian otomatis merupakan langkah terbaik yang dapat dilakukan ketika tes yang dibutuhkan tidak cukup untuk dijalankan hanya satu kali. Pengujian secara otomatis dapat memberikan hasil yang maksimal dan juga mengurangi kesalahan

penginputan ataupun meminimalisir kurangnya efektivitas yang dilakukan oleh manusia sehingga mengakibatkan menurunnya kinerja sistem yang sedang berjalan (Cahyono dan Kosasih, 2021).

2.3 Sistem

Menurut Madiyanto & Ridho (2021), Sistem adalah hasil penggabungan dari berbagai elemen, komponen, atau variabel yang saling berinteraksi guna mencapai suatu tujuan tertentu. Sistem dapat diartikan sebagai kumpulan jaringan proses kerja yang saling terkait dan berkolaborasi untuk melaksanakan suatu tindakan. Sistem diperkenalkan dan diimplementasikan di dalam organisasi yang menggunakannya. Keberhasilan implementasi sistem dapat diukur berdasarkan penerimaan dan penggunaannya oleh para pengguna. Jika sistem diterima dan digunakan dengan baik, maka implementasinya bisa dianggap berhasil. Namun, jika pengguna menolak sistem yang diterapkan, maka bisa dikatakan bahwa sistem tersebut gagal (Agustin Hamdi 2018).

2.4 Pajak Kendaraan Bermotor

Pajak adalah kewajiban pembayaran yang harus dipenuhi oleh setiap wajib pajak terhadap objek pajak yang dimilikinya, dan penerimaan pajak tersebut dialokasikan kepada pemerintah sebagai sumber pendapatan negara utama saat ini (Pieterz, picauy, et.all, 2021). Pajak kendaraan bermotor (PKB) merupakan salah satu pajak daerah yang digunakan untuk membiayai pembangunan daerah provinsi. Pajak kendaraan bermotor merupakan jenis pajak yang dipungut oleh provinsi namun setiap kabupaten diberikan kewenangan untuk memungut PKB sendiri melalui Kantor Samsat seperti yang dituangkan dalam peraturan presiden Nomor 5 Tahun 2015 tentang Penyelenggaraan Sistem Administrasi Manunggal Satu Atap (SAMSAT). Tujuan kebijakan tersebut untuk memudahkan masyarakat dalam pembayaran pajak kendaraan bermotor di setiap kabupaten yang ada di Indonesia (Koel, et.all, 2019).

2.5 Kepabeanan

Kepabeanan adalah segala sesuatu yang berhubungan dengan pengawasan atas lalu lintas barang yang masuk atau keluar daerah pabean serta pemungutan bea masuk dan bea keluar (Undang-Undang Nomor 10 Tahun 1995 Tentang

Kepabeanan Dalam Satu Naskah - Pasal 1). Fokus utama kepabeanan adalah mengawasi barang dan mengenakan bea pada barang. Dalam pelaksanaannya, tugas pengawasan dan pemungutan bea ini dilakukan oleh Direktorat Jenderal Bea dan Cukai, sebuah bagian di bawah Kementerian Keuangan. Selain mengawasi barang impor dan ekspor, pejabat bea dan cukai juga berwenang untuk mengawasi barang tertentu (Sukarno & Arfin, A, 2022)

2.6 Katalon Studio

Katalon Studio adalah suatu alat pengujian otomatis yang menggunakan kemampuan *library Selenium* dan *Appium* yang keduanya sudah diinstal sebelumnya untuk menjalankan pengujian pada *mobile*. Katalon mendukung 3 *platform* pengujian secara automasi, yaitu *Web testing*, *API testing*, dan *Mobile testing*. Katalon sudah terintegrasi dengan beberapa teknologi luar seperti *qtest*, *JIRA*, *kobiton*, *github*, dan lain-lain (Hendri, dkk. 2022).

Studio Katalon menyediakan antarmuka ganda yang dapat digunakan untuk membuat kasus uji, termasuk tampilan manual dan tampilan skrip. Tampilan manual digunakan untuk pengguna yang kurang teknis dan tampilan skrip digunakan untuk pengguna yang berpengalaman dalam menulis tes otomasi sintaks yang kompleks. Sehingga tidak ada lagi kendala bagi tester dalam melakukan otomatisasi test (Hasanah, 2022).

2.3.1 Test suite

Menurut Ardi dan Putro (2020), *Test suite* merupakan sekumpulan *test case* yang siap untuk diuji secara bersamaan dengan *data binding* atau *data files*. Dalam Testing, tidak dapat menentukan suite di kode sumber pengujian, tetapi diwakili oleh satu file XML, sebagai suite adalah fitur eksekusi. Hal ini juga memungkinkan konfigurasi yang fleksibel dari tes yang akan dijalankan. Sebuah suite dapat berisi satu atau lebih tes (Pratiwi, 2021).

2.3.2 Test Case

Test Case atau juga bisa disebut dengan kasus uji adalah suatu rancangan atau rangkaian mengenai tindakan yang dilakukan oleh user untuk melakukan verifikasi terhadap fitur atau fungsi tertentu dari sebuah perangkat lunak (Praza, 2019). *Test case* sendiri bertindak sebagai titik mula dari sebuah pengujian perangkat lunak,

dengan menggunakan *test case*, user dapat mengetahui apakah fitur pada sistem berjalan dengan normal atau tidak (Muhammad, 2022).

2.7 Bug

Bug adalah masalah umum yang sering terjadi pada perangkat lunak. Ini merujuk pada kesalahan, cacat, kegagalan, atau kerusakan dalam perangkat lunak yang menyebabkan sistem berjalan tidak sesuai dengan yang diharapkan. *Bug* dalam perangkat lunak sering terjadi karena kesalahan yang tidak disengaja oleh para pengembang, baik dalam sumber kode (*source code*) maupun dalam desain perangkat lunak itu sendiri (Ridwan&Rochimah, 2018).





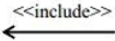
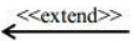
2.8 Web Browser

Web browser adalah perangkat lunak yang dibuat dengan menyimpan informasi apa pun seperti *URL history*, *search keyword*, *timestamp*, *password*, dan segala sesuatu yang dilakukan pengguna saat menjelajah di internet (Abdiati, et.all, 2021). Saat ini, terdapat banyak variasi web browser dengan berbagai mesin browser yang terintegrasi dalam perangkat lunak. Beberapa yang paling populer dan banyak digunakan meliputi *Chrome*, *Firefox*, *Seamonkey*, *Safari*, dan *Opera* (Rochmadi Tri, 2018).

2.9 Use Case Diagram

Pemodelan *Use Case* digunakan untuk merancang kebutuhan sistem fungsional. Dalam pendekatan ini, setiap *Use Case* diilustrasikan sebagai skenario yang dilakukan oleh aktor dalam batas sistem yang telah ditentukan. Tiap *Use Case* dihubungkan dengan garis notasi yang menggambarkan hubungan antara mereka (aliman wilyanti, 2021). Berikut adalah komponen *use case* diagram yang disajikan pada tabel 1.

Tabel 1. Komponen *Use Case Diagram*

simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat Ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>Use cas</i> lainnya jika suatu kondisi terpenuhi

2.10 Penelitian Terdahulu

Penelitian terdahulu merupakan riset yang telah dilakukan oleh peneliti sebelumnya untuk mengumpulkan data, informasi, atau bukti yang relevan dengan topik atau masalah yang sedang diteliti. Jurnal terkait disajikan pada Tabel. 2

Tabel 2. Penelitian Terdahulu

Nama dan Tahun	Judul	Metode Pengujian Sistem	Hasil
(Agustian, et.all, 2020)	“Implementasi Teknik <i>Equivalence Partitioning</i> pada Pengujian Aplikasi <i>E-learning</i> Berbasis Web”	<i>Equivalence Partitioning</i>	Hasil yang didapat dapat disimpulkan bahwa tehnik ini yang paling sesuai sebab metode ini menguji sistem dengan membagi ruang menjadi pilihan. Dalam hal ini apabila tidak dilakukan pengujian dapat membuat sistem tidak maksimal dan dalam menentukan tingkat integritas serta kerahasiaan metode lain

Lanjutan tabel 2. Penelitian Terdahulu

(Nirmala,Putra & Sudana, 2021)	“Penguujian <i>Black Box</i> Pada Sistem Terintegrasi Pembayaran Uang Kuliah Pada Universitas X Dengan Metode <i>Equivalence Partitions</i> ”	<i>Equivalence Partitioning</i>	Hasil pengujian yang telah dilakukan, keseluruhan Sistem Terintegrasi Pembayaran Uang Kuliah berjalan dengan baik. Namun, terdapat beberapa fitur yang kurang dan beberapa output sistem yang tidak sesuai
(Hendri, et.all, 2020)	“Penguujian Black Box pada Aplikasi Sistem Informasi Pengelolaan Masjid Menggunakan Teknik <i>Equivalence Partitions</i> ”	<i>Equivalence Partitioning</i>	disimpulkan dalam pengujian tidak ditemukan kesalahan
(Ikhlashi & Putro, 2019)	“Komparasi Dua Teknik Black Box Testing: <i>Equivalence Partitioning</i> dan <i>Boundary Value Analysis</i> ”	<i>Equivalence Partitioning & Boundary Value Analysis</i>	Hasil Perbandingan antara kedua metode tersebut adalah, <i>equivalence partitioning</i> dapat menguji pada tipe data range dan bebas, sedangkan <i>boundary value analysis</i> hanya dapat menguji tipe data range, <i>equivalence partitioning</i> cocok untuk mengeksplorasi semua kemungkinan berdasarkan kriteria, sedangkan <i>boundary value analysis</i> cocok untuk sistem yang kritis dan mengekspos masalah masukan user
(Amalia,et.all, 2022)	“Penguujian Black Box Menggunakan Teknik <i>Equivalence Partitions</i> Pada Aplikasi E-Learning Berbasis Web”	<i>Equivalence Partitioning</i>	Hasil penelitian menunjukkan bahwa aplikasi e-learning ini layar digunakan sebagai media pembelajaran jarak jauh atau online di Institut Teknologi Telkom Surabaya