

Rekayasa Ulang Sistem Pajak BPHTB Berbasis Agile dan DevOps Untuk E-Government

Hilyah Magdalena ^{1)*}, Jihan Jessika ²⁾

Institut Sains dan Bisnis Atmaluhur ^{1,2)}

hilyah@atmaluhur.com ^{1)*}, 2022500117@mahasiswa.atmaluhur.ac.id ²⁾

Abstrak

Layanan pungutan Bea Perolehan Hak atas Tanah dan Bangunan sebagai salah satu sumber pemasukan daerah yang ada di BAKUEDA Pangkalpinang. Layanan ini masih berjalan manual yang mengharuskan wajib pajak melengkapi berbagai berkas, lalu menunggu petugas lapangan survei objek pajak, menunggu verifikasi berkas dan hasil survei oleh Kabid/Kasi yang terkait, serta menunggu pelunasan dan pencetakan SSPD-BPHTB untuk wajib pajak, sampai menyiapkan laporan tahunan. Sistem yang selama ini ada direkayasa ulang dengan pendekatan Agile dan DevOps. Pendekatan ini adalah gabungan metode Agile yang mampu membangun sistem dengan identifikasi kebutuhan, sprint agile, dan pengembangan sistem iteratif. Sedangkan DevOps mendukung proses implementasi, pengujian dan evaluasi yang berkesinambungan. Hasil rekayasa sistem ini meningkatkan kualitas layanan pembayaran BPHTB, seperti kemudahan akses bagi wajib pajak, dan layanan yang lebih efisien bagi beberapa bagian yang terkait, seperti bagian pelayanan, petugas survei, dan Kabid/Kasi terkait. Layanan sistem berbasis web juga memperluas kesempatan bagi wajib pajak yang mengajukan BPHTB, sekaligus mampu memantau perkembanganajuan BPHTB sekaligus mengetahui estimasi waktu selesainya proses ajuan BPHTB secara online.

Kata kunci: BPHTB, Agile, DevOps

Abstract

[Agile and Development-Based Reengineering of the BPHTV Tax System for E-Government] The Land and Building Acquisition Tax (BPHTB) collection service is one of the regional revenue sources in the Pangkalpinang Regional Revenue Agency (BAKUEDA). This service is still run manually, requiring taxpayers to complete various documents, then wait for field officers to survey the taxable object, wait for file verification and survey results by the relevant Head of Division/Section, and wait for payment and printing of the SSPD-BPHTB for taxpayers, until preparing the annual report. The existing system was reengineered using an Agile and DevOps approach. This approach is a combination of Agile methods that are able to build a system through needs identification, agile sprints, and iterative system development. Meanwhile, DevOps supports the continuous implementation, testing, and evaluation process. The results of this system engineering improve the quality of BPHTB payment services, such as ease of access for taxpayers, and more efficient services for several related departments, such as the service department, survey officers, and the relevant Head of Division/Section. The web-based system service also expands opportunities for taxpayers who apply for BPHTB, while also allowing them to monitor the progress of BPHTB applications and find out the estimated completion time for the BPHTB application process online.

Keywords: BPHTB, Agile, DevOps

1. PENDAHULUAN

Badan Keuangan Daerah (BAKEUDA) Kota Pangkalpinang memegang peran penting dalam memastikan pengelolaan keuangan daerah berjalan tertib dan transparan. Lembaga ini menjadi pusat perencanaan, pelaksanaan, dan pengawasan anggaran, sekaligus mengatur administrasi keuangan yang mendukung roda pemerintahan kota. Dengan dukungan 220 personel yang terdiri dari aparatur sipil negara dan tenaga honorer, BAKEUDA berupaya menjalankan amanah sesuai ketentuan perundangan serta

mendorong terwujudnya tata kelola keuangan daerah yang efektif.

Salah satu layanan penting dalam tugas mengelola keuangan daerah yang menjadi tanggung jawab BAKEUDA adalah layanan pungutan Bea Perolehan Hak atas Tanah dan Bangunan (BPHTB). Pajak ini dikenakan atas setiap perolehan hak atas tanah maupun bangunan, sebagaimana diatur dalam ketentuan perundang-undangan. Regulasi terkait tertuang dalam Undang-Undang Nomor 28 Tahun 2009 tentang Pajak Daerah dan Retribusi Daerah, yang

kemudian diperbarui melalui Undang-Undang Nomor 1 Tahun 2022 tentang Hubungan Keuangan antara Pemerintah Pusat dan Pemerintah Daerah[1][2].

Wajib pajak yang melakukan proses pembayaran Bea Perolehan Hak atas Tanah dan Bangunan (BPHTB) akan menjalani proses perhitungan pajak berdasarkan selisih antara Nilai Perolehan Objek Pajak (NPOP) dan Nilai Perolehan Objek Pajak Tidak Kena Pajak (NPOPTKP), lalu mengisi Surat Setoran Pajak Daerah (SSPD-BPHTB), membayar pelunasan melalui bank atau lewat pembayaran online, kemudian verifikasi oleh instansi terkait, hingga penerbitan tanda bukti pelunasan sebagai syarat pembuatan akta peralihan hak. Meskipun alurnya telah diatur dengan jelas, namun pelaksanaan di lapangan masih terkedala beberapa hal, seperti dokumen pendukung yang tidak lengkap, pemahaman warga yang minim terhadap aturan BPHTB, sistem online yang tidak selalu tersedia, proses verifikasi yang lama, serta perubahan peraturan yang tak disadari oleh wajib pajak. Kondisi ini menunjukkan bahwa tata laksana BPHTB tidak hanya bergantung pada regulasi yang ada, tetapi juga pada efektivitas penerapan dan tingkat literasi pajak masyarakat [3][4].

Dalam praktiknya, pembayaran BPHTB di Kota Pangkalpinang umumnya menjadi tanggung jawab pembeli dan harus diselesaikan sebelum proses peralihan hak dicatatkan di Badan Pertanahan Nasional. Namun, pelaksanaannya di lapangan masih menghadapi kendala teknis, terutama pada proses administrasi dan pencarian data yang cenderung lambat karena masih banyak dilakukan secara manual. Kondisi ini menjadi perhatian tersendiri mengingat tingginya volume transaksi yang ditangani BAKEUDA, sehingga keterlambatan dalam pengelolaan data dapat berdampak pada pelayanan publik dan akurasi pengawasan pajak.

Layanan pembayaran pajak BPHTB sebagai salah satu sumber pemasukan daerah perlu ditingkatkan dengan sistem informasi. Sistem layanan berbasis web diharapkan mampu mempercepat administrasi mulai dari kemudahan unggah file pendukung, proses survei lapangan oleh petugas, sampai proses pencetakan dokumen bukti pembayaran.

Sistem layanan berbasis web pengelolaan BPHTB ini dirancang dengan pendekatan *Agile* dan *DevOps* karena mampu mengakomodasi perubahan kebutuhan sistem serta kebutuhan pengguna secara iteratif, mempercepat integrasi dan penerapan sistem, serta menjaga kualitas layanan yang efektif dalam meningkatkan layanan publik dan penerimaan pajak daerah.

Alasan lain memanfaatkan metode *Agile* dan *DevOps*, karena metode ini terbukti sesuai untuk sistem pembayaran di beberapa instansi, baik itu pembayaran pajak BPHTB ataupun layanan pembayaran lain yang bertujuan mendukung tata kelola pemerintah yang baik, seperti telah dilakukan di beberapa penelitian berikut ini.

Pendekatan *Agile Governance* pada Sistem E-Palapa Pajak Hotel di Bapenda Badung, menunjukkan layanan yang responsif, walaupun masih ada kendala teknis dan beberapa laporan keterlambatan pembayaran pajak, sehingga perlu memperhatikan upaya meningkatkan kepatuhan wajib pajak[5].

Berikutnya penerapan *agile government* digunakan pemerintah Indonesia untuk analisis ancaman resesi global 2023, menunjukkan implementasi *agile governance* mampu mendorong kebijakan penting seperti digitalisasi UMKM, desain APBN antisipatif, dan perbaikan iklim investasi[6].

SIPD berhasil menaikkan tingkat efisiensi layanan administrasi daerah agar menjadi layanan yang lebih cepat, minim kesalahan, laporan yang dapat disajikan dengan akurat, dan mudah diakses oleh warga. Kendala yang masih terasa adalah kurangnya dukungan infrastruktur dan sumber daya manusia[7].

Sistem layanan bantuan hukum di Sekretariat Daerah dikembangkan menjadi sistem online dengan pendekatan *DevOps*. Sistem layanan bantuan hukum online ini untuk mengurangi kendala jarak dan geografis bagi warga di daerah terpencil[8].

Penelitian berikut ini fokus pada penerapan metode *agile* untuk mengembangkan aplikasi SIMDA oleh BPKP dan kontribusinya terhadap pengawasan internal. Metode ini menghasilkan sistem yang efektif dalam pengawasan, walaupun belum mampu sepenuhnya memberikan penilaian akurat[9].

Penerapan e-BPHTB di BAKEUDA DKI Jakarta masih digunakan terbatas karena kendala sumber daya manusia dan rumitnya sistem persetujuan dalam layanan e-BPHTB[10].

Metode *DevOps* menjadi metode yang dipilih untuk mengembangkan dan mengelola sistem layanan publik. Hal ini karena metode *DevOps* mampu menggabungkan otomatisasi, kerja sama tim, monitoring sistem yang *real-time*. Semua keunggulan tersebut menjadi kunci bagi sektor publik untuk mengatasi tantangan pengelolaan sistem. Sehingga sistem yang dihasilkan lebih adaptif, aman, dan berkelanjutan di era digital[11].

Layanan e-BPHTB di Lombok Barat fokus pada penerapan prinsip *self-assessment* dan perlindungan hukum dalam proses validasi. Sistem berjalan efektif, memberikan kepastian hukum dan perlindungan bagi wajib pajak serta PPAT melalui mekanisme pengawasan, kerahasiaan data, dan prosedur keberatan. [12].

Penelitian ini mengkaji peran kecerdasan buatan (AI) dalam mencermati alasan gagalnya proyek yang menggunakan metode *Agile*. Hasil penelitian ini membuktikan peran AI dalam membantu memperbaiki berbagai fase *Agile*, sehingga meningkatkan efektivitas dan keberhasilan manajemen proyek *Agile*[13].

Secara teoritis, penelitian berikut ini menjelaskan bahwa metode *Agile* adalah metode yang menawarkan pendekatan *iteratif* dan adaptif dalam pengembangan perangkat lunak, namun mempunyai kendala dalam menangani perubahan kebutuhan sistem

dan koordinasi anggota tim, terutama pada pengembangan skala besar[14].

Penelitian berikut ini adalah rangkuman dari 92 studi berkesinambungan yang berlangsung dari 2010 sampai 2021. Penelitian tersebut mengamati proses evaluasi web yang menghasilkan kesimpulan bahwa web yang dihasilkan adalah web yang inklusif dan ramah pengguna. Hasil ini menjadi pedoman untuk menghasilkan serta web yang lebih mudah diakses di masa depan[15].

Berdasarkan beberapa penelitian tersebut, maka pengembangan sistem BPHTB berbasis online dengan metode *Agile* dan *DevOps* di BAKUEDA di beberapa Provinsi lain dapat menghasilkan sistem layanan yang lebih cepat, mudah, dan transparan.

2. BAHAN DAN METODE

Pengembangan sistem layanan pembayaran BPHTP di BAKUEDA Pangkalpinang dengan metode *Agile* dan *DevOps*, didasarkan pada beberapa keunggulan kolaborasi *Agile* dan *DevOps* yang diperkuat dengan beberapa penelitian berikut.

Metode *Agile* dalam pengembangan perangkat lunak dilakukan melalui langkah berulang meliputi lima tahap yaitu perencanaan, pengembangan, kolaborasi, perbaikan, dan penyesuaian[16].

Penelitian berikutnya, mengkaji secara sistematis 191 studi di 134 organisasi untuk membandingkan lima metode *Agile* skala besar. Hasilnya adalah, terdapat 31 tantangan dan 27 faktor keberhasilan yang dapat menjadi acuan pengembang lain saat ingin memilih *Agile* sebagai metode dalam mengembangkan sistem[17].

Metode *Agile* mempunyai beberapa keunggulan saat digunakan dalam pengembangan sistem, seperti, mudah beradaptasi, iteratif, kolaboratif, dan respon terhadap perubahan kebutuhan. Selain itu *Agile* juga meningkatkan keterlibatan pemangku kepentingan, sehingga menurunkan resiko keterlambatan dan kesalahan dengan komunikasi intensif[18].

Menurut penelitian berikut, metode *Agile* mempunyai dua belas faktor keberhasilan pengembangan sistem, yang terbagi dalam tiga dimensi: manusia, organisasi, dan teknis. Faktor terpenting adalah faktor manusia dalam bentuk kapabilitas tim, disusul oleh komunikasi yang efektif dan keterlibatan pengguna, selain itu budaya organisasi tetap relevan sebagai landasan penerapan *Agile*[19].

Tantangan utama penerapan *Agile* adalah estimasi waktu yang kurang akurat, dokumentasi minim, arsitektur belum matang, dan rendahnya keterlibatan pengguna. Selain itu tingginya perubahan kebutuhan menambah kompleksitas, hal ini memerlukan komunikasi lintas fungsi dan perbaikan estimasi untuk keberhasilan proyek[20].

Keunggulan metode *Agile* dalam pengembangan perangkat lunak adalah kemampuan fleksibel menghadapi perubahan, kolaborasi intens lintas fungsi, dan mampu rilis produk lebih cepat dengan iterasi proses yang singkat. Kelemahannya

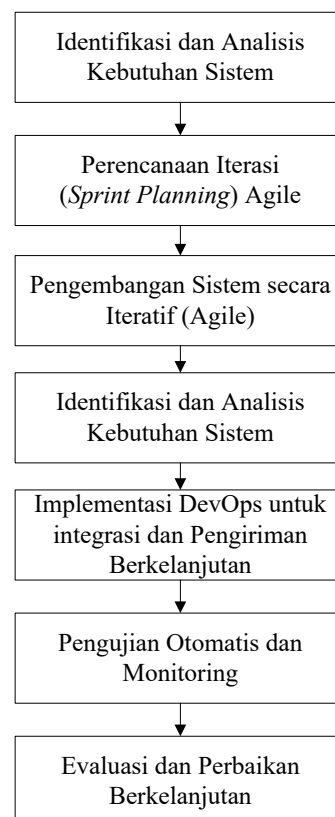
adalah ketergantungan pada komunikasi efektif, minimnya dokumentasi, dan tantangan pengelolaan proyek besar dengan kebutuhan yang dinamis[21][22].

DevOps, yang lahir dari *Agile*, adalah metode yang memadukan pengembangan dan operasi melalui otomatisasi dan kolaborasi berkelanjutan. Keunggulannya adalah kemampuan rilis sistem lebih cepat dengan kualitas yang lebih baik. Tantangan yang dihadapi adalah kurangnya pemahaman konsep, sulit integrasi sistem, terutama yang berkaitan dengan resistensi budaya [23].

Berdasarkan beberapa uraian tentang metode *Agile* dan *DevOps*, terlihat kedua metode saling terkait dalam pengembangan perangkat lunak dengan fokus pada kecepatan, kolaborasi, dan adaptasi. Metode *Agile* mengembangkan sistem secara iteratif dan responsif terhadap perubahan kebutuhan pengguna, sementara *DevOps* fokus pada integrasi pengembangan dan operasional melalui otomatisasi dan penyerahan berkelanjutan. Keduanya mempercepat proses dan meningkatkan kualitas sistem sesuai kebutuhan pengguna.

Berikut adalah gambar yang menunjukkan langkah-langkah penelitian untuk rekayasa sistem pembayaran BPHTP dengan metode *Agile* dan *DevOps*

Rekayasa Ulang Sistem Pajak BPHTB Berbasis *Agile* dan *DevOps* untuk E-government



Gambar 1. Langkah Penelitian

Gambar 1 adalah menampilkan langkah gabungan metode *Agile* pada tahap identifikasi kebutuhan, *sprint planning*, pengembangan iteratif. Sedangkan tahap

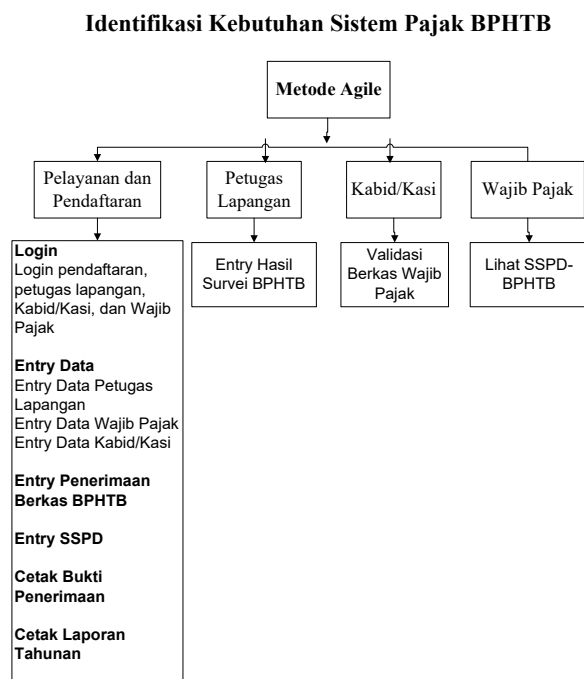
berikutnya adalah metode *DevOps* yang terdiri dari implementasi, pengujian dan *monitoring*, serta evaluasi berkelanjutan.

3. HASIL DAN PEMBAHASAN

Berdasarkan metode *Agile* dan *DevOps*, rekayasa sistem pembayaran BPHTB, dimulai dari,

3.1. Identifikasi Dan Analisis Kebutuhan Sistem.

Tahap awal penelitian meliputi identifikasi serta sistem analisis kebutuhan dari proses bisnis BPHTB BAKEUDA Kota Pangkalpinang. Berdasarkan analisis proses bisnis yang ada, ditemukan bahwa mekanisme administrasi (entri data, validasi, hingga pencetakan dokumen) masih manual dan dominan berbasis dokumen fisik. Hal ini menyebabkan risiko kehilangan data, duplikasi, keterlambatan proses, dan rendahnya efisiensi pelayanan. Identifikasi kebutuhan sistem ditampilkan dalam bentuk hirarki berikut,

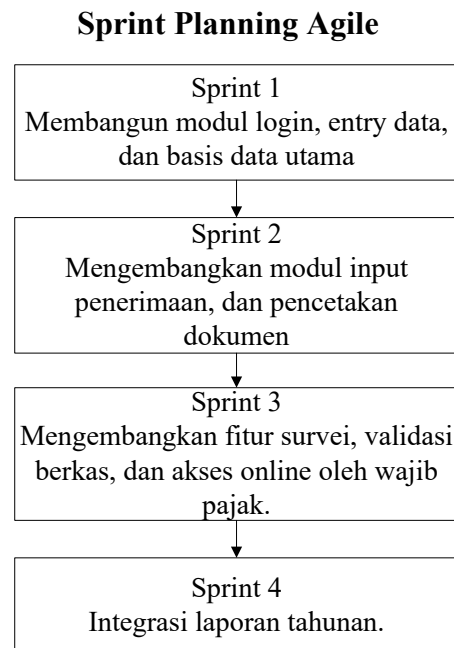


Gambar 2. Diagram Hirarki Identifikasi Kebutuhan Sistem

Gambar 2 menunjukkan hasil identifikasi kebutuhan sistem untuk mengembangkan sistem berbasis web layanan BPHTB di BAKUEDA Pangkalpinang. Identifikasi kebutuhan di bagian pelayanan dan pendaftaran, kebutuhan untuk petugas lapangan, kebutuhan untuk Kabid/Kasi yang terkait, serta kebutuhan untuk wajib pajak.

3.2. Perencanaan Iterasi (*Sprint Planning*) Agile

Tahap kedua adalah *sprint planning*, pada tahap ini pengembang sistem akan membagi hasil identifikasi kebutuhan menjadi tiga kelompok *sprint*, agar pengembangan sistem lebih cepat dan dapat menjaga ritme koordinasi lintas fungsi dengan baik. Berikut gambar yang menampilkan *sprint planning* Agile.

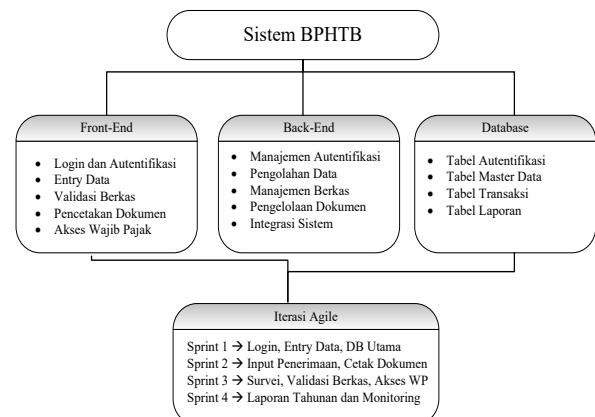


Gambar 3. Diagram *Sprint Planning* Agile

Gambar 3 menunjukkan aktivitas pengembangan sistem pada sprint 1 yang terdiri dari proses pembangunan modul *login*, *entry* data dan basis data. Lanjut pada sprint 2 yang fokus pada bagian survei, validasi berkas, dan akses oleh wajib pajak. Sedangkan sprint 3 mengintegrasikan laporan tahunan dan monitoring ke dalam sistem web. *Backlog*, prioritas pengembangan, dan *sprint plan* disusun diawal setiap siklus iterasi dengan memperhatikan umpan balik dari pemangku kepentingan yang terlibat dan kebutuhan pengguna.

3.3. Pengembangan Sistem Secara Iteratif (*Agile*)

Langkah selanjutnya adalah proses pengembangan sistem secara iteratif dengan metode *Agile* disajikan dalam gambar 4 berikut ini,



Gambar 4. Gambar arsitektur modular sistem BPHTB

Gambar 4 adalah gambar arsitektur modular sistem BPHTB yang menggambarkan pembagian komponen utama sistem ke dalam tiga lapisan: *Front-End*, *Back-End*, dan *Database*, yang dikembangkan secara *iteratif* menggunakan metode Agile.

1. Lapisan *Front-End* sebagai antarmuka yang menghubungkan pengguna dengan sistem. Modul yang termasuk di dalamnya adalah, *login* dan autentikasi keamanan akses untuk proses *entry* data bagi petugas, wajib pajak, dan pejabat (KaBid/KaSi). Selanjutnya adalah validasi berkas untuk memeriksa kelengkapan dan kebenaran dokumen. Lalu pencetakan dokumen sebagai bukti penerimaan dan laporan tahunan. Terakhir akses wajib pajak yang memungkinkan pengguna melihat status SSPD-BPHTB secara daring.

2. Lapisan *Back-End*

Lapisan ini membahas logika bisnis, analisis data, dan integrasi sistem. Modul-modul dalam lapisan ini, meliputi manajemen autentikasi untuk mengatur akses pengguna. Lalu pengolahan data & validasi otomatis untuk memastikan akurasi input. Selanjutnya adalah manajemen berkas & pelacakan untuk penyisipan dokumen. Terakhir, mengelola laporan, seperti riwayat cetak dan laporan tahunan. Serta Mengintegrasikan sistem menggunakan API internal dan eksternal.

3. Lapisan *Database*

Lapisan ketiga ini menyimpan dan mengelola semua data dalam sistem, lengkap dengan struktur tabel, meliputi, tabel autentikasi *user* untuk hak akses, tabel master, tabel transaksi, dan tabel laporan.

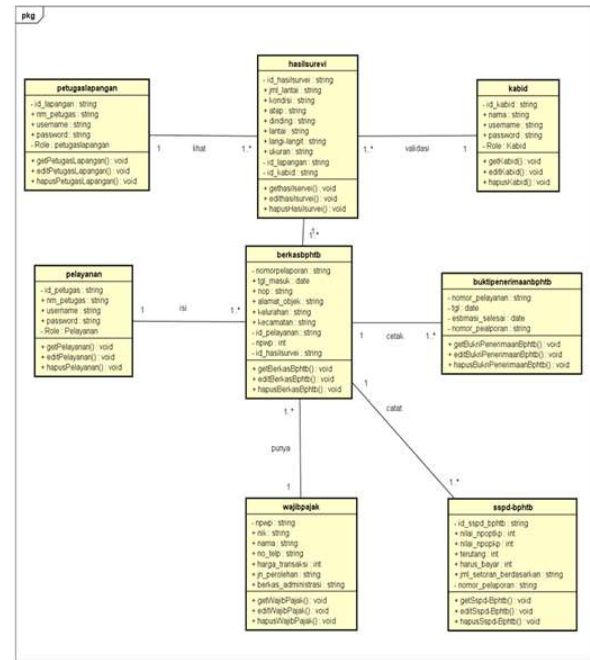
4. Iterasi *Agile*

Pengembangan sistem melalui empat sprint, yaitu sprint 1 untuk *login*, *entry data*, dan *database*. Sprint 2 untuk input penerimaan dan cetak dokumen. Sprint 3 untuk modul survei, validasi berkas, dan akses bagi wajib pajak ke sistem. Sprint 4 modul untuk mencetak laporan tahunan dan *monitoring*.

Pendekatan ini memastikan bahwa pengembangan sistem dapat beradaptasi dengan perubahan kebutuhan, meminimalkan risiko, dan menghasilkan sistem yang teruji secara bertahap sebelum implementasi penuh.

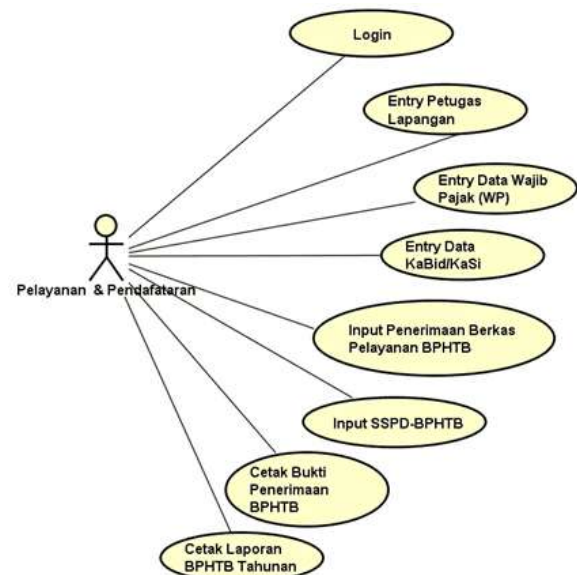
Berikut adalah pembangunan modul *login*, *entry data*, dan skema basis data utama, yang digambarkan lewat diagram UML, seperti *class diagram* dan *use case diagram*.

Gambar 5 adalah *class diagram* yang menggambarkan hubungan *class* dalam sistem. Rancangan ini akan menghasilkan rancangan basis data untuk sistem berorientasi objek.



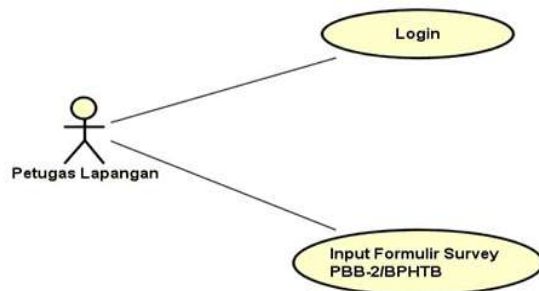
Gambar 5. *Class Diagram*

Gambar 5 menampilkan asosiasi delapan *class* yang akan menjadi pola dalam membangun relasi dalam basis data. Berikutnya adalah gambar 6 sampai 9 yang merupakan *use case diagram*. Gambar 6 adalah *use case diagram* pelayanan dan pendaftaran.



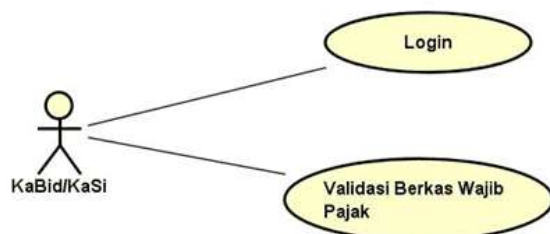
Gambar 6 *UseCase Diagram* Pelayanan Dan Pendaftaran

Gambar 6 menunjukkan *actor* bagian pelayanan dan pendaftaran bertanggungjawab untuk beberapa proses *entry*, seperti *entry data* petugas lapangan, *entry data* wajib pajak, *entry data* Kabid/Kasi, lalu *input* berkas pemohon, *input* SSPD-BPHTB, sampai cetak bukti penerimaan BPHTB dan cetak laporan tahunan. Berikutnya adalah gambar 7, *use case diagram* petugas lapangan.



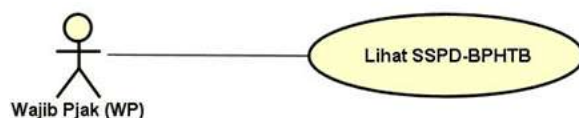
Gambar 7. UseCase Diagram Petugas Lapangan

Gambar 7 menampilkan interaksi *actor* petugas lapangan dengan sistem, yaitu pada proses *entry* proses formulir survei BPHTB. Selanjutnya adalah gambar 8, yaitu *use case diagram* Kabid/Kasi.



Gambar 8. UseCase Diagram KaBid/KaSi

Gambar 8 adalah *usecase diagram* yang menunjukkan interaksi *actor* Kabid/Kasi dengan sistem, khususnya pada proses validasi berkas wajib pajak yang mengajukan BPHTB. Gambar 9 berikut ini adalah *use case diagram* untuk wajib pajak.



Gambar 9. UseCase Diagram Wajib Pajak (WP)

Gambar 9 adalah gambar *usecase diagram* yang menunjukkan interaksi wajib pajak dalam sistem layanan BPHTB berbasis web. Dalam sistem ini, wajib pajak dapat melihat SSPD-BPHTB yang sedang ia ajukan.

Setelah rancangan *use case diagram*, selanjutnya adalah merancang tampilan antar muka yang akan menjadi *prototype* bagi semua pengguna sistem. Gambar 10 sampai 17 adalah beberapa desain antar muka yang menampilkan model sistem layanan BPHTB berbasis web. Gambar 10 adalah desain *login* untuk semua *actor* yang terlibat dalam sistem ini, yaitu staf bagian pelayanan, petugas lapangan, Kabid/Kasi, dan Wajib Pajak.



Gambar 10. Desain Login Bagian Pelayanan, Petugas Lapangan, KaBid/KaSi, Wajib Pajak.

Gambar 10 menampilkan laman login sistem untuk bagian yang terlibat dan wajib pajak. Desain menampilkan *text box* untuk input *user name* dan *password*.



Gambar 11. Desain Entry Data Pelayanan dan Pendaftaran Bagian Tampilan data

Gambar 11 adalah desain tampilan data untuk memantau semua user yang login ke sistem.



Gambar 12. Desain Entry Wajib Pajak Bagian Tampilan Data

Gambar 12 adalah tampilan untuk input data wajib pajak yang mengajukan layanan BPHTB.

Gambar 13 Desain Input Penerimaan Bagian Tambahan Data

Gambar 13 adalah tampilan layar untuk bagian pelayanan input berkas pendukung pengajuan BPHTB.

Gambar 16. Desain Verifikasi BPHTB Bagian Validasi Data

Gambar 16 adalah tampilan verifikasi dan validasi data ajuan BPHTB.

Gambar 14. Desain Input Penerimaan Bagian Tampilan Data

Gambar 14 menampilkan tampilan data bagian pelayanan untuk memantau progres layanan BPHTB, mulai dari tanggal masuk berkas sampai selesai.

Gambar 15. Desain Survei Objek BPHTB Tampilan Bagian isi survei

Gambar 15 menunjukkan tampilan untuk petugas lapangan saat input hasil survei objek BPHTB.

Gambar 17. Desain Bagian Wajib Pajak Bagian Tampilan Detail SSPD-BPHTB

Gambar 17 menampilkan desain untuk wajib pajak melihat progres ajuan BPHTB di sistem web.

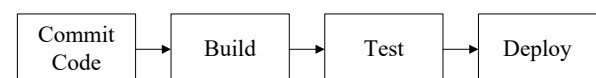
3.4. Implementasi *DevOps* untuk Integrasi dan Pengiriman Berkelanjutan

Tahap selanjutnya adalah implementasi untuk integrasi dan berkesinambungan dengan tahap menyerahkan sistem kepada *stakeholder*.

Untuk meningkatkan efisiensi pengujian dan *deployment*, dilakukan otomatisasi *pipeline DevOps*, dengan implementasi *CI/CD* (*Continuous Integration/Continuous Deployment*).

Gambar 18 menggambarkan bagaimana alur kerja *DevOps* *CI/CD* berjalan secara otomatis dan berkelanjutan dari tahap pengembangan hingga ke produksi dan monitoring.

DevOps CI/CD



Gambar 18. Diagram Implementasi *DevOps* *CI/CD*

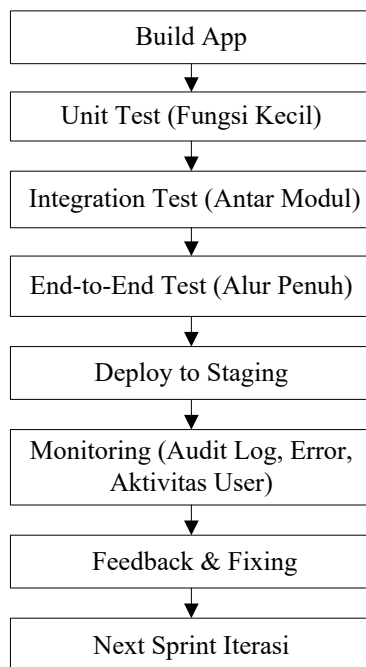
Gambar 18 menggambarkan alur *CI/CD* pada pengembangan sistem BPHTB. Tahap *commit code* adalah saat pengembangan sistem mengirimkan kode ke

repositori kontrol. Lanjut ke tahap *build* adalah tahap pengkodean sistem secara otomatis untuk memastikan kode bisa dikompilasi dan bisa dijalankan. Lalu tahap *test*, yaitu tahap pengujian kode secara otomatis untuk memastikan tidak ada *bug*. Lanjut ditahap *deploy* yaitu tahap pengiriman kode yang sudah lulus uji ke tahap produksi secara otomatis. Dan terakhir adalah *monitoring*, yaitu sistem memantau aplikasi yang telah diserahkan untuk memastikan mampu berfungsi dengan performa yang stabil.

3.5. Pengujian Otomatis dan Monitoring

Tahap berikutnya adalah pengujian otomatis dan monitoring yang tampak pada gambar 19 berikut ini.

Pengujian Otomatis dan Monitoring yang berulang tiap sprint



Gambar 19. Diagram Pengujian Otomatis dan Monitoring

Gambar 19 adalah diagram alur yang menggambarkan dalam setiap *sprint* Agile selalu diikuti siklus pengujian berlapis. Siklus pengujian mulai dari unit *test* untuk fungsi kecil, lalu lanjut untuk pengujian yang terintegrasi (*integration test*), dan diakhiri dengan pengujian akhir (*End-to-End Test*). Setelah lulus tahap pengujian, lalu sistem di-*deploy* ke *staging* yang di dukung pengawasan (*monitoring*) berupa *audit log*, pencatatan *error* dan aktivitas pengguna. Hasil pengawasan menjadi umpan balik perbaikan untuk *sprint* berikutnya

Berikut tabel ringkas bentuk pengujian yang relevan untuk Sistem BPHTB berbasis *Agile* dan *DevOps* :

Tabel 1. Tabel Pengujian untuk Sistem BPHTB berbasis *Agile* dan *DevOp*

Jenis Pengujian	Tujuan	Indikator Keberhasilan
<i>Unit Testing</i>	Memastikan fungsi atau modul kecil bekerja sesuai spesifikasi	Semua fungsi (perhitungan BPHTB, validasi NOP) mengembalikan output sesuai data uji
<i>Integration Testing</i>	Memastikan semua modul dapat berinteraksi tanpa error.	Alur login → input data → simpan DB → cetak dokumen berjalan lancar tanpa kehilangan data.
<i>End-to-End (E2E) Testing</i>	Memastikan keseluruhan proses dari awal hingga akhir sesuai skenario bisnis	Simulasi pengguna (petugas/wajib pajak) dapat menyelesaikan transaksi tanpa error mayor
<i>Performance Testing</i>	Mengukur waktu respon dan kapasitas sistem di bawah beban tertentu.	Waktu respon < 2 detik untuk transaksi normal; sistem stabil pada beban ≥ 100 user aktif.
<i>Security Testing</i>	Memastikan keamanan akses dan data	Login aman, enkripsi data aktif, hak akses sesuai role pengguna.
<i>Usability Testing</i>	Menilai kemudahan penggunaan antarmuka.	Skor kepuasan pengguna ≥ 80% dari survei usability.

3.6. Evaluasi dan Perbaikan Berkelanjutan

Evaluasi dilakukan dengan membandingkan kondisi sebelum dan sesudah sistem digital diterapkan.

Tahap Evaluasi dan Perbaikan Berkelanjutan dilakukan dengan membandingkan kinerja sebelum dan sesudah penerapan sistem digital. Hasilnya menunjukkan peningkatan efisiensi proses, dengan waktu input dan pencarian dokumen yang lebih cepat. Keamanan dan keteraturan data juga membaik, bebas dari risiko kehilangan dokumen fisik, mudah dilacak, dan aman secara elektronik.

Umpan balik pengguna positif, di mana petugas dan wajib pajak merasakan kemudahan akses serta pemantauan. Temuan ini menjadi dasar untuk pengembangan fitur dan peningkatan layanan di tahap selanjutnya.

4. KESIMPULAN

Pengembangan sistem layanan BPHTB menjadi sistem layanan berbasis web di BAKEUDA Pangkalpinang yang didesain ulang dengan metode *Agile* dan *DevOps* bertujuan untuk meningkatkan kualitas layanan kepada wajib pajak yang mengajukan

BPHTB. Dengan menyediakan layanan BPHTB berbasis web, wajib pajak dapat memahami alur proses, mempersiapkan berkas pendukung sesuai ketentuan, dan memantau progres secara daring. Pilihan menggunakan metode *Agile* dan *DevOps* berdasarkan keunggulan *iteratif Agile* yang memungkinkan sistem berkembang dan beradaptasi sesuai kebutuhan pengguna dan perubahan peraturan terkait. Dan integrasi *DevOps* memudahkan proses integrasi dan pengujian otomatis yang relatif lebih minim gangguan. Rekayasa ulang sistem layanan BPHTB ini meningkatkan efisiensi layanan, meningkatkan keamanan dan menurunkan resiko kehilangan data, sinkronisasi data antar bagian, meningkatkan kemudahan akses wajib pajak dalam memantau penyelesaian layanan BPHTB yang diajukannya. Dengan demikian, metode gabungan *Agile-DevOps* layak diimplementasikan sebagai strategi efektif pengembangan sistem pemerintahan berbasis digital. Sebagai bagian dari *E-Government*, sistem layanan pajak BPHTB yang direkayasa ulang dengan metode *Agile* dan *DevOps* mendorong transparansi sistem dengan tersedianya *dashboard real-time* dan pelaporan digital yang mudah diakses. Rekayasa ulang sistem memperkuat digitalisasi pajak daerah untuk ikut serta mewujudkan *E-Government* yang responsif, inklusif, dan berkelanjutan, agar lingkungan administrasi lebih efisien dalam melayani kebutuhan publik.

5. DAFTAR PUSTAKA

- [1] bpk.go.id, *UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 28 TAHUN 2009 TENTANG PAJAK DAERAH DAN RETRIBUSI DAERAH*. 2009.
- [2] djpk.kemenkeu.go.id, *Undang-Undang Nomor 1 Tahun 2022 Tentang Hubungan Keuangan Antara Pemerintah Pusat dan Pemerintahan Daerah*. 2022.
- [3] Provinsi Kepulauan Bangka Belitung, *Peraturan Walikota Pangkalpinang No.1 Tahun 2024 Tentang Perubahan Atas Peraturan Walikota Pangkalpinang No. 65 Tahun 2023 Tentang Penjabaran Anggaran Pendapatan dan Belanja Daerah Tahun Anggaran 2024*. 2024.
- [4] W. K. P. PINANG, *Peraturan Walikota Pangkalpinang No.30 Tahun 2024 Tentang Penjabaran Pertanggungjawaban Pelaksanaan Anggaran Pendapatan dan Belanja Daerah Tahun Anggaran 2023*, no. April. 1967.
- [5] A. D. K. Wardani, I. P. D. Yudharta, and J. P. Lukman, "Agile Governance dalam Sistem Elektronik Pelayanan Administrasi Pajak Daerah (E-Palapa) pada Pajak Hotel di Badan Pendapatan Daerah (Bapenda) Kabupaten Badung," *Socio-political Commun. Policy Rev.*, vol. 1, no. 2, pp. 1–16, 2024, doi: 10.61292/shkr.103.
- [6] M. I. Muttaqin and A. Permatasari, "Agile Government: Langkah Strategis Pemerintah Negara Indonesia Dalam Menghadapi Ancaman Resesi Global 2023," *JlIP J. Ilm. Ilmu Pemerintah.*, vol. 8, no. 2, pp. 187–201, 2023, doi: 10.14710/jiip.v8i2.18875.
- [7] D. A. N. Dampaknya and T. Efisiensi, "Implementasi Sistem Informasi Pajak Daerah (SIPD) dan Dampaknya Terhadap Efisiensi Administrasi," *Judge J. Huk.*, vol. 06, no. 02, pp. 267–276, 2025.
- [8] S. Riyadi, "Adopsi Metode DevOps Sebagai Acuan Pengembangan Aplikasi Bantuan Hukum," *Gener. J.*, vol. 7, no. 1, pp. 23–30, 2023, doi: 10.29407/gj.v7i1.19629.
- [9] W. K. Ary Suharyanto, "Studi Implementasi Agile Method dalam Pengembangan Aplikasi SIMDA pada Badan Pengawas Keuangan dan Pembangunan," *Pros. Semin. Nas. Pelayanan Publik*, no. March, 2023, [Online]. Available: https://www.researchgate.net/profile/Ary-Suharyanto/publication/369189279_Studi_Implementasi_Agile_Method_dalam_Pengembangan_Aplikasi_SIMDA_pada_Badan_Pengawasan_Keuangan_dan_Pembangunan/links/640f2a5c66f8522c38a020f5/Studi-Implementasi-Agile-Method-dalam.
- [10] S. Fadhilah, A. Habib, and S. Adang, "Analysis of the Implementation of E-BPHTB Online and Its Impact on Local Tax Revenue and Public Compliance in DKI Jakarta Province," *Account. Res. J. Sutaatmadja*, vol. 09, no. 01, pp. 73–83, 2025.
- [11] Pramod Kumar Muppala, "Resilient government services: adopting DevOps for public sector efficiency," *World J. Adv. Res. Rev.*, vol. 25, no. 1, pp. 2188–2201, 2025, doi: 10.30574/wjarr.2025.25.1.3565.
- [12] I. Y. Saputri, G. Asmara, and E. B. Sili, "Implementation of the Principle of the Self-Assessment System For Taxpayers and PPAT in the Validation of E-BPHTB in West Lombok Regency, Indonesia," *Path Sci.*, vol. 10, no. 12, p. 2052, 2024, doi: 10.22178/pos.112-21.
- [13] H. L. Lumbanraja, T. Raharjo, and A. N. Fitriani, "Artificial Intelligence Implementation in Agile Project Management Addressing Challenges and Maximizing Impact," *Indones. J. Comput. Sci.*, vol. 13, no. 4, pp. 5023–5042, 2024, doi: 10.33022/ijcs.v13i4.4155.
- [14] A. Rasheed *et al.*, "Requirement Engineering Challenges in Agile Software Development," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/6696695.
- [15] J. Ara, C. Sik-Lanyi, and A. Kelemen, *Accessibility engineering in web evaluation process: a systematic literature review*, vol. 23, no. 2. Springer Berlin Heidelberg, 2024.
- [16] E. Altameem, "Impact of Agile Methodology

- on Software Development,” *Comput. Inf. Sci.*, vol. 8, no. 2, pp. 9–14, 2015, doi: 10.5539/cis.v8n2p9.
- [17] H. Edison, X. Wang, and K. Conboy, “Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review,” *IEEE Trans. Softw. Eng.*, vol. 48, no. 8, pp. 2709–2731, 2022, doi: 10.1109/TSE.2021.3069039.
- [18] H. Lamsellak and M. G. Belkasmi, “Global software development agile planning model: challenges and current trends,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 32, no. 3, pp. 1774–1784, 2023, doi: 10.11591/IJEECS.V32.I3.PP1774-1784.
- [19] A. Muhammad, A. Siddique, Q. N. Naveed, U. Saleem, M. A. Hasan, and B. Shahzad, “Investigating crucial factors of agile software development through composite approach,” *Intell. Autom. Soft Comput.*, vol. 27, no. 1, pp. 15–34, 2021, doi: 10.32604/iasc.2021.014427.
- [20] M. Batra and A. Bhatnagar, “A Research Study on Critical Challenges in Agile Requirements Engineering,” *Int. Res. J. Eng. Technol.*, no. June, p. 1214, 2008, [Online]. Available: www.irjet.net.
- [21] I. A. Edet, “Empirical Studies on Agile Software Development Benefits and Challenges: A Systematic Literature Review,” University of Oulu, 2019.
- [22] Meera Baby, “Hybrid Agile in Software Engineering: A Systematic Literature Review,” University of Oulu, 2024.
- [23] G. B. Ghantous and A. Q. Gill, “DevOps: Concepts, practices, tools, benefits and challenges,” 2017.