

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Profile Kampus

Institut Teknologi Bisnis Riau (ITB Riau) merupakan perguruan tinggi yang berada di bawah naungan Yayasan At-Thoiba Riau dan diresmikan pada 17 Februari 2022. Perguruan tinggi ini memiliki visi untuk menjadi institusi pendidikan teknologi terbaik di Sumatera yang berlandaskan nilai-nilai Islam. ITB Riau menawarkan berbagai program studi berbasis teknologi dan bisnis, seperti Teknik Informatika, Bisnis Digital, dan Sistem Informasi. Program studi yang ditawarkan dirancang untuk membekali mahasiswa dengan keterampilan yang sesuai dengan kebutuhan industri.

Dalam upaya meningkatkan mutu pendidikan dan penelitian, ITB Riau telah menjalin kerja sama dengan beberapa institusi, seperti Politeknik Negeri Bengkalis dan Universitas Pasir Pangaraian. Kerja sama ini bertujuan untuk memperluas cakupan riset, meningkatkan kualitas pembelajaran, serta mempersiapkan lulusan yang memiliki daya saing tinggi di dunia kerja dan industri berbasis teknologi.



Gambar 2. 1 Kampus ITB Riau

2.1.2 Sistem Informasi Akademik

Sistem Informasi Akademik adalah perangkat lunak yang digunakan oleh institusi pendidikan untuk mengelola informasi terkait aktivitas akademik. Sistem ini juga berfungsi untuk menyimpan data civitas akademika, seperti data mahasiswa, dosen, serta administrasi akademik lainnya. Sistem Informasi

Akademik merupakan perangkat lunak yang berperan dalam menyajikan informasi dan menata administrasi terkait kegiatan akademik. Dengan adanya sistem ini, proses administrasi akademik seperti pendataan, pencarian informasi, dan pengelolaan pembayaran menjadi lebih efisien dan mudah (Pandu Pratama & Kamisutara, 2021).

2.1.3 Agile Software Development

Agile Software Development adalah metode pengembangan perangkat lunak yang berfokus pada iterasi, kolaborasi tim, dan kemampuan beradaptasi terhadap perubahan secara cepat dan terukur. Metode ini membagi proses pengembangan ke dalam siklus pendek yang disebut *sprint*, di mana pada setiap *sprint* tim menghasilkan *working software* yang bisa langsung diuji dan dievaluasi. Tidak seperti metode tradisional yang perencanaannya kaku, *Agile* mengandalkan *adaptive planning*, *incremental delivery*, dan *continuous feedback*. Pengembangan sistem dalam metode ini biasanya dimulai dari *user story*, lalu dipecah ke dalam *task* yang dapat diselesaikan dalam waktu singkat. Salah satu kerangka kerja paling populer dari *Agile* adalah *Scrum*, yang melibatkan peran-peran seperti *product owner*, *scrum master*, dan *development team*. Prosesnya dimulai dengan menyusun *product backlog*, kemudian dilanjutkan ke *sprint planning* untuk menentukan fitur yang akan dikembangkan dalam 1-4 minggu. Selama *sprint*, tim melakukan *daily scrum* untuk membahas progres dan hambatan, lalu di akhir *sprint* dilakukan *sprint review* dan *sprint retrospective* sebagai bentuk evaluasi dan perbaikan proses kerja. Siklus ini terus berulang, sehingga pengembangan berjalan secara bertahap (*incremental*) dan selalu dapat disesuaikan dengan kebutuhan yang berkembang. Dengan pendekatan yang modular dan berkelanjutan, metode ini terbukti efektif dalam membangun perangkat lunak yang fleksibel dan responsif (Gumay dkk., 2023).

2.1.4 Scrum

Scrum merupakan kerangka kerja yang digunakan dalam pengelolaan pengembangan produk yang kompleks dengan mengintegrasikan berbagai proses dan teknik di dalamnya. Kerangka kerja ini mampu mengidentifikasi ketidakefisienan dalam manajemen produk dan teknik kerja, sehingga dapat terus

meningkatkan kualitas produk, efektivitas tim, serta lingkungan kerja. Seiring dengan meningkatnya kompleksitas teknologi, pasar, dan lingkungan bisnis, *Scrum* terbukti mampu mengoptimalkan produktivitas serta efisiensi waktu dalam proses pengembangan. Hal ini didukung oleh adanya *backlog*, yaitu daftar tugas yang disusun berdasarkan prioritas, ukuran, serta kesiapan untuk dikerjakan (Schwaber & Sutherland, 2020). Pratama dkk., (2022) menjelaskan bahwa metode *Scrum* mendorong kolaborasi tim yang lebih baik melalui aktivitas harian seperti *daily scrum* dan sesi *retrospective*, serta meningkatkan keterlibatan *stakeholder* secara langsung dalam setiap tahapan pengembangan. Hal ini menghasilkan proses kerja yang lebih adaptif terhadap perubahan dan lebih terkoordinasi secara keseluruhan.

Menurut Satpathy (2022), *Scrum* memiliki beberapa tahapan utama dalam pengembangan aplikasi, di antaranya:



Gambar 2. 2 Alur *Scrum*

1) *Initiate* (Inisiasi)

Fase *Initiate* merupakan tahap awal dalam pengembangan menggunakan metode *Scrum*. Pada fase ini, dilakukan berbagai proses yang terkait dengan inisiasi proyek, yaitu:

a) *Create Project Vision*

Proses ini bertujuan untuk meninjau ulang *business case* proyek dan menyusun *Project Vision Statement* yang akan menjadi inspirasi dan fokus utama bagi seluruh tim dalam pengembangan. Selain itu, pada tahap ini juga dilakukan penetapan siapa yang akan menjadi *Product Owner*.

b) *Identify Scrum Master and Business Stakeholder(s)*

Dalam proses ini, *Scrum Master* dan pemangku kepentingan bisnis diidentifikasi menggunakan kriteria yang telah ditentukan. Pemilihan ini penting untuk memastikan peran dan tanggung jawab mereka dapat mendukung kesuksesan proyek.

c) *Form Scrum Team*

Proses ini dilakukan untuk memilih dan menetapkan anggota tim Scrum. Biasanya, *Product Owner* memiliki peran utama dalam pemilihan anggota, namun sering kali dilakukan bersama *Scrum Master* agar tercapai keseimbangan antara keahlian dan kolaborasi tim.

d) *Develop Epic(s)*

Proses ini menggunakan *Project Vision Statement* sebagai dasar untuk menyusun *Epics*, yaitu gambaran umum mengenai kebutuhan pengguna yang akan dikembangkan lebih lanjut. Diskusi bersama pengguna juga dapat dilakukan melalui *User Group Meetings* untuk merumuskan *Epics* yang sesuai.

e) *Create Prioritized Product Backlog*

Pada tahap ini, *Epics* yang telah disusun diperinci dan disusun berdasarkan prioritas untuk membentuk *Prioritized Product Backlog*. Selain itu, kriteria penyelesaian atau *Done Criteria* juga ditetapkan sebagai standar keberhasilan setiap *item backlog*.

f) *Conduct Release Planning*

Proses ini dilakukan oleh *Product Owner* dengan dukungan dari tim *Scrum* untuk menyusun jadwal rilis (*Release Planning Schedule*) yang menjadi panduan implementasi bertahap produk. Jadwal ini juga akan disampaikan kepada pemangku kepentingan. Selain itu, panjang durasi *sprint* juga ditentukan pada tahap ini.

1) *Plan and Estimate*

Fase ini terdiri dari proses-proses yang berkaitan dengan perencanaan dan estimasi pekerjaan yang akan dilakukan, yang meliputi beberapa tahap berikut:

a) *Create User Stories*

Pada proses ini, *Product Owner* membuat *User Stories* beserta *Acceptance Criteria* yang relevan, lalu memasukkannya ke dalam *Prioritized Product Backlog*. *User Stories* dirancang untuk memastikan kebutuhan pelanggan dapat digambarkan dengan jelas dan dapat dipahami oleh seluruh pemangku kepentingan bisnis.

b) *Estimate User Stories*

Proses ini dilakukan oleh tim *Scrum* dengan dukungan dari *Scrum Master*, di mana mereka memberikan estimasi *effort* yang diperlukan untuk mengembangkan fungsionalitas yang dijelaskan dalam setiap *User Story*. Menurut Hadji dan Taufik (2019), perkiraan kecepatan tim dapat dihitung dengan rumus berikut:

$$Velocity = \frac{(Hari Kerja per Sprint \times Jam Kerja per Sprint \times Focus Factor)}{Estimasi Waktu per Task}$$

Dengan keterangan sebagai berikut:

- Menentukan total hari kerja per *sprint*
- Menentukan jam kerja per hari
- *Focus Factor* adalah tingkat efektivitas kerja tim (antara 70% hingga 80%)
- Menghitung total waktu estimasi per *task*

c) *Commit User Stories*

Pada tahap ini, tim *Scrum* berkomitmen untuk menyelesaikan *User Stories* yang telah disetujui oleh *Product Owner* dalam satu *Sprint*. Hasil dari proses ini adalah *User Stories* yang sudah disepakati (*Committed User Stories*) serta terbentuknya *Sprint Backlog*.

d) *Identify Task*

Proses ini dilakukan untuk memecah *User Stories* yang sudah dikomitmenkan menjadi tugas-tugas yang lebih spesifik dan detail, yang kemudian disusun dalam *Task List*.

e) *Estimate Tasks*

Proses ini bersifat opsional, di mana tim inti *Scrum* memberikan estimasi *effort* yang diperlukan untuk menyelesaikan setiap tugas yang ada di *Task List*.

f) *Update Sprint Backlog*

Pada proses ini, tim inti *Scrum* memperbarui *Sprint Backlog* dengan detail tambahan mengenai tugas-tugas yang telah diidentifikasi. Kegiatan ini umumnya dilakukan pada saat rapat *Sprint Planning Meeting*.

2) *Implement*

Fase *Implement* berkaitan dengan pelaksanaan tugas dan aktivitas yang diperlukan untuk menciptakan produk dari suatu proyek. Pada fase ini terdiri dari beberapa tahapan, yaitu:

a) *Create Deliverables*

Pada proses ini, tim *Scrum* mengerjakan tugas-tugas yang ada dalam *Sprint Backlog* untuk menghasilkan *Sprint Deliverables*. Sebagai alat bantu pemantauan pekerjaan dan aktivitas yang sedang berjalan, biasanya digunakan *Scrum Board*. Selain itu, jika tim *Scrum* menghadapi masalah atau hambatan, hal tersebut dicatat dan diperbarui di dalam *Impediment Log*.

b) *Conduct Daily Standup*

Pada proses ini, setiap hari diadakan pertemuan singkat yang sangat terfokus dan memiliki batas waktu yang jelas, yang dikenal sebagai *Daily Standup Meeting*. Forum ini digunakan oleh tim *Scrum* untuk saling memberi pembaruan mengenai kemajuan individu masing-masing, serta mengidentifikasi kendala yang mungkin dihadapi.

c) *Refine Prioritized Product Backlog*

Pada proses ini, *Prioritized Product Backlog* secara berkelanjutan diperbarui dan dipelihara. Biasanya dilakukan *Prioritized Product Backlog Review Meeting*, yaitu pertemuan untuk mendiskusikan perubahan atau pembaruan apa pun yang diperlukan. Hasilnya kemudian dimasukkan kembali ke dalam *Product Backlog* sesuai kebutuhan.

3) Review and Retrospect

Fase *Review and Retrospect* berkaitan dengan proses peninjauan terhadap *deliverables* dan pekerjaan yang telah dilakukan. Selain itu, fase ini juga fokus pada penentuan cara-cara untuk meningkatkan praktik dan metode yang digunakan dalam pekerjaan proyek.

a) *Demonstrate and Validate Sprint*

Pada proses ini, tim Scrum mempresentasikan hasil *Sprint Deliverables* kepada *Product Owner* dan pemangku kepentingan bisnis yang relevan melalui *Sprint Review Meeting*. Tujuan dari pertemuan ini adalah untuk mendapatkan persetujuan dan penerimaan dari *Product Owner* atas *User Stories* yang telah diselesaikan dalam *sprint* tersebut.

b) *Retrospect Sprint*

Proses ini dilakukan oleh *Scrum Master* dan tim *Scrum* dengan mengadakan pertemuan untuk mendiskusikan pelajaran yang telah diperoleh selama pelaksanaan *sprint*. Hasil diskusi ini kemudian didokumentasikan dan menjadi dasar untuk peningkatan *sprint* berikutnya. Selain itu, diskusi ini biasanya menghasilkan kesepakatan untuk melakukan perbaikan yang dapat ditindaklanjuti.

4) Release

Fase *Release* menekankan pada penyerahan *Accepted Deliverables* kepada pengguna, serta identifikasi, pendokumentasian, dan internalisasi pelajaran yang telah didapatkan selama proyek berlangsung.

a) *Ship Deliverables*

Pada proses ini, semua *deliverables* yang berasal dari *User Stories* yang telah diterima pada *sprint-sprint* sebelumnya diserahkan atau dialihkan kepada pemangku kepentingan bisnis yang relevan. Untuk mendokumentasikan penyelesaian rilis secara formal, dibuat *Working Deliverables Agreement* yang memuat kesepakatan mengenai *deliverables* yang telah diselesaikan.

b) *Retrospect Release*

Proses ini menjadi tahap akhir dari sebuah rilis, di mana pemangku kepentingan bisnis dan anggota tim inti *Scrum* berkumpul untuk merefleksikan proses rilis yang telah dilaksanakan. Dalam pertemuan ini, pelajaran yang didapat diidentifikasi, didokumentasikan, dan diinternalisasi untuk meningkatkan kualitas pada proyek selanjutnya. Seringkali, hasil diskusi ini berupa kesepakatan mengenai langkah-langkah perbaikan yang akan diterapkan pada proyek-proyek di masa mendatang.

2.1.5 *Laravel*

Laravel adalah sebuah kerangka kerja (*framework*) berbasis *open-source* yang dirancang untuk mempermudah pengembangan aplikasi *web* menggunakan PHP. Laravel menawarkan berbagai fitur unggulan, seperti *template engine* Blade untuk membangun antarmuka yang efisien, *routing* untuk pengaturan alur data, dan *migration* yang memungkinkan pengelolaan *database* secara terstruktur melalui skrip kode. Selain itu, Laravel memiliki dokumentasi yang lengkap, modularitas yang tinggi, dan dukungan komunitas yang luas, menjadikannya salah satu *framework* yang paling populer di kalangan pengembang *website* (Alpina & Witriyono, 2022).



Gambar 2. 3 Laravel

2.1.6 *Hypertext Preprocessor (PHP)*

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman berbasis *server-side scripting* yang digunakan secara luas dalam pengembangan aplikasi *web* dinamis. PHP memungkinkan pengolahan dan pemrosesan data langsung di server, dengan hasil yang dikirimkan ke *client* melalui *browser*. Sebagai bahasa *open-source*, PHP menawarkan fleksibilitas tinggi dalam penggunaannya dan

mendukung integrasi yang mudah dengan berbagai basis data, termasuk MySQL. PHP dikenal karena kemampuannya untuk menyatu dengan tag HTML dan menghasilkan halaman *website* yang interaktif dan dinamis. Keunggulan utama PHP meliputi kemudahan penggunaannya, kompatibilitas lintas *platform*, dan dukungan komunitas yang luas, menjadikannya pilihan populer dalam pengembangan *web* modern (Hermiati dkk., 2021).



Gambar 2. 4 PHP

2.1.7 My Structure Query Language (MySQL)

MySQL adalah sistem manajemen basis data relasional (*Relational Database Management System* atau RDBMS) berbasis *open-source* yang digunakan untuk menyimpan dan mengelola data dalam bentuk tabel. MySQL dikenal dengan kinerjanya yang cepat, stabil, dan fleksibel, serta mendukung pengurangan pengulangan data dan peningkatan integritas melalui pengaturan relasi antar tabel. Dengan sifatnya yang *open-source* dan fitur seperti keamanan akses melalui *user authentication*, MySQL menjadi pilihan utama dalam pengembangan aplikasi berbasis *web*, termasuk yang menggunakan *framework* seperti Laravel (Alpina & Witriyono, 2022).



Gambar 2. 5 MySQL

2.1.8 Usability Testing

Usability testing merupakan proses evaluasi yang bertujuan untuk mengetahui tingkat kemudahan, efisiensi, dan kepuasan pengguna dalam menggunakan suatu sistem. Pengujian ini penting untuk memastikan bahwa sistem dapat dioperasikan dengan baik oleh pengguna akhir serta mengurangi tingkat kesalahan dalam penggunaan (Nugroho dkk., 2022). Dalam evaluasi *usability*, terdapat lima indikator utama, yaitu *learnability* (kemudahan dalam mempelajari sistem), *memorability* (kemudahan dalam mengingat cara penggunaan sistem), *efficiency* (kecepatan dalam menyelesaikan tugas), *errors* (jumlah dan tingkat kesalahan yang terjadi), dan *satisfaction* (tingkat kepuasan pengguna terhadap sistem) (Sukmasetya dkk., 2020).

Pertanyaan pada kuesioner dapat dijawab menggunakan Skala Likert. Skala Likert merupakan salah satu alat ukur yang digunakan untuk mengetahui tingkat persetujuan responden terhadap suatu pernyataan. Pada penelitian ini, skala yang digunakan memiliki lima pilihan jawaban, yaitu Sangat Tidak Setuju (STS), Tidak Setuju (TS), Netral (N), Setuju (S), dan Sangat Setuju (SS). Setiap pilihan jawaban memiliki skor yang berbeda, seperti ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Skala *Likert* pada Kuesioner SUS

Skor	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Kuesioner dibagikan kepada responden yang telah ditentukan. Hasil jawaban responden kemudian dihitung untuk mengetahui nilai kelayakan sistem. Perhitungan dilakukan dengan membandingkan skor yang diperoleh dari responden dengan skor maksimal yang diharapkan. Rumus yang digunakan adalah sebagai berikut:

$$\text{Hasil(\%)} = \frac{\text{Jumlah Skor Total}}{\text{Jumlah Skor Maksimal} \times \text{Jumlah Pertanyaan} \times \text{Jumlah Responden}} \times 100\%$$

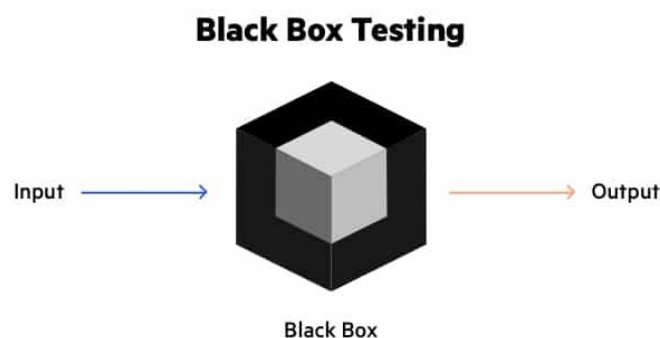
Setelah nilai persentase diperoleh, hasil tersebut dapat dikategorikan berdasarkan tingkat kelayakan sistem. Adapun kategori kelayakan dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Kategori Kelayakan *Usability Testing*

Angka (%)	Kategori Kelayakan
< 21	Sangat Tidak Layak
21-40	Tidak Layak
41-60	Cukup
61-80	Layak
81-100	Sangat Layak

2.1.9 Black Box Testing

Black Box Testing adalah metode pengujian perangkat lunak yang difokuskan pada pengujian aspek fungsional sistem tanpa memeriksa struktur internal kode. Pendekatan ini bertujuan untuk memastikan bahwa sistem bekerja sesuai dengan spesifikasi yang telah ditentukan, dengan menguji masukan dan keluaran berdasarkan skenario yang telah dirancang. *Black Box Testing* berfungsi sebagai alat untuk memvalidasi bahwa sistem dapat menangani berbagai kondisi masukan, termasuk data yang tidak valid atau tidak terduga, sehingga memastikan sistem dapat digunakan dengan aman tanpa kendala (Muslimin dkk., 2020).



Tabel 2. 3 *Black Box Testing*

2.2 Penelitian Terdahulu

Penelitian ini tentu tidak lepas dari penelitian-penelitian terdahulu yang dijadikan sebagai pembandingan. Hasil penelitian yang dijadikan pembandingan

tentunya tidak terlepas dari topik penelitian yang akan dilakukan, yaitu Pengembangan Sistem Informasi Akademik Berbasis *Web* Menggunakan Metode Scrum Untuk Pengolaan Modul BAAK (Studi Kasus: Institut Teknologi Bisnis Riau).

Febrianto Widyoutomo dkk. (2021) melakukan penelitian dengan judul “Pengembangan *Web Service* Modul Mahasiswa Pada Sistem Informasi Akademik Universitas Negeri Jakarta”. Penelitian ini berfokus pada pengembangan *web service* pada modul mahasiswa untuk menciptakan sistem informasi yang dapat mengelola berbagai data, baik data akademik maupun non-akademik. Sistem ini dirancang dengan menggunakan metode *Spiral*. Penelitian ini bertujuan untuk meningkatkan arsitektur sistem SIAKAD agar lebih adaptif terhadap perubahan dan pembaruan fitur. Sistem ini dibangun menggunakan *framework* Express.js dengan 25 *endpoint* yang mendukung integrasi sisi *frontend*. Dengan sistem ini, pengelolaan data akademik menjadi lebih terstruktur dan mudah untuk dikembangkan.

Penelitian kedua dilakukan oleh Aryanti dkk. (2021) dengan judul “Penerapan Metode *Rapid Application Development* Dalam Pengembangan Sistem Informasi Akademik Berbasis *Web*”. Penelitian ini bertujuan untuk mengembangkan sistem informasi akademik yang dapat memudahkan pengelolaan data akademik, seperti data absensi siswa, absen mengajar masuk dan absen mengajar guru, mengelola data nilai, jadwal Pelajaran, jadwal mengajar guru, serta dapat menerbitkan informasi-informasi yang berhubungan dengan akademik. Sistem ini akan dikembangkan dengan menggunakan metode *Rapid Application Development (RAD)*, metode ini digunakan karena model ini dianggap mengutamakan waktu, sehingga pengerjaannya relative singkat. Dengan adanya sistem ini, pelajar dan pengajar dapat mengakses informasi akademik secara cepat, praktis, dan efisien tanpa harus datang ke sekolah.

Penelitian ketiga dilakukan oleh Setiadana dkk. (2021) dengan judul “Pengembangan Sistem Penagihan Biaya Kuliah Dengan Fitur *WhatsApp* Menggunakan Metode *Scrum*”. Penelitian ini bertujuan untuk mempermudah proses penagihan biaya kuliah, seperti mengirimkan notifikasi tagihan kepada mahasiswa melalui *WhatsApp*, pengelolaan data pembayaran, serta penyimpanan

histori pembayaran. Dalam pengembangan sistem ini akan digunakan metode *Scrum* yang didukung dengan beberapa teknologi seperti *Subline Text*, dan *MySQL*. Dengan adanya sistem ini, petugas keuangan dapat lebih mudah mengelola tagihan kuliah secara efisien dan terorganisir.

Penelitian keempat dilakukan oleh Etrariadi & A'inunisya (2023) dengan judul “Pengembangan *Website* Manajemen Proyek Menggunakan Metode *Agile Scrum* (Studi Kasus Diskopindag Kota Malang)”. Sistem ini dirancang untuk mempermudah pengelolaan proyek, seperti penjadwalan, pemantauan, dan pelaporan kegiatan. Metode yang digunakan dalam penelitian ini adalah metode *Agile Scrum*, salah satu metodologi dari *System Development Life Cycle (SDLC)* berbasis *incremental model* dan iteratif. Dengan adanya sistem ini, manajemen proyek di Diskopindag dapat dilakukan secara terstruktur, cepat, dan lebih akurat dibandingkan dengan cara manual yang dilakukan sebelumnya.

Berdasarkan beberapa penelitian terdahulu, proyek akhir ini akan mengembangkan Sistem Informasi Akademik Berbasis *Web* Menggunakan Metode *Scrum* untuk Pengelolaan Modul BAAK (Studi Kasus: Institut Teknologi Bisnis Riau). Sistem ini akan dibangun menggunakan *framework* *Laravel* dan memanfaatkan *MySQL* sebagai basis data utama. Sistem ini bertujuan untuk mempermudah pengelolaan data akademik secara terintegrasi, mempercepat proses administrasi akademik, serta menyediakan informasi yang akurat dan *realtime* bagi mahasiswa, dosen, dan staf kampus. Metode *Scrum* dipilih karena bersifat iteratif dan kolaboratif, sehingga meminimalkan risiko kesalahan selama proses pengembangan dan memungkinkan perbaikan sistem secara bertahap berdasarkan masukan dari pengguna.

Tabel 2.1 Perbandingan Variabel Penelitian

Kategori	Peneliti 1	Peneliti 2	Peneliti 3	Peneliti 4
Peneliti	Febrianto Widyoutomo dkk. (2021)	Aryanti dkk. (2021)	Setiadana dkk. (2021)	Etrariadi & A'inunisya (2023)
Judul	Pengembangan <i>Web Service</i> Modul Mahasiswa Pada Sistem Informasi Akademik Universitas Negeri Jakarta	Penerapan Metode <i>Rapid Application Development</i> Dalam Pengembangan Sistem Informasi Akademik Berbasis <i>Web</i>	Pengembangan Sistem Penagihan Biaya Kuliah Dengan Fitur <i>WhatsApp</i> Menggunakan Metode <i>Scrum</i>	Pengembangan <i>Website</i> Manajemen Proyek Menggunakan Metode <i>Agile Scrum</i> (Studi Kasus Diskopindag Kota Malang)
Platform / Jenis Aplikasi	<i>Web service</i>	<i>Web base</i>	<i>Web base</i>	<i>Web base</i>
Metode	Spiral	<i>Rapid Application Development (RAD)</i>	<i>Scrum</i>	<i>Agile Scrum</i>

Kategori	Peneliti 1	Peneliti 2	Peneliti 3	Peneliti 4
Data	Data akademik dan non-akademik pada modul mahasiswa SIAKAD Universitas Negeri Jakarta	Data absensi, nilai, jadwal pelajaran, jadwal mengajar guru, dan informasi akademik	Data tagihan biaya kuliah, pembayaran, notifikasi WhatsApp, dan histori pembayaran	Data proyek, jadwal, pemantauan pekerjaan, dan laporan kegiatan
Fokus Utama	Meningkatkan arsitektur SIAKAD agar lebih adaptif terhadap perubahan dan integrasi <i>frontend</i>	Mempermudah pengelolaan data akademik berbasis <i>web</i> agar informasi akademik dapat diakses dengan cepat	Mempermudah proses penagihan biaya kuliah dan pengelolaan pembayaran mahasiswa	Mempermudah pengelolaan proyek sektor publik secara terstruktur, cepat, dan akurat
Teknologi	Express.js, REST API, dan MySQL	PHP, MySQL, HTML, dan CSS	PHP, MySQL, dan WhatsApp API	PHP, MySQL, dan JavaScript
Keterkaitan dengan Penelitian Ini	Sama-sama membahas sistem informasi akademik, tetapi penelitian ini berfokus pada pengembangan modul mahasiswa dalam bentuk <i>web service</i>	Sama-sama mengembangkan sistem informasi akademik berbasis <i>web</i> , tetapi menggunakan metode RAD	Sama-sama menggunakan Scrum, tetapi objek penelitian berfokus pada penagihan biaya kuliah	Sama-sama menggunakan <i>Agile Scrum</i> , tetapi objek penelitian berfokus pada manajemen proyek