

Perancangan Aplikasi *Mobile* Sistem Informasi Akademik Mahasiswa Sebagai Salah Satu Tahapan Metode *DevOps*

Satrio Muslim Wibowo¹, Erma Susanti^{2*}, Erfanti Fatkhiah³

^{1, 2*, 3}Fakultas Teknologi Informasi dan Bisnis, Institut Sains & Teknologi AKPRIND, Yogyakarta, Indonesia

Email: ¹satriomuslim2627@gmail.com, ²erma@akprind.ac.id, ³erfanti@akprind.ac.id

Abstract

The student service information system is important for processing information at an educational institution. IST AKPRIND campus has implemented information services using mobile devices. In student information services, there is some information about lectures, such as study result cards, course schedules, student profiles, and student attendance. However, there are still areas for improvement in application, such as the lack of academic features in the service. The solutions in this study in the form of additions, improvements, and updates include adding a dynamic announcement feature, improving the payment history feature, improving the scan feature on self-presence, adding features to display the cumulative assessment index, semester credit units taken, active student status and updating the display design. The DevOps approach is used by the system development method because it can reduce the time between software creation and operation without sacrificing software quality. This study aims to apply the DevOps method design phase to the development of student service applications. The following stage of development is built upon the outcomes of this design.

Keywords: *Information System, Student, Service, Mobile Apps, Devops*

Abstrak

Sistem informasi layanan mahasiswa adalah sistem penting untuk mengolah informasi pada sebuah institusi pendidikan. Kampus IST AKPRIND telah mengimplementasikan layanan informasi dengan menggunakan perangkat *mobile*. Pada layanan informasi mahasiswa terdapat beberapa informasi mengenai perkuliahan seperti kartu hasil studi, jadwal mata kuliah, profil mahasiswa, dan presensi mahasiswa. Namun, masih terdapat kekurangan pada aplikasi seperti pada fitur-fitur akademik di layanan tersebut. Solusi penelitian ini berupa penambahan, perbaikan, dan pembaharuan meliputi menambahkan fitur pengumuman dinamis, memperbaiki fitur *history* pembayaran, memperbaiki fitur pemindai pada presensi mandiri, menambahkan fitur untuk menampilkan indeks penilaian kumulatif, satuan kredit semester tempuh, status aktif mahasiswa dan memperbarui desain tampilan. Metode pengembangan sistem menggunakan metode *DevOps* dipilih karena dapat mempersingkat waktu dalam pengembangan dan pengoperasian perangkat lunak, namun dengan tidak mengurangi kualitas perangkat lunak. Penelitian ini bertujuan untuk mengimplementasikan fase perancangan pada metode *DevOps* yang digunakan untuk pengembangan aplikasi layanan mahasiswa. Hasil perancangan ini merupakan dasar bagi fase pengembangan berikutnya.

Kata Kunci: Sistem Informasi, Layanan, Mahasiswa, Aplikasi Mobile, *Devops*

1. PENDAHULUAN

Sistem informasi layanan mahasiswa pada sebuah institusi pendidikan merupakan sebuah sistem yang dapat dimanfaatkan untuk menyimpan data dan juga mengolah informasi akademik dari pengguna (Pratama & Kamisutara, 2021). Adapun sistem informasi layanan mahasiswa yang kebanyakan dipakai oleh institusi pendidikan hanya

menggunakan web. Namun, untuk mempermudah pengguna sebagian besar institusi pendidikan sudah mendesain dan mengimplementasi sistem informasi layanan mahasiswa bukan hanya menggunakan platform web tetapi juga di desain dengan menggunakan perangkat Mobile. Dengan menggunakan perangkat Mobile pengguna akan lebih mudah mengakses informasi layanan kapan saja dan dimana saja secara lebih efektif. Berdasarkan data GlobalState, pengguna perangkat *mobile* di Indonesia hingga September 2022 mencapai 89,77% (GlobalState, 2022). Data tersebut menjelaskan bahwa terdapat banyak pengguna perangkat Mobile di Indonesia.

Salah satu institusi pendidikan di Yogyakarta yaitu Institut Sains & Teknologi AKPRIND (IST AKPRIND) sudah mengimplementasikan sistem informasi layanan mahasiswa versi 1 dengan menggunakan perangkat *mobile*. Sistem informasi layanan mahasiswa tersebut memiliki beberapa informasi mengenai perkuliahan seperti kartu hasil studi, jadwal mata kuliah, profil universitas, presensi mahasiswa dan informasi akademik lainnya. Namun, masih terdapat beberapa fitur akademik dalam sistem informasi *mobile* untuk mahasiswa yang perlu dikembangkan kembali sesuai kebutuhan. Adapun solusi untuk mengatasi permasalahan ketersediaan fitur aplikasi yang sesuai kebutuhan pengguna yaitu dengan membuat sistem informasi layanan mahasiswa berbasis Mobile menggunakan metode *DevOps*. Metode *DevOps* adalah sebuah pendekatan kolaboratif dan integratif dalam proses pengembangan infrastruktur perangkat lunak yang terjadi antara *dev* dan *ops* (pengembang dan operasional) perangkat lunak, sehingga dapat mengurangi tahapan pengembangan dibandingkan dengan menggunakan metode lama (Tohirin, Utami, Widianto, & Mauludyansah, 2020). Aplikasi ini dibangun untuk memberikan kemudahan pada mahasiswa dalam mengakses layanan akademik. Adapun kemudahan yang diberikan seperti menambahkan fitur pengumuman dinamis, memperbaiki fitur *history* pembayaran, memperbaiki fitur pemindai pada presensi mandiri, menambahkan fitur untuk menampilkan indeks penilaian kumulatif, satuan kredit semester tempuh dan status aktif mahasiswa serta pembaruan desain tampilan. Penambahan dan perbaikan fitur serta pembaruan desain tampilan aplikasi sistem informasi layanan mahasiswa berbasis *mobile* dalam pengembangannya menggunakan metode *DevOps* untuk mendapatkan aplikasi yang efisien sesuai kebutuhan pengguna. Metode *DevOps* banyak digunakan oleh peneliti karena dapat mempercepat waktu dalam pengembangan dan pengoperasian suatu perangkat lunak, namun tetap dapat mempertahankan kualitas perangkat lunak yang dihasilkan.

Berbagai penelitian tentang penerapan aplikasi *mobile* layanan siswa atau mahasiswa pernah dilakukan oleh berbagai peneliti. Penelitian (RG Hattari, Lutfi, & Khairan, 2018) bertujuan untuk mengembangkan sistem informasi berbasis *mobile* web untuk Universitas Khairun Ternate. Pengembangan dilakukan menggunakan metode *prototype* dalam perancangannya terdiri dari 3 proses antara lain pertama fase mendengarkan pelanggan, kedua fase membangun dan merancang sistem, serta ketiga melakukan fase pengujian pelanggan. Hasil perancangan sistem informasi akademik berbasis Android pada universitas tersebut ini masih belum dilengkapi dengan akses untuk pengguna lain, sehingga masih diperlukan pengembangan pada aplikasi. Penelitian (Riyadli & Arliyana, 2019) mengembangkan sistem informasi akademik yang menampilkan hasil absensi, memberikan informasi jadwal pelajaran, dan memberikan akses kepada guru untuk memberikan penilaian para siswa-siswi melalui sebuah aplikasi *mobile*. Peneliti melakukan pengembangan seperti *Research & Development* yang berkaitan dengan bidang ilmu teknologi informasi dan menggunakan metode pengumpulan data berupa wawancara, observasi, studi pustaka, dokumentasi, dan menggunakan kuesioner. Berdasarkan hasil pengujian, aplikasi menghasilkan informasi

akademik untuk digunakan oleh guru dan siswa dalam pembelajaran di SMAN 1 Kumai Kotawaringin Barat.

Penelitian (Setiawan, Suratno, & Lutfi, 2018) menerapkan metodologi analisis dan desain sistem dengan metode *waterfall*. Tujuannya adalah untuk melakukan analisis, membuat desain, dan juga merancang sistem aplikasi *mobile* informasi akademik pada Universitas Jambi. Penelitian tersebut berhasil merancang sistem informasi akademik untuk menyimpan transaksi akademik antara mahasiswa, dosen dan program studi. Tidak hanya itu, aplikasi mobile akan mempermudah akses terhadap aktivitas-aktivitas yang terjadi pada perguruan tinggi selama berada dalam jaringan internet. Penelitian lainnya tentang sistem informasi akademik *mobile* dilakukan (Nuraeni, Setiawan, Nurhakim, & Mubarok, 2022) sebagai media informasi akademik *online*. Penelitian dilakukan untuk akses siswa dan orang tua pada sistem informasi akademik (SIAKAD) melalui perangkat *smartphone*. Hal ini disebabkan terdapat perubahan sistem pembelajaran dari luring dimana siswa pergi mendatangi sekolah secara langsung untuk mendapatkan informasi, menjadi sistem pembelajaran daring. Pengembangan sistem informasi akademik (SIAKAD) menggunakan metode *Rational Unified Process* (RUP) memiliki empat fase mulai dari fase *inception* (permulaan), fase *elaboration* (perluasan), fase *construction* (pembangunan) dan terakhir fase *transition* (transisi). Hasil dari penelitian ini selain memudahkan siswa dan orang tua juga memudahkan pihak sekolah terutama staf administrasi untuk menginput data file Excel serta meminimalisir kehilangan data.

Kekurangan media informasi yang terdapat pada penggunaan teknologi *sms-gateway* di SMK Negeri 1 Depok membuat (Fauzi & Harli, 2019) meneliti tentang sistem informasi *mobile*. Penelitian dilakukan menggunakan metode *Rapid Application Development* (RAD). Sistem dibangun dengan cepat dan dilakukan pengujian perangkat lunak untuk menguji fungsionalitas aplikasi menggunakan metode *Black Box*. Sedangkan penelitian (Aminudin, Gusti, Putra, & Prismane, 2021) berusaha memecahkan permasalahan penilaian kompetensi siswa dengan subjek penelitian siswa kelas XI Multimedia yang masih dilakukan dengan cara manual di SMK Negeri 1 Driyorejo Gresik. Hasilnya sebuah sistem informasi penilaian portofolio berbasis web digunakan untuk membantu pendidik dalam melakukan penilaian dan melakukan pemantauan kompetensi siswa untuk mencapai standar kompetensi sekolah. Metode *DevOps* terbagi menjadi beberapa tahapan yaitu mulai dari tahap observasi dengan melakukan pengamatan untuk mengetahui kondisi dan permasalahan yang terjadi di sekolah, sehingga bisa diberikan solusi yang tepat. Tahap akhir yaitu pengembangan proyek, uji coba sistem dan pengambilan sampel sesuai populasi. Sistem informasi web penilaian portofolio siswa ini merupakan hasil produk akhir berupa media informasi pengumpulan ketrampilan dan penilaian hasil karya siswa.

Tabel 1 menampilkan beberapa perbandingan penelitian tentang aplikasi layanan siswa/mahasiswa. Perbedaan antara penelitian sebelumnya dan penelitian ini adalah pada objek pengguna yang diteliti dan metode pengembangan sistem. Pengguna sistem adalah mahasiswa IST AKPRIND, sedangkan pada penelitian sebelumnya yaitu sekolah. Metode pengembangan yang digunakan adalah *DevOps*, sedangkan pada penelitian sebelumnya adalah *Prototype*, *Research & Development*, *Waterfall*, *Rational Unified Process*, dan *Rapid Application Development*. Adanya kekurangan pada fitur-fitur aplikasi *mobile* sistem informasi layanan mahasiswa di IST AKPRIND menyebabkan perlunya penambahan dan perbaikan fitur serta pembaruan desain. Penambahan, perbaikan, dan pembaharuan tersebut dibuat dengan metode *DevOps* meliputi menambahkan fitur pengumuman dinamis, memperbaiki fitur *history* pembayaran, memperbaiki fitur pemindai pada presensi mandiri, menambahkan fitur untuk

menampilkan indeks penilaian kumulatif, satuan kredit semester tempuh, dan status aktif mahasiswa serta memperbarui desain tampilan.

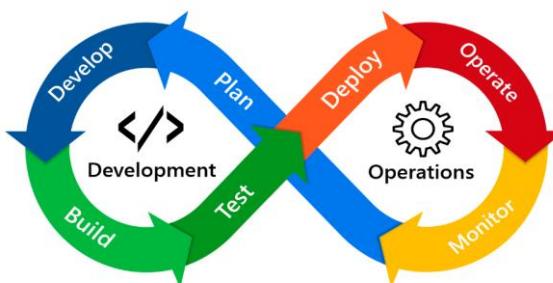
Tabel 1. Tabel Perbandingan Beberapa Penelitian Sistem Layanan Siswa/Mahasiswa

No	Peneliti	Tempat	Metode Pengembangan Sistem
1	(RG Hattari et al., 2018)	Universitas Khairun Ternate	<i>Prototype</i>
2	(Setiawan et al., 2018)	Universitas Jambi	<i>Waterfall</i>
3	(Riyadli & Arliyana, 2019)	SMAN 1 Kumai Kotawaringin Barat	<i>Research and Development</i>
4	(Fauzi & Harli, 2019)	SMK Negeri 1 Depok	<i>Rapid Application Development</i>
6	(Aminudin et al., 2021)	SMK Negeri 1 Driyorejo Gresik	DevOps
7	(Nuraeni et al., 2022)	SMPN 1 Banyuresmi	<i>Rational Unified Process (RUP)</i>

2. METODOLOGI PENELITIAN

2.1 Metode *Devops*

DevOps adalah gerakan budaya yang mengubah cara individu berpikir tentang pekerjaan mereka, menghargai keragaman pekerjaan yang dilakukan, mendukung proses yang disengaja yang mempercepat laju bisnis mewujudkan nilai, dan mengukur efek perubahan sosial dan teknis. Ini adalah cara berpikir dan cara bekerja yang memungkinkan individu dan organisasi untuk mengembangkan dan mempertahankan praktik kerja yang berkelanjutan. Ini adalah kerangka kerja budaya untuk berbagi cerita dan mengembangkan empati, memungkinkan orang dan tim untuk mempraktikkan keahlian mereka dengan cara yang efektif dan langgeng (Davis & Daniels, 2016).



Gambar 1. Siklus Metode *DevOps*

Sumber: <https://agus-hermanto.com/>

Siklus metode *DevOps* (Yuanita, Wijayanto, & Cahyono, 2022) disajikan pada Gambar 1. Penjelasan siklus dan langkah dari metode *DevOps* yaitu antara lain:

1. Fase *Plan* (Perencanaan)

Fase perencanaan merupakan proses mengidentifikasi tujuan dan persyaratan untuk desain dan pengembangan perangkat lunak aplikasi. Fase ini juga memiliki aktivitas lain seperti perencanaan rilis, studi literatur, kebijakan persyaratan, pembuatan model UML dan rencana pengujian untuk pembaruan aplikasi.

2. Fase *Develop* (Perancangan)

Fase perancangan berfokus pada pengembangan dan peninjauan kode perangkat lunak aplikasi. Kode dikembangkan menggunakan bahasa pemrograman dan IDE yang sesuai. Kode dikelola menggunakan sistem kontrol versi (*version control*). Proses integrasi dan pengujian unit didasarkan pada alat otomasi.

3. Fase *Build* (Pembuatan)

Fase pembuatan merupakan integrasi modul-modul perangkat lunak. Fase ini dilakukan untuk membuat fitur produk atau file eksekusi produk yang dikembangkan. Selain itu, ada proses evaluasi yang mengukur penerapan hasil perangkat lunak terhadap persyaratan yang diberikan.

4. Fase *Test* (Pengujian)

Fase pengujian otomasi terus dilakukan untuk memastikan pengukuran kualitas perangkat lunak yang telah dikembangkan telah baik. Tujuan pengujian dilakukan untuk memastikan bahwa potensi kesalahan dalam perangkat lunak yang dikembangkan telah dihilangkan. Selain itu juga memastikan aplikasi/perangkat lunak telah dikirimkan dengan andal.

5. Fase *Deploy* (Penerapan)

Fase penerapan berfokusnya pada implementasi ulang secara berkelanjutan produk perangkat lunak di lingkungan produksi. Fase ini juga termasuk mengkonfigurasi platform dan mengelola target sumber daya. Selanjutnya, jika fitur atau produk telah berhasil diterapkan, maka siap untuk dirilis.

6. Fase *Operate* (Pengoperasian)

Fase pengoperasian terdiri dari pemasangan infrastruktur, perubahan skalabilitas infrastruktur, manajemen dan konfigurasi infrastruktur, pemulihan data, serta manajemen basis data.

7. Fase *Monitor* (Pemantauan)

Fase pemantauan digunakan untuk memantau performa aplikasi yang digunakan dengan cara mengumpulkan dan menganalisis data penggunaan aplikasi, sehingga dapat membantu proses deteksi, identifikasi anomali dan memberikan umpan balik untuk peningkatan perangkat lunak yang dikembangkan. Pelacakan masalah dan diagnosis yang berkelanjutan penting dilakukan untuk mendorong pengembangan aplikasi selama siklus rilis aplikasi berlangsung.

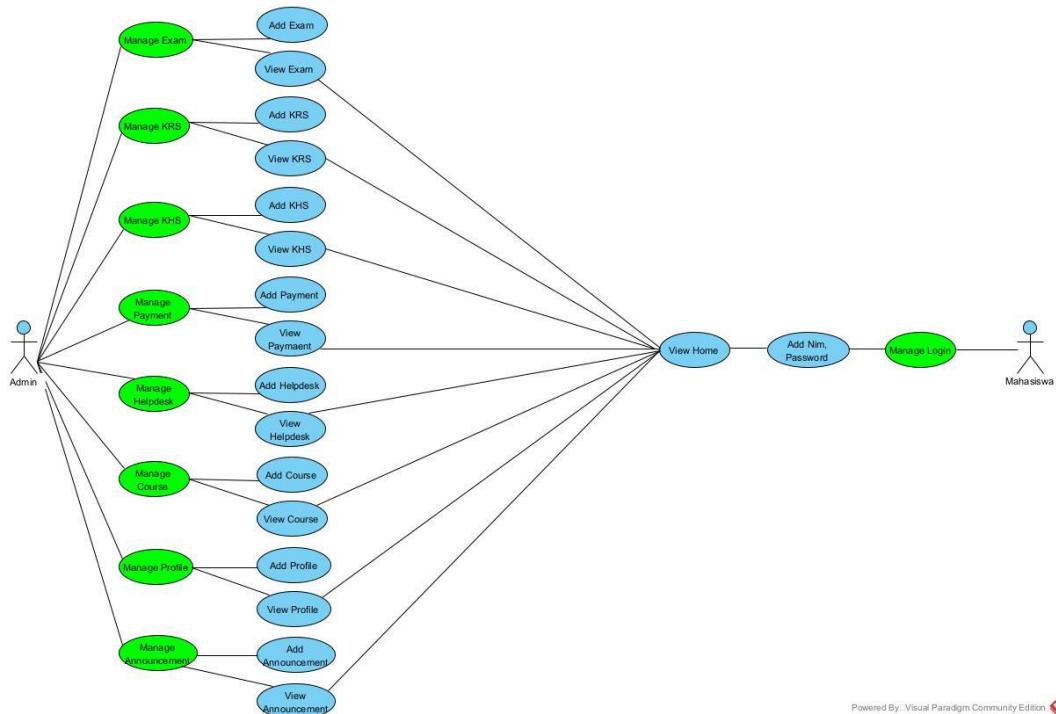
Metode *DevOps* adalah singkatan gabungan dari dua kata yaitu *development* (*dev*) dan *operations* (*ops*). Metode ini berfokus pada proses kolaborasi dan komunikasi antara pengembang dan pengguna profesional teknologi informasi. Adapun tujuan metode *DevOps* yaitu untuk merancang perangkat lunak melalui kolaborasi kerja sama antara tim-tim teknologi informasi untuk pengembangan yang lebih terorganisir, ideal, dan pemeliharaan perangkat lunak yang cepat (Indra & Rizky, 2020). Pola pikir pada metode *DevOps* yaitu adanya koordinasi antar tim secara singkat, sehingga tidak membutuhkan banyak pertanyaan dalam pengembangannya. Tim pengembangan dan operasi cukup hanya mengkonfigurasi komponen-komponen yang dibutuhkan melalui prosedur yang disusun, sehingga hasil proses pengembangan dapat dilakukan secara efektif dan efisien (Zalukhu & Arinal, 2021).

3. HASIL DAN PEMBAHASAN

Pemodelan merupakan tahap merancang suatu sistem. Setelah pemodelan selesai selanjutnya dilakukan tahap pembuatan program (*coding*). Perancangan sistem dalam riset ini dimulai dengan memodelkan permasalahan-permasalahan ke dalam bentuk diagram *Unified Modelling Language* (UML). *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram* digunakan untuk menyusun rancangan aplikasi layanan mahasiswa di IST AKPRIND.

3.1. Use Case Diagram

Use Case Diagram menggambarkan hubungan dalam sistem antara sejumlah *external actor* dengan *use case* sistem. Fungsi diagram ini untuk mendeskripsikan bagaimana sistem akan terlihat di mata pengguna layanan. Sistem yang dikembangkan ini melibatkan dua aktor yaitu admin dan mahasiswa. Admin berperan dalam melakukan aksi terhadap data. Sedangkan mahasiswa dalam penelitian ini berperan menggunakan aplikasi dalam pembelajaran di kampus. Sebelum melakukan proses pada menu-menu tersebut, mahasiswa harus melakukan login terlebih dahulu untuk bisa mengakses menu-menu dalam aplikasi. Hasil rancangan *Use Case Diagram* disajikan pada Gambar 2.



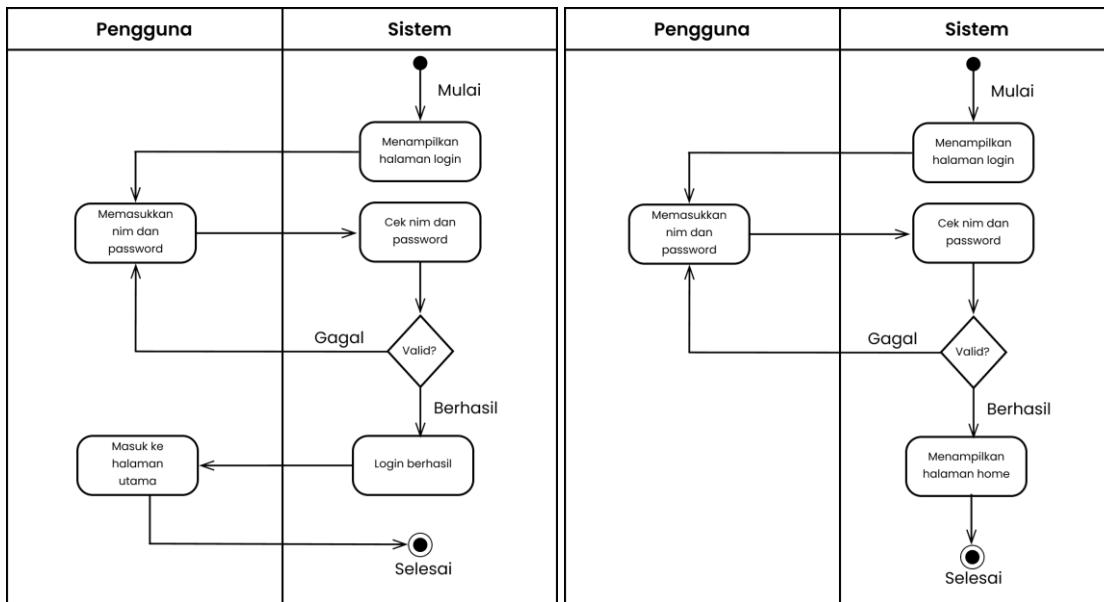
Gambar 2. *Use Case Diagram*

Powered By: Visual Paradigm Community Edition

3.2. Activity Diagram

Activity Diagram menggambarkan aktivitas yang dilakukan *user* terhadap aplikasi. *Activity Diagram* berfungsi untuk mendeskripsikan prosedural, proses dan aliran kerja aplikasi. Kegunaan diagram ini yaitu menunjukkan tahapan, melakukan pengambilan keputusan dan percabangan. Selain itu, *activity diagram* dapat menunjukkan cara menjalankan aplikasi dengan tahap-tahap yang disarankan.

Alur *Activity Diagram* untuk *login user* pada sistem informasi layanan mahasiswa pada penelitian ini dimulai dengan aktivitas *user* harus melakukan *login* terlebih dahulu dengan memasukkan nim dan *password*. Nim dan *password* akan divalidasi. Jika proses validasi berhasil, maka *user* dapat melakukan aktivitas pada sistem, tetapi jika validasi tersebut gagal *user* harus mengulangi proses *login* sampai berhasil dan menampilkan halaman sistem yang dituju. Halaman *home* dapat digunakan apabila *user* sistem berhasil *login*. Setelah *user* berhasil *login* ke sistem, selanjutnya sistem akan menampilkan halaman utama, kemudian *user* dapat memilih beberapa menu. Beberapa menu pada halaman *home* antara lain terdapat menu *exam*, KRS, KHS, *present*, *payment*, *helpdesk*, dan menu pengumuman. *Activity diagram login* dan halaman *home* ditunjukkan pada Gambar 3.

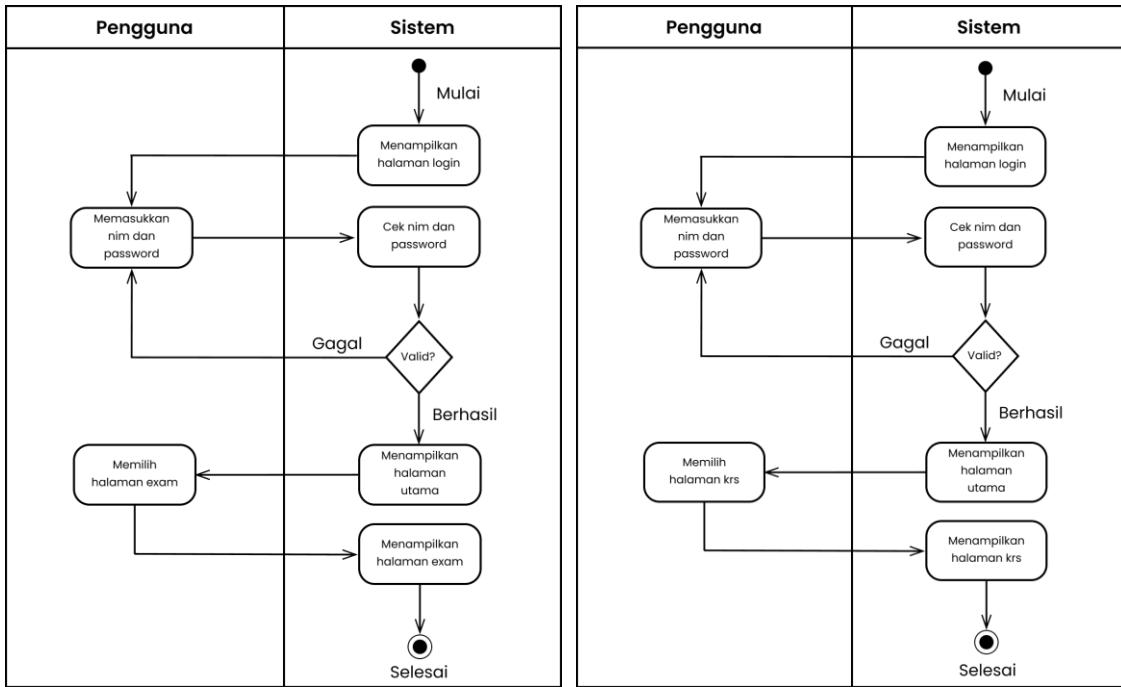


Gambar 3. Activity Diagram Halaman Login dan Halaman Home

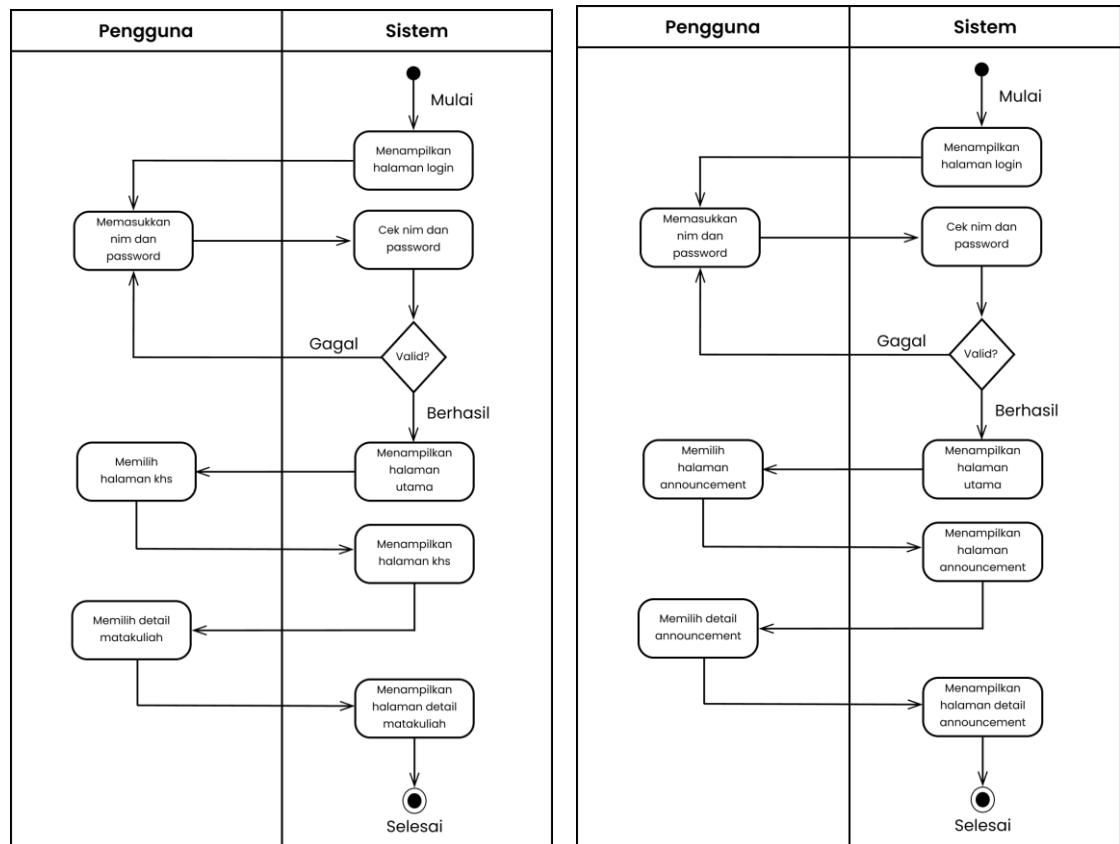
Halaman *exam* dapat digunakan apabila *user* sistem telah berhasil *login*. Selanjutnya sistem akan menampilkan halaman utama dan memilih menu *exam*. Pada halaman *exam*, *user* dapat melihat jadwal ujian tengah semester (UTS) dan ujian akhir semester (UAS). Detail tampilan UTS dan UAS berupa penjelasan mata kuliah yang diujikan, tanggal pelaksanaan ujian, dan ruangan ujian berlangsung. Halaman KRS dapat digunakan apabila *user* sistem berhasil *login*. Selanjutnya sistem akan menampilkan halaman utama dan memilih menu KRS. Pada halaman KRS, *user* dapat melihat kartu rencana studi selama satu semester yang sudah diinputkan pada web. *Activity diagram* halaman *exam* dan halaman KRS ditunjukkan pada Gambar 4.

Halaman KHS dapat digunakan apabila *user* sistem berhasil *login* dan akan menampilkan halaman utama. *User* dapat memilih menu KHS sehingga dapat melihat kartu hasil studi sesuai semester tempuh masa perkuliahan. Selain itu, *user* dapat melihat detail hasil studi dari setiap mata kuliah pada halaman ini. Halaman *announcement* dapat digunakan apabila *user* sistem berhasil *login*. Selanjutnya *user* dapat memilih menu *announcement*. Selanjutnya sistem akan menampilkan halaman pengumuman. *User* dapat melihat pengumuman dinamis pada aplikasi. Hasil rancangan *Activity Diagram* halaman KHS dan halaman pengumuman disajikan pada Gambar 5.

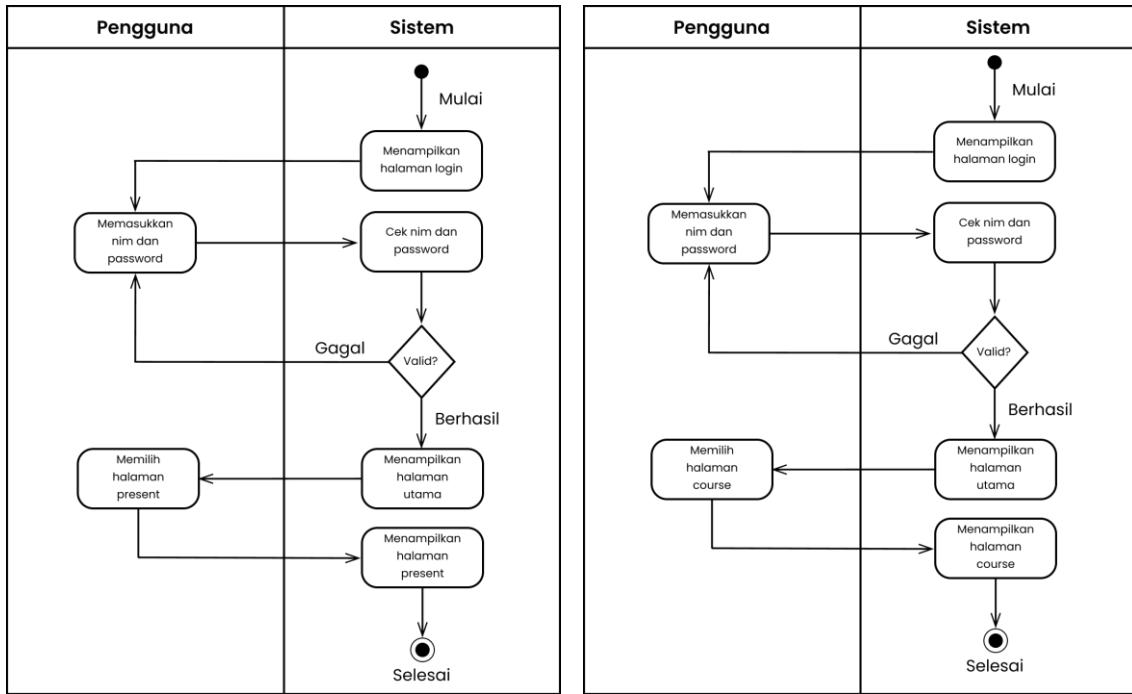
Halaman *present* dapat dilakukan apabila *user* sistem berhasil *login*. Setelah memilih menu *present* sistem akan menampilkan halaman *present*. *User* dapat melihat jumlah kehadiran selama satu semester perkuliahan dari setiap mata kuliah. Halaman *course* dapat digunakan apabila *user* sistem berhasil *login*. Selanjutnya *user* dapat memilih menu *course* pada *bottom navigation*. Setelah memilih menu *course* sistem akan menampilkan halaman *course*. *User* dapat melihat jadwal kuliah pada hari tersebut yang meliputi detail mata kuliah, nama dosen, ruangan perkuliahan, dan jam berlangsungnya perkuliahan. *Activity diagram* halaman *present* dan halaman *course* ditunjukkan pada Gambar 6.



Gambar 4. Activity Diagram Halaman Exam dan Halaman KRS

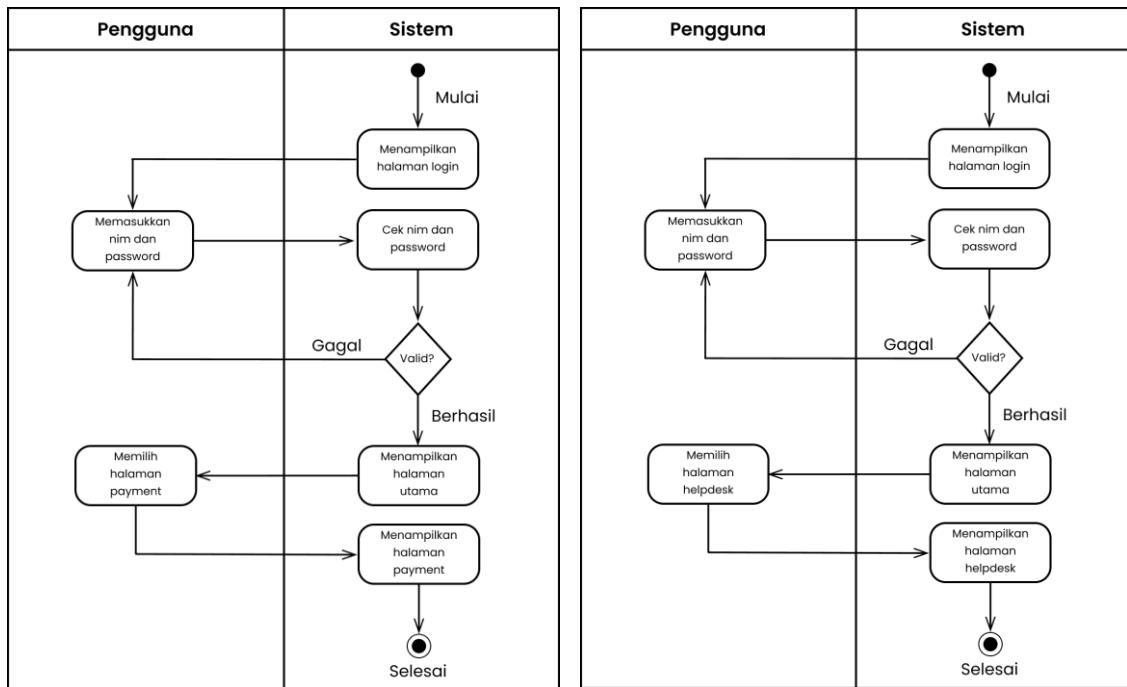


Gambar 5. Activity Diagram Halaman KHS dan Halaman Announcement



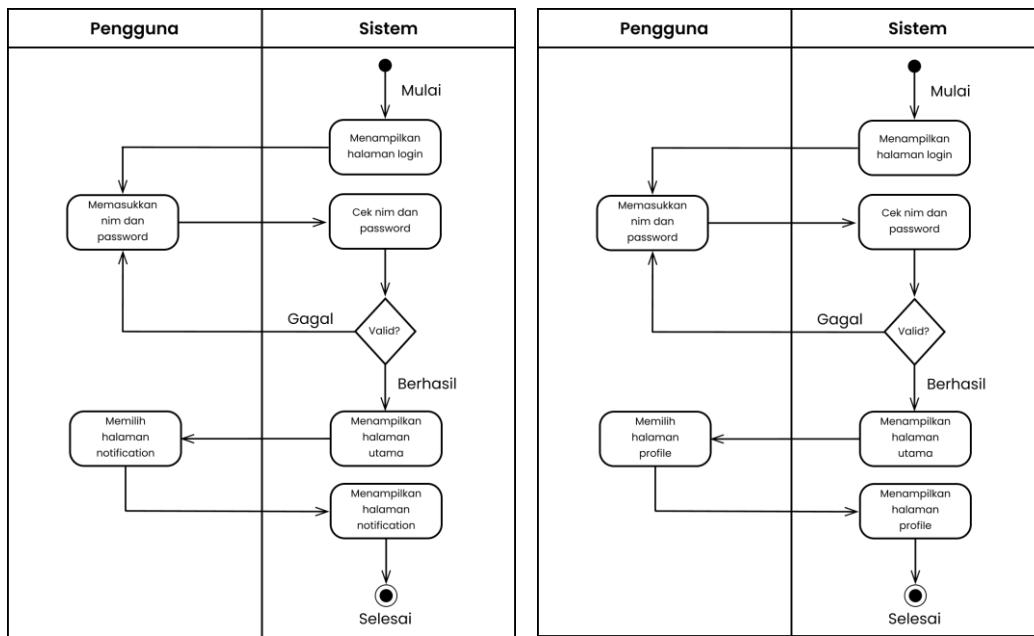
Gambar 6. *Activity Diagram* Halaman Halaman Present dan Halaman Course

Halaman *payment* dapat dilakukan apabila *user* sistem berhasil *login*. Setelah memilih menu *payment* sistem akan menampilkan halaman *payment*. *User* dapat melihat *history* pembayaran selama masa perkuliahan. Halaman *helpdesk* dapat digunakan apabila *user* sistem berhasil *login*. Selanjutnya sistem akan menampilkan halaman utama dan memilih menu *helpdesk*. Pada halaman *helpdesk*, *user* dapat melihat berbagai pertanyaan dan pernyataan sesuai dengan kebutuhan. Hasil rancangan *Activity Diagram* halaman *payment* dan halaman *helpdesk* ditunjukkan pada Gambar 7.

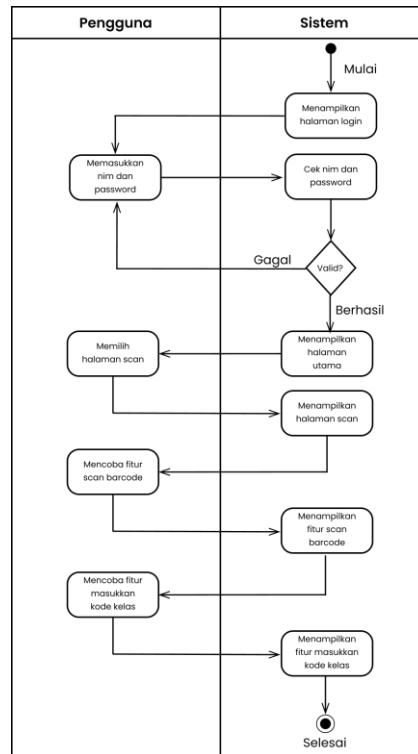


Gambar 7. *Activity Diagram* Halaman Payment dan Halaman Helpdesk

Halaman *notification* dapat digunakan apabila *user* sistem berhasil *login*. Selanjutnya *user* dapat memilih menu *notification* pada *bottom navigation*. Setelah memilih menu *notification* sistem akan menampilkan halaman *notification*. *User* dapat melihat *list notif* yang muncul pada aplikasi. Halaman *profile* dapat digunakan apabila *user* sistem berhasil *login*. Selanjutnya *user* dapat memilih menu *profile* pada *bottom navigation*. Pada halaman *profile*, *user* dapat melihat informasi data diri seperti nama, nim, email, jurusan dan sebagainya. Selain itu, *user* juga dapat mengakses menu ubah *password*, notifikasi, petunjuk, versi aplikasi, dan *logout*. Hasil rancangan *Activity Diagram* halaman *notification* dan halaman *profile* ditunjukkan pada Gambar 8.



Gambar 8. *Activity Diagram* Halaman *Notification* dan Halaman *Profile*

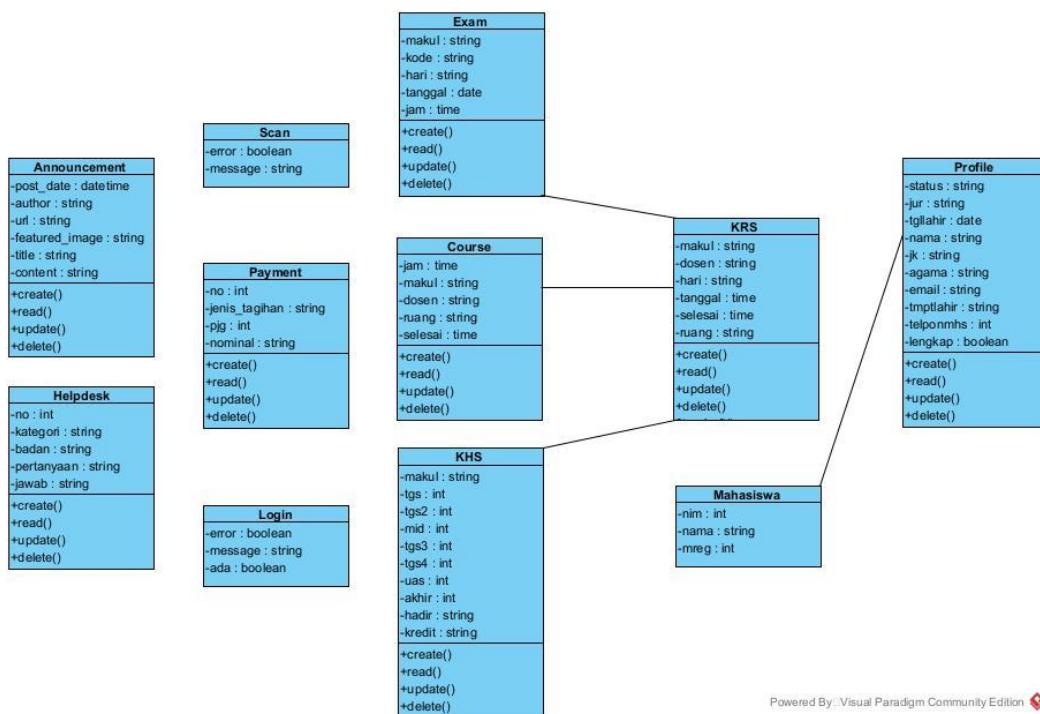


Gambar 9. *Activity Diagram* Halaman *Scan*

Hasil rancangan *Activity Diagram* halaman *scan* ditampilkan pada Gambar 9. Halaman *scan* dapat digunakan apabila *user* sistem berhasil *login*. Selanjutnya *user* dapat memilih menu *scan* pada *bottom navigation*. Setelah memilih menu *scan* sistem akan menampilkan halaman *scan*. *User* dapat melakukan presensi menggunakan metode *scan barcode* atau dengan memasukkan kode kelas pada perkuliahan.

3.3. Class Diagram

Class Diagram adalah konstruksi untuk memperlihatkan gambaran dari *database* pada suatu program. Gambaran *class diagram* terdiri dari beberapa *class* (kelas) yang di dalamnya terdapat *attribute* dan *operation*. *Attribute* dimiliki oleh *class*, sedangkan *operation* dapat dilakukan oleh *class* tersebut. Dalam penelitian ini, *class diagram* digambarkan melalui *API server* berbentuk *JSON* yang menggunakan relasi *one to one* antara entitas mahasiswa dan *profile* serta relasi *one to many* antara entitas krs ke *exam*, *course*, dan khs. Hasil rancangan *Class Diagram* ditampilkan pada Gambar 10.



Gambar 10. Class Diagram

4. KESIMPULAN

Perancangan sistem yang dibuat telah memenuhi tujuan awal pembuatan aplikasi berbasis *mobile* dengan menerapkan metode DevOps sebagai metode pengembangan aplikasi. Aplikasi ini didukung dengan data yang valid dari *API server*, sehingga data yang digunakan sesuai dengan kebutuhan *user*. Perancangan yang berhasil dikembangkan pada penelitian ini memiliki tiga belas fitur yaitu fitur *login*, fitur *home*, fitur *exam*, fitur *krs*, fitur *khs*, fitur *present*, fitur *payment*, fitur *helpdesk*, fitur *announcement*, fitur *course*, fitur *scan*, fitur *notification*, dan fitur *profile*. Hasil rancangan yang telah berhasil dibuat pada penelitian ini didasarkan pada kebutuhan pengguna dan hasil evaluasi dari kekurangan dan kelemahan yang ada pada aplikasi sistem informasi akademik mahasiswa versi 1 yang sudah diimplementasikan sebelumnya. Semua fitur tersebut telah berhasil dirancang untuk memperbaiki dan mengembangkan aplikasi *mobile* layanan mahasiswa.

Penelitian tentang sistem informasi akademik berbasis *mobile* ke depan masih dapat dikembangkan lagi seperti memperbarui data, memperbaiki *bug* aplikasi, menstabilkan koneksi ke API, memperbaiki fitur presensi QR Code yang masih sering mengalami kegagalan dalam membaca data, penambahan *notifikasi realtime* di dalam aplikasi untuk mengetahui aktivitas aplikasi yang sedang berjalan, dan menambahkan fitur keamanan aplikasi.

5. REFERENCES

- Aminudin, H., Gusti, I., Putra, L., & Prisman, E. (2021). Pengembangan Sistem Informasi Penilaian Portofolio Siswa (SIPPS) Berbasis Website untuk Mengetahui Tingkat Kompetensi Siswa di SMK Negeri 1 Driyorejo. *Jurnal Information Technology and Education*, 05(02), 584–591. Retrieved from <https://ejournal.unesa.ac.id/index.php/it-edu/article/view/39716>
- Davis, J., & Daniels, R. (2016). *Effective DevOps: building a culture of collaboration, affinity, and tooling at scale*. O'Reilly Media, Inc.
- Fauzi, A., & Harli, E. (2019). Rancang Bangun Sistem Informasi Akademik Smk Negeri 1 Depok Berbasis Android Dengan Pendekatan Rapid Application Development. *Jurnal Teknik Informatika*, 12(2), 129–136. <https://doi.org/10.15408/jti.v12i2.10939>
- GlobalState. (2022). Mobile Operating System Market Share Indonesia. Retrieved from <https://gs.statcounter.com/os-market-share/mobile/indonesia>
- Indra, E., & Rizky, A. (2020). Sistem Informasi Manajemen Kampus dengan Pengembangan Model Smart Campus:(Studi Kasus Di Universitas Prima Indonesia). *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, 3(2), 15–25. Retrieved from <http://ejournal.sisfokomtek.org/index.php/jikom/article/view/42>
- Nuraeni, F., Setiawan, R., Nurhakim, W., & Mubarok, M. S. (2022). Sistem Informasi Akademik Berbasis Mobile Apps Sebagai Media Informasi Akademik Online. *Jurnal Algoritma*, 18(2), 358–366. <https://doi.org/10.33364/ALGORITMA/V.18-2.951>
- Pratama, A. P., & Kamisutara, M. (2021). Pengembangan Sistem Informasi Akademik Berbasis Mobile Menggunakan Flutter Di Universitas Narotama Surabaya. *Jurnal Ilmiah NERO*, 6(2), 145. Retrieved from <http://repository.narotama.ac.id/id/eprint/1122>
- RG Hattari, M., Lutfi, S., & Khairan, A. (2018). Perancangan Aplikasi Android Sistem Informasi Akademik Universitas Khairun Ternate. *JIKO (Jurnal Informatika Dan Komputer)*, 2(2), 76–84. <https://doi.org/10.33387/jiko.v1i2.773>
- Riyadli, H., & Arliyana, A. (2019). Sistem Informasi Akademik Siswa Berbasis Web Mobile. *Jurnal Sains Komputer Dan Teknologi Informasi*, 2(1), 11–15. <https://doi.org/10.33084/JSAKTI.V2I1.1158>
- Setiawan, D., Suratno, T., & Lutfi, L. (2018). Analisis, Desain dan Rancang Bangun Sistem Informasi Akademik Mobile Berbasis Android. *Elkha*, 10(2), 73–77. <https://doi.org/10.26418/elkha.v10i2.28272>
- Tohirin, T., Utami, S. F., Widianto, S. R., & Mauludyansah, W. Al. (2020). Implementasi DevOps Pada Pengembangan Aplikasi e-Skrining Covid-19. *Jurnal Multinetics*, 6(1), 15–20. <https://doi.org/10.32722/multinetics.v6i1.2764>
- Yuanita, H., Wijayanto, B., & Cahyono, T. (2022). Frontend Development of Course Scheduling System Integrated SIA at Engineering Faculty University of Jenderal Sudirman. *Jurnal Teknik Informatika (JUTIF)*, 3(2), 305–319. <https://doi.org/10.20884/1.jutif.2022.3.2.227>
- Zalukhu, F., & Arinal, V. (2021). Implementasi Sistem Persediaan Barang Berbasis Web dengan Metode DevOps pada PT. Heinz ABC Indonesia. *Jurnal Sosial Dan Teknologi*, 1(7), 671–681. Retrieved from <https://sostech.greenvest.co.id/index.php/sostech/article/view/135>