Jurnal Minfo Polgan
Volume 12, Nomor 1, Agustus 2023 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v12i1.12870 p-ISSN : 2089-9424

**Terbit** : 01 Agustus 2023

# Pemodelan Sistem Penerimaan Anggota Baru dengan *Unified Modeling Language* (UML) (Studi Kasus: *Programmer Association of Battuta*)

[1]M. Rhifky Wayahdi, [2] Fahmi Ruziq
[1,2]Program Studi Sistem Informasi, Fakultas Teknologi, Universitas Battuta

[1]muhammadrhifkywayahdi@gmail.com, [2]fahmiruziq89@gmail.com

**ABSTRAK**

Sistem yang baik adalah sistem yang dirancang dengan proses analisis mendalam serta terdokumentasi dengan standar yang ditetapkan, serta memperhatikan siklus hidup pengembangan sistem. UML dapat membantu dalam proses pemodelan pengembangan sistem atau perangkat lunak. Dalam penelitian ini, pemodelan dengan UML dilakukan pada sistem penerimaan anggota baru di *Programmer Association of Battuta* yang bertujuan untuk memudahkan dalam pengembangan aplikasi dikemudian hari. Pemodelan sistem penerimaan anggota baru dimulai dengan tahap pendaftaran, ujian tahap 1, ujian tahap 2, sampai laporan akhir diterima atau tidaknya menjadi anggota baru. Penelitian ini hanya membahas alur penerimaan anggota baru dan tidak membahas alur sistem yang lainnya. Pengumpulan data yang digunakan dalam penelitian ini adalah dengan teknik observasi dan studi literatur. Penelitian ini menggunakan pemodelan sistem dengan UML yang mencakup diagram-diagram tertentu, yaitu *use case diagram* digunakan untuk memodelkan hubungan/interaksi antara pengguna sistem, *activity diagram* memperlihatkan alur atau aktivitas sistem, s*equence diagram* menjelaskan sistem secara lebih detail, dan *class diagram* menggambarkan hubungan antar *class* dari sistem yang dirancang. Setiap diagram dalam UML memiliki tugas dan fungsi masing-masing. Pemodelan sistem dengan UML memberikan beberapa keunggulan, baik dari segi perancangan dan pengembangan, ataupun pada tahap pemahaman dan komunikasi antara *developer* dan *client*. UML akan memungkinkan *developer* untuk melakukan analisis yang lebih mendalam dan perancangan yang lebih baik dan maksimal sebelum memulai merancang aplikasi.

**Kata Kunci**: Pemodelan Sistem, UML, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

## PENDAHULUAN

Untuk merancang sebuah sistem yang baik diharuskan memperhatikan siklus hidup pengembangan sistem (*system development life cycle*) atau lebih dikenal dengan sebutan SDLC. SDLC menjadi pendekatan yang sangat penting dalam pengembangan perangkat lunak (sistem) karena membantu untuk mengurangi risiko kesalahan dan memastikan bahwa perangkat lunak (sistem) yang dihasilkan sesuai dengan kebutuhan dan harapan pengguna (*user*). Dalam tahapan SDLC terdapat tahapan pemodelan (*modeling*), yang bertujuan untuk lebih memahami persyaratan (*requirement*) perangkat lunak (sistem) dan desain yang akan mencapai persyaratan tersebut (Pressman & Maxim, 2019). Pemodelan (*modeling*) perangkat lunak (sistem) dapat menggunakan bahasa pemodelan visual *Unified Modeling Language* (UML).

UML telah dirancang untuk berbagai pemodelan sistem perangkat lunak, sehingga disediakan konstruksi untuk berbagai sistem dan aktivitas (Bruegge & Dutoit, 2014). UML sejauh ini sudah menjadi bahasa pemodelan perangkat lunak paling sukses yang distandarisasi oleh *Object Management Group* (OMG) (Gotti & Mbarki, 2016). Sehingga UML sangat cocok untuk semua

Jurnal Minfo Polgan
Volume 12, Nomor 1, Agustus 2023 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v12i1.12870 p-ISSN : 2089-9424

fase siklus hidup perangkat lunak dari analisis kebutuhan (Jin & Liang, 2017). UML dapat membantu dengan mengumpulkan dan menganalisis persyaratan perangkat lunak dan menggabungkannya ke dalam desain program dengan cara independen berupa *technology* dan *methodology* (Castillo, Navajas, & Piattini, 2021). Namun, keberhasilan UML sebagai standar *de facto* untuk pemodelan sistem perangkat lunak tidak berarti menutup pintu eksplorasi atau eksperimen ilmiah dengan pemodelan di lapangan (Al-Fedaghi, UML Modeling to TM Modeling and Back, 2021).

Menurut Jin & Liang (2017) dalam penelitiannya tentang pemodelan manajemen administrasi kendaraan berbasis sistem, menjelaskan bahwa UML merupakan bahasa pemodelan visual yang kaya semantik yang mendukung prinsip pengembangan inkremental dan iteratif dari metode pengembangan berorientasi objek. UML membantu untuk membangun model sistem yang jelas dan intuitif, menyatukan komunikasi tim pengembangan, mengurangi kompleksitas pengembangan perangkat lunak, meningkatkan penggunaan kembali dan pemeliharaan sistem, sehingga dapat meningkatkan efisiensi pengembangan perangkat lunak (Jin & Liang, 2017).

Menurut Nugroho, et al. (2018) dalam penelitiannya tentang *mobile cloud learning* menggunakan UML, menjelaskan bahwa perancangan *mobile cloud learning* menggunakan UML sangat membantu dalam menghasilkan rancangan berbasis *object* yang akan membantu proses pembuatan program aplikasi (Nugroho, Sumardi, & Murdowo, 2018).

Menurut Liang & Jin (2020) dalam penelitiannya tentang pemodelan sistem penjualan komputer dengan UML, menjelaskan bahwa pemodelan sistem dengan UML merupakan metode yang efektif, karena tidak hanya mempertimbangkan hubungan antara sistem dan lingkungan, tetapi juga hubungan antara sistem dan pengguna. Serta pada saat yang sama, mempelajari secara mendalam hubungan antar komponen di dalam sistem (Liang & Jin, 2020).

Menjelajahi pemodelan dalam konteks UML penting untuk kemajuan dalam pemodelan konseptual dalam rekayasa perangkat lunak atau sistem (Al-Fedaghi, UML Modeling to TM Modeling and Back, 2021). Dari beberapa kesimpulan yang diperoleh dari penelitian-penelitian terkait dalam pemodelan sistem menggunakan UML, penulis akan mengimplementasikan pemodelan dengan UML untuk merancang sistem penerimaan anggota baru pada salah satu Unit Kegiatan Mahasiswa di Universitas Battuta, yaitu *Programmer Association of Battuta* (Pro.Asta). Penulis akan merepresentasikan alur sistem penerimaan anggota baru dengan beberapa diagram UML yang umum digunakan seperti *Use case diagram*, *Activity diagram*, *Sequence diagram*, dan *Class diagram*.

## TINJAUAN PUSTAKA

### *Use Case Diagram*

*Use case diagram* merupakan *tools* untuk mengembangkan sistem dengan menjelaskan hubungan antara *actor* yang terlibat dalam sistem (Nugroho, Sumardi, & Murdowo, 2018). Diagram ini biasanya digunakan untuk mengkomunikasikan fungsi *high-level* dari sistem dan ruang lingkup sistem yang menggambarkan beberapa ekspektasi (harapan) dari sistem oleh praktisi di luar sistem (Liang & Jin, 2020). Diagram ini berguna untuk orang di luar sistem yang biasanya digunakan untuk menentukan perilaku sistem dari sudut pandang pengguna (Al-Fedaghi, TMUML: A Singular TM Model with UML Use Cases and Classes, 2021).

Sebuah *use case* menjelaskan sebuah fungsi yang disediakan oleh sistem yang menghasilkan *output* yang terlihat untuk seorang *actor*. Sedangkan seorang *actor* menjelaskan setiap entitas yang berinteraksi dengan sistem. Identifikasi *actor* dan *use case* menghasilkan definisi batas sistem (*system boundary*), yaitu membedakan tugas/kegiatan yang diselesaikan oleh sistem dan tugas/kegiatan yang diselesaikan oleh lingkungannya. *Actor* berada di luar *system boundary*, sedangkan *use case* berada di dalam *system boundary* (Bruegge & Dutoit, 2014).

### *Activity Diagram*

*Activity diagram* merupakan salah satu yang paling banyak digunakan di antara semua diagram UML. Alur *activity diagram* dapat digunakan untuk mengontrol urutan tindakan yang dieksekusi atau untuk merepresentasikan komunikasi data. Biasanya diagram ini digunakan untuk desain tingkat tinggi seperti memodelkan proses bisnis, dan desain tingkat rendah seperti untuk

Jurnal Minfo Polgan
Volume 12, Nomor 1, Agustus 2023 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v12i1.12870 p-ISSN : 2089-9424

mendeskripsikan algoritma yang akan diimplementasikan (Lima, Tavares, & Nogueira, 2019).

*Activity diagram* menunjukkan adanya alur kerja atau aktivitas yang berjalan dari suatu sistem (Nugroho, Sumardi, & Murdowo, 2018). Diagram ini berfokus pada aliran kontrol (*control flow*) dari satu aktivitas ke aktivitas lainnya, berupa sebuah proses yang didorong oleh *internal processing* (Liang & Jin, 2020). Diagram aktivitas bersifat hierarkis, maksudnya adalah suatu aktivitas yang dibuat dari tindakan atau grafik sub-aktivitas dan aliran objek yang terkait (Bruegge & Dutoit, 2014).

***Sequence Diagram***

*Sequence diagram* adalah diagram UML paling umum kedua yang merepresentasikan bagaimana objek berinteraksi dan bertukar pesan dari waktu ke waktu (*over time*). *Sequence diagram* menggambarkan bagaimana peristiwa (aktivitas) dalam *use case* yang dipetakan ke dalam operasi kelas objek dalam *class diagram*. Penerimaan umum *sequence diagram* dapat dikaitkan dengan sifatnya yang relatif intuitif dan kemampuan untuk menggambarkan perilaku parsial (Al-Fedaghi, UML Sequence Diagram: An Alternative Model, 2021).

Pokok dari *sequence diagram* yaitu menunjukkan interaksi antar objek dan indikasi hubungan antar objek (Nugroho, Sumardi, & Murdowo, 2018). Sequence diagram mewakili interaksi antara bagian-bagian dari sistem. Tujuan utama dari *sequence diagram* adalah untuk mengungkapkan bagaimana objek berinteraksi satu sama lain untuk mengimplementasikan suatu fungsi yang dapat mewakili *object collaboration model* yang ada dalam pikiran perancang sistem untuk program masa depan saat *runtime* (Liang & Jin, 2020).

***Class Diagram***

*Class diagram* adalah *class* yang menggambarkan struktur dan penjelasan dari *class*, *package*, dan *object* serta hubungan satu sama lain seperti *containment*, *inheritance*, *associations*, dan lain-lain (Nugroho, Sumardi, & Murdowo, 2018). *Class* adalah abstraksi yang merepresentasikan struktur dan perilaku umum dari sekumpulan *object*. *Object* adalah turunan dari *class* yang dibuat (*created*), dimodifikasi (*modified*), dan dihancurkan (*destroyed*) selama eksekusi sistem. Sebuah *object* memiliki status yang menyertakan nilai *attribute*-nya dan hubungannya dengan *object* lain (Bruegge & Dutoit, 2014). *Class diagram* berfungsi sebagai cara untuk merencanakan dan berkomunikasi antar *developers* (Yang & Sahraoui, 2022).

*Class diagram* merepresentasikan struktur statis dari sistem yang mewakili *object class* dalam sistem dan *relationship* pada setiap *class* (Liang & Jin, 2020). Selain itu, *class diagram* digunakan untuk memodelkan tampilan struktur sistem yang mencakup kemampuan untuk membawa data dan menjalankan tindakan (*execute actions*), juga sebagai dasar untuk pengembangan sistem perangkat lunak lebih lanjut, misalnya *design phase* (Al-Fedaghi, TMUML: A Singular TM Model with UML Use Cases and Classes, 2021).

## METODE PENELITIAN

**Ruang Lingkup dan Batasan**

Penelitian ini membahas pemodelan sistem penerimaan anggota baru pada UKM (Unit Kegiatan Mahasiswa) *Programmer Association of Battuta* dengan menggunakan UML. Perancangan atau pemodelan alur sistem dimulai dari awal penerimaan yang mencakup pendaftaran dan ujian, sampai akhir sistem berupa hasil penilaian dan keputusan diterima atau tidaknya menjadi anggota baru *Programmer Association of Battuta*. Yang menjadi aktor dalam sistem ini adalah Mahasiswa, Koordinator UKM, dan Dekan Fakultas Teknologi.

Batas permasalahan dalam penelitian ini adalah peneliti hanya membahas alur penerimaan anggota baru dan tidak membahas alur sistem yang lainnya. Diagram-diagram UML yang digunakan dalam pemodelan yaitu *Use case*, *Activity*, *Sequence*, dan *Class* di mana masing-masing diagram akan menggambarkan alur sistem yang dirancang. Pemodelan dengan diagram UML menggunakan *software* Microsoft Visio versi terbaru.

**Teknik Pengumpulan Data**

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah dengan teknik

Jurnal Minfo Polgan
Volume 12, Nomor 1, Agustus 2023 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v12i1.12870 p-ISSN : 2089-9424

observasi dan studi literatur. Teknik observasi yang dilakukan adalah dengan mengamati langsung bagaimana alur sistem yang sedang berjalan sehingga memperoleh pemahaman secara visual, serta dapat memberikan hipotesis berupa kelemahan sistem yang harus diperbaiki. Studi literatur juga digunakan dengan mencari dan mengumpulkan bahan pustaka yang relevan dengan tema penelitian sehingga diperoleh pemahaman yang lebih mendalam mengenai konsep, metode, atau kerangka kerja yang baik atau relevan dalam pemodelan sistem.

**Konsep Pemodelan Sistem**

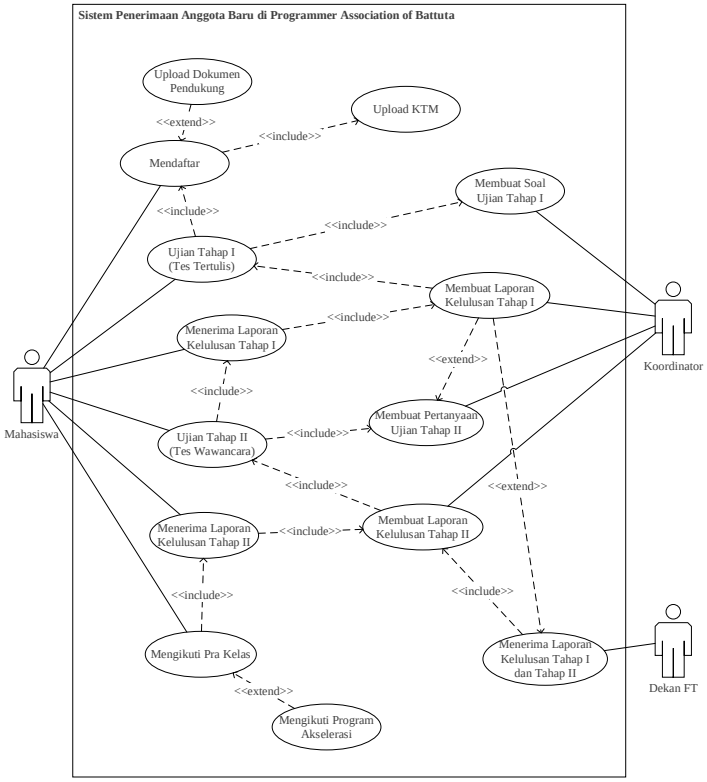
Penelitian ini menggunakan pemodelan sistem dengan UML yang mencakup diagram-diagram tertentu. *Use case diagram* akan merepresentasikan peran dari masing-masing *actor* yang mencakup Mahasiswa, Koordinator, dan Dekan di dalam sistem di mana peran atau kegiatan yang dilakukan oleh *actor* dinotasikan dengan simbol oval yang beraturan (*elips*) yang disebut dengan *use case*. *Activity diagram* akan merepresentasikan penjabaran dari *use case diagram*, di mana setiap *use case* seharusnya memiliki masing-masing *activity diagram*. *Activity diagram* akan digunakan untuk merinci aktivitas dari hubungan *actor* dengan *use case* di dalam *use case diagram*.

Selain *use case* dan *activity*, *sequence diagram* dan *class diagram* juga digunakan dalam pemodelan sistem pada penelitian ini. *Sequence diagram* hampir sama dengan *activity diagram*, namun *sequence diagram* lebih merepresentasikan interaksi antara *actor* dengan sistem, di mana dalam *sequence diagram* dapat dilihat bagaimana interaksi dengan sistem dinotasikan dengan objek-objek tertentu. Objek-objek dalam *sequence diagram* menggambarkan *actor*, *data control*, ataupun *data store*. Sedangkan *class diagram* akan merepresentasikan struktur dari sistem yang dimodelkan berupa kelas-kelas yang di dalamnya terdapat atribut/properti/variabel apa saja yang digunakan dan metode berupa fungsi (*function*) dan prosedur (*procedure*) apa saja yang dapat dilakukan pada kelas tersebut.

## HASIL DAN PEMBAHASAN

**Hasil**

a) Pemodelan sistem dengan *use case diagram*

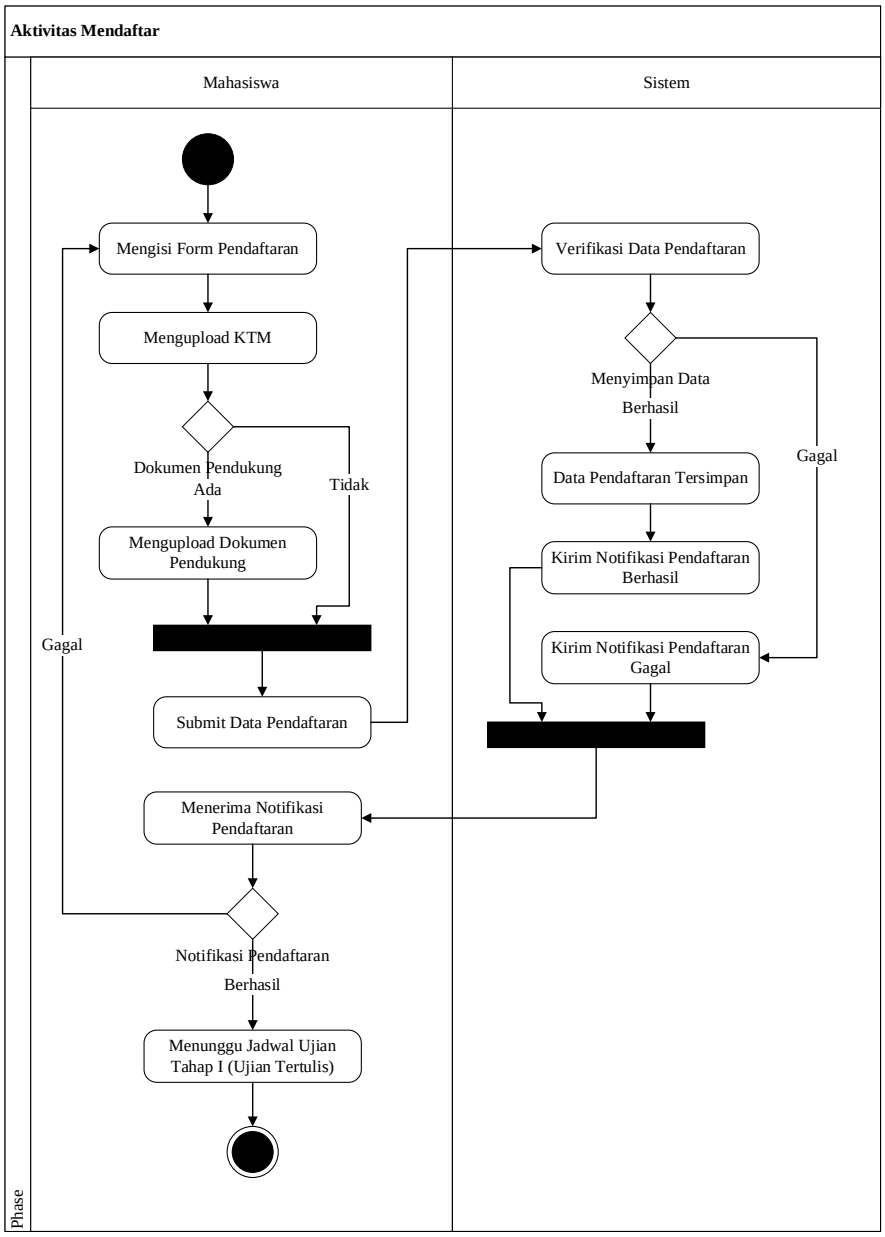



Gambar 1. Pemodelan Sistem dengan *Use Case Diagram*

Jurnal Minfo Polgan
Volume 12, Nomor 1, Agustus 2023 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v12i1.12870 p-ISSN : 2089-9424

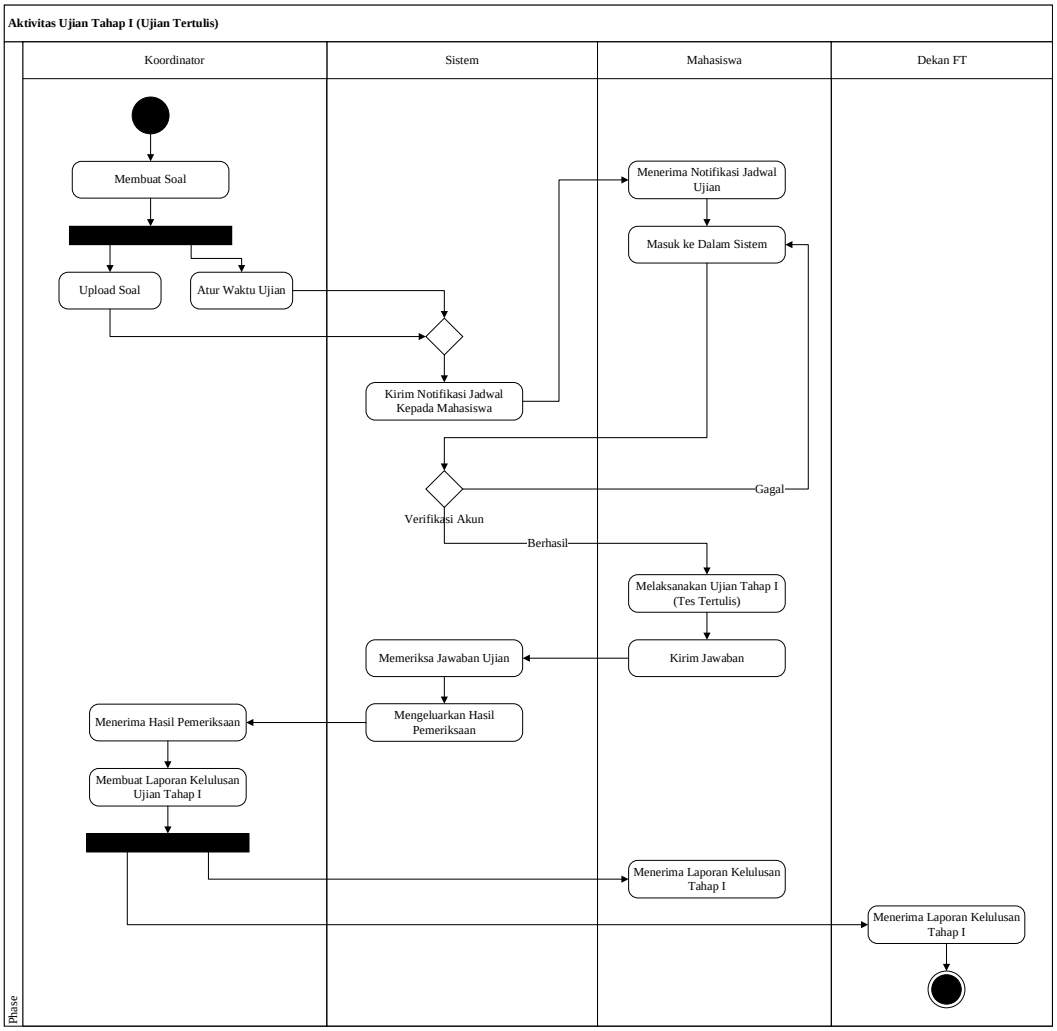
Pada Gambar 1 dapat dilihat hasil pemodelan sistem penerimaan anggota baru pada *Programmer Association of Battuta* menggunakan *use case diagram*. Pada *use case diagram* tersebut terdapat 3 *actor* yang terlibat yaitu Mahasiswa, Koordinator, dan Dekan FT di mana ketiga *actor* tersebut memiliki tugas dan aktivitas yang berbeda sesuai dengan otoritasnya masing-masing yang digambarkan dengan notasi *use case* yang saling terhubung.

b) Pemodelan sistem dengan *activity diagram*




Gambar 2. Pemodelan Sistem dengan *Activity Diagram* (Aktivitas Mendaftar)

Jurnal Minfo Polgan
Volume 12, Nomor 1, Agustus 2023 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v12i1.12870 p-ISSN : 2089-9424



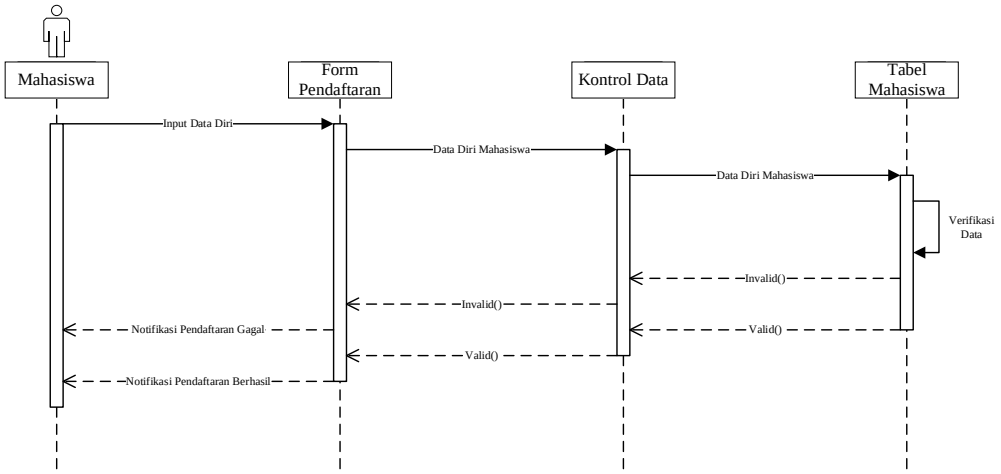

Gambar 3. Pemodelan Sistem dengan *Activity Diagram* (Aktivitas Ujian Tahap I)

Pada Gambar 2 dan Gambar 3 dapat dilihat hasil pemodelan sistem penerimaan anggota baru pada *Programmer Association of Battuta* menggunakan *activity diagram*. Gambar 2 merupakan aktivitas mendaftar, yaitu di mana Mahasiswa melakukan proses pendaftaran (registrasi) ke dalam sistem yang dimulai dari mengisi form pendaftaran, mengupload dokumen, sampai tahap akhir yaitu menunggu jadwal ujian tertulis tahap I (ujian tertulis).

Sedangkan Gambar 3 merupakan aktivitas ujian tahap I (ujian tertulis), di mana terdapat aktivitas antara Koordinator yang membuat soal dan menentukan waktu ujian, Mahasiswa yang melakukan ujian tertulis, Dekan FT yang menerima laporan, dan Sistem sebagai jembatan penghubung semua entitas. Untuk aktivitas ujian tahap II hampir sama atau memiliki perbedaan yang tidak terlalu signifikan dengan aktivitas ujian tahap I, di mana pada ujian tahap II (wawancara) dilakukan secara manual (tatap muka), sehingga Penulis tidak menggambarkan *activity diagram*-nya.

c) Pemodelan sistem dengan *sequence diagram*

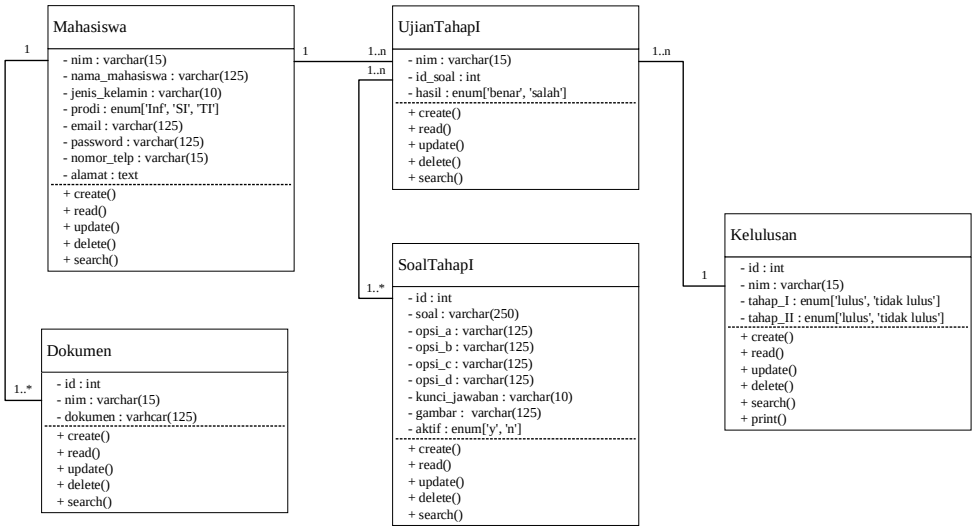
Jurnal Minfo Polgan
Volume 12, Nomor 1, Agustus 2023 e-ISSN : 2797-3298
DOI : https://doi.org/10.33395/jmp.v12i1.12870 p-ISSN : 2089-9424




Gambar 4. Pemodelan Sistem dengan *Sequence Diagram*

Pada Gambar 4 dapat dilihat hasil pemodelan sistem penerimaan anggota baru pada *Programmer Association of Battuta* menggunakan *sequence diagram*, khususnya untuk bagian pendaftaran. Penulis hanya memodelkan bagian pendaftaran karena hal tersebut sudah cukup menggambarkan bagaimana memodelkan sistem dengan menggunakan *sequence diagram*. Untuk proses lainnya sudah cukup digambarkan atau dimodelkan dengan diagram-diagram UML sebelumnya (*use case diagram* dan *activity diagram*).

d) Pemodelan sistem dengan *class diagram*




Gambar 5. Pemodelan Sistem dengan *Class Diagram*

Pada Gambar 5 dapat dilihat hasil pemodelan sistem penerimaan anggota baru pada *Programmer Association of Battuta* menggunakan *class diagram*. *Class diagram* dimodelkan secara sederhana agar memudahkan untuk pengembangan sistem di tahap selanjutnya.

**Pembahasan**

Pemodelan sistem dengan UML akan memudahkan pengembang sistem dalam menentukan langkah selanjutnya. *Use case diagram* digunakan untuk memodelkan hubungan/interaksi antara pengguna sistem, yang dalam hal ini adalah *actor* dengan ruang lingkup

sistem, sehingga akan terlihat jelas keterlibatan setiap entitas atau *actor* dengan sistem yang dirancang. *Activity diagram* akan memperlihatkan alur atau aktivitas sistem. *Sequence diagram* menjelaskan sistem secara lebih detail. Sedangkan *class diagram* menggambarkan hubungan antar *class* dari sistem yang dirancang. Semua diagram UML yang digambarkan memiliki peran dan fungsi masing-masing dalam memodelkan sistem penerimaan anggota baru pada *Programmer Association of Battuta*.

## KESIMPULAN

*Unified Modeling Language* (UML) dapat memodelkan sistem yang dirancang dengan notasi standar, sehingga tahapan setelah pemodelan dapat diproses dengan lebih mudah dan cepat. Pemodelan sistem penerimaan anggota baru pada *Programmer Association of Battuta* menggunakan UML memberikan beberapa keuntungan yang cukup banyak dalam pengembangan sistem, baik dari segi perancangan dan pengembangan, ataupun pada tahap pemahaman dan komunikasi antara *developer* dan *client.* Pemodelan dengan UML memungkinkan para *developer* untuk melakukan analisis yang lebih mendalam dan perancangan yang lebih baik dan maksimal sebelum memulai merancang aplikasi.

## REFERENSI

Al-Fedaghi, S. (2021). TMUML: A Singular TM Model with UML Use Cases and Classes. *International Journal of Computer Science and Network Security (IJCSNS), XXI*(6), 127-136.

Al-Fedaghi, S. (2021). UML Modeling to TM Modeling and Back. *International Journal of Computer Science and Network Security (IJCSNS), XXI*(1), 1-13.

Al-Fedaghi, S. (2021). UML Sequence Diagram: An Alternative Model. *International Journal of Advanced Computer Science and Applications (IJACSA), XII*(5), 635-645.

Bruegge, B., & Dutoit, A. H. (2014). *Object-Oriented Software Engineering using UML, Patterns, and Java.* United States: Pearson Education Limited.

Castillo, R. P., Navajas, L. J., & Piattini, M. (2021). Modelling Quantum Circuits with UML. *IEEE/ACM 2nd International Workshop on Quantum Software Engineering (Q-SE)*, 7-12. doi:10.1109/Q-SE52541.2021.00009

Gotti, S., & Mbarki, S. (2016). UML Executable: A Comparative Study of UML Compilers and Interpreters. *International Conference on Information Technology for Organizations Development (IT4OD)* (pp. 1-5). Fez, Morocco: IEEE. doi:10.1109/IT4OD.2016.7479251

Jin, L., & Liang, J. (2017). Modeling of Vehicle Administrative Management System Based on Unified Modeling Language. *IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)* (pp. 50-54). Chongqing, China: IEEE. doi:10.1109/ITOEC.2017.8122372

Liang, J., & Jin, L. (2020). Multi-perspective modeling of computer sales system Based on Unified Modeling Language. *IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)* (pp. 109-113). Chongqing, China: IEEE. doi:10.1109/ITOEC49072.2020.9141934

Lima, L., Tavares, A., & Nogueira, S. C. (2019, October 31). A Framework for Verifying Deadlock and Nondeterminism in UML Activity Diagrams based on CSP. *Brazilian Symposium on Formal Methods (SBMF)*, 1-29.

Nugroho, K., Sumardi, & Murdowo, S. (2018). Mobile Cloud Learning Using UML and Feature Driven Development On Higher Education. *International Conference on Computer Science and Engineering Technology (ICCSET)* (pp. 627-634). Kudus: IOP Publishing. doi:10.4108/eai.24-10-2018.2280537

Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (9th ed.). New York: McGraw-Hill Education.

Yang, S., & Sahraoui, H. (2022). Towards Automatically Extracting UML Class Diagrams from Natural Language Specifications. *International Conference on Model Driven Engineering Languages and Systems*, 1-8. doi:10.1145/3550356.3561592