

## DESAIN SISTEM SMART CAMPUS UNTUK MONITORING AKTIVITAS MAHASISWA BERBASIS SMS GATEWAY MENGGUNAKAN UNIFIED MODELING LANGUAGE

Rezania Agramanisti Azdy

Teknik Informatika STMIK PalComTech

Jl. Basuki Rahmat No. 05, Palembang 30129, Indonesia

e-mail: rezania.azdy@gmail.com

**Abstrak** – Pada kegiatan pengembangan perangkat lunak, metodologi menjabarkan urutan tahapan yang jelas dan apa saja yang harus dilakukan dalam mengembangkan sebuah perangkat lunak yang sesuai dengan kebutuhan penggunaanya. Tahapan desain merupakan tahapan yang paling penting dalam proses pengembangan perangkat lunak karena pada tahap inilah didefinisikan bagaimana cara membangun perangkat lunak yang dapat memenuhi kebutuhan user. Salah satu teknik yang dapat digunakan untuk menggambarkan desain perangkat lunak adalah Unified Modeling Language (UML).

UML merupakan pemodelan desain yang umum digunakan untuk pembangunan perangkat lunak dengan pendekatan berorientasi objek. Makalah ini membahas penggunaan diagram pada UML untuk memodelkan sistem *smart campus* pada tahapan desain. *Monitoring* dilakukan dalam mengawasi aktivitas mahasiswa berupa daftar kehadiran di setiap perkuliahan dan proses pembayaran yang dilakukan oleh mahasiswa. Pembahasan pada makalah ini menjabarkan 3 buah diagram yang digunakan untuk pemodelan pada tahapan desain, yaitu *class diagram*, *sequence diagram*, dan *activity diagram*. Pada *class diagram* ditemukan 5 komponen yang dapat dijadikan landasan untuk tahapan pengkodean, *sequence* dan *activity diagram* dapat dijadikan gambaran rangkaian proses dan aktivitas yang terjadi dalam sistem *smart campus* yang akan dibangun.

**Kata kunci** – sistem *smart campus*, *Unified Modeling Language*, desain berorientasi objek, *sms gateway*

### I. PENDAHULUAN

Dalam kegiatan pengembangan perangkat lunak, terdapat metodologi yang dapat digunakan sebagai acuan dalam membangun perangkat lunak. Metodologi menjabarkan urutan tahapan yang jelas dan apa saja yang harus dilakukan dalam mengembangkan sebuah perangkat lunak yang sesuai dengan kebutuhan penggunaanya. Meskipun setiap metodologi memiliki tahapan dan urutannya yang berbeda-beda, namun terdapat tahapan yang hampir sama dimiliki oleh setiap metodologi, yaitu analisis kebutuhan, desain, pengkodean, dan pengujian [1]. Analisis kebutuhan digunakan untuk memperoleh apa yang diharapkan *user* dari perangkat lunak yang dibuat. Tahapan desain merupakan jembatan antara tahapan analisis kebutuhan dengan pembuatan perangkat lunak menggunakan aktivitas pengkodean. Pada tahapan ini, *developer*

merancang perangkat lunak yang akan dibuat dengan menerjemahkan kebutuhan *user* kedalam bentuk fungsionalitas yang harus dimiliki oleh perangkat lunak. Desain menentukan bagaimana perangkat lunak akan berfungsi dengan menggambarkan aliran proses yang terjadi, data yang diolah di dalam sistem, bentuk antarmuka perangkat lunak, hingga cara penulisan kode program. Tahapan desain merupakan tahapan yang paling penting dalam proses pengembangan perangkat lunak karena pada tahap inilah didefinisikan bagaimana cara membangun perangkat lunak yang dapat memenuhi kebutuhan *user* sehingga dapat mengurangi resiko baik dari segi biaya, tenaga, maupun waktu pengerjaan perangkat lunak.

Beberapa teknik dapat digunakan untuk menggambarkan desain perangkat lunak. Beberapa diantaranya adalah *flowchart*, *data flow diagram* (DFD), dan *unified modeling language* (UML). [2] mendefinisikan *flowchart* sebagai bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah dan urutan-urutan dari suatu program komputer. Kelemahan *flowchart* terletak pada semakin rumitnya diagram untuk dipahami seiring dengan semakin besarnya sistem atau perangkat lunak yang dibangun. Sama seperti *flowchart*, DFD juga digunakan untuk memodelkan logika proses sebuah sistem. Andi Kristanto dalam [3] menyatakan pemodelan pada DFD tidak hanya pada alur prosesnya saja tapi juga menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang disimpan, serta proses yang dikenakan pada data tersebut.

Kedua teknik desain diatas umum digunakan untuk memodelkan perangkat lunak yang akan dirancang dengan pengkodean menggunakan pemrograman yang terstruktur, dimana prosedur logis perangkat lunak hanya dilihat sebagai rangkaian proses berikut: input data, proses atau olah data, kemudian menghasilkan output data. Akan tetapi, pengkodean menggunakan pemrograman terstruktur hanya sesuai digunakan jika perangkat lunak yang dibangun berskala kecil. Jika perangkat lunak lebih kompleks maka pendekatannya pemrograman yang berorientasi terhadap objek lebih tepat untuk digunakan. Dimana dengan pendekatan tersebut, logika dari proses tidak hanya fokus terhadap bagaimana mendefinisikan sebuah data melainkan memandang sebuah perangkat lunak dari objek yang ada dan bagaimana objek-objek tersebut terhubung satu dengan lainnya.

UML adalah sebuah notasi grafis yang secara *de facto* telah menjadi standar untuk memodelkan sebuah sistem dengan pendekatan berbasis objek. UML menggunakan diagram untuk mendokumentasikan dekomposisi sebuah sistem berdasarkan objeknya dan memperlihatkan interaksi dan dinamika dari objek-objek tersebut [4]. UML terdiri dari 9 jenis diagram yang dapat digunakan untuk memodelkan perangkat lunak baik dalam tahapan analisis, desain, pengkodean, maupun pengujian.

Pada makalah ini akan membahas penggunaan diagram pada UML untuk memodelkan sistem *smart campus* pada tahapan desain. Diagram yang digunakan yaitu *class diagram*, *interaction diagram*, dan *state diagram*. Bagian 2 mengutarkan beberapa penelitian sejenis yang telah dilakukan. Bagian 3 memberikan penjelasan lebih lengkap tentang perangkat lunak (sistem) yang akan dimodelkan. Bagian 4 merupakan pembahasan tentang pemodelan sistem menggunakan diagram yang telah disebutkan diatas, dan Bagian 5 dan 6 berisi tentang kesimpulan dan saran.

## II. KAJIAN LITERATUR

Terdapat beberapa penelitian sebelumnya yang telah menggunakan UML untuk memodelkan sistem yang akan dibangun.

Penelitian pertama menggunakan UML untuk memodelkan sistem pengolahan parkir pada sebuah Perseroan Terbatas (PT) [5]. Model yang digambarkan terdiri dari 2 jenis, yaitu model bisnis yang menggambarkan proses bisnis perparkiran yang terjadi dan model rekayasa perangkat lunak untuk menggambarkan model sistem perangkat lunak pengolahan lahan parkir. Model bisnis dikelompokkan menjadi 2, yaitu business usecase model yang menjelaskan hubungan antara aktor yang melaksanakan bisnis dengan proses bisnisnya dan setiap aktor dengan usecase-nya digambarkan lebih jelas lagi menggunakan activity diagram. Model yang kedua adalah business object model yang menjelaskan objek-objek yang mendukung proses bisnis, yaitu struktur dan unit organisasi yang digambarkan dengan class diagram. Pada model rekayasa perangkat lunak menggambarkan interaksi user dengan menggunakan usecase diagram, analisa dari perangkat lunak tentang realisasi perilaku pada usecase menggunakan class diagram, sequence diagram, dan collaboration diagram. Dan model lainnya berupa model database, desain antarmuka, model fisik perangkat lunak, model perangkat kerasnya. Hasil pengujian menunjukkan bahwa pemodelan menggunakan UML menghasilkan gambaran yang jelas dengan memberikan kemudahan dalam analisa, desain, dan implementasi, serta dapat memenuhi keserasian dalam proses pengkodean.

Penelitian berikutnya adalah memodelkan protokol jaringan dengan menggunakan UML, yaitu memodelkan *Voice over Internet Protocol* yang akan dibangun pada sebuah PT untuk dievaluasi oleh pihak manajemen sebelum diimplementasikan [6]. Diagram yang digunakan adalah *usecase diagram*, *activity diagram*, *interaction diagram*, *class diagram*, *component diagram*, dan *deployment diagram*.

*Usecase diagram* digunakan untuk menggambarkan perilaku sistem dari sudut pandang pengguna dengan scenario yang hamper sama dengan sistem komunikasi telepon. *Activity diagram* menggambarkan proses yang terjadi jika *user* menghubungi *user* lain pada cabang yang berbeda. Urutan proses yang terjadi mulai dari *user* mengangkat *handset* untuk memulai komunikasi hingga *user* memutuskan hubungan digambarkan menggunakan *interaction diagram*. Hasil dari penelitian adalah pemodelan menggunakan UML yang semakin rinci akan mendekatkan model pada sistem yang sesungguhnya dan menggambarkan sistem agar dapat dipahami dan dikoreksi sebelum akhirnya diimplementasikan secara langsung.

Dari tinjauan pustaka diatas disimpulkan bahwa UML dapat memodelkan sebuah perangkat lunak atau sistem dengan lebih rinci dan langsung dapat disesuaikan dengan pengkodean yang akan dilakukan untuk membuat perangkat lunak tersebut.

## III. SISTEM SMART CAMPUS MONITORING AKTIVITAS MAHASISWA BERBASIS SMS GATEWAY

Perkembangan teknologi informasi dan komunikasi dalam segala bidang, khususnya di dunia pendidikan tinggi, telah menjadikan penyebaran informasi dalam bidang akademik semakin mudah untuk dilakukan. Dengan adanya penerapan teknologi informasi dan komunikasi yang baik diharapkan dapat meningkatkan kualitas pelayanan terhadap mahasiswa dan masyarakat sehingga informasi yang dulunya hanya dapat diperoleh dengan mendatangi pihak kampus secara langsung, kini dapat mudah diperoleh dengan mengakses segala *device* yang telah terhubung ke Internet.

Akan tetapi tidak seluruh daerah baik di perkotaan maupun di desa telah terjangkau Internet dengan baik. Terlebih lagi minimnya pengetahuan orang tua atau wali mahasiswa akan kegunaan Internet yang tidak lebih baik dibandingkan kegunaan *handphone* yang telah umum digunakan sebagai media komunikasi. Maka dari itu penelitian ini memfokuskan tentang penyebaran informasi yang dapat diakses melalui *short message service* (SMS).

Kegiatan pengawasan terhadap peserta didik pun menjadi prioritas utama khususnya dalam hal dimana orang tua atau wali menaruh kepercayaan sepenuhnya terhadap mahasiswa untuk melakukan kegiatan akademik. Adapun kegiatan akademik yang menjadi fokus *monitoring* adalah:

### 1. Kehadiran mahasiswa pada setiap perkuliahan.

Salah satu indikator keberhasilan proses belajar mengajar pada sebuah perkuliahan bisa dilihat dari keaktifan mahasiswa dalam menghadiri perkuliahan.. Pengawasan dilakukan untuk mengendalikan tingkat kehadiran mahasiswa pada setiap mata kuliah yang diambilnya. Sehingga pemantauan dapat dilakukan tidak hanya oleh pihak kampus namun juga oleh orang tua atau wali mahasiswa. Hal ini dilakukan untuk mengantisipasi adanya kemungkinan

ketidakhadiran mahasiswa tanpa diketahui oleh orangtua maupun wali mahasiswa tersebut dan orangtua atau wali dapat mendapat informasi bahwa mahasiswa yang bersangkutan telah menghadiri perkuliahan atau tidak.

Terdapat 2 cara untuk mengimplementasikan proses *monitoring* ini dengan sistem yang sedang berjalan. Pada beberapa perguruan tinggi yang telah memberlakukan absensi *online* pada setiap perkuliahan, maka *monitoring* dapat diintegrasikan pada sistem absensi yang telah ada dan secara langsung keterangan yang dimasukkan akan diteruskan ke nomor *handphone* melalui SMS.

Absensi Online Mahasiswa			
Nama Materi	Algoritma Dan Struktur Data (KK14028)		
Grup	SIP2A		
Ruangan	4.2		
Hari/Waktu	Senin (08:00 - 11:20)		
Jumlah Mahasiswa	32 Orang		
Absensi Pertemuan	5		
No	NPM	Nama Mahasiswa	Status Hadir
1	011150045	Angga Pratama	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir
2	011150016	Ayu Meylia Tirta	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir
3	021150083	Bayu Aji	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir
4	021150094	Heykal Teranova	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir
5	021150019	Ilin Putriani Msy	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir
6	011150066	M. Yoan Gilananda	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir
7	011150018	Muhammad Ridho	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir
8	021150082	Muhammad Syaiful Aulia	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir
9	021150117	Muntas Agustin	<input checked="" type="radio"/> Hadir <input type="radio"/> Tidak Hadir

Gambar 1. Absensi Online pada STMIK PalComTech

Gambar 1 memperlihatkan absensi *online* mahasiswa pada sebuah Perguruan Tinggi di Palembang. Absensi diakses oleh dosen di setiap pertemuan mata kuliah sehingga daftar kehadiran selalu *update* untuk mata kuliah yang telah berlangsung. Sistem *monitoring* yang terintegrasi pada absensi *online* ini akan secara langsung meneruskan informasi kehadiran mahasiswa melalui SMS ke nomor *handphone* yang telah didaftarkan.

Pada perguruan tinggi yang masih memberlakukan absensi manual, yaitu absensi dilakukan dengan cara mahasiswa menandatangani secara langsung daftar hadir perkuliahan, maka sistem *monitoring* dapat diintegrasikan kepada sebuah aplikasi yang dapat digunakan untuk memasukkan data keterangan hadir mahasiswa dan di-*input* oleh seorang administrator.

## 2. Transaksi pembayaran biaya kuliah.

Pemantauan terhadap kegiatan pembayaran biaya kuliah oleh orang tua perlu dilakukan untuk mengantisipasi terjadinya penyelewengan dana yang telah diberikan. Selain dengan menunjukkan kuitansi pembayaran secara langsung kepada orang tua atau wali, informasi mengenai pembayaran juga dapat dilakukan dengan mengirimkan SMS kepada orang tua atau wali bahwa mahasiswa yang bersangkutan telah melakukan proses pembayaran biaya kuliah.

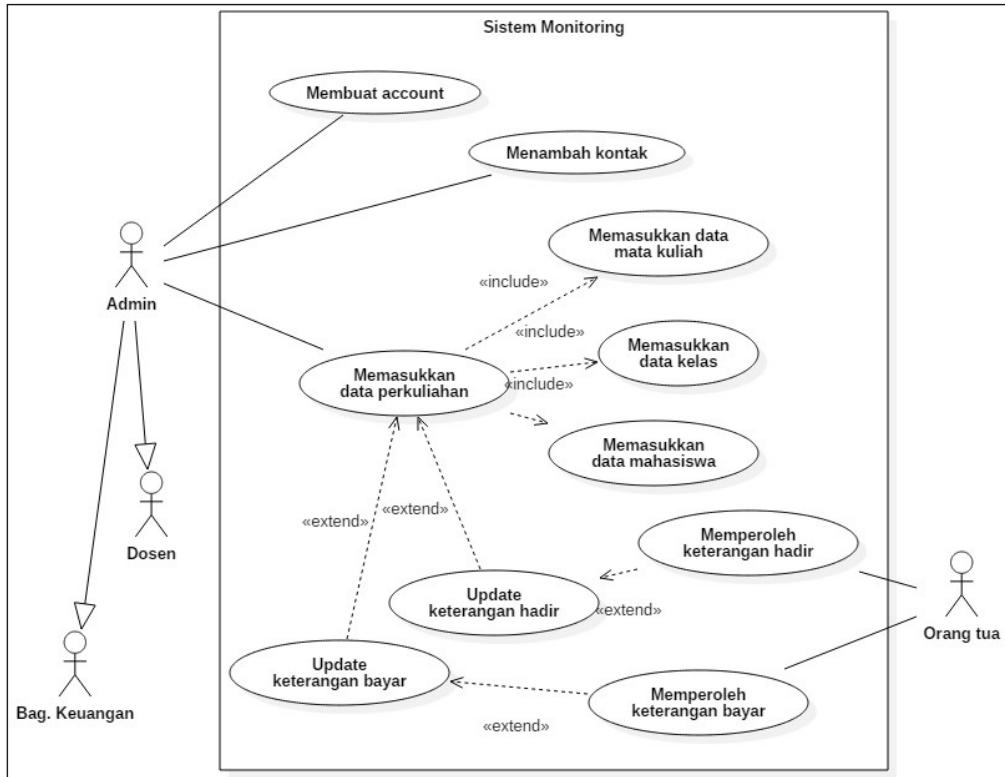
Proses pembayaran biaya kuliah kini dipermudah dengan adanya sistem *host-to-host* antara perguruan tinggi dengan Bank mitra yang telah bekerjasama. Sistem ini menghubungkan dua *server* yang dimiliki oleh Bank dengan *server* yang dimiliki oleh perguruan tinggi. Koneksi terjadi secara *realtime* dan setiap ada perubahan data pada salah satu *server*, *update* data secara otomatis akan terjadi pada *server* yang lainnya.

Setelah bagian administrasi/keuangan memverifikasi pembayaran yang dilakukan oleh mahasiswa, sistem akan secara otomatis menyampaikan informasi keterangan pembayaran tersebut melalui SMS ke nomor *handphone* yang telah terdaftar.

Pada makalah ini, sistem monitoring aktivitas kegiatan mahasiswa yang akan dibuat adalah sebuah sistem mandiri yang tidak terhubung dengan sistem yang telah berjalan. Oleh karena itu diperlukan seorang administrator yang akan memasukkan data awal berupa data mata kuliah, data kelas, data ruang, dan data mahasiswa.

## IV. DESAIN SISTEM SMART CAMPUS YANG DIUSULKAN

Untuk melakukan perancangan desain terhadap sistem yang akan dibuat, perlu dilakukan analisa kebutuhan sistem terlebih dahulu agar desain yang dihasilkan dapat memenuhi spesifikasi perangkat lunak yang diharapkan. Salah satu diagram UML yang dapat digunakan untuk memodelkan apa yang bisa dilakukan oleh perangkat lunak adalah usecase diagram.



Gambar 2. Usecase Diagram Sistem Smart Campus

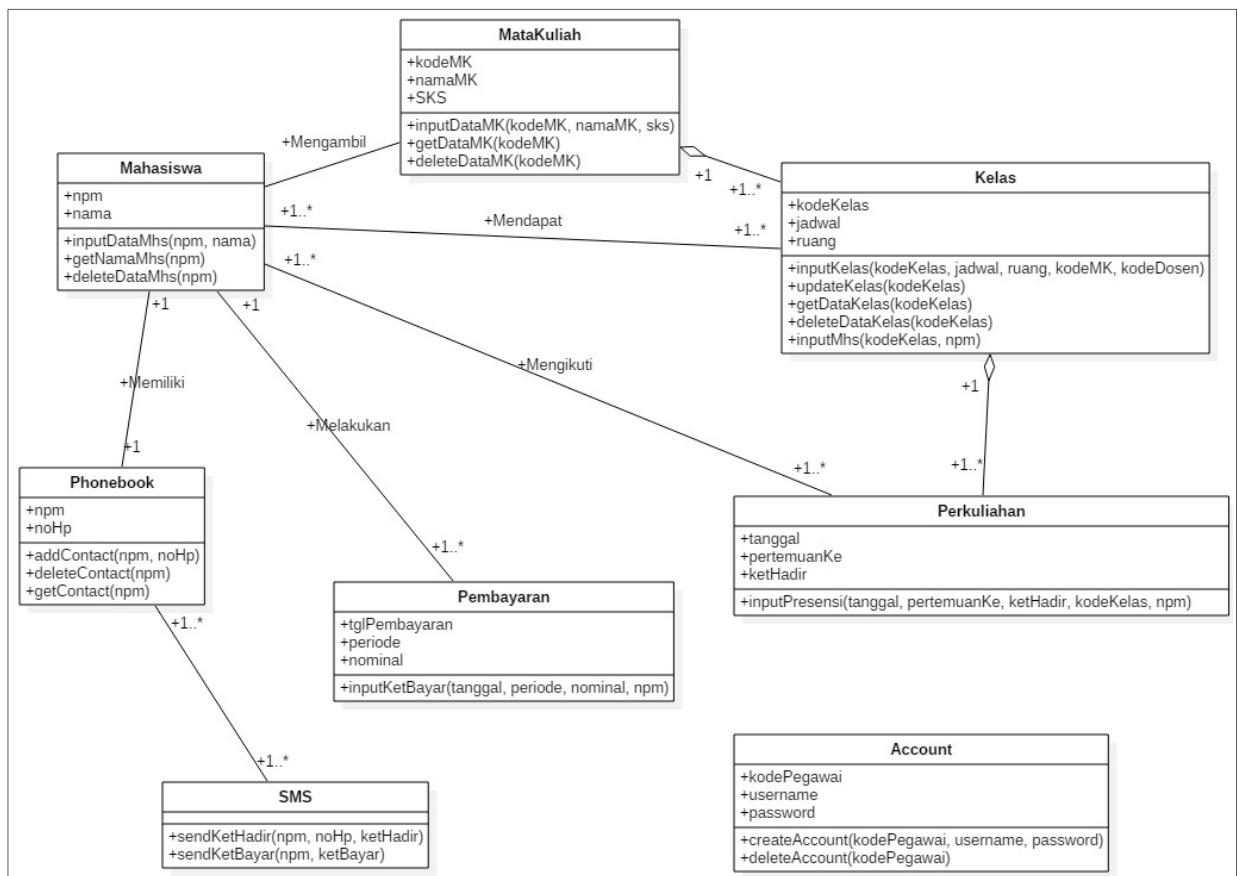
#### A. Usecase Diagram

*Usecase diagram* merupakan salah satu cara yang efektif untuk memodelkan kebutuhan sistem yang bersifat fungsional. Dengan kata lain, *usecase diagram* dapat digunakan untuk menjelaskan bagaimana sistem bekerja dengan menggambarkan apa saja yang dapat dilakukan sistem. [7] memaparkan 3 komponen penting pada *usecase diagram*, yaitu *aktor*, *use cases*, dan *subject*. *Subject* merupakan sistem dimana *use case* akan diterapkan. *Aktor* merupakan pemodelan dari entitas yang berada di luar sistem, sehingga seluruh *user* maupun sistem lain yang berinteraksi dengan *subject* adalah *aktor*. Dan *use case* menentukan perilaku dari sistem yang didefinisikan sesuai kebutuhan dari *aktor*.

Gambar 2 merupakan *usecase diagram* yang memperlihatkan *aktor* yang berinteraksi dengan sistem. Interaksi *aktor* terhadap sistem dapat berbeda dan berhubungan satu sama lain. Seorang *administrator* dapat memasukkan data perkuliahan dengan memasukkan data mata kuliah, data kelas, dan data mahasiswa sebelumnya ke dalam sistem. Seorang dosen atau *administrator* dapat melakukan *update* berupa data kehadiran mahasiswa pada setiap perkuliahan, dan bagian administrasi/keuangan dapat melakukan *update* data pembayaran pada sistem. *Use case update* baik untuk keterangan hadir maupun keterangan bayar akan memicu *use case* yang lainnya yaitu milik orang tua atau wali untuk dapat memperoleh informasi berupa keterangan hadir atau keterangan bayar.

#### B. Class Diagram

Selain dapat digunakan untuk memodelkan analisa kebutuhan pada tahapan analisis, *class diagram* juga dapat digunakan untuk pemodelan sistem pada tahapan desain. Hal ini dikarenakan pada pemrograman menggunakan pendekatan berorientasi objek, fokus utama terletak pada objek-objek yang saling berinteraksi satu sama lain. Dan sebuah *class* dapat disebut sebagai sebuah *template* yang dapat digunakan untuk membuat objek. Sehingga pemodelan desain menggunakan *class diagram* lebih memudahkan translasi model dalam bentuk desain ke tahapan pengkodean yang menggunakan bahasa pemrograman berorientasi objek. *Class diagram* merepresentasikan aspek statis dari sistem, yaitu struktur dari sistem yang akan dibuat.



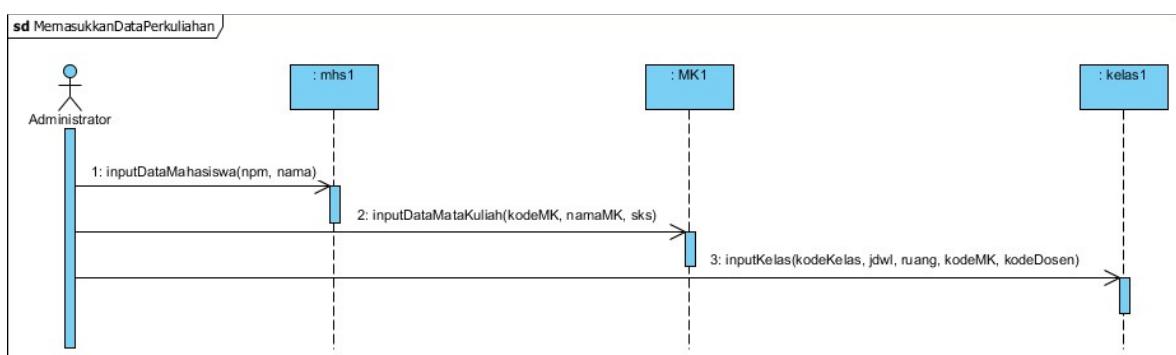
Gambar 3 Class Diagram Sistem Smart Campus

*Class diagram* pemodelan desain sistem monitoring diperlihatkan pada Gambar 3. Sistem memiliki 7 kelas yang saling terhubung, yaitu kelas mahasiswa, kelas mata kuliah, kelas kelas untuk keterangan jadwal dan ruang sebuah mata kuliah, kelas perkuliahan untuk *update* daftar kehadiran mahasiswa di setiap kelas mahasiswa tersebut terdaftar, kelas dosen, dan kelas pembayaran untuk keterangan pembayaran yang dilakukan mahasiswa. Kelas phonebook digunakan untuk menyimpan kontak para orang tua atau wali murid yang telah terdaftar, dan kelas SMS digunakan untuk mengirimkan keterangan hadir atau bayar melalui SMS gateway.

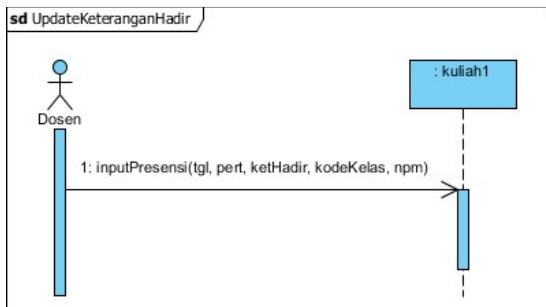
### C. Sequence Diagram

*Sequence diagram* mendeskripsikan interaksi antar komponen pada sistem dalam hal pertukaran pesan pada setiap waktu. Pemodelan menggunakan *sequence diagram* dapat dikelompokkan untuk setiap *usecase* yang telah didefinisikan sebelumnya.

Gambar 4 memperlihatkan *sequence diagram* untuk *usecase* yang dilakukan oleh aktor *administrator*, yaitu **memasukkan data perkuliahan**. Untuk memasukkan data perkuliahan, diperlukan interaksi antar 3 kelas, yaitu kelas mahasiswa yang mengambil mata kuliah, serta kelas yang sesuai dengan mata kuliah yang diambil.

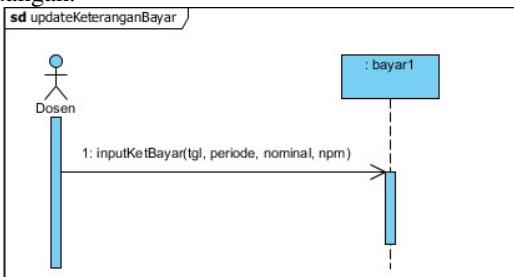


Gambar 4 Sequence diagram untuk usecase Memasukkan Data Perkuliahan

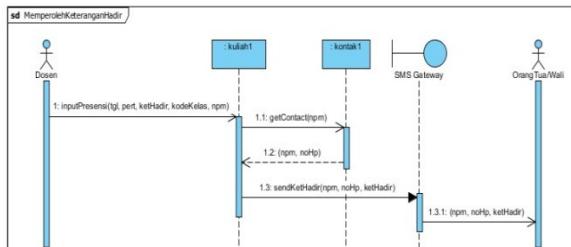


Gambar 5 Sequence diagram untuk usecase Update Keterangan Hadir

Gambar 5 memperlihatkan *Sequence diagram* untuk *usecase Update Keterangan Hadir* yang dilakukan oleh dosen, serta Gambar 6 merupakan *Sequence diagram* untuk *usecase Update Keterangan Bayar* yang dilakukan oleh Bagian administrator atau keuangan.

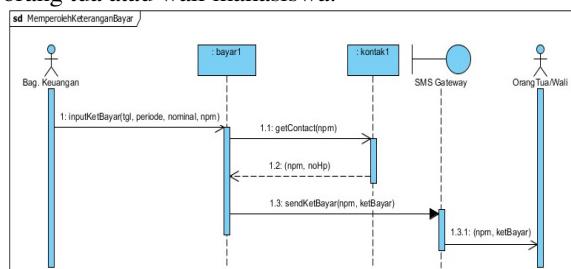


Gambar 6 Sequence diagram untuk usecase Update Keterangan Bayar



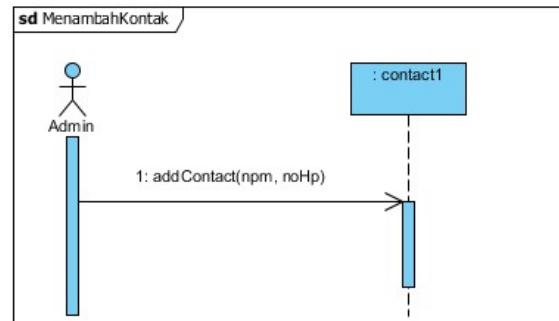
Gambar 7 Sequence diagram untuk usecase Memperoleh Keterangan Hadir

Gambar 7 merupakan *Sequence diagram* untuk *usecase Memperoleh Keterangan Hadir*, dimana dosen akan memanggil objek dari kelas Perkuliahan, yang kemudian akan memanggil *method* *sendNotif* dengan parameter berupa tanggal, pertemuan, dan keterangan hadir mahasiswa. Kemudian melalui SMS gateway akan mengirimkan info kehadiran tersebut ke orang tua atau wali mahasiswa.



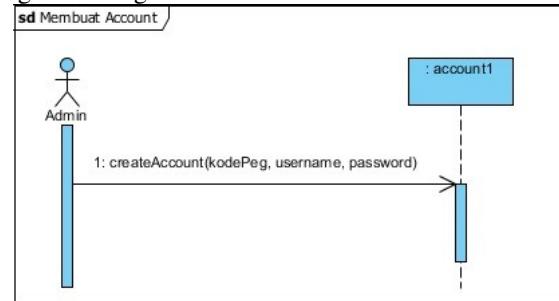
Gambar 8 Sequence diagram untuk usecase Memperoleh Keterangan Bayar

Gambar 8 merupakan *Sequence diagram* untuk *usecase Memperoleh Keterangan Bayar*, dimana bagian keuangan/administrasi akan memanggil objek dari kelas Pembayaran, dan kemudian akan memanggil *method* *sendNotif* dengan parameter berupa tanggal, periode, dan keterangan bayar mahasiswa. Kemudian melalui SMS gateway akan mengirimkan info pembayaran tersebut ke orang tua atau wali mahasiswa.



Gambar 9 Sequence diagram untuk usecase Menambah Kontak

Gambar 9 memperlihatkan *sequence diagram* untuk mendaftarkan kontak orang tua atau wali ke dalam *phonebook* dan Gambar 10 memperlihatkan *sequence diagram* memperlihatkan aliran proses untuk membuat akun baik untuk dosen maupun untuk bagian keuangan/administrasi.

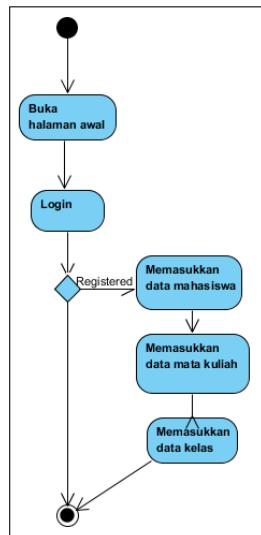


Gambar 10 Sequence diagram untuk usecase Membuat Account

#### D. Activity Diagram

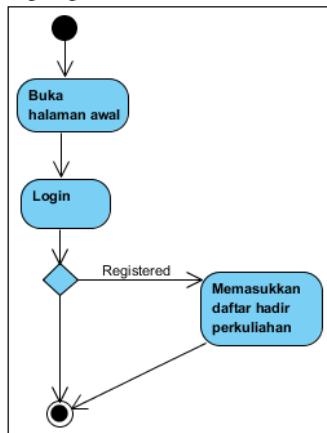
Jika *class diagram* digunakan untuk memodelkan aspek statis dari sistem, maka *activity diagram* dapat digunakan untuk memodelkan aspek dinamisnya, yaitu perilaku dari sistem. *Activity diagram* merupakan serangkaian aktivitas yang menggambarkan aliran kontrol pada sebuah sistem.

Sama seperti pada *sequence diagram*, pemodelan menggunakan *activity diagram* juga dikelompokkan untuk setiap *usecase* yang telah dijabarkan sebelumnya.



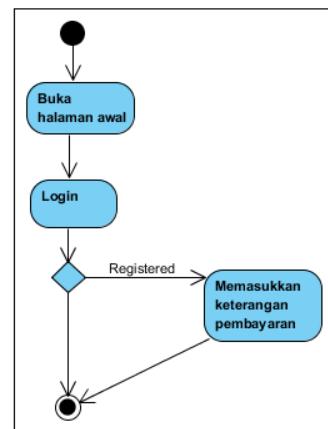
Gambar 11 Activity diagram untuk usecase Memasukkan Data Perkuliahan

Gambar 11 merupakan *activity diagram* untuk *usecase* memasukkan data perkuliahan. Setelah membuka halaman awal, *administrator* harus melakukan *login* terlebih dahulu ketika akan memasukkan data yang diperlukan. Data yang dimasukkan dalam aktivitas ini adalah data mata kuliah, data dosen, serta data kelas yang berisi jadwal dan ruang yang digunakan.



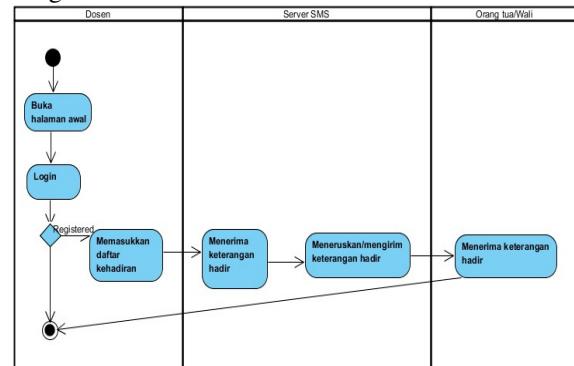
Gambar 12 Activity diagram untuk usecase Update Keterangan Hadir

Gambar 12 memperlihatkan *activity diagram* untuk *usecase update* keterangan hadir. Aktivitas ini dilakukan oleh dosen dengan melakukan autentifikasi terlebih dahulu. Gambar 13 merupakan *activity diagram* untuk *usecase update* keterangan bayar yang dilakukan oleh bagian administrasi atau keuangan. Sama seperti pada aktivitas yang dilakukan dosen, bagian administrasi/keuangan diharuskan melakukan autentifikasi terlebih dahulu untuk melanjutkan ke aktivitas berikutnya.



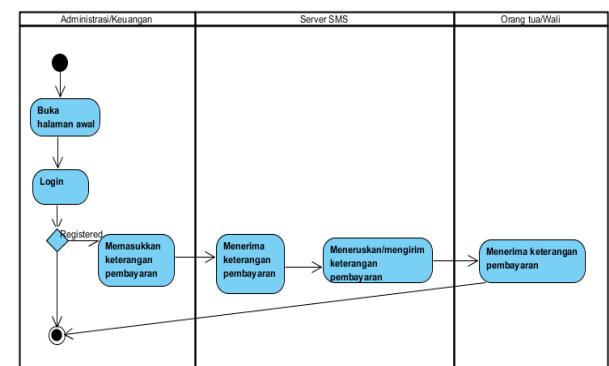
Gambar 13 Activity diagram untuk usecase Update Keterangan Bayar

Gambar 14 memperlihatkan *activity diagram* untuk *usecase* memperoleh keterangan hadir. Aktivitas yang terjadi saling terkait antar dosen, server SMS, dan orang tua atau wali.



Gambar 14 Activity diagram untuk usecase Memperoleh Keterangan Hadir

Gambar 15 memperlihatkan *activity diagram* untuk *usecase* memperoleh keterangan bayar. Aktivitas yang terjadi merupakan interaksi antara *aktor* bagian administrasi/keuangan, server SMS, dan orang tua/wali.



Gambar 15 Activity diagram untuk usecase Memperoleh Keterangan Bayar

## V. KESIMPULAN

Setelah melakukan penelitian dan menghasilkan desain sistem *smart campus* maka kesimpulan yang diperoleh adalah:

1. Penggunaan UML pada tahapan UML tidak hanya memodelkan sistem yang akan dibuat agar sesuai dengan keinginan pengguna, namun juga dapat memberikan gambaran kepada pengembang tentang bagaimana sistem tersebut akan dibuat atau bagaimana pengkodean yang akan dilakukan oleh pengembang untuk membangun sistem tersebut.
2. Pada kasus pengembangan sistem *smart campus* untuk monitoring aktivitas mahasiswa, *developer* telah memperkirakan terdapat kurang lebih 8 kelas yang akan digunakan dalam membangun sistem, yaitu kelas mata kuliah, kelas mahasiswa, kelas kelas untuk setiap mata kuliah beserta jadwal dan ruangnya, serta kelas perkuliahan, kelas pembayaran, kelas *phonebook*, kelas SMS, dan kelas *account*. Interaksi antar masing-masing kelas dimungkinkan dengan mengakses *method* yang saling terhubung pada setiap kelas.
3. *Activity diagram* pada pemodelan makalah ini digunakan untuk menggambarkan aliran aktivitas yang dilakukan masing-masing aktor yang berperan pada sistem ini.

## VI. SARAN

Pemodelan desain menggunakan UML pada makalah ini hanya fokus pada penyampaian informasi ke orang tua atau wali mahasiswa saja. Pemodelan tidak menggambarkan cara kerja sistem secara keseluruhan, seperti bagaimana skema autentifikasi agar dosen dapat memasukkan daftar absensi dan bagian keuangan dapat memasukkan keterangan pembayaran.

Meskipun demikian, dengan mengabaikan kekurangan diatas, pemodelan desain pada makalah ini tetap dapat digunakan sebagai landasan dalam pengkodean untuk membangun sistem *smart campus* yang ditawarkan.

## REFERENSI

- [1] A. Mishra dan D. Dubey, “A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios”, International Journal of Advance Research in Computer Science and Management Studies, vol.1, no.5, Oktober 2013.
- [2] D.Prasetyo dan A.Dahlan, “Pembuatan Aplikasi Pengolahan Data Transaksi Dan Penjualan Aksesoris Handphone Pada Toko Yoyo’ Cell”, *Jurnal Ilmiah DASI*, vol.16, no.1, pp.30-33, Maret 2015.
- [3] A.D.Yulianto, W.W.Winarno, dan A.Suwastono, “Pemodelan Sistem Informasi Kemetrologian Dalam Pendekatan Business Process Reengineering Untuk Pelayanan Tera Dan Tera Ulang Balai Metrologi Dinas Perindag Provinsi Jawa Tengah”, Proc.Seminar Nasional Teknologi Informasi dan Multimedia, 2014, Yogyakarta, Indonesia. Yogyakarta: STMIK AMIKOM Yogyakarta, 2014.
- [4] F. Solms, “Object-Oriented Analysis and Design using UML”, 2012. [Online]. Available: <http://myuce.url.ph/ebooks/s7cse/ObjectOrientedAnalysisAndDesignUsingUML.pdf>. [Diakses pada: 17 April 2016].
- [5] T. Sumarta, “Perancangan Model Berorientasi Objek Menggunakan Unified Modeling Language (UML) Studi Kasus Sistem Pengolahan Parkir Pada PT. TRIKARYA ABADI”, 2004. [Online]. Available: [http://rio\\_wirawan.staff.gunadarma.ac.id/Downloads/files/29293/jptunikompp-gdl-s1-2004-tsumartani-792-JURNAL.pdf](http://rio_wirawan.staff.gunadarma.ac.id/Downloads/files/29293/jptunikompp-gdl-s1-2004-tsumartani-792-JURNAL.pdf). [Diakses pada: 18 Maret 2016].
- [6] W.S.W, A.B.Prasetijo, Wahyudi, Pemodelan Voip Dengan Menggunakan UML (Unified Modeling Language), 2011. [Online]. Available: <http://eprints.undip.ac.id/25848/1/ML2F300575.pdf>. [Diakses pada: 15 April 2016]
- [7] O.M.Group, “OMG Unified Modeling Language™ (OMG UML), Superstructure Version 2.4.1”, 2011. [Online]. Available: <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>. [Diakses pada: 17 April 2016].