

Analisis dan Perancangan Sistem Informasi Kesiswaan Kursus Pemrograman (Studi Kasus: Clevio Coder Camp)

Aries Satryadhi Aldana¹, Adam Hendra Brata², Eriq Muhammad Adams Jonemaro³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹alda_07@student.ub.ac.id, ²adam@ub.ac.id, ³eriq.adams@ub.ac.id

Abstrak

Berdasarkan hasil wawancara dari peneliti dengan pemilik Clevio Coder Camp, salah satu kendala yang dihadapi adalah komunikasi dari pihak Clevio Coder Camp dengan orang tua siswa, terutama siswa yang berasal dari kelas kerjasama, karena pada beberapa sekolah yang berkerjasama dengan Clevio Coder Camp tidak memberikan akses langsung antara orang tua dan pihak Clevio Coder Camp. Clevio Coder Camp butuh suatu gambaran sistem untuk menangani permasalahan manajemen siswa maupun komunikasi terhadap orang tua siswa sehingga dapat terjalin komunikasi yang baik untuk memastikan siswa belajar secara sangkil dan mendapatkan yang terbaik bagi perkembangan karakter dan pribadi mereka. Dalam penelitian ini, dibahas bagaimana gambaran merancang suatu sistem yang cocok untuk Clevio Coder Camp. Perancangan yang dilakukan adalah perancangan arsitektur, komponen, basis data, dan antarmuka. Perancangan aplikasi ini telah diuji menggunakan *Traceability Matrix*, CBO, RFC, LCOM1 metrics, dan LCOM2 metrics. Hasil perancangan aplikasi ini memiliki kopling rendah dan *understandability* tinggi berdasarkan CBO dan RFC, dan kohesi yang sedang berdasarkan LCOM1. Hasil perancangan juga memiliki *Adaptability* tinggi karena memenuhi nilai *adaptable* dari CBO dan RFC, dan memenuhi nilai LCOM2 0 hingga 1. Selain itu, semua artefak dibuat berdasarkan *use case* yang telah dibuat sebelumnya, karena setiap kolom dan baris pada *traceability matrix* memiliki nilai.

Kata kunci: kelas, coach, parent, student, clevio, OOAD

Abstract

Based on the results of interviews of researchers with the owner of Clevio Coder Camp, one of the obstacles faced was communication from Clevio Coder Camp with parents of students, especially students who came from the collaboration class, because in some schools working with Clevio Coder Camp did not provide access directly between parents and Clevio Coder Camp. Clevio Coder Camp needs an overview of the system to deal with student management problems and communication with students' parents so that good communication can be established to ensure students learn skillfully and get the best for their character and personal development. In this research, we discuss how to design a system that is suitable for Clevio Coder Camp. The design carried out is the design of architecture, components, databases, and interfaces. The design of this application has been tested using Traceability Matrix, CBO, RFC, LCOM1 metrics, and LCOM2 metrics. The results of this application design have a low coupling and high understandability based on CBO and RFC, and cohesion that is based on LCOM1. The design results also have a high adaptability because it meets the adaptable values of CBO and RFC, and meets LCOM2 values 0 to 1. In addition, all artifacts are made based on the use case that was made before, because every column and row in the traceability matrix has a value.

Keywords: class, coach, parent, student, clevio, OOAD

1. PENDAHULUAN

Bermain gim merupakan hal yang menyenangkan, dan bisa membuat siapa saja lupa waktu. Baik anak-anak, remaja, maupun dewasa rela menghabiskan waktu berjam-jam

didepan gawai hanya untuk bermain gim. Namun, walaupun bermain gim dapat membuat orang senang dan mengurangi rasa stres, berlama-lama bermain gim dapat membuat kecanduan gim pada anak, sehingga menimbulkan berbagai masalah seperti

produktivitas yang menurun maupun masalah kesehatan jika tidak diberikan penanganan yang tepat (Halodoc, 2018). Hal – hal positif dari bermain gim akan terbuang percuma jika tidak diolah dengan tepat, salah satunya imajinasi dan kreatifitas berfikir pada anak yang telah diasah dari gim. Anak-anak memiliki kemampuan imajinasi kreatif yang tak terbatas. Pikiran mereka penuh dengan gagasan luar biasa tanpa batas. Namun umumnya untuk mewujudkan imajinasi tersebut baru mereka miliki setelah mereka dewasa, saat imajinasi kreatif dan gagasan luar biasa sudah digantikan dengan pandangan pragmatis, realistik, terbatas (Clevio Coder Camp, 2017). Oleh karena itu, untuk merealisasikan imajinasi kreatif anak –anak, Clevio Coder Camp memiliki misi untuk merealisasikan imajinasi kreatif anak pada bidang teknologi informasi.

Clevio Coder Camp merupakan sekolah pemrograman di daerah DKI Jakarta yang mengajarkan pemrograman bagi anak-anak dan keluarga. Clevio Coder Camp telah mendidik ribuan siswa dan berkembang hingga memiliki 5 cabang pada 2015. Clevio memberikan wadah bermain dengan mengajak anak-anak membangun perusahaan gimnya sendiri. Clevio Coder Camp, umumnya memiliki 2 jenis kelas, yaitu kelas Reguler dan kelas yang bekerja sama dengan sekolah – sekolah di daerah DKI Jakarta. Kelas Reguler biasanya memiliki 15 siswa per kelas, sedangkan banyaknya siswa kelas kerjasama bergantung pada pihak sekolah yang bekerjasama.

Berdasarkan hasil wawancara dari peneliti dengan pemilik Clevio Coder Camp, salah satu kendala yang dihadapi Clevio Coder Camp adalah komunikasi dari pihak Clevio Coder Camp dengan orang tua siswa, terutama siswa yang berasal dari kelas kerjasama, karena pada beberapa sekolah yang berkerjasama dengan Clevio Coder Camp tidak memberikan akses langsung antara orang tua dan pihak Clevio Coder Camp. Padahal banyak kegiatan yang dilakukan oleh siswa pada saat kegiatan Clevio Coder Camp, namun orang tua tidak tahu apa kegiatan siswa tersebut. Bahkan beberapa orang tua dari kelas kerjasama hanya tahu Clevio Coder Camp sebagai kursus pembuat gim. Padahal komunikasi antara pihak Clevio Coder Camp dan orang tua sangat penting untuk memastikan siswa belajar secara sangkil dan mendapatkan yang terbaik bagi perkembangan karakter dan pribadi mereka.

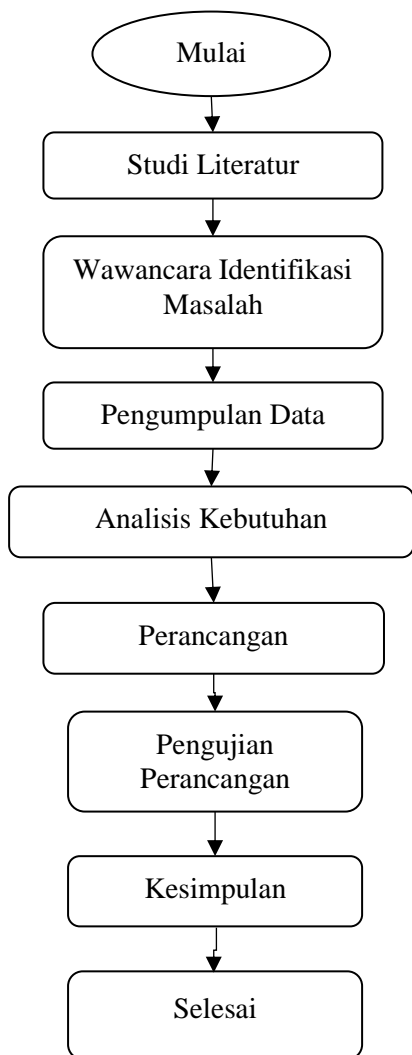
Selain itu ukuran prestasi di Clevio Coder

Camp bukan hal yang simpel. Tidak adanya patokan nilai dan banyaknya parameter membuat penyampaian laporan yang diberikan ada orang tua semakin rumit. Clevio Coder Camp menggunakan berbagai catatan yang dicatat oleh Coach saat mengobservasi berbagai kemajuan dan keberhasilan yang dialami siswa selama mengikuti kegiatan Coder Camp. Walaupun Clevio Coder Camp adalah sekolah pemrograman untuk anak-anak, Coach tidak hanya terfokus pada kemampuan programming, tetapi juga pada berbagai pengembangan diri Contohnya yaitu kemampuan dalam mengumpulkan dan mengolah pengetahuan; kemampuan bersosialisasi dan bekerjasama; kreatifitas dalam berimajinasi, berekspresi, dan pemecahan masalah; pengembangan karakter positif seperti kegigihan, pengendalian diri, dan kepemimpinan; dan lain-lain.

Karena itu, Clevio Coder Camp butuh suatu gambaran sistem untuk menangani permasalahan manajemen siswa maupun komunikasi terhadap orang tua siswa sehingga dapat terjalin komunikasi yang baik untuk memastikan siswa belajar secara sangkil dan mendapatkan yang terbaik bagi perkembangan karakter dan pribadi mereka. Dalam penelitian ini, akan dibahas bagaimana merancang suatu sistem yang cocok untuk Clevio Coder Camp. Urgensi dari perancangan ini adalah agar dapat menghasilkan perancangan yang baik dan sesuai dengan permasalahan yang dialami Clevio Coder Camp agar dapat diimplementasikan dengan baik, karena aplikasi yang baik dihasilkan dari perancangan yang baik juga..

2. METODOLOGI PENELITIAN

Penelitian yang dilakukan oleh peneliti menggunakan basis OOAD (*Object Oriented Analysis and Design*) dan MVC (*Model View Controller*). Basis ini dipilih untuk memudahkan pemahaman tentang perancangan *class* dan bagaimana interaksi *class* tersebut menghasilkan kebutuhan untuk aplikasi sistem informasi kesiswaan. Diagram alir penelitian diejawantahkan oleh gambar 1.1



Gambar 1. Diagram alir metodologi penelitian

Studi Literatur diperoleh berdasarkan dari penelitian sebelumnya, Wawancara dan pengumpulan data dilakukan dengan wawancara dari pemilik Clevio Coder Camp.

Mengenai analisis kebutuhan, penjelasan yang dilakukan yaitu penjabaran proses bisnis dalam BPMN, identifikasi aktor dan kebutuhan yang nantinya akan diejawantahkan dalam diagram *use case* dan skenario *use case*.

Pada tahap perancangan aplikasi dilakukan desain arsitektur aplikasi untuk menjelaskan arsitektur, perancangan komponen, dan, perancangan antarmuka untuk menggambarkan interaksi antara aktor dan sistem, perancangan data untuk pengelolaan data, serta perancangan arsitektur menggunakan diagram UML untuk membantu dalam membangun perangkat lunak sehingga memahami spesifikasi sistem dan rancangan sistem (Pressman, 2010).

Pada tahap pengujian dilakukan pengujian *traceability matrix* dilakukan untuk mengetahui apakah perancangan dilakukan dengan konsisten dan sesuai dengan kebutuhan dan pengujian kualitas untuk mengukur kohesi, kopling, *adaptability*, dan *understandability* dari aplikasi yang dirancang.

Di tahap kesimpulan dan saran berisi kesimpulan akhir yang menjawab rumusan masalah dan berisi saran untuk perancangan selanjutnya di waktu yang akan datang.

3. ANALISIS KEBUTUHAN

3.1. Wawancara Identifikasi Masalah

Wawancara yang dilakukan menghasilkan informasi bahwa Clevio mempunyai masalah terhadap komunikasi pada orang tua siswa. Informasi kegiatan yang dilakukan siswa pun tak tersampaikan pada orang tua terutama pada kelas kerjasama dengan sekolah, karena beberapa sekolah tidak memberikan Clevio akses komunikasi langsung pada orang tua.

3.2. Identifikasi Kebutuhan

Identifikasi kebutuhan terdiri dari kebutuhan fungsional dan non fungsional. Kebutuhan fungsional yaitu kebutuhan yang dibutuhkan secara langsung oleh Aktor. Berikut daftar kebutuhan fungsional pada tabel dibawah ini.

Tabel 1. Identifikasi Aktor

No.	Aktor	Deskripsi
1	User	Merupakan Aktor umum sistem termasuk pada pengguna yang belum melakukan login pada sistem.
2	Coach	Merupakan Aktor sistem yang mempunyai akses untuk melakukan absensi pada student, memberi tugas pada student, serta memberikan nilai pada student
3	Student	Merupakan Aktor sistem yang mempunyai akses pelaksanaan tugas
4	Parent	Merupakan Aktor sistem yang menerima hasil laporan kegiatan siswa.
5	Admin	Merupakan Aktor sistem yang memiliki hak akses penuh terhadap aplikasi dimana Aktor dapat melakukan seluruh ktifitas alam sistem

Sedangkan untuk identifikasi kebutuhan fungsional untuk perancangan sistem ini

diejawantahkan pada tabel 2.

Tabel 2. Identifikasi Kebutuhan

No.	Kode Kebutuhan	Deskripsi
1	KF-SIMKSP-001	Sistem dapat mengelola data master siswa, coach, orang tua dan report
2	KF-SIMKSP-002	Sistem dapat memberi laporan pada orang tua tentang kegiatan siswa
3	KF-SIMKSP-003	Sistem memiliki daftar absensi
4	KF-SIMKSP-004	Sistem dapat menyampaikan informasi tentang kegiatan yang dilakukan pada saat kurus
5	KF-SIMKSP-005	Sistem mampu menghubungkan coach dengan orang tua
6	KF-SIMKSP-006	Sistem mampu menyampaikan informasi hasil dari kegiatan siswa
7	KF-SIMKSP-007	Sistem dapat mempermudah akses interaksi informasi tentang kegiatan yang dilakukan dan yang akan dilakukan para aktor.
8	KF-SIMKSP-008	Sistem dapat menyusun sebuah Kelas yang terdiri dari coach dan studentnya

Sedangkan untuk kebutuhan non-fungsional disebutkan dalam tabel 3.

Tabel 3. Identifikasi Kebutuhan non Fungsional

No.	Aktor	Deskripsi
1	KNF-SIMKSP-001	Sistem dapat dengan mudah dipahami oleh Aktor

3.3. Diagram Use Case

Diagram *use case* berisi merupakan kumpulan interaksi antara aktor dengan sistem untuk mencapai tujuan tertentu. Diagram *use case* yang terbentuk berjumlah 56 *use case* yaitu *use case* Login, Pendaftaran Student, Pendaftaran Parent, Tambah Student, Tampilkan Student, Ubah Student, Tambah Coach, Tampilkan Coach, Ubah Coach, Tambah Kelas, Tampilkan Kelas, Ubah Kelas, Hapus Kelas, Tambah Peserta Kelas KBM, Tampilkan List Kelas KBM, Ubah Peserta Kelas KBM, Penggantian Coach, Setujui

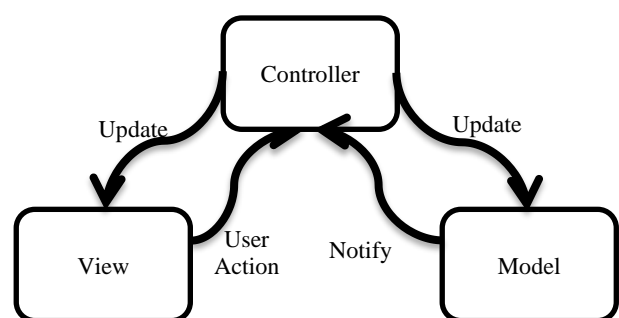
Penggantian Coach, Arsipkan Kelas KBM, Tambah Grup, Tampilkan Kelas KBM, Ubah Grup, Hapus Grup, Absensi Student, Rapor Absensi, Penilaian Kegiatan, Reset Penilaian Kegiatan, Rapor Kegiatan, Tambah Skor, Ubah Skor, Hapus Skor, Tambah Tugas, Tampilkan Tugas, Ubah Tugas, Serahkan Tugas, Hapus Tugas, Tambah Post, Setujui Post, Tampilkan Post, Ubah Post, Hapus Post, Tambah Komentar, Ubah Komentar, Hapus Komentar, Obrolan, Tampilkan Arsip, Kembalikan Arsip, Hapus Arsip, Buat Event, Tampilkan Event, Ubah Event, Hapus Event, Pendaftaran Event, Konfirmasi Peserta Event, Hapus Peserta Event, dan Logout.

4. PERANCANGAN

Perancangan yang dibuat berdasarkan dari hasil analisis kebutuhan. Pembahasan mengenai perancangan sendiri terdiri dari perancangan arsitektur, perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, pseudocode, dan perancangan antarmuka.

4.5. Perancangan Arsitektur

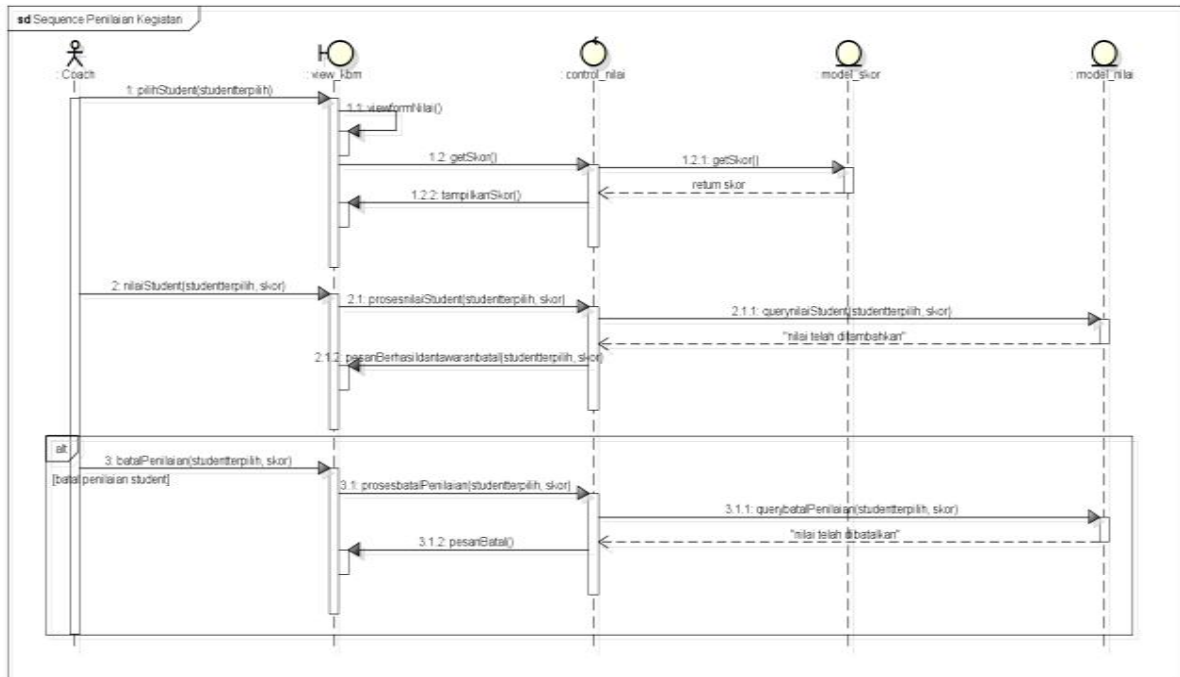
Arsitektur yang digunakan yaitu arsitektur *Model View Controller* yang disingkat *MVC*. Arsitektur ini memisahkan antara kelas yang bertanggung jawab pada data aplikasi (*Model*), kelas yang bertanggung jawab pada *user interface* (*View*), serta kelas yang bertanggungjawab pada proses aplikasi (*Controller*).



Gambar 2. Arsitektur MVC

4.2. Perancangan Sequence Diagram

Sequence Diagram dibuat berdasarkan dari perancangan Use Case dan skenarionya yang telah dibuat pada bab sebelumnya. Salah satu contoh dari Sequence diagram yaitu sequence Penilaian Kegiatan pada Gambar 3.

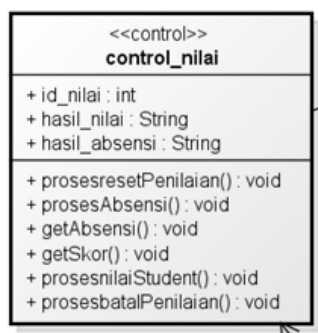


Gambar 3. Sequence Diagram Penilaian Kegiatan

4.3. Perancangan Class Diagram

Berdasarkan penggunaan arsitektur MVC atau yang disebut sebagai Model View Controller, maka perancangan class diagram dibagi menjadi 3 jenis class, yaitu class View, class Model, class Controller. Class juga dikelompokkan ke dalam package yang disesuaikan dengan kemiripan masing – masing class secara fungsi. Pembagian class dibagi menjadi 6 package, yaitu package Sesi Akun, package Navigasi, package Master Data, package List Kelas KBM, package Rapor, dan package Komunikasi.

Berikut adalah salah satu contoh class diagram yaitu control_nilai



Gambar 4. Class diagram control_nilai

Dalam package sesi_akun, terdapat class view_login yang digunakan untuk menanggapi halaman login dan control_login bertugas untuk mengecek data

agar User dapat melakukan login. Sedangkan class view_daftar digunakan untuk menanggapi dari halaman daftar dan control_daftar digunakan untuk mengatur pendaftaran. Lalu view_logout merupakan menanggapi dari tombol logout dan control_logout bertugas menghapus sesi dari user.

Dalam package navigasi, terdapat 2 class yang berhubungan dengan navigasi. Kelas view_selasar digunakan untuk menanggapi bar selasar dan control_selasar bertugas untuk mengarahkan perpindahan halaman yang dilakukan oleh user.

Dalam package yang ada pada gambar di atas, terdapat class – class yang berhubungan dengan data – data Master yang ada pada perancangan sistem. Class view_student, view_coach, dan view_kelas digunakan untuk menanggapi halaman tampilan data student, coach, dan Kelas. Untuk control_student, control_coach, dan control_kelas bertugas untuk mengatur penambahan data, perubahan data, penghapusan data atau menjadikannya sebagai arsip. Model_student, model_coach, model_kelas digunakan untuk menyimpan masing – masing data tersebut. Sedangkan class view_arsip digunakan untuk menanggapi dari halaman arsip yang bertugas menampilkan data yang berstatus arsip, untuk control_arsip digunakan untuk mengatur pengembalian maupun penghapusan arsip.

Dalam *package* List Kelas KBM tercantum *class – class* yang berhubungan dengan List Kelas KBM. Kelas *view_listkelaskbm* digunakan untuk menajawantahkan halaman List Kelas KBM, *control_listkelaskbm* bertugas untuk mengatur fungsi fungsi yang ada pada halaman List kelas KBM, seperti fungsi tambah peserta kelas KBM, pengajuan ganti coach, dan status dari kelas KBM dan *model_listkelaskbm* digunakan untuk emnyimpan data – data dari List Kelas KBM. Selain itu terdapat *class view_ganticoach* digunakan untuk menajawantahkan dari form penggantian *coach* dan *control_ganticoach* digunakan untuk mengatur penggantian coach.

Dalam *package* Rapor tercantum *class – class* yang berhubungan dengan kegiatan kelas KBM hingga penyampaian rapor. Kelas *view_kbm* digunakan untuk menajawantahkan halaman kelas KBM dan *control_kbm* bertugas untuk mengatur data dari *student* dan grup. Model *kbm* juga digunakan untuk menyimpan *student* yang dicantumkan dalam grup.

Dalam gambar ini terlihat banyak controller yang tersambung pada *view_kbm* karena digunakan untuk menjadi fungsi pada kelas KBM. Diantaranya adalah *class control_skor*, dan *class control_nilai*. Dua *class* tersebut megatur bagaimana fungsi pengaturan pada skor dan fungsi pengaturan untuk memberikan nilai skor pada *student* atau grup maupun memberikan *state* absensi pada *student* dan mempunyai model untuk masing – masing data dari fungsi tersebut yaitu *model_skor* dan *model_nilai*. *Class control_tugas* walaupun memiliki *view_tugas*, tapi banyak fungsi yang diejawantahkan pada *class view_kbm* seperti penambahan tugas dan menampilkan tugas, sedangkan *view_tugasnya* sendiri digunakan untuk *student* dalam mengunggah tugas yang diberikan oleh *coach*. Model dari *model_tugas* digunakan untuk menyimpan data –data yang berkaitan dengan tugas.

Selain fungsi – fungsi yang berkaitan dengan *view_kbm*, ada *view_rapor* dan *view_absensi* yang bertugas untuk menampilkan laporan dari kegiatan yang diberikan pada *student* dan menampilkan absensi dari *student*. dan diatur oleh *control_rapor* dan *control_absensi*.

Dalam *package* komunikasi, tercantum - *class – class* yang berhubungan dengan komunikasi. Kelas *view_post* digunakan untuk menajawantahkan halaman post, *control_post* bertugas untuk mengatur fungsi – fungsi yang

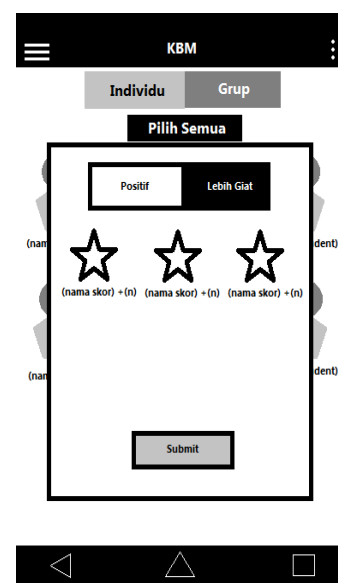
ada dalam halaman post. Data – data yang berkaitan dengan fungsi tersebut disimpan pada *model_post* yang digunakan untuk menyimpan post – post yang ada pada sistem dan *model_komentar* yang digunakan untuk menyimpan komentar – komentar yang menempel pada suatu post. *Class view_obrolan*, *control_obrolan* dan *model_obrolan* digunakan untuk menajawantahkan fungsi obrolan dengan *view_obrolan* sebagai halamannya, *control_obrolan* sebagai pengaturnya, dan *model_obrolan* sebagai tempat penyimpanannya.

4.4. Perancangan Data

Physical Data Model (PDM) digunakan dalam perancangan data. Perancangan data ini memiliki 16 tabel, yang 2 diantaranya tidak memiliki *primary key* pada atributnya (*weak entity*). Tabel yang dirancang diantaranya yaitu tabel *model_coach*, *model_admin*, *model_listkelaskbm*, *model_kelas*, *model_kbm*, *model_student_has_model_kbm*, *model_datauser*, *model_obrolan*, *model_komentar*, *model_post*, *model_skor*, *model_nilai*, *model_parent*, *model_event*, *model_student*, dan *model_tugas*.

4.3. Perancangan Antarmuka

Pada perancangan antarmuka, dibuat mock up halaman yang terdiri dari 74 halaman. Salah satu dari antarmuka tersebut ialah Antarmuka Penilaian Kegiatan yang ditunjukkan pada gambar



Gambar 5. Antarmuka Penilaian Kegiatan

Gambar di atas merupakan gambar untuk akses

Penilaian Kegiatan. Secara *default*, ketika *coach* memilih salah satu *student* dalam kelas, sistem langsung menampilkan halaman penilaian dengan skor yang telah didefinisikan oleh *coach*. *Coach* dapat langsung memberi skor apa saja yang diberikan pada *student* yang dipilihnya. Setelah itu *coach* memilih *Submit* untuk memberi skor pada *student* tersebut. Penilaian ini juga berlaku pada grup.

5. PENGUJIAN

5.1 Pengujian Traceability Matrix

Traceability Matrix pada setiap kolom dan baris pada *traceability matrix* memiliki nilai 1, yang menandakan bahwa *use case* terhubung secara konsisten pada artefaknya sehingga dapat dilacak antara *use case* dan artefaknya. Perancangan yang dibuat termasuk konsisten dan mangkus karena semua *use case* menghasilkan artefak menurut baris dan *use case* tidak ada yang kehilangan sumber artefak atau berasal dari requirement menurut kolom.

5.2 Pengujian Kualitas

Pengujian dilakukan menggunakan perhitungan (*Coupling Between Object Classes*) CBO, (*Response for a Class*) RFC, (*Lack of cohesion in methods1*) LCOM1 metrics, dan (*Lack of cohesion in methods2*) LCOM2 metrics.

Menurut Akwukwuma-Udo (2015), tingkat *adaptability* jika menggunakan metrik CBO, RFC, LCOM1, LCOM2 akan ditunjukkan pada Tabel 4.

Tabel 4. Tingkat *adaptability* berdasarkan *metric* CBO, RFC, LCOM1, dan LCOM22

No.	Metric	Adaptable	Fairly Adaptable	Poorly Adaptable
1	CBO	1-3	4-5	>5
2	RFC	1-69	70-100	>100
3	LCOM1	1	-	>1
4	LCOM2	0-1	2	>2

Pada Tabel 4. nilai *adaptability* sesuai dengan satuan CBO, RFC, LCOM1, LCOM2 dapat dibagi menjadi *adaptable* (mudah untuk diadaptasi), *fairly adaptable* (cukup dapat diadaptasi), dan *poorly adaptable* (susah untuk diadaptasi). Selain *adaptability*, nilai dari CBO dan RFC dapat mendefinisikan nilai *coupling*, dan LCOM1 dan LCOM2 memperlihatkan tingkat kohesi, dimana *coupling* yang sedikit akan terjadi hanya jika CBO kurang dari atau sama dengan 5 dan RFC kurang dari atau sama dengan 100. Sehingga pada kondisi selain itu

akan menghasilkan *coupling* yang banyak. Kohesi tinggi terlihat ketika LCOM1 lebih rendah atau sama dengan 1 dan LCOM2 lebih rendah atau sama dengan 2. Selain dari kondisi tersebut, kohesi yang ditunjukkan memiliki nilai yang rendah (Akwukwuma & Udo, 2015).

Rata-rata pada nilai CBO yaitu 1.052, berarti penilaian CBO berada diantara 1 hingga 3, sedangkan rata-rata nilai RFC 8.263 berarti nilai RFC diantara 1 hingga 69. Dalam hal ini berarti ikatan antar fungsi pada satu komponen (*coupling*) menunjukkan kemudahan *adaptability* saat ada perubahan dalam sistem (hal ini memperlihatkan bahwa nilai dari *adaptability* tinggi). Rendahnya nilai RFC membuat mudah untuk dilakukan testing dan debug *class* akibat dari kompleksitas yang rendah. Nilai CBO yang rendah berarti memiliki *coupling* yang rendah, sehingga menunjukkan bahwa perancangan *class* mudah dipahami (hal ini memperlihatkan bahwa nilai dari *understandability* tinggi) karena modul mudah untuk dipahami dan dirubah.

Coupling pada perancangan *class* bernilai rendah adalah akibat dari setiap controller hampir semuanya terhubung pada 1 kelas view (kecuali *control_tugas* yang berhubungan dengan *view_tugas* dan *view_kbm*) sehingga membantu mengurangi *coupling*.

Semua kelas untuk LCOM2 berada pada nilai antara 0 hingga 1. Ini menunjukkan bahwa perancangan ini memiliki tingkat *adaptability* tinggi (yakni kemudahan dalam beradaptasi) karena 8 *class* memenuhi LCOM1=1 dan LCOM2 0 hingga 1. Untuk LCOM1>1 dan LCOM2 0 hingga 1 terdapat 6 *class*. Pada 8 *class* memiliki kohesi yang tinggi karena LCOM<=1 dan LCOM2<=2 sedangkan 6 *class* lainnya memiliki kohesi yang rendah, karena memiliki kondisi lainnya, tidak memenuhi LCOM<=1, hanya memenuhi LCOM2<=2 saja. Semakin rendah nilai LCOM1, maka akan semakin tinggi kohesi sedangkan pada LCOM2, semakin rendah nilai maka semakin tinggi nilai *adaptability*.

6. KESIMPULAN DAN SARAN

Dari penelitian ini menghasilkan beberapa hal, sehingga dapat ditarik beberapa kesimpulan, yaitu:

1. Berdasarkan hasil rekayasa kebutuhan yang dilakukan, perancangan sistem informasi kesiswaan kursus pemrograman ini mempunyai 8 kebutuhan fungsional

dan 1 kebutuhan non fungsional yang dapat membantu kegiatan belajar mengajar dalam kursus Clevio Coder Camp. Penentuan kebutuhan fungsional diperoleh berdasarkan permasalahan terkait kegiatan belajar mengajar yang dapat disampaikan pada orang tua siswa. Kemudian, dilakukan beberapa pemodelan kebutuhan untuk memudahkan pengembang dalam menggambarkan aplikasi, seperti Pemodelan Proses Bisnis, use case diagram, dan Skenario use case

2. Berdasarkan hasil rekayasa kebutuhan, diperoleh perancangan arsitektur, perancangan data, perancangan komponen, dan perancangan antarmuka. Dalam perancangan arsitektur menggunakan arsitektur MVC (Model View Controller). *class* diagram, dan *package* diagram yang dijelaskan dengan rinci. Pada perancangan data digunakan PDM (Physical Data Model). Pada perancangan komponen terdapat pengejawantahan berupa pseudocode pada *class* controller yang akan digunakan pada aplikasi. Pada perancangan antarmuka berisi faktitius atau rekaan gambaran aplikasi.
3. Berdasarkan hasil pengujian dasar, dapat disimpulkan bahwa semua artefak dibuat berdasarkan use case yang telah dibuat sebelumnya karena setiap kolom dan baris pada traceability matrix memiliki nilai 1. Pada hasil pengujian kualitas dapat disimpulkan bahwa aplikasi memiliki kopling rendah karena rata-rata nilai CBO \leq 5 dan RFC \leq 100, kohesi rendah karena pada mayoritas *class* memiliki LCOM1 $>$ 1, understandability tinggi karena kopling rendah, dan *adaptability* tinggi karena rata-rata nilai CBO diantara 1 hingga 3, sedangkan rata-rata nilai RFC diantara 1 hingga 69, serta semua *class* memenuhi nilai LCOM2 0 hingga 1.

Berikut merupakan saran yang diberikan untuk perancangan Sistem informasi Kesiswaan Kursus Pemrograman selanjutnya antara lain:

1. Perancangan modul pemrosesan agar *Student* dapat memiliki lebih dari 1 *Parent* agar kedua orang tua *Student* dapat lebih bisa memantau anaknya dalam kegiatan kursus Clevio Coder Camp.

7. DAFTAR PUSTAKA

- Budi Sutedjo Dharma Utomo, S. M. (2006). Perencanaan & Pembangunan Sistem Informasi. Yogyakarta: Penerbit Andi.
- Clevio Coder Camp. (2017, February 20). RAGAM CARA MENILAI PRESTASI CODER CILIK. Retrieved August 30, 2018, from Clevio Coder Newsletter: <http://clev.io/camp/newsletter-detail/2/>
- Freeman, E. F. (2004). Head First Design Pattern. O'Reilly.
- Halodoc. (2018, March 28). Anak-Anak Sering Main Game? Hati-hati 7 Dampak Ini. Retrieved April 12, 2018, from Halodoc.com: <https://www.halodoc.com/anak-anak-sering-main-game-hati-hati-7-dampak-ini>
- Leffingwell, D., & Widrig, D. (2002). The Role of Requirements Traceability in System Development. The Rational Edge, 13.
- Pressman, R. S. (2010). Software Engineering - A Practitioners Approach - 7th Edition. McGraw-Hill.
- Pribadi, E. (2012, Agustus). Karakteristik Sistem Informasi. Dipetik Februari 28, 2016, dari Eko_P90: <http://pribadi-eko.blogspot.co.id/2012/08/karakteristik-sistem-informasi.html>
- R. Trianto, N. N. (2013). Konsep Dasar Sistem Pakar dan Perbedaan Sistem Pendukung Keputusan & Sistem Informasi. Malang: Program Studi Teknik Informatika, Universitas Brawijaya.
- Sommerville, I. (2011). Software Engineering 9th Edition. Boston: Addison-Wesley.