

Pengembangan Sistem Penentuan Prioritas Kebutuhan Perangkat Lunak Dengan Metode MVGB (*Majority Voting Goal Based*)

Fikri Arroisi¹, Bayu Priyambadha², Denny Sagita Rusdianto³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹fikrirrs@gmail.com, ²bayu_priyambadha@ub.ac.id, ³denny.sagita@ub.ac.id

Abstrak

Rekayasa kebutuhan adalah tahapan awal dari kegiatan rekayasa perangkat lunak, dimana dilakukan penggalan kebutuhan, dan analisis kebutuhan. Kegiatan penentuan prioritas kebutuhan adalah hal penting dalam proses rekayasa kebutuhan yang dapat menentukan keberhasilan sebuah proyek pengembangan perangkat lunak. Kegiatan penentuan prioritas kebutuhan dapat memilih mana kebutuhan yang penting dan kurang penting, sehingga sumber daya yang dimiliki tim pengembang dapat difokuskan untuk menyelesaikan masalah kunci terlebih dahulu, agar sumber daya tidak terbuang untuk menyelesaikan masalah yang tidak terlalu mendesak. Metode prioritisasi kebutuhan MVGB (*Majority Voting Goal Based*) adalah metode prioritisasi kebutuhan yang mempertimbangkan nilai yang diberikan masing-masing pemangku kepentingan dan juga nilai ketergantungan sebuah kebutuhan. Sistem penentuan prioritas kebutuhan perangkat lunak dengan metode MVGB ini adalah sistem yang menyediakan tempat untuk melakukan penggalan dan penentuan prioritas kebutuhan dengan melibatkan pemangku kepentingan yang dipilih dengan menggunakan teknologi web. Sistem ini memiliki 46 kebutuhan fungsional yang berguna untuk proses penggalan, pengamatan, dan prioritisasi kebutuhan, sistem telah diuji dengan pengujian unit untuk 3 fungsi utama sistem, pengujian integrasi untuk 3 pasang kelas utama sistem, dan pengujian validasi untuk semua kebutuhan, pengujian menghasilkan nilai valid untuk semua kasus uji.

Kata kunci: MVGB, prioritas kebutuhan, rekayasa kebutuhan, penggalan kebutuhan

Abstract

Requirement engineering is the initial stages of software engineering activities, where requirements elicited and analyzed. Requirement prioritization is an important process in requirement engineering that can determine the success of a software development project. Requirement prioritization activities can choose which requirements are important and less important, so the resource of the development team can focus on solving the key problem first so that resources are not wasted to solve a less urgent problem. MVGB (Majority Voting Goal Based) is a requirements prioritization method that takes into account the value given by each stakeholder as well as the dependency value of a requirement. This system of prioritizing software requirements using MVGB method is a system that provides a place to elicit and prioritize requirements by involving selected stakeholders using web technologies. The system has 46 functional requirements that are useful for elicitation, observation, and prioritization of requirements, the system has been tested with unit testing for 3 major system functions, integration testing for 3 pairs of system main classes, and validation testing for all requirements, the test yield a valid value for all test cases.

Keywords: MVGB, requirement prioritization, requirement engineering, requirement elicitation

1. PENDAHULUAN

Rekayasa perangkat lunak atau *software engineering* adalah disiplin ilmu rekayasa yang mempelajari aspek-aspek dalam produksi atau pengembangan perangkat lunak mulai dari tahap

pertama pengembangan perangkat lunak sampai perawatan sistem (Sommerville, 2011). Pembuatan daftar kebutuhan adalah tahap awal dalam pengembangan sebuah perangkat lunak. Daftar kebutuhan berisikan informasi kebutuhan-kebutuhan yang harus dipenuhi oleh perangkat lunak yang dibangun dan akan

menjadi dasar untuk tahap pengembangan perangkat lunak selanjutnya. Rekayasa kebutuhan adalah salah satu faktor keberhasilan dalam proses pengembangan perangkat lunak, prioritas kebutuhan berdampak positif pada proses pengembangan perangkat lunak pada tahap selanjutnya (liaqat, 2016).

Kebutuhan utama dapat diidentifikasi melalui kegiatan prioritas kebutuhan. Pada tahun 2000 FBI (*Federal Bureau of Investigation*) membangun sebuah proyek yang bernama *Virtual Case File* yang menghabiskan dana lebih dari 170 miliar dollar namun proyek ini dihentikan pada April 2005 dapat menjadi salah satu contoh kasus kegagalan proyek perangkat lunak akibat proses rekayasa kebutuhan yang tidak tepat (Marchewka, 2010). Kebutuhan yang terus berkembang dan melebar menjadi sumber utama kegagalan proyek tersebut. Perencanaan rilis, pencegahan pelebaran kebutuhan, dan pencegahan konflik antar *stakeholder* dapat tercapai dengan melakukan kegiatan prioritas kebutuhan (liaqat, 2016).

Terdapat beberapa metode prioritas kebutuhan yang telah dikembangkan seperti HCV (*Hierarchy Cumulative Voting*), AHP (*Analytic Hierarchy Process*), dan *Numerical Assignment*. Metode *Numerical Assignment* bekerja dengan mengelompokkan kebutuhan menjadi kelompok *optional*, *standard*, dan *critical*, kekurangan metode ini adalah masing-masing kebutuhan secara individual kehilangan prioritasnya karena pengelompokkan. Metode HCV bekerja dengan setiap *stakeholder* menerima uang 100 dollar kemudian setiap kebutuhan diberikan uang dengan dengan pertimbangan kepentingannya, kekurangan metode ini adalah semakin banyak kebutuhan maka proses distribusi uang menjadi semakin kompleks. Metode AHP bekerja dengan cara memasang masing-masing kebutuhan dengan kebutuhan yang lain kemudian diberikan nilai skala kepentingan dari 1 sampai 9 yang akan membentuk *weight matrix*, kekurangan metode ini adalah karena perlu dilakukannya perbandingan setiap kebutuhan dengan semua kebutuhan yang lain metode ini memakan waktu yang lama untuk dilakukan (liaqat, 2016).

Prioritas kebutuhan dengan metode MVGB (*Majority Voting Goal Based*) yang diteliti oleh liaqat et. al (2016) dapat mencegah terjadinya masalah yang ada pada metode lain yang telah disebutkan sebelumnya, metode MVGB bekerja dengan melibatkan semua

stakeholder untuk ikut dalam kegiatan prioritas kebutuhan. Metode MVGB bekerja dengan cara setiap kebutuhan diberikan nilai evaluasi dengan rentang 1 sampai 10 oleh masing-masing *stakeholder* yang akan menghasilkan nilai *Evaluation Value (E)* yang dirumuskan dengan Persamaan (1).

$$E_{ro} = \frac{\sum_{i=1}^{i=n} S_{ei}}{n} \tag{1}$$

Dimana ro merepresentasikan kebutuhan dari 1 sampai jumlah total kebutuhan, n adalah jumlah *stakeholder*, dan S_{ei} adalah nilai evaluasi yang diberikan oleh setiap *stakeholder*. Kemudian *expert* menentukan jumlah *dependency* masing-masing kebutuhan atau yang disebut dengan *Dependency Level (d)*, setelah ditentukan nilai evaluasi dan nilai *dependency* setiap kebutuhan, dapat ditentukan nilai prioritas masing-masing kebutuhan tersebut atau *Requirement Prioritization Value (rpv)* yang dirumuskan dalam Persamaan 2.

$$rpv_{ro} = (E_{ro} / (d_{ro} + 1)) / SD_{ro} \tag{2}$$

Dimana ro menunjukkan kebutuhan dari 1 sampai jumlah total kebutuhan, d_{ro} menunjukkan nilai *dependency level* kebutuhan, dan SD_{ro} menunjukkan standar deviasi dari nilai evaluasi (E) untuk kebutuhan ro , dan pada tahap terakhir kebutuhan dapat diurutkan sesuai nilai rpv -nya dimana semakin tinggi nilai rpv maka prioritasnya semakin tinggi pula.

Menurut Duarte et al. (2012) untuk meningkatkan pemahaman tentang kebutuhan yang didapatkan dapat digunakan visualisasi kebutuhan, kegiatan mencari dan melihat kebutuhan yang disajikan dalam bentuk daftar yang tidak saling terkait dapat berdampak pada berkurangkannya pemahaman *stakeholder* terhadap kebutuhan yang sudah ditemukan. Untuk menampilkan kebutuhan yang sedang hangat dibicarakan atau yang menonjol dan menyediakan informasi kegiatan dalam proyek kepada setiap *stakeholder* yang terlibat, perlu disajikan beberapa grafik sebagai berikut :

1. 10 kebutuhan dengan rata-rata *vote* tertinggi.
2. 10 kebutuhan dengan jumlah komentar tertinggi.

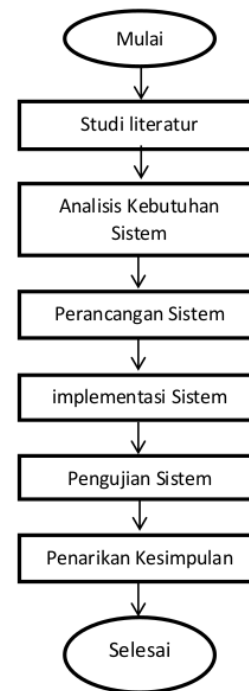
3. 10 *stakeholder* yang mengajukan kebutuhan terbanyak.
4. 10 kebutuhan dengan komentar terbanyak dalam 24 jam terakhir.
5. Grafik aktivitas proyek berdasarkan waktu dengan melihat jumlah kegiatan *voting*, jumlah komentar, dan jumlah pengajuan kegiatan perhari.

Pengembangan sistem penentuan prioritas kebutuhan dengan metode MVGB (*Majority Voting Goal Based*) bertujuan untuk membangun perangkat lunak yang dapat membantu mempercepat dan mempermudah proses *elicitation*, *monitoring*, dan penentuan prioritas kebutuhan pada sebuah proyek perangkat lunak.

2. METODE PENELITIAN

Model *Software Development Life Cycle* (SDLC) yang digunakan dalam penelitian ini adalah model waterfall, yang mana dimulai dengan proses analisis kebutuhan sampai akhirnya masuk dalam tahap pengujian. Diagram alir metodologi penelitian ini digambarkan pada Gambar 1.

Pada penelitian ini, studi literatur adalah tahap awal yang dilakukan yang mana digunakan untuk mempelajari teori dasar yang digunakan dalam penelitian ini yang didapatkan dari jurnal, artikel, konferensi, buku, dan penelitian lain yang terkait. Kemudian dilanjutkan dengan analisis kebutuhan sistem, yang berguna untuk menggali kebutuhan-kebutuhan yang dibutuhkan untuk membangun sistem ini, tahap ini menghasilkan *use case* diagram dan skenarionya untuk memperjelas dan menambahkan informasi dari setiap *use case* yang terbentuk. Kemudian tahap ketiga adalah perancangan sistem, yang bertujuan membentuk rancangan yang digunakan sebagai acuan atau dasar dari implementasi dan pengujian sistem, tahap ini menghasilkan perancangan arsitektur dalam bentuk diagram *sequence* dan diagram *class*, perancangan komponen, perancangan data, dan perancangan antarmuka. Tahap selanjutnya adalah implementasi sistem, pada tahap ini sistem mulai dibangun dengan berdasarkan pada perancangan yang telah dibuat, tahap ini



Gambar 1. Diagram alir metode penelitian

menghasilkan implementasi *program code*, *database*, dan *interface*. Tahap selanjutnya yaitu tahap pengujian sistem, pada tahap ini sistem yang sudah dibangun diuji untuk mengukur sistem apakah sesuai dengan analisis dan perancangan yang telah dilakukan, pengujian yang dilakukan disini adalah *unit testing*, *integration testing*, dan *validation testing*. Tahap yang terakhir adalah kesimpulan, kesimpulan didapat dengan berdasar pada tahap studi literatur, analisis kebutuhan, perancangan, implementasi, dan pengujian yang telah dilakukan.

3. ANALISIS KEBUTUHAN SISTEM

Tahap awal dalam pengembangan perangkat lunak adalah menggali kebutuhan yang diperlukan dalam sebuah sistem, tahap ini dinamakan dengan analisis kebutuhan. Pada penelitian ini dihasilkan 46 kebutuhan fungsional dengan beberapa kebutuhan fungsional pentingnya dapat dilihat pada Tabel 1, dan diagram *use case* dari penelitian ini digambarkan pada Gambar 2. Ada 5 aktor yang terlibat dalam sistem ini, yang pertama yaitu *Guest* yaitu *user* yang belum *login*, yang kedua *Member* adalah *user* yang sudah *login* kedalam sistem, kemudian *Client* adalah member yang menginginkan dibuatnya sebuah *software*, yang keempat adalah *Developer* sebagai tim pengembang inti, dan terakhir adalah *Expert*

sebagai ahli teknis dalam proyek.

Tabel 1. Daftar beberapa kebutuhan fungsional penting

No	Kebutuhan	Aktor	Use Case
1	Client, Developer, Expert dapat melihat daftar kebutuhan.	Client, Developer, Expert	Melihat Daftar Kebutuhan
2	Client, Developer, Expert dapat melihat daftar kebutuhan final.	Client, Developer, Expert	Melihat Daftar Kebutuhan Final
3	Client, Developer, Expert dapat menambahkan kebutuhan.	Client, Developer, Expert	Menambahkan Kebutuhan
4	Client, Developer, Expert dapat memberikan vote pada sebuah kebutuhan.	Client, Developer, Expert	Memberikan Vote Kebutuhan
5	Expert dapat menambahkan dependency sebuah kebutuhan dengan kebutuhan yang lain.	Expert	Menambahkan Dependency Kebutuhan

4. PERANCANGAN SISTEM

Setelah dilakukan analisis kebutuhan, tahap selanjutnya adalah perancangan sistem, tahap ini menghasilkan perancangan arsitektur, perancangan komponen, perancangan data, dan perancangan antarmuka.

Perancangan arsitektur menghasilkan diagram *sequence* yang menunjukkan pesan yang dikirim dan diterima antar objek dalam sistem, dan diagram *class* yang menunjukkan *class-class* yang ada dalam sistem, diagram *class* dari sistem ini dapat dilihat pada Gambar 3. Perancangan komponen menghasilkan algoritma-algoritma fungsi utama yang akan digunakan dalam sistem. Perancangan data menghasilkan struktur *database* yang digunakan sistem yang digambarkan dalam ERD (*Entity Relationship Diagram*) dan CDM (*Conceptual Data Model*), diagram CDM digambarkan pada Gambar 4. Perancangan antarmuka menghasilkan *wireframe* dari antarmuka yang akan diimplementasikan dalam sistem.

5. IMPLEMENTASI SISTEM

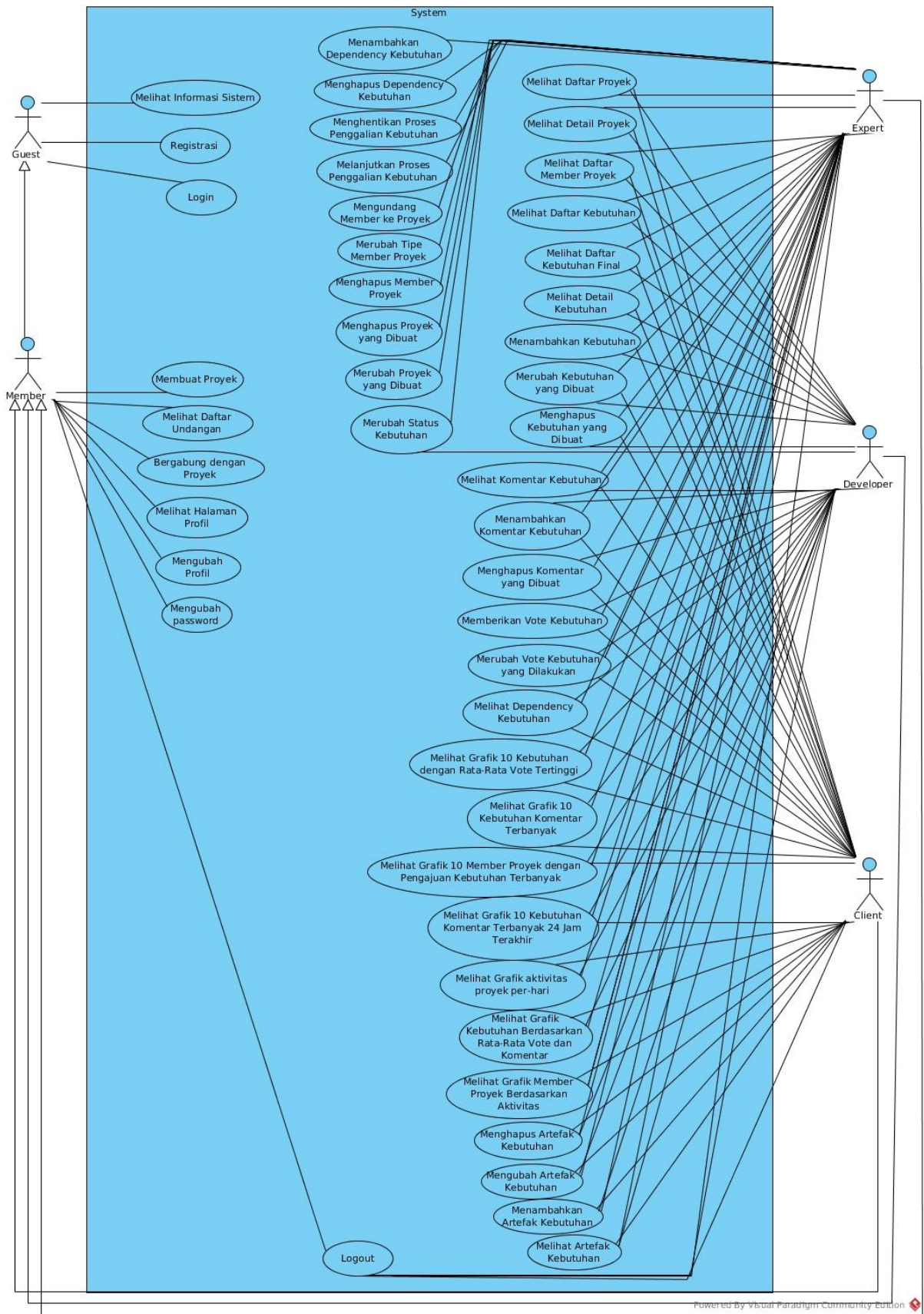
Implementasi sistem dilakukan berdasarkan pada perancangan sistem yang telah dilakukan. Pada tahap ini terdapat implementasi kode program, implementasi data, dan implementasi antarmuka. Implementasi kode program menghasilkan *source code* yang didasarkan pada algoritma yang sudah dirancang, *source code* diimplementasi dengan menggunakan bahasa pemrograman PHP dengan *framework* Laravel. Implementasi data menghasilkan *database* dengan yang dimodelkan dalam PDM (*Physical Data Model*). Implementasi antarmuka menghasilkan antarmuka yang dibuat menggunakan HTML, CSS, dan Javascript dengan *framework* VueJs sesuai dengan rancangan yang sudah dibuat. Contoh hasil implementasi daftar kebutuhan yang belum diprioritaskan ditampilkan pada Gambar 5, dan implementasi daftar kebutuhan yang telah diprioritisasi ditampilkan pada Gambar 6.

6. PENGUJIAN

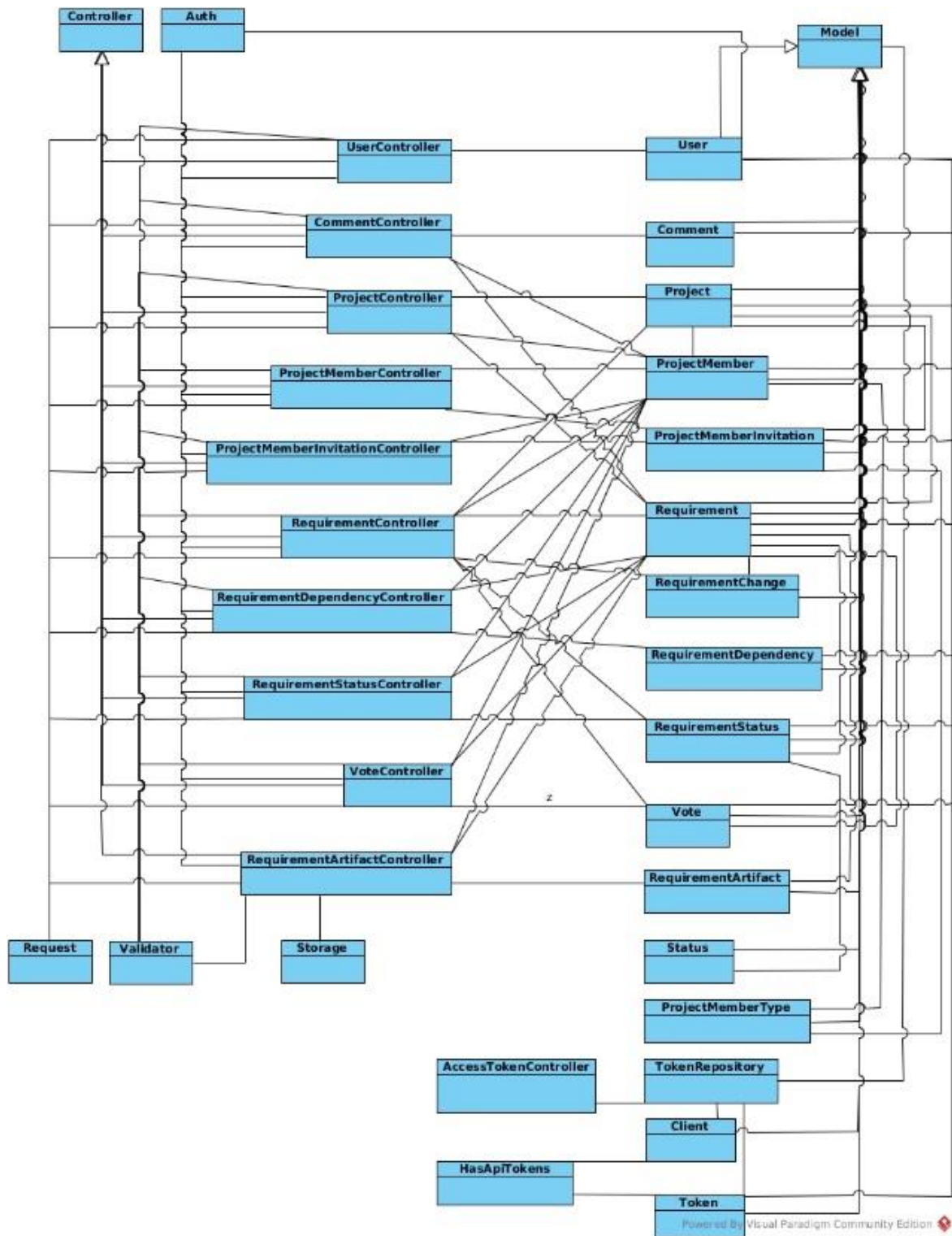
Pada tahap ini sistem diuji untuk menentukan apakah sistem yang dibangun sudah layak dan sesuai dengan analisis kebutuhan dan hasil perancangan yang telah dibuat. Terdapat tiga jenis pengujian yang dilakukan pada penelitian ini yaitu pengujian unit, pengujian integrasi, dan pengujian validasi. Pengujian unit dilakukan *whitebox testing* dengan metode pengujian *basis path testing* yang menguji semua kemungkinan jalur yang dilalui dalam sebuah komponen.

Pengujian unit dilakukan terhadap 3 *method* utama dalam sistem yaitu *method* untuk menambahkan kebutuhan, menambahkan *vote*, dan melihat kebutuhan final, masing-masing *method* tersebut memiliki nilai *Cyclomatic Complexity* 5, 7, dan 6, dimana setelah diuji semua jalurnya pengujian menghasilkan nilai valid untuk semua kasus uji.

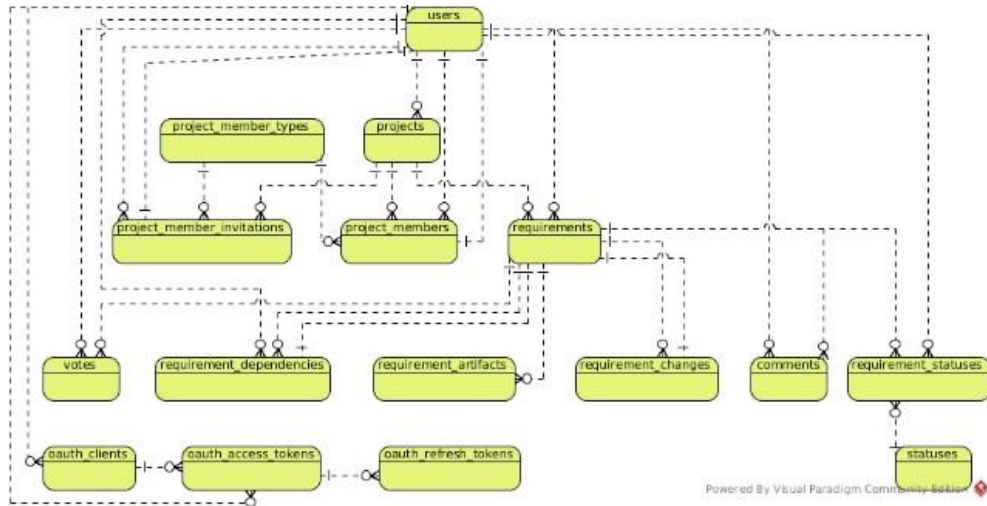
Pengujian integrasi dilakukan dengan menggunakan metode *whitebox testing*, pada penelitian dilakukan pengujian integrasi terhadap 3 pasang *class* utama yaitu *class* yang mengatur kebutuhan, *vote* kebutuhan, dan *dependency* kebutuhan, dimana untuk semua kasus uji menghasilkan nilai valid.



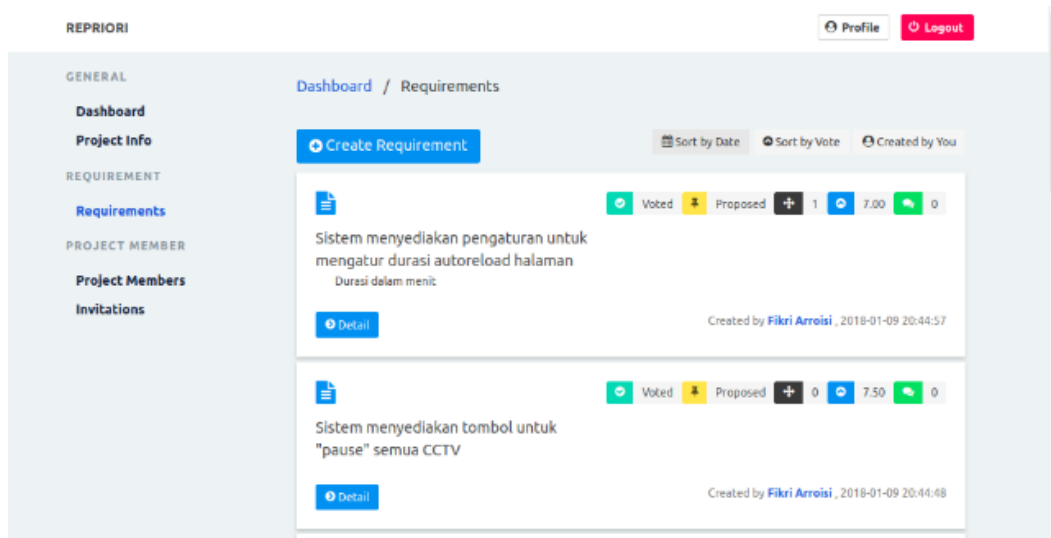
Gambar 2. use case diagram



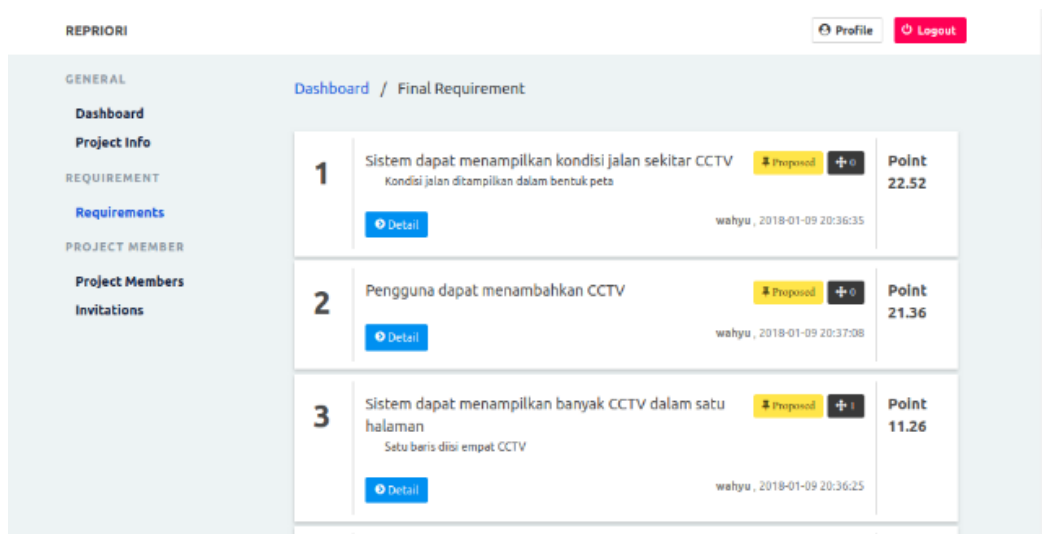
Gambar 3. Class diagram



Gambar 4. Conceptual Data Model (CDM)



Gambar 5. Hasil implementasi daftar kebutuhan yang belum diprioritisasi



Gambar 6. Hasil implementasi daftar kebutuhan yang sudah diprioritisasi

Pengujian validasi dilakukan dengan menggunakan metode *blackbox testing*, dimana semua kebutuhan fungsional dan non-fungsional yang telah didefinisikan diuji ditambahkan dengan skenario alternatif dari setiap kebutuhan sehingga dihasilkan 79 kasus uji yang setelah diuji seluruhnya menghasilkan nilai valid.

7. KESIMPULAN

Kesimpulan yang didapat pada penelitian ini :

1. Dari hasil analisis kebutuhan didapatkan 46 kebutuhan fungsional dan 2 kebutuhan non-fungsional.
2. Dari hasil perancangan didapatkan dalam perancangan arsitektur menghasilkan rancangan *sequence diagram* dan *class diagram*. Dalam perancangan komponen diperoleh rancangan algoritma-algoritma utama yang digunakan dalam sistem. Dalam perancangan data diperoleh rancangan basisdata berupa *conceptual data model*. Dan perancangan antarmuka berisi rancangan *layout wireframe*.
3. Dari hasil pengujian didapatkan, dari pengujian unit untuk 3 *method* utama menghasilkan nilai *Cyclomatic Complexity* masing-masing adalah 6, 5, dan 7 yang semua kasus ujinya menghasilkan nilai valid, dari pengujian integrasi dengan 3 pasang *class* utama dihasilkan 4 kasus uji yang semuanya menghasilkan nilai valid, dan dari pengujian validasi dihasilkan 79 kasus uji yang semuanya menghasilkan nilai valid.

8. DAFTAR PUSTAKA

- Duarte, D., Farinha, C., Silva, M. M. D., Silva, A. R. D. 2012. *Collaborative Requirements Elicitation with Visualization Techniques. Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*
- Liaqat, R. M., Azam, F., Ahmed, M. A., Mehboob, B. 2016. *A Majority Voting Goal Based Technique for Requirement Prioritization. 2016 IEEE 22nd International Conference on Automation and Computing.*
- Marchewka, J. T., 2010. *The FBI virtual case file: A case study. Communications of the IIMA: Vol. 10: Iss. 2, Article 1.*