

Implementasi Metode *Waterfall* pada Rancang Bangun Sistem Informasi Kerja Praktik Industri

Studi Kasus: Program Studi D IV Teknik Informatika Politeknik Harapan Bersama

Muhammad Fikri Hidayattullah¹, Yustia Hapsari²

¹ Program Studi DIV Teknik Informatika, Politeknik Harapan Bersama, Tegal, Indonesia

¹ muhammadfikri.uad@gmail.com

² Program Studi S1 Sistem Informasi, STMIK YMI, Tegal, Indonesia

Diterima 05 Mei 2020

Disetujui 18 November 2020

Abstract—One of the courses in D IV Informatics Engineering Department is Industrial Practices Work. This course is held to equip students of praxis in the world of software development. Unfortunately, so far the mechanism for implementing Industrial Practices Work is still in the conventional form. One of them is, for example, a student registers with the administration department of a study program, then requests a letter to be made to several software houses. In fact, sometimes the drafting of the request can be repeated in each group. This kind of process is very ineffective and slows down the performance of study programs. Therefore, through this research an information system was developed that could help to facilitate the implementation of Industrial Practices Work. The system development methodology used in this study is the waterfall method.

Index Terms— informatic engineering, information systems, waterfalls

I. PENDAHULUAN

Perkembangan teknologi informasi semakin hari semakin pesat. Data dan informasi menjadi sesuatu yang sangat bernilai tinggi, disebabkan keduanya digunakan dalam penentuan keputusan dan pengambilan kebijakan pada suatu organisasi. Untuk itulah dikembangkan berbagai macam sistem informasi. Hari ini sistem informasi merupakan hiperstruktur utama dari sebuah perusahaan dan memainkan peran penting dalam menentukan keberhasilannya [1]. Lebih jauh lagi sistem informasi digunakan untuk menunjang kompetensi utama dan diselenggarakan dengan strategi perusahaan untuk bisa terus berkompetisi [2][3]. Selain digunakan di perusahaan dalam melakukan manajemen, sistem informasi juga banyak digunakan di sekolah dan kampus [4][5]. Sistem informasi tersebut digunakan untuk mendukung pembelajaran, monitoring kegiatan, melakukan sistem penilaian terintegrasi dan kebutuhan pendidikan lainnya.

Program Studi D IV Teknik Informatika Politeknik Harapan Bersama Tegal merupakan salah satu program studi yang menyelenggarakan mata kuliah Kerja Praktik Industri (KPI). Mata kuliah ini bertujuan mengasah dan membekali mahasiswa kemampuan praksis di dunia *software development*. Mereka dituntut untuk mampu mengimplementasikan berbagai ilmu yang telah diperoleh selama menempuh studi di bangku kuliah. Oleh karena itu dalam pelaksanaan KPI tersebut Program Studi D IV Teknik Informatika mengarahkan para mahasiswa agar melaksanakan KPI di beberapa *software house* ternama di kota-kota besar seperti Jakarta, Yogyakarta, Semarang, Bandung dan sebagainya.

Prosedur pendaftaran KPI dimulai dari pendaftaran tiap kelompok mahasiswa yang akan melakukan KPI. Setelah masing-masing kelompok mendaftarkan diri ke koordinator KPI, proses selanjutnya adalah dengan pengiriman surat permohonan dari program studi kepada perusahaan atau *software house* yang hendak dituju dengan melampirkan daftar mahasiswa yang akan melaksanakan KPI. Bagian administrasi prodi akan membuat surat sebanyak yang diajukan ke perusahaan atau *software house* yang dituju mahasiswa. Jika salah satu *software house* menolak, maka akan dilanjut dengan pembuatan surat permohonan baru yang ditujukan ke *software house* lain. Begitu seterusnya. Tahap selanjutnya adalah ketika mahasiswa sudah mulai melaksanakan KPI di tempat masing-masing, setiap mahasiswa wajib mengisi daftar aktivitas harian sebagai bentuk monitoring. Daftar aktivitas tersebut nantinya akan diserahkan ke pembimbing KPI setelah selesai melaksanakan KPI. Seluruh proses di dalam tahapan tersebut masih bersifat konvensional sehingga sangat tidak efektif dan memperlambat aktivitas program studi. Model konvensional semacam ini bisa berakibat kepada duplikasi data mahasiswa disebabkan tidak ada penyaringan data

secara otomatis. Kendala lain adalah pihak administrasi prodi harus membuat surat permohonan satu per satu yang tentunya membutuhkan waktu yang lama dan menyebabkan antrian. Monitoring aktivitas harian mahasiswa juga tidak bisa terpantau setiap hari karena dicatat secara manual di buku aktivitas harian KPI.

Salah satu metodologi pengembangan perangkat lunak yang sering digunakan dalam pengembangan perangkat lunak adalah metode *waterfall*. Metode ini memiliki beberapa keunggulan diantaranya sederhana dan mudah diimplementasikan. Setiap tahapan di dalamnya didefinisikan dengan tepat [6] dan mampu bekerja dengan sangat baik dalam waktu yang singkat [7].

Oleh karena itu, melalui penelitian ini akan dikembangkan sebuah sistem informasi yang mampu membantu mengelola pelaksanaan KPI di Program Studi D IV Teknik Informatika dengan menggunakan metode *waterfall*. Sistem informasi KPI ini akan diberi nama Sikepik (Sistem Informasi Kerja Praktik).

II. TINJAUAN PUSTAKA

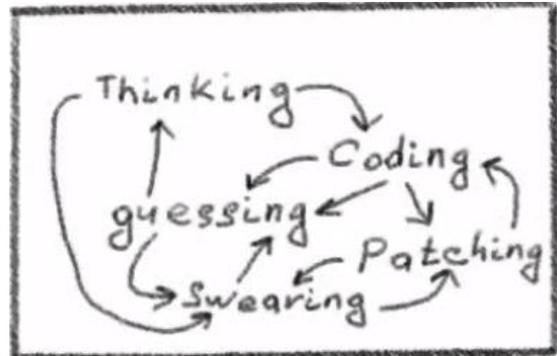
A. Sistem Informasi

Fokus utama disiplin sistem informasi adalah berkaitan dengan cara orang membangun dan menggunakan sistem berbasis komputer untuk menghasilkan informasi yang berguna. Oleh karena itu dalam implementasinya, sistem informasi selalu melibatkan orang dengan mesin yang meliputi berbagai entitas manusia dan non-manusia yang saling terhubung dalam sebuah sistem informasi [8]. Kroenke mendefinisikan sistem informasi sebagai sekelompok komponen yang saling berinteraksi untuk menghasilkan informasi. Lima komponen sistem informasi adalah perangkat keras, perangkat lunak, data, prosedur, dan orang-orang [9]. Sedangkan Laudon & Laudon di dalam bukunya *Management Information System* memberi definisi sistem informasi yaitu komponen yang saling terkait bekerja sama untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, kontrol, analisis, dan visualisasi dalam suatu organisasi [10].

B. Metode Waterfall

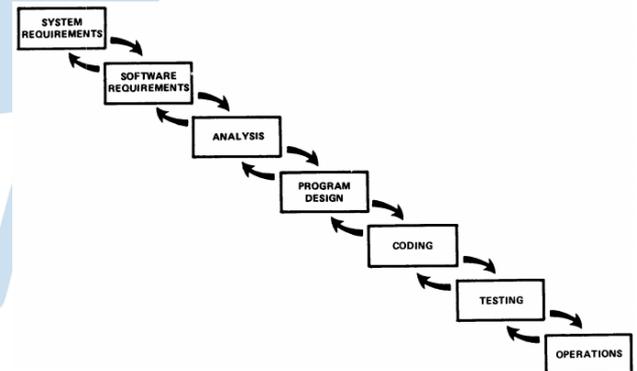
Waterfall merupakan sebuah metodologi pengembangan perangkat lunak yang sangat evolusioner, terutama di awal kelahirannya. Metode ini diprakarsai oleh seorang ilmuwan komputer yang bernama Winston W. Royce pada tahun 1970 dalam makalah publikasinya yang berjudul *Managing the Development of Large Software Systems* [11]. Penamaan *waterfall* bukanlah dari Royce sendiri, melainkan dari peserta konferensi ilmiah yang waktu itu ikut hadir dan menyimak presentasi dari Royce.

Kemunculan metode *waterfall* disebabkan pada waktu itu belum ada metodologi pengembangan perangkat lunak yang jelas dan dapat dijadikan acuan yang valid. Gambar 1 merupakan kondisi metodologi pengembangan perangkat lunak sebelum kemunculan *waterfall*.



Gambar 1. Ketidajelasan tahapan pengembangan perangkat lunak sebelum kemunculan *waterfall*

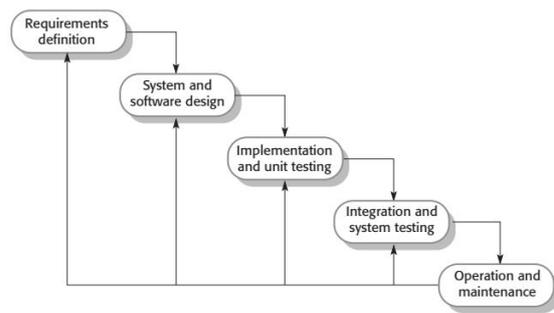
Untuk itulah Winston W. Royce mengembangkan sebuah metodologi yang dapat dijadikan solusi pengembangan perangkat lunak, terutama dalam skala besar di zamannya.



Gambar 2. Model iterasi *waterfall* yang diajukan Winston W. Royce dalam publikasinya [11]

III. METODE PENELITIAN

Metodologi penelitian yang digunakan dalam penelitian ini mengikuti alur proses pada metode *waterfall*. Berikut rincian tahapannya [12]:



Gambar 3. Metode Waterfall

A. Analisis Kebutuhan dan Definisi (Requirements Analysis and Definition)

Tahap ini merupakan langkah pertama yang sangat penting, karena berkaitan dengan informasi mengenai kebutuhan apa saja yang dibutuhkan untuk membuat sebuah perangkat lunak. Melalui tahapan ini akan digali segala informasi mengenai layanan sistem, kendala yang ada dan tujuan yang hendak dicapai. Hasil yang dicapai di dalam tahap ini adalah spesifikasi sistem yang akan dibuat atau dikembangkan.

B. Desain Sistem dan Perangkat Lunak (System and Software Design)

Selain mengalokasikan kebutuhan *software* dan *hardware*, pada tahap ini juga bertujuan membangun arsitektur sistem secara keseluruhan. Secara umum pekerjaan yang dilakukan dalam tahap ini seperti pembuatan *flowchart* sistem, perancangan UML (*Unified Modelling Language*), desain *database* hingga perancangan antarmuka sistem.

C. Implementasi dan Pengujian Unit (Implementation and Unit Testing)

Tahap ini bertugas mengimplementasikan hasil perancangan sistem pada tahap desain sistem dan perangkat lunak. Implementasi dilakukan dengan melakukan penulisan serangkaian kode program. Adapun proses pengujian unit yang dimaksud adalah proses verifikasi dari masing-masing unit atau bagian dari keseluruhan sistem. Apabila sewaktu proses pengujian unit terdapat ketidaksesuaian dari salah satu unit, maka unit tersebut harus diperbaiki terlebih dahulu.

D. Integrasi dan Pengujian Sistem (Integration and System Testing)

Secara umum sistem yang dikembangkan dalam sebuah perangkat lunak terdiri dari beberapa unit. Unit-unit yang sudah lolos verifikasi pada tahap sebelumnya perlu untuk diintegrasikan menjadi satu kesatuan sistem yang utuh. Setelah seluruh unit terintegrasi menjadi sebuah sistem, langkah selanjutnya adalah menguji sistem tersebut. Sistem yang telah lolos tahap pengujian berarti siap untuk dikirimkan ke pengguna.

E. Operasi dan Pemeliharaan (Operation and Maintenance)

Meskipun sistem yang dihasilkan telah melewati serangkaian tahapan panjang dan sudah melalui proses pengujian, namun seringkali ketika digunakan secara nyata ada beberapa *bug* ataupun *error* yang menyusul. Untuk itulah pentingnya tahap operasi dan pemeliharaan. Tahap ini diperlukan untuk menjaga keberlangsungan penggunaan perangkat lunak yang telah dihasilkan.

IV. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

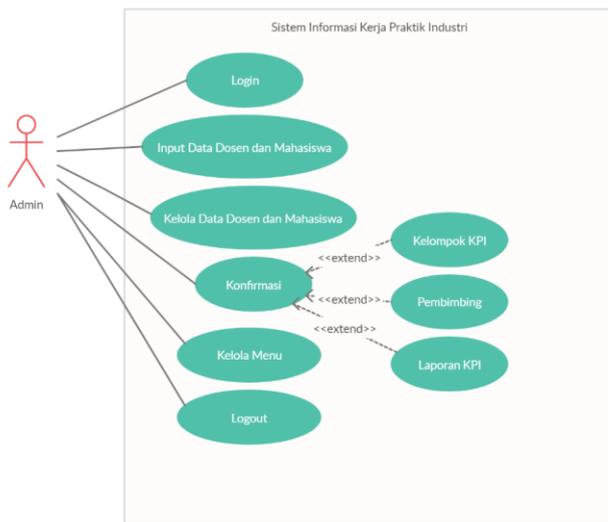
Proses analisis dilakukan untuk menemukan kebutuhan apa saja yang diperlukan dalam pembuatan sistem informasi kerja praktik (Sikepik). Analisis dilakukan mulai dari tahap registrasi mahasiswa yang akan melaksanakan KPI, proses bisnis yang sedang berjalan, mekanisme bimbingan antara mahasiswa dengan dosen pembimbing hingga pengumpulan laporan KPI. Berdasarkan hasil analisis yang dilakukan, maka sistem yang akan dikembangkan harus mampu mendata seluruh data mahasiswa yang diperlukan. Sistem juga harus mampu mempercepat proses bisnis dengan menambahkan fitur pembuatan surat permohonan yang dapat diunduh secara otomatis, fitur bimbingan *online* dengan dosen pembimbing, fitur pengisian data aktivitas harian dan beberapa fitur otomatis lainnya.

B. Desain Sistem

Sistem yang dikembangkan menggunakan pendekatan berorientasi objek, oleh karena itu desain sistem menggunakan UML (*Unified Modelling Language*). Salah satu diagram UML yang digunakan untuk menggambarkan secara global interaktivitas pengguna dengan sistem adalah *use case diagram*. Berikut ini *use case diagram* dari tiga level pengguna sistem:

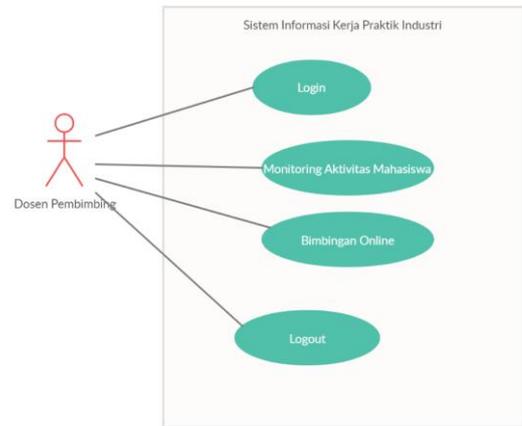
- *Use case diagram* untuk Admin

Use case ini menggambarkan aktivitas Admin selaku level pengguna tertinggi di dalam Sikepik. Admin mengelola secara keseluruhan fungsi-fungsi yang terdapat di Sikepik.



Gambar 4. Use case diagram untuk Admin

selama melaksanakan KPI dan pelaksanaan bimbingan *online*. Dua fitur ini diharapkan dapat memudahkan proses monitoring dan bimbingan dari yang sebelumnya bersifat konvensional.



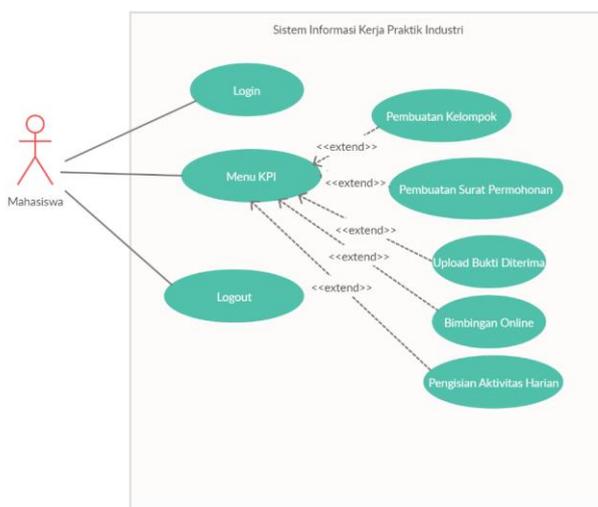
Gambar 6. Use case diagram untuk Dosen Pembimbing

• Use case diagram untuk Mahasiswa

Use case ini menggambarkan interaksi mahasiswa terhadap sistem. Mahasiswa mampu membuat kelompok secara langsung di dalam sistem dengan menambahkan secara otomatis rekannya. Mahasiswa juga bias membuat secara langsung surat permohonan pelaksanaan KPI di *software house* tanpa datang langsung ke bagian administrasi program studi.

C. Implementasi

Implementasi Sikepik menggunakan *Code Igniter framework*. *Framework* ini dikenal sangat efektif dan cepat untuk membangun sebuah sistem berbasis *website*. Berikut merupakan tampilan halaman Admin dari Sikepik:



Gambar 5. Use case diagram untuk Mahasiswa

No	ID KP	Nama	Perusahaan	Daftar Mahasiswa
1	170900240	Dilgo Digital Valley	1. AKHMAD TANTOWI (170900331) 2. KHARUDIN (17090120) 3. RENDI KUSUMA (17090045) 4. Dimas Rayandra (170900240)	
2	EulhT#_Nv	Karya Surya Gemilang	1. SYAHRU NAJI (17090023) 2. DWI PRYANTO (17090110) 3. KHIBAR PUSAKA (17090204) 4. HANZAH RIZKI PRAYOGA (17090043)	
3	Vkz7HefRq	CV. Karya Surya Gemilang	1. RIZKY TAPAYOGA (17090223) 2. FARHAN HANIF (17090115) 3. YONIEQ PRABA SAPUTRA (17090131)	
4	7dL5Nu_Syp	Daniel Co.	1. Saiful Labib Marzuqi Hidayat (17090144) 2. FITRA NURZQI HANIDANANI (17090017) 3. SITI MACHMURAH (17090105) 4. DEWI MURTI KUSUMA WARDHANI (17090149)	
5	-36o7a_uWz	Izza Web	1. MURRI KURNIASARI (17090013) 2. TRI INKATUL AMANAH (17090046) 3. ATIKA DEWI LESTARI (17090133) 4. MIFTACHUL WJATI (17090020)	
6	00Z7DLvHl	Imersa Development	1. FAJARAJI GILANG (17090128) 2. ARDU FIRMAN MAULANA (17090081) 3. ARI ABDURRAHMAN GHUFRON (17090125)	

Gambar 7. Halaman Admin Sikepik

• Use case diagram untuk Dosen Pembimbing

Fitur utama yang tergambar di dalam use case ini adalah monitoring aktivitas mahasiswa

D. Pengujian

Pengujian sistem menggunakan metode *black box testing*. Setelah melalui serangkaian proses pengujian

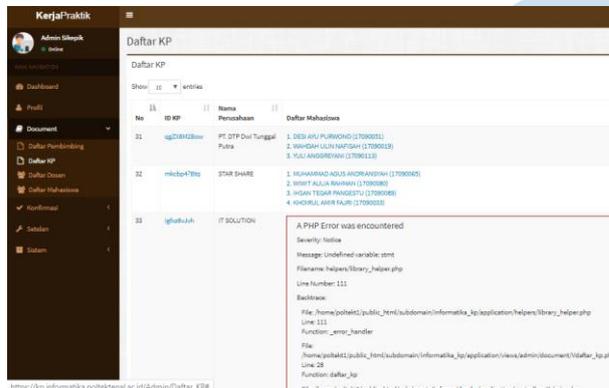
seluruh fitur dan menu yang ada, sistem terbukti telah terverifikasi dengan baik dan siap untuk diluncurkan.

E. Launching

Sistem telah di-launching di <https://kp.informatika.poltektegal.ac.id/> dan juga telah digunakan untuk membantu pelaksanaan KPI di Program Studi D IV Teknik Informatika Politeknik Harapan Bersama Tegal pada periode tahun ajaran 2019/2020. Sikepik terbukti mampu mempercepat dan mempermudah seluruh rangkaian pelaksanaan KPI dibandingkan dengan periode sebelumnya.

F. Perawatan

Sikepik dalam penggunaannya secara nyata terdapat beberapa *bug* yang muncul. *Bug* tersebut muncul biasanya pada kasus tabrakan data mahasiswa yang terduplikasi ke dalam dua kelompok yang berbeda. Disinilah perlunya upaya perawatan untuk menjaga stabilitas dan keberlangsungan sistem.



Gambar 8. Contoh *bug* pada Sikepik

V. SIMPULAN

Sikepik terbukti mampu mempermudah proses pengelolaan kegiatan KPI di Prodi DIV Teknik Informatika Politeknik Harapan Bersama Tegal setelah digunakan untuk membantu pelaksanaan KPI pada periode tahun ajaran 2019/2020. Selain itu juga metode *waterfall* yang digunakan sebagai metodologi pengembangan Sikepik juga terbukti efektif untuk pengembangan sistem informasi.

DAFTAR PUSTAKA

- [1] M. H. F. Zarandi, M. Tarimoradi, M. A. Shirazi, and I. B. Turksan, "Fuzzy intelligent agent-based expert system to keep Information Systems aligned with the strategy plans: A novel approach toward SISF," *Annu. Conf. North Am. Fuzzy Inf. Process. Soc. - NAFIPS*, vol. 2015-Sept, pp. 1–5, 2015.
- [2] M. B. Mohdzain and J. M. Ward, "A study of subsidiaries' views of information systems strategic planning in multinational organisations," *J. Strateg. Inf. Syst.*, vol. 16, no. 4, pp. 324–352, 2007.
- [3] T. Bechor, S. Neumann, M. Zviran, and C. Glezer, "A contingency model for estimating success of strategic information systems planning," *Inf. Manag.*, vol. 47, no. 1,

pp. 17–29, 2010.

- [4] R. Yanhua, "On construction of information system in college management based on team collaboration," *3rd Int. Symp. Intell. Inf. Technol. Appl. Work. IITAW 2009*, pp. 452–455, 2009.
- [5] S. Zhou and L. Chen, "The design and practice of management information system of college teaching," *Proc. - 2011 6th IEEE Jt. Int. Inf. Technol. Artif. Intell. Conf. ITAIC 2011*, vol. 1, pp. 283–285, 2011.
- [6] V. Chandra, "Comparison between Various Software Development Methodologies," *Int. J. Comput. Appl.*, vol. 131, no. 9, pp. 7–10, 2015.
- [7] J. Kasser, "The Cataract Methodology for Systems and Software Acquisition 1," *Syst. Eng.*, no. October, 2002.
- [8] A. Tatnall, "Actor-Network Theory in Information Systems Research," in *Encyclopedia of Information Science and Technology, First Edition*, vol. 19, no. 2, IGI Global, 2005, pp. 42–46.
- [9] S. Alter, "Defining information systems as work systems: implications for the IS field," *Eur. J. Inf. Syst.*, vol. 17, no. 5, pp. 448–469, Oct. 2008.
- [10] K. C. Laudon and J. P. Laudon, *Management Information Systems Managing The digital Firm Thirteen Edition Global Edition*. 2014.
- [11] D. W. W. Royce, "Managing the Development of large Software Systems," *IEEE Wescon*, no. August, pp. 1–9, 1970.
- [12] I. Sommerville, *Software engineering (10th edition)*. 2016.