

Estimasi Biaya Proyek Pengembangan Aplikasi E-Government Di Indonesia

Anung Asmoro^{#1}, Lukito Edi Nugroho^{*2}, Sujoko Sumaryono^{*3}

*#Fakultas Informatika, Telkom University
Bandung, Indonesia*

¹anungasmoro@telkomuniversity.ac.id

**Fakultas Teknik, Universitas Gajah Mada
Yogyakarta, Indonesia*

²lukito@ugm.ac.id

³sujoko@ugm.ac.id

Abstract

Proyek yang sukses adalah yang berhasil menyelesaikan semua fitur yang diperlukan sesuai dengan jadwal dan anggaran yang telah ditetapkan. Kesalahan dalam memperkirakan kebutuhan proyek dapat mengakibatkan masalah seperti kurangnya sumber daya manusia, lingkup proyek yang tidak memadai, dan penjadwalan yang terlalu ketat. Penerapan teknologi digital telah membawa perubahan signifikan dalam bentuk e-government di negara maju. Namun, untuk menerapkan e-government secara efektif, perlu penyesuaian dalam berbagai aspek seperti birokrasi, regulasi, visi pimpinan daerah, alokasi anggaran, dan peningkatan keterampilan SDM. Tantangan utama dalam implementasi e-government meliputi birokrasi yang kompleks, regulasi yang belum sesuai, dan kurangnya pemahaman tentang teknologi informasi oleh DPRD. Pengelolaan anggaran dan proyek e-government menjadi faktor kunci dalam keberhasilan implementasi e-government. Namun, masih sering terjadi masalah seperti pemotongan anggaran yang tidak tepat dan pengalokasian anggaran yang kurang efisien. Penelitian ini bertujuan untuk melakukan analisis metode estimasi yang cocok untuk proyek perangkat lunak e-government di Indonesia dan merekomendasikan model estimasi yang sesuai dengan kebutuhan.

Keywords: Estimasi proyek, E-government, Birokrasi, Pengelolaan anggaran

Abstrak

A successful project is one that completes all required features on schedule and within budget. Underestimating project estimates can lead to understaffing, under-scoping, and overly tight schedules. Overestimating project estimates tends to result in higher costs and project extensions, hindering the use of resources for subsequent projects. The utilization of digital technology has created e-government, a new form of government bureaucracy in developed countries. In the implementation of e-government, adjustments to bureaucracy, regulations, leadership vision, budget allocation, and improvement of human resource capabilities are necessary. Key challenges in e-government implementation include complex bureaucracy, inadequate regulations, and a lack of understanding of information technology by legislative bodies. Budget and project management for e-government are key to its success, yet budget cuts and improper allocations still occur frequently. This research aims to analyze estimation methods for e-government software projects in Indonesia and recommend suitable estimation models..

Kata Kunci: Project estimates, E-government, Bureaucracy, Budget management

I. INTRODUCTION

Proyek yang sukses adalah proyek yang berhasil mengirimkan semua fitur yang diperlukan sesuai dengan jadwal dan anggaran yang telah ditetapkan. Penelitian oleh Standish Group pada tahun 1995 menunjukkan bahwa hanya 16% dari proyek perangkat lunak yang berhasil mencapai hal ini [1]. Sebaliknya, 31% proyek gagal total, sementara 53% lainnya mengalami keterlambatan besar dalam waktu dan biaya, dan hanya memberikan fungsionalitas yang kurang dari yang dibutuhkan.

Menurut Paters (1999), menaksir proyek perangkat lunak adalah salah satu aspek yang paling penting dan menantang dalam pengembangan perangkat lunak. Tanpa estimasi yang tepat dan dapat diandalkan, perencanaan dan pengendalian proyek akan menjadi sulit dilakukan. Estimasi yang kurang akurat dapat mengarah pada masalah seperti kurangnya sumber daya manusia, pengaturan lingkup yang tidak memadai, dan jadwal yang terlalu ketat, yang semuanya dapat berdampak negatif pada proyek.

Untuk menghindari masalah ini, banyak organisasi cenderung untuk memperkirakan lebih tinggi dari yang seharusnya. Meskipun demikian, memperkirakan terlalu tinggi juga dapat memiliki konsekuensi buruk, termasuk penggunaan anggaran yang berlebihan dan penundaan proyek yang dapat menghambat penggunaan sumber daya untuk proyek-proyek masa depan.

Di negara-negara maju, adopsi teknologi digital telah memunculkan bentuk baru dari birokrasi pemerintahan yang disebut e-government. E-government adalah serangkaian perangkat lunak yang dibuat untuk mendukung proses layanan publik yang melibatkan pengguna akhir, seperti petugas pemerintah dan masyarakat umum. Namun, penerapan e-government memerlukan penyesuaian dalam birokrasi, regulasi, komitmen pemimpin daerah, alokasi anggaran yang memadai, peningkatan keterampilan SDM, dan penguasaan teknologi informasi.

Kesalahan dalam pengelolaan anggaran dan proyek e-government juga dapat menjadi hambatan bagi keberhasilan implementasi e-government. Pemotongan anggaran yang tidak tepat, kurangnya pemahaman tentang teknologi informasi di kalangan legislatif, dan pengalokasian anggaran yang tidak efektif dapat menghambat pengembangan perangkat lunak yang berkualitas, peningkatan keterampilan SDM, dan pemeliharaan sistem yang diperlukan dalam penerapan e-government.

Dalam konteks implementasi e-government di Indonesia, penelitian ini bertujuan untuk melakukan analisis dan memberikan rekomendasi model dan metode estimasi yang tepat untuk memperkirakan biaya dan usaha yang diperlukan dalam pengembangan perangkat lunak e-government.

II. LITERATURE REVIEW AND THEORETICAL FOUNDATION

A. Literature Review

Salah satu pertanyaan yang sering muncul di awal proyek yaitu: "Seberapa besar proyek yang akan dikerjakan?". Untuk proyek yang sifatnya real misalnya pembangunan rumah tinggal, jawaban yang muncul biasanya sangat terukur, misalnya: "Rumah tinggal satu lantai seluas 120 m² terdiri dari 1 kamar utama 20 m², 2 kamar anak 12 m², 1 kamar pembantu 9 m², 1 ruang makan 20 m², 1 ruang keluarga 20 m², 2 kamar mandi 6 m², sisanya dapur, carport, dan taman". Dan untuk proyek yang sifatnya real seperti ini jika ditanyakan hal yang sama untuk obyek yang sama kepada beberapa kontraktor/pengembang, jawabannya akan relatif serupa.

Namun untuk proyek pengembangan software, ukuran merupakan hal yang berbeda pada tim pengembang yang berbeda. Pengembang software biasanya sangat memperhatikan usaha yang diperlukan untuk menyelesaikan tugas-tugasnya. Hal ini menyebabkan mereka menggambarkan ukuran sebagai sesuatu yang dihasilkan untuk mengimplementasikan requirement. Unit-unit ukur tersebut meliputi : tampilan layar, report, tabel-tabel basis data, halaman-halaman web, scripts, kelas-kelas object, dan lainlain. Hal-hal seperti ini sifatnya abstrak, sehingga menimbulkan perbedaan persepsi atas kompleksitas unit yang bervariasi untuk tiap orang. Ini berpotensi menghadirkan jawaban yang berbeda atas pertanyaan terhadap seberapa besar ukuran proyek yang akan dikembangkan.

Di sisi estimator software, ukuran adalah hasil kuantifikasi dari proyek yang akan diserahkan atau yang diusulkan untuk diserahkan. Unit-unit ukuran yang dipakai estimator biasanya konkret seperti Source Line of code (SLOC) atau yang lebih abstrak seperti function point. Estimator biasanya menggunakan tool estimasi komersial. Apa yang pasti dalam pengembangan software adalah bahwa kata ukuran bisa berarti usaha, artifact-artifact pemrograman, unit-unit dasar dari pekerjaan, tergantung pada siapa yang menggunakannya. Ini menjadi sumber potensi kebingungan dan miskomunikasi. Kemerer dalam Touesnard (2004) menyatakan bahwa yang

paling menarik dari perbedaan-perbedaan model-model estimasi adalah antara model yang menggunakan SLOC sebagai input dengan model-model yang tidak menggunakannya [2]. Oleh karena itu, yang pertama dibahas dalam penelitian ini adalah estimasi SLOC.

Albrecht dalam Touesnard (2004) menyatakan bahwa model function point bisa menjadi alat bantu yang powerful untuk melakukan estimasi software baik di fase awal maupun setelah fase awal dalam siklus hidup pengembangan software. Di fase awal, dengan dokumen formal requirement software, dapat digunakan untuk menghitung estimasi fungsi dari software [3].

Rules of Thumb (RoT) adalah metode estimasi yang sederhana. Versi pertama model ini muncul pada 6 April 1997, versi kedua pada 13 Juni 2003 dan versi ketiga pada 20 Maret 2007. Kesederhanaan menjadi kelebihan model ini karena kebanyakan metode estimasi yang digunakan adalah yang sederhana [4]. Menurut Pascal (2003) metode RoT ini sangat cepat dan sederhana, dengan menggunakan 12 aturan yang mengatur bagaimana menghitung metrik berbasis function point.

Model Constructive Cost Model (COCOMO) dikenalkan oleh Dr. Barry Boehm pada tahun 1981. Merupakan paket software estimasi yang memodelkan estimasi biaya, usaha dan jadwal untuk pengembangan produk software baru maupun upgrade [5].

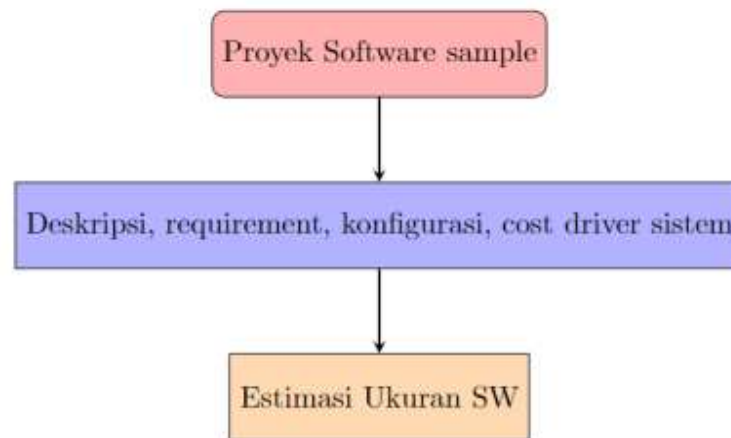
Brian, L.C. dan Wiczorek, I. (2001) menyatakan bahwa yang paling umum digunakan untuk asesmen kualitas model dan estimasi adalah mean magnitude of relative error [6]. Sedangkan Koten (2005) menyatakan bahwa MMore secara de facto menjadi standar dalam literatur estimasi dan belum ada standar alternative yang ada [7].

McConnell (2006) menyatakan bahwa keakuratan estimasi berkaitan dengan fase pengembangan software. Cooper dalam McConnell (2006) menyatakan bahwa biasanya tim pengembang mempunyai standar proses pengembangan atau Software Development Live Cycle (SDLC) [8]. Proses siklus hidup pengembangan software terdiri atas langkah-langkah (stages) dan gerbang-gerbang (gates).

B. Theoretical Foundation

Jones dalam Pressman (2006) mengatakan bahwa ada 6 fungsi yang bersifat generik untuk mengembangkan tool / sistem software estimasi [9]. Keenam fungsi tersebut adalah:

- 1) Pengukuran proyek yang akan diserahkan. Ukuran dari satu atau lebih software sebagai work product diestimasi. Representasi eksternal dari software termasuk dalam work product ini. Beberapa yang termasuk dalam representasi eksternal ini, misalnya :
 - a. tampilan layar / screens, reports, form isian, dan lain-lain.
 - b. Fungsionalitas yang diserahkan (function point).
 - c. Software itu sendiri (LOC).
 - d. Dokumentasi.
- 2) Pemilihan aktivitas-aktivitas proyek. Framework proses yang sesuai dipilih, dan rangkaian tugas rekayasa software dispesifikasikan.
- 3) Memprediksi staffing levels. Menentukan banyaknya personel yang tersedia untuk melaksanakan pekerjaan tersebut. Karena hubungan antara personel yang tersedia dan pekerjaan adalah nonlinear, ini menjadi input penting.
- 4) Memprediksi software effort. Tool estimasi menggunakan satu atau lebih model estimasi. Model estimasi ini akan menghubungkan ukuran proyek yang diserahkan ke usaha / effort yang diperlukan.
- 5) Memprediksi software cost. Dari hasil langkah d, biaya / cost bisa dihitung dengan mengalokasikan rates dari pekerja ke aktivitas-aktivitas proyek di langkah 2.
- 6) Memprediksi software schedule. Ketika usaha, staffing level, dan aktivitas-aktivitas proyek diketahui, maka draft jadwal dapat dihasilkan dengan mengalokasikan personel ke dalam aktivitas.



Gambar 1. Alur Penelitian.

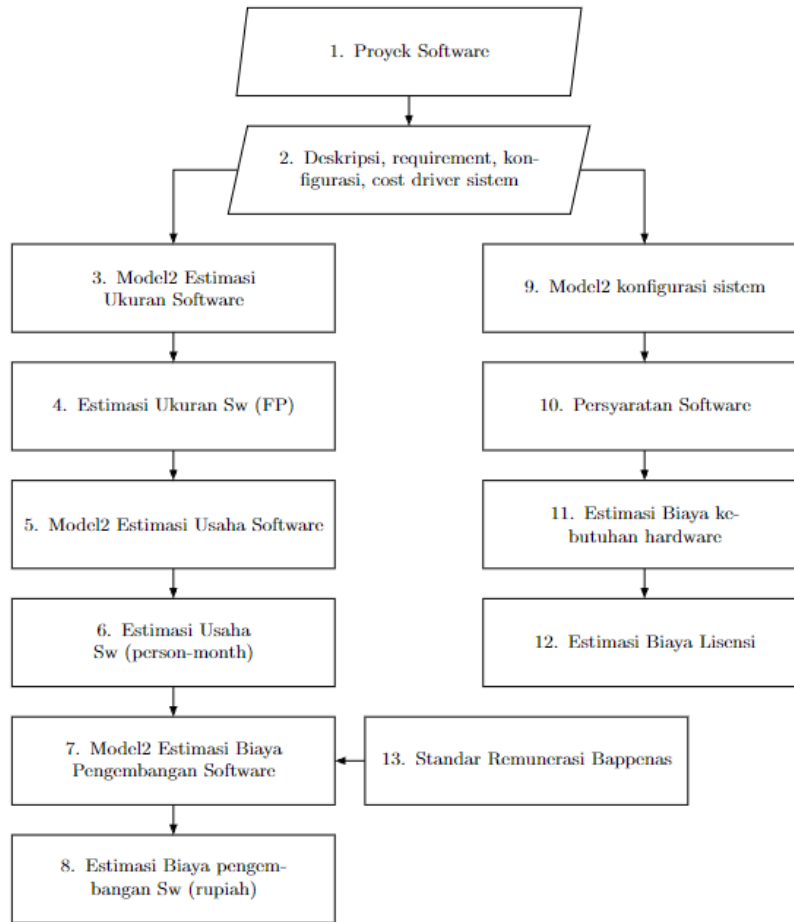
III. RESEARCH METHOD

Dalam penelitian ini jalan penelitian yang akan dilakukan tergambar dalam Gambar 1. Proyek Software sample menunjukkan keseluruhan proses atau proyek yang sedang dikerjakan. Bagian "Deskripsi, requirement, konfigurasi, cost driver sistem" merupakan elemen-elemen yang harus disertakan dalam proposal proyek. Sumber Kode merujuk pada kode sumber yang sudah ada atau yang akan dikembangkan sebagai bagian dari proyek. Practiline Source Code Line Counter adalah metode untuk menghitung jumlah baris kode sumber (SLOC - Source Lines of Code) [10]. Ukuran SW (SLOC) digunakan untuk mengukur ukuran perangkat lunak berdasarkan jumlah baris kode. Gearing Factor digunakan untuk mengatur atau mengkalibrasi ukuran perangkat lunak terhadap metrik lainnya. Ukuran SW sample FP (Function Points) adalah metode lain untuk mengukur ukuran perangkat lunak, berbeda dengan SLOC, yang berfokus pada fungsi dan fitur perangkat lunak [11].

Usaha SW sample (person-month) mengukur usaha yang dibutuhkan untuk mengembangkan perangkat lunak dalam satuan orang-bulan. Biaya SW sample (rupiah) - Merupakan estimasi biaya pengembangan perangkat lunak dalam mata uang Rupiah. Analisis hasil estimasi vs data sample melibatkan perbandingan antara hasil estimasi dengan data yang ada (sampel) untuk mengevaluasi keakuratan estimasi. Standar Remunerasi Bappenas adalah standar yang ditetapkan oleh sebuah lembaga atau badan pemerintahan untuk penentuan bayaran atau biaya tenaga kerja dalam proyek. Persyaratan Software dan Estimasi Biaya Lisensi berkaitan dengan biaya tambahan untuk perangkat lunak, termasuk biaya untuk lisensi perangkat lunak yang dibutuhkan.

IV. RESULTS AND DISCUSSION

Hasil penelitian meliputi hasil estimasi ukuran software yang akan dikembangkan, serta usaha, dan biaya yang dibutuhkan dalam proyek pengembangan software, dengan menggunakan beberapa metode dan model estimasi yang sudah ditentukan. Kemudian diukur keakuratan hasil estimasi-estimasi tersebut. Model estimasi yang menghasilkan estimasi yang akurat direkomendasikan sebagai model estimasi yang akan digunakan.



Gambar 2. Rekomendasi Proses Estimasi

Hasil dari penelitian ini adalah rekomendasi proses, model estimasi dan penerapan model estimasi. Rangkaian proses estimasi yang direkomendasikan adalah sebagaimana digambarkan dalam Gambar 2. Proyek Software: Titik awal diagram ini, kemungkinan merujuk pada inisiasi dari proyek pengembangan perangkat lunak. Deskripsi, requirement, konfigurasi, cost driver system adalah Komponen yang berisi rincian mengenai kebutuhan proyek, konfigurasi yang dibutuhkan, dan faktor-faktor yang mempengaruhi biaya. Estimasi Ukuran SW (FP) melakukan estimasi ukuran perangkat lunak yang dihitung dengan menggunakan 'Function Points', suatu teknik umum dalam manajemen proyek perangkat lunak.

Estimasi Usaha Sw (person-month) adalah hasil konkret dari estimasi usaha yang dibutuhkan untuk proyek, diukur dengan jumlah bulan kerja yang diperkirakan. Estimasi Biaya pengembangan Sw (rupiah) merupakan jumlah biaya pengembangan perangkat lunak yang diestimasi, dinyatakan dalam mata uang Rupiah. Persyaratan Software berisi daftar atau penilaian dari persyaratan perangkat lunak yang harus dipenuhi. Estimasi Biaya kebutuhan hardware adalah estimasi biaya untuk hardware yang diperlukan dalam pengembangan atau implementasi perangkat lunak. Estimasi Biaya Lisensi adalah estimasi biaya untuk lisensi dari perangkat lunak yang mungkin diperlukan dalam proyek. Standar Remunerasi Bappenas merujuk pada standar remunerasi yang ditetapkan oleh Badan Perencanaan Pembangunan Nasional di Indonesia, yang digunakan sebagai acuan dalam menentukan biaya tenaga kerja untuk proyek. Akhir dari proses ini adalah estimasi total biaya yang menggabungkan estimasi usaha, biaya perangkat lunak, hardware, dan lisensi.

V. CONCLUSION

Estimasi awal yang akurat terhadap usaha dan biaya yang diperlukan untuk pengembangan perangkat lunak sangat penting bagi alokasi yang tepat dari anggaran, waktu, dan sumber daya dalam proyek pengembangan perangkat lunak. Beberapa faktor, termasuk ukuran produk perangkat lunak, jenis perangkat lunak yang akan dikembangkan, kemampuan tim pengembang, bahasa pemrograman, serta metode dan model estimasi yang digunakan, memengaruhi hasil estimasi. Di Indonesia, patuh pada praktik remunerasi standar seperti yang dijelaskan oleh Bappenas sangat penting untuk mengestimasi biaya pengembangan perangkat lunak e-government. Model estimasi adjusted function point sangat direkomendasikan untuk mengestimasi ukuran, usaha, dan biaya proyek yang melibatkan pengembangan aplikasi perangkat lunak untuk situs web pemerintah daerah, layanan publik elektronik, dan Sistem Informasi Administrasi Keuangan (SIAP). Selain itu, metode estimasi algoritmik berbasis function point direkomendasikan untuk mengestimasi biaya dan usaha proyek pengembangan perangkat lunak e-government di Indonesia. Pendekatan ini berkontribusi pada perencanaan yang lebih akurat dan alokasi sumber daya, yang pada akhirnya meningkatkan kesuksesan upaya pengembangan perangkat lunak.

ACKNOWLEDGEMENT

This paper is supported by Telkom University, PPM under scheme Basic and Applied Research with title “Model government knowledge management system untuk mewujudkan transparansi dan partisipasi publik pada Instansi Pemerintah”.

REFERENCES

- [1] C. Bilir and E. Yafez, “Project success/failure rates in Turkey,” *International Journal of Information Systems and Project Management*, vol. 9, no. 4, pp. 24–40, 2021, doi: 10.12821/ijispm090402.
- [2] S. Nastic *et al.*, “SLOC: Service level objectives for next generation cloud computing,” *IEEE Internet Comput*, vol. 24, no. 3, pp. 39–50, 2020, doi: 10.1109/MIC.2020.2987739.
- [3] J. Zhao, “Quantum Software Engineering: Landscapes and Horizons,” Jul. 2020, [Online]. Available: <http://arxiv.org/abs/2007.07047>
- [4] S. U. Rehman, S. Bresciani, K. Ashfaq, and G. M. Alam, “Intellectual capital, knowledge management and competitive advantage: a resource orchestration perspective,” *Journal of Knowledge Management*, vol. 26, no. 7, pp. 1705–1731, Jul. 2022, doi: 10.1108/JKM-06-2021-0453.
- [5] M. A. Murad, N. A. S. Abdullah, and M. M. Rosli, “Software cost estimation for mobile application development-A comparative study of COCOMO models,” in 2021 IEEE 11th International Conference on System Engineering and Technology (ICSET), 2021, pp. 1-6.
- [6] A. Najm, A. Marzak, and A. Zakrani, “Systematic review study of decision trees based software development effort estimation,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, pp. 542–552, 2020, doi: 10.14569/IJACSA.2020.0110767.

- [7] C. van Koten, "Bayesian statistical models for predicting software development effort," Apr. 2011, Accessed: Dec. 11, 2023. [Online]. Available: <https://ourarchive.otago.ac.nz/handle/10523/933>
- [8] S. McConnell, *Software estimation: demystifying the black art*, Microsoft Press, 2006.
- [9] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, Pressman and Associates, 2005.
- [10] K. Alpernas, Y. M. Y. Feldman, and H. Peleg, "The wonderful wizard of LoC: Paying attention to the man behind the curtain of lines-of-code metrics," in *Onward! 2020 - Proceedings of the 2020 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Co-located with SPLASH 2020*, Association for Computing Machinery, Inc, Nov. 2020, pp. 146–156. doi: 10.1145/3426428.3426921.
- [11] K. Lind and R. Heldal, "On the relationship between functional size and software code size," in *Proceedings - International Conference on Software Engineering*, 2010, pp. 47–52. doi: 10.1145/1809223.1809230.