

SIRA: AN ENERGY-EFFICIENCY AND TECHNICAL DEBT ORIENTED PROCESS FOR SOFTWARE ARCHITECTURE RECONSTRUCTION oleh Daniel Alejandro Guaman Coronel. 2021. Archivo Digital UPM – Universidad Politecnica De Madrid. Halaman 263. <https://doi.org/10.20868/UPM.thesis.69219>

Tesis doktor falsafah yang dihasilkan oleh Daniel Alejandro Guaman Coronel ini hadir pada masa yang sangat sesuai dan relevan dengan perkembangan teknologi maklumat terutamanya dalam bidang kejuruteraan perisian. Dunia hari ini digerakkan oleh sistem perisian yang semakin kompleks dari telefon pintar, aplikasi media sosial, sistem pengurusan perniagaan, sehinggalah kepada sistem yang kritikal seperti kesihatan, keselamatan, dan pengangkutan pintar. Namun di sebalik kepesatan ini, terdapat isu besar yang sering kali terabai iaitu bagaimana memastikan sistem yang kita bangunkan bukan sahaja dapat digunakan dengan baik, tetapi juga tahan lama, mudah diselenggara, dan tidak membazirkan sumber tenaga. Dalam konteks inilah tesis ini menjadi penting, kerana ia cuba menjawab persoalan asas tentang kelestarian pembangunan perisian dengan menghubungkan tiga aspek utama yang sebelum ini dikaji secara berasingan iaitu seni bina perisian, hutang teknikal, dan kecekapan tenaga.

Bahagian awal tesis memberi gambaran tentang dilema industri perisian masa kini. Ramai pembangun perisian dan organisasi terpaksa berdepan dengan apa yang disebut sebagai “hutang teknikal”, iaitu kompromi dalam reka bentuk atau kod untuk mencapai matlamat jangka pendek. Konsep ini mula diperkenalkan oleh Ward Cunningham pada awal 1990-an, dan sejak itu banyak perbincangan akademik serta praktikal telah timbul mengenainya. Hutang teknikal boleh dianalogikan seperti hutang kewangan iaitu kita mendapat sesuatu secara segera, contohnya produk lebih pantas disiapkan, tetapi pada masa hadapan kita terpaksa membayar balik dengan kos tambahan seperti kod sukar diselenggara, prestasi menurun, dan risiko ralat yang lebih tinggi (Brown et al., 2010). Penulis tesis ini menegaskan bahawa hutang teknikal bukan lagi isu kecil, tetapi satu ancaman serius kepada daya saing industri, kerana kos penyelenggaraan sistem perisian lama kini jauh melebihi kos membina sistem baharu.

Hutang teknikal merujuk kepada kompromi yang dibuat dalam proses pembangunan perisian, sama ada disebabkan oleh kekangan masa, sumber, atau keputusan reka bentuk yang tidak optimum. Kompromi ini mungkin mempercepatkan pembangunan dalam jangka pendek, namun menimbulkan “hutang” yang perlu dibayar semula dalam bentuk kos penyelenggaraan, kerumitan sistem, serta risiko kepada prestasi dan kebolehskalaan perisian pada masa hadapan (Cunningham, 1992). Hutang teknikal bukan sahaja memberi kesan kepada kualiti kod dan seni bina sistem, tetapi juga terhadap kecekapan operasi keseluruhan perisian.

Dalam konteks ini, tesis ini melangkah setapak lebih jauh dengan mengaitkan hutang teknikal kepada isu penggunaan tenaga. Apabila hutang teknikal meningkat, misalnya melalui kod yang tidak cekap, seni bina yang tidak dioptimumkan, atau algoritma yang kompleks tanpa keperluan, ia boleh menyebabkan sistem menggunakan lebih banyak sumber pengkomputeran dan seterusnya tenaga elektrik. Dalam dunia di mana pusat data menguasai sebahagian besar penggunaan elektrik global, dan aplikasi mudah alih digunakan oleh berbilion pengguna setiap hari, implikasi tenaga ini menjadi semakin kritikal. Oleh itu, penyelidikan ini meneliti hubungan antara hutang teknikal

dan keberkesanan tenaga sebagai satu langkah untuk menyumbang kepada pembangunan perisian yang lebih mampan dan sejajar dengan prinsip kejuruteraan perisian hijau (Hindle, 2012).

Setelah meneliti literatur seperti yang diterangkan di dalam tesis ini, belum jelas pendekatan yang benar-benar menyatukan ketiga-tiga aspek iaitu seni bina perisian, hutang teknikal, dan kecekapan tenaga. Fowler (2018), dalam karyanya tentang *Refactoring*, memberi fokus kepada amalan memperbaiki kod agar lebih bersih dan mudah diurus, tetapi tidak terlalu menyentuh isu tenaga. Li et al. (2015) dalam kajian pemetaan sistematik mereka mengenai hutang teknikal, berjaya mengkategorikan pelbagai strategi dan teknik pengurusan hutang teknikal, tetapi tetap tidak menghubungkannya dengan penggunaan tenaga. Kajian lain pula menekankan isu kecekapan tenaga, misalnya Hindle (2012), tetapi tanpa memberi perhatian kepada struktur seni bina atau hutang teknikal. Maka jelas, tesis ini menempati ruang kosong yang kritikal dengan mengemukakan satu metodologi yang bersepadu, yang dikenali sebagai SIRA.

SIRA atau *Software Improvement in the Reconstruction of Architectures* merupakan salah satu sumbangan terbesar tesis ini. Ia bukan sekadar satu kaedah analisis, tetapi satu proses rujukan yang boleh dijadikan panduan oleh jurutera perisian untuk menilai semula kod sumber, mengenal pasti masalah dalam seni bina, serta mencadangkan penambahbaikan. Proses ini digarap dengan teliti, berasaskan analisis literatur dan kajian sistematik, serta diuji dalam konteks aplikasi sebenar. Kekuatan utama SIRA ialah ia cuba menyatukan tiga objektif yang sering kali terpisah iaitu menambah baik kualiti kod, mengurangkan hutang teknikal, dan menjimatkan tenaga. Ini menjadikan SIRA sebagai model yang lebih holistik berbanding kebanyakan metodologi lain yang hanya memberi fokus kepada satu aspek. Namun begitu, persoalan keberkesanan SIRA di dunia sebenar masih bergantung kepada tahap penerimaan industri. Seperti ditegaskan oleh Li et al. (2015), industri sering bergerak dengan logik pemahaman kos jangka pendek. Maka, penulis perlu meyakinkan pembuat keputusan bahawa pelaksanaan SIRA bukan sekadar kos tambahan, tetapi pelaburan yang memberi pulangan melalui penjimatan kos penyelenggaraan dan tenaga pada masa hadapan.

Untuk menyokong SIRA, penulis membangunkan ARCMEL, iaitu satu metamodel baharu. Metamodel ini berperanan sebagai peta konseptual yang dapat menggambarkan hubungan kompleks antara komponen perisian dengan lebih konsisten. Dalam dunia di mana seni bina sistem sering kali terlalu rumit untuk difahami sepenuhnya oleh manusia, ARCMEL membantu menyediakan abstrak yang jelas. ARCMEL bukan sahaja menjadi asas teori, tetapi juga asas teknikal bagi pembangunan alat SCAT. Dengan kata lain, ARCMEL membolehkan jurutera menilai seni bina kod dengan lebih sistematik, yang seterusnya membuka jalan untuk analisis automatik.

SCAT atau *Source Code Analysis Tool* ialah sumbangan praktikal utama yang dibangunkan berasaskan ARCMEL. Alat ini mampu menganalisis kod Java dan menghasilkan metrik berkaitan kualiti, hutang teknikal, serta penggunaan tenaga. Dari segi inovasi, SCAT sangat bernilai kerana ia mengurangkan kebergantungan pada analisis manual yang memakan masa dan sering dipengaruhi oleh subjektiviti manusia. Walau bagaimanapun, seperti ditegaskan Fowler (2018), alat pembangunan perisian hanya akan berjaya sekiranya ia benar-benar mesra pengguna. Dalam hal ini, SCAT masih mempunyai ruang penambahbaikan, khususnya dari segi antara muka

pengguna. Bagi jurutera berpengalaman, SCAT mungkin berguna untuk analisis mendalam, tetapi bagi jurutera baharu atau syarikat perisian kecil, kerumitan alat ini boleh menjadi penghalang. Maka, penulis perlu mempertimbangkan bagaimana untuk menjadikan SCAT lebih memberi kesan intuitif dan mesra pengguna agar potensinya dapat dimanfaatkan dengan lebih meluas.

Setakat ini, tiga sumbangan utama iaitu SIRA, ARCMEL, dan SCAT menunjukkan kekuatan penyelidikan ini. Ia bukan sahaja mengemukakan teori, tetapi juga membangunkan alat praktikal yang diuji dalam aplikasi sebenar. Namun, seperti mana-mana kajian akademik, ia juga mempunyai batasan. Dari segi teori, persoalan penerimaan industri masih menjadi tanda tanya. Dari segi praktikal, SCAT perlu dipertingkatkan kebolehgunaannya. Walaupun begitu, langkah yang diambil oleh penulis untuk menggabungkan teori, model, dan alat praktikal dalam satu rangka kerja adalah sesuatu yang jarang dilakukan dalam tesis peringkat PhD, dan ini menambah nilai unik kepada penyelidikan ini.

Selepas membina asas metodologi melalui SIRA, ARCMEL, dan SCAT, penulis menguji keberkesanan pendekatan ini melalui kajian kes yang terperinci. Kajian kes dijalankan menggunakan aplikasi Java yang mengikuti corak Model-View-Controller (MVC), iaitu salah satu corak seni bina yang paling banyak digunakan dalam pembangunan perisian. Pemilihan MVC bukanlah secara kebetulan, tetapi kerana ia merupakan contoh tipikal sistem berskala sederhana yang menekankan pemisahan tanggungjawab antara komponen. Dalam konteks kajian ini, pendekatan tersebut memudahkan analisis kod secara sistematik, tetapi pada masa yang sama menonjolkan isu-isu praktikal yang sering muncul dalam pembangunan sebenar.

Penulis menggunakan teknik Self-Organizing Maps (SOM), satu bentuk pembelajaran mesin tidak terselia, untuk mengkluster aplikasi berdasarkan kualiti kod. Pendekatan ini membuktikan bahawa penilaian kualiti tidak semestinya subjektif atau bergantung semata-mata pada pengalaman jurutera. Dengan bantuan algoritma, aplikasi dapat dikategorikan secara objektif, memberikan gambaran yang lebih jelas tentang sejauh mana kod memenuhi piawaian tertentu. Kajian ini menunjukkan bahawa hasil daripada SIRA adalah lebih konsisten berbanding kaedah penilaian manual yang sering kali berbeza-beza mengikut individu.

Namun demikian, skop kajian kes ini mempunyai batasanannya. Tesis ini hanya menguji pada aplikasi Java, yang meskipun popular, tidak mewakili keseluruhan bahasa pengaturcaraan moden. Dalam bidang pengaturcaraan, organisasi menggunakan pelbagai bahasa seperti Python, C++ atau JavaScript. Setiap bahasa mempunyai keunikan. Misalnya, pengurusan memori dalam C++ jauh berbeza daripada Java, manakala JavaScript lebih terdedah kepada isu prestasi dalam aplikasi web. Oleh itu, meskipun kajian kes ini membuktikan keberkesanan metodologi dalam satu konteks, generalisasi kepada bahasa lain masih perlu dilakukan. Li et al. (2015) juga menegaskan bahawa untuk memastikan sesuatu metodologi benar-benar praktikal, ia harus diuji dalam pelbagai persekitaran dan tidak bergantung kepada satu bahasa sahaja.

Selain itu, kajian kes ini memberi gambaran penting bahawa pelaksanaan kaedah analisis seperti SIRA tidak bebas daripada cabaran dunia sebenar. Data kod yang digunakan dalam aplikasi sebenar sering kali bercampur dengan kod warisan (legacy code) yang rumit, dokumentasi tidak lengkap, dan kebergantungan antara komponen yang tidak jelas. Justeru, walaupun SCAT dan

ARCMEL menyediakan alat bantu yang canggih, keberkesanan sebenar di lapangan masih memerlukan usaha penyesuaian tambahan. Di sinilah letaknya peranan pengurusan projek dan budaya organisasi, yang mungkin tidak dibincangkan secara mendalam dalam tesis tetapi tetap menjadi faktor penentu kejayaan penerapan.

Salah satu bahagian yang paling inovatif dalam tesis ini ialah pembangunan model *Complexity and Code Smells Energy Model* (CCsEM). Model ini dirangka untuk menganggarkan penggunaan tenaga berdasarkan kerumitan kod, saiz, serta kehadiran code smells. Selama ini, pengukuran tenaga dalam perisian biasanya dilakukan dengan menjalankan aplikasi dalam persekitaran terkawal dan mengukur penggunaan CPU, memori, serta tenaga elektrik. Pendekatan itu bukan sahaja memakan masa, tetapi juga memerlukan peralatan khusus. CCsEM mengatasi kekangan ini dengan membolehkan anggaran tenaga dibuat tanpa perlu menjalankan perisian, iaitu melalui analisis statik kod. Dari perspektif kecekapan, ini adalah sumbangan yang signifikan kerana ia menjimatkan kos dan masa.

Walaupun begitu, persoalan utama yang timbul ialah soal kebolehskalaan (*scalability*). Model CCsEM diuji dalam konteks aplikasi kecil dan sederhana, tetapi belum dibuktikan keberkesanannya dalam sistem industri berskala besar. Dalam dunia sebenar, sistem perisian boleh terdiri daripada jutaan baris kod dengan beratus modul yang saling bergantung. Dalam keadaan sebegini, kompleksiti meningkat secara eksponen dan tidak semestinya dapat diramalkan hanya melalui model linear. Oleh itu, untuk menjadikan CCsEM benar-benar relevan, kajian lanjut perlu dilakukan dalam kolaborasi dengan industri. Hanya dengan menguji model ini pada sistem sebenar, barulah keyakinan dapat diberikan kepada organisasi bahawa ia benar-benar boleh membantu mengurangkan kos tenaga dalam operasi berskala besar.

Selain isu kebolehskalaan, perlu juga ditanya sama ada model ini cukup teliti untuk menangani perbezaan konteks perkakasan. Penggunaan tenaga bukan sahaja ditentukan oleh kod, tetapi juga oleh faktor seperti kecekapan pemproses, sistem operasi, dan konfigurasi pelayan. Misalnya, kod yang sama mungkin menggunakan tenaga yang berbeza pada CPU berkuasa rendah berbanding CPU moden yang lebih cekap. Oleh itu, meskipun CCsEM membuka jalan baharu, ia masih memerlukan pelengkap dalam bentuk penentuan yang mengambil kira faktor perkakasan. Hindle (2012) juga menegaskan bahawa hubungan antara kod dan tenaga adalah kompleks dan memerlukan pendekatan multi-dimensi.

Dari sudut kekuatan, CCsEM jelas membawa nilai tambah kerana ia memberi perspektif baharu kepada isu hutang teknikal. Selama ini, hutang teknikal dilihat semata-mata sebagai isu penyelenggaraan dan kos kewangan. Namun dengan adanya CCsEM, hutang teknikal kini boleh dilihat juga sebagai isu tenaga. Kod yang penuh dengan *code smells* bukan sahaja menyukarkan penyelenggaraan, tetapi juga meningkatkan penggunaan tenaga, sekali gus menyumbang kepada pembaziran sumber dan kesan alam sekitar. Perspektif ini mengubah cara kita memahami konsep hutang teknikal. Ia tidak lagi hanya soal kecekapan ekonomi organisasi, tetapi juga soal tanggungjawab sosial dan alam sekitar.

Disamping itu, penulis juga membincangkan bagaimana CCsEM boleh digunakan bersama SIRA dan SCAT untuk membentuk satu rangka kerja lengkap. SIRA memberi proses, ARCMEL

memberi abstraksi teori, SCAT memberi alat analisis, dan CCsEM memberi dimensi tenaga. Kombinasi ini menjadikan tesis ini lebih komprehensif berbanding penyelidikan lain. Walaupun begitu, kejayaan sebenar masih bergantung kepada pelaksanaan di dunia sebenar. Tanpa sokongan industri, rangka kerja ini mungkin kekal sebagai sumbangan akademik yang baik tetapi sukar diterjemahkan kepada impak praktikal.

Di sinilah cadangan masa depan tesis ini menjadi penting. Penulis menyarankan agar kajian lanjutan dilakukan dengan melibatkan kolaborasi industri. Cadangan ini amat wajar kerana ia membolehkan model diuji dalam sistem sebenar yang mempunyai kompleksiti dan cabaran yang lebih nyata. Selain itu, tesis ini juga boleh diperkayakan dengan metrik tambahan seperti keselamatan (*security*) atau kebolehpercayaan (*reliability*). Dalam dunia moden, aspek keselamatan tidak boleh diketepikan kerana serangan siber semakin meningkat. Jika rangka kerja ini dapat menggabungkan metrik keselamatan bersama metrik tenaga dan hutang teknikal, ia akan menjadi lebih lengkap dan bernilai. Fowler (2018) juga menyokong pandangan bahawa aspek kualiti perisian tidak boleh dilihat secara terpisah, tetapi perlu ditangani secara menyeluruh.

Apabila dibandingkan dengan penyelidikan terdahulu, kajian ini jelas menonjol kerana berjaya menyatukan beberapa bidang yang biasanya dikaji secara berasingan. Martin Fowler (2018) melalui bukunya *Refactoring* menekankan teknik memperbaiki kod agar lebih bersih dan mudah diurus, tetapi tidak menyentuh langsung tentang isu tenaga. Li et al. (2015) pula menghasilkan kajian pemetaan sistematik yang luas mengenai hutang teknikal, dengan memberi tumpuan kepada cara mengenal pasti, mengukur, dan menguruskan hutang teknikal. Namun, kajian tersebut berhenti pada tahap penyelenggaraan kod dan keberkesanan pembangunan semata-mata, tanpa mengaitkannya dengan isu kelestarian tenaga. Hindle (2012) pula memperkenalkan konsep *Green Mining*, iaitu metodologi untuk menghubungkan perubahan perisian dengan penggunaan tenaga, tetapi kajiannya tidak memberi tumpuan kepada hutang teknikal atau rekonstruksi seni bina.

Dalam hal ini, tesis yang dikaji benar-benar melengkapkan jurang yang wujud. Ia menyatukan tiga dimensi iaitu kualiti kod, hutang teknikal, dan tenaga dalam satu rangka kerja yang lebih utuh. Penulis tidak hanya berhenti pada peringkat teori, tetapi turut membangunkan alat praktikal seperti SCAT dan model seperti CCsEM. Hal ini menjadikan penyelidikan ini lebih kukuh kerana ia menghubungkan antara sumbangan teori dan aplikasi praktikal. Dengan kata lain, tesis ini bukan sekadar sebuah sumbangan akademik, tetapi juga berpotensi membawa perubahan kepada cara industri membangunkan dan menilai sistem perisian.

Namun demikian, ada juga beberapa kelemahan yang boleh diketengahkan. Seperti yang telah diterangkan, kajian ini masih terlalu terikat pada satu bahasa pengaturcaraan, iaitu Java. Meskipun Java merupakan salah satu bahasa paling popular, ia tidak semestinya mencerminkan cabaran unik dalam bahasa lain. Misalnya, Python mempunyai pengurusan memori yang berbeza, manakala C++ memberikan kawalan yang lebih langsung ke atas perkakasan, yang mungkin memberi kesan berbeza terhadap penggunaan tenaga. Justeru, secara umumnya hasil kajian ini masih terbatas. Kedua, meskipun SCAT berupaya menjalankan analisis automatik, keberkesanan alat ini bergantung kepada sejauh mana ia boleh digunakan dengan mudah oleh pelbagai jenis jurutera. Jika antara muka pengguna terlalu kompleks, ia mungkin hanya digunakan oleh kumpulan kecil penyelidik atau jurutera berpengalaman, sekali gus mengehadkan impaknya.

Selain itu, cadangan penulis untuk melibatkan industri dalam kajian masa depan adalah amat penting tetapi juga penuh cabaran. Dalam persekitaran industri, faktor-faktor seperti kos, masa, dan budaya kerja sering menjadi penghalang kepada penerapan metodologi baharu. Ramai pengurus projek lebih menekankan penghantaran pantas berbanding pelaburan jangka panjang untuk mengurangkan hutang teknikal atau menjimatkan tenaga. Maka, kajian lanjutan perlu dilaksanakan bukan sahaja untuk membuktikan keberkesanan teknikal, tetapi juga menyediakan justifikasi ekonomi yang kukuh. Dengan menunjukkan bahawa SIRA, SCAT, dan CCsEM boleh menjimatkan kos dalam jangka panjang melalui pengurangan masa penyelenggaraan dan penggunaan tenaga, barulah penerimaan industri dapat diperoleh.

Dari perspektif akademik, kajian doktor falsafah ini memberikan sumbangan yang cukup besar. Ia bukan sahaja menambah teori melalui pembangunan metamodel ARCMEL, tetapi juga melengkapkan perbincangan akademik dengan data empirikal daripada kajian kes. Di samping itu, penyelidikan ini membuka jalan untuk bidang baharu, iaitu integrasi antara pemuliharaan senibina perisian, pengurusan hutang teknikal, dan kejuruteraan perisian hijau. Ketiga-tiga bidang ini sebelum ini sering dianggap terpisah, tetapi tesis ini menunjukkan bahawa penggabungan mereka boleh menghasilkan nilai tambah yang signifikan. Bagi penyelidik masa hadapan, tesis ini menjadi rujukan penting untuk membina kajian lanjutan, sama ada dalam bentuk pengembangan kepada bahasa lain, pengujian berskala besar, atau penambahan metrik baharu seperti keselamatan dan kebolehpercayaan.

Dari perspektif industri, penyelidikan ini berpotensi membawa impak yang signifikan. Dengan adanya metodologi seperti SIRA, syarikat perisian boleh menilai semula sistem sedia ada dan mengurangkan kos penyelenggaraan. Dengan SCAT, mereka boleh mengautomasikan analisis kod untuk mengenal pasti masalah dengan lebih cepat. Dengan CCsEM, mereka boleh menganggarkan penggunaan tenaga tanpa perlu menjalankan sistem, sekali gus menjimatkan kos operasi. Jika digabungkan, ketiga-tiga ini boleh membantu syarikat perisian bukan sahaja meningkatkan kualiti sistem, tetapi juga menyokong agenda kelestarian. Dalam era di mana syarikat teknologi semakin ditekan untuk mengurangkan jejak karbon, rangka kerja yang ditawarkan tesis ini boleh menjadi satu kelebihan kompetitif.

Namun, tesis ini juga menimbulkan persoalan lebih luas tentang arah tuju kejuruteraan perisian pada masa depan. Adakah pembangunan perisian akan terus menekankan hanya pada aspek fungsi dan masa penghantaran atau ia akan bergerak ke arah paradigma baharu yang lebih menekankan kelestarian, penjimatan kos, dan tanggungjawab sosial? Tesis ini seolah-olah menjadi suara yang mendesak kita untuk memilih jalan kedua. Dengan menekankan bahawa hutang teknikal juga bermakna hutang tenaga, penulis mengingatkan bahawa setiap keputusan pembangunan perisian mempunyai implikasi bukan sahaja kepada organisasi, tetapi juga kepada alam sekitar.

Kesimpulannya, kajian doktor falsafah oleh Daniel Alejandro Guaman Coronel ini boleh dianggap sebagai penyelidikan yang matang, kritikal, dan relevan. Ia memperkenalkan proses SIRA, metamodel ARCMEL, alat SCAT, serta model CCsEM yang kesemuanya saling melengkapi dalam menangani isu kualiti kod, hutang teknikal, dan kecekapan tenaga. Walaupun terdapat keterbatasan dari segi skop kajian kes dan kebolehgunaan alat, sumbangan utama tesis ini tetap

jelas. Ia berjaya menghubungkan teori dengan amali, menambah dimensi tenaga kepada konsep hutang teknikal, serta menyediakan landasan untuk penyelidikan dan pembangunan masa depan. Dari perspektif akademik, tesis ini menunjukkan kepelbagaian literatur iaitu dari perspektif industri, ia menawarkan potensi praktikal yang besar. Justeru, tesis ini wajar diberi pengiktirafan sebagai salah satu penyelidikan penting dalam usaha membentuk masa depan pembangunan perisian yang lebih mampan dan bertanggungjawab.

Rujukan

- Brown, N., Cai, Y., Guo, Y., Kazman, R., Kim, M., Kruchten, P., ... & Seaman, C. (2010). Managing Technical
- Cunningham, W. (1992). The WyCash Portfolio Management System. ACM OOPSLA Experience Report.
- Debt in Software-Reliant Systems. Proceedings of FSE/SDP Workshop on Future of Software Engineering Research.
- Fowler, M. (2018). Refactoring: Improving the Design of Existing Code. Addison-Wesley.
- Hindle, A. (2012). Green Mining: A Methodology of Relating Software Change and Configuration to Power Consumption. Proceedings of the 34th International Conference on Software Engineering (ICSE).
- Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. Journal of Systems and Software, 101, 193–220.

Muhamad Adham Hambali

Fakulti Multimedia Kreatif dan Komputeran
Universiti Islam Selangor