MODEL SENI BINA PENYIMPANAN KUERI DI JABATAN PERKHIDMATAN AWAM MALAYSIA

HERRNIWATI NGATIMAN
KAMSURIAH AHMAD

Fakulti Teknologi dan Sistem Maklumat, Universiti Kebangsaan Malaysia

ABSTRAK

Proses pembangunan sistem amat berkait rapat dengan isu pengaturcaraan dan pengkodan yang tidak terancang. Pelbagai isu yang kerap ditimbulkan termasuk masalah pembinaan kueri yang meliputi proses penyimpanan dan perkongsian kueri. Dua (2) faktor yang telah dikenalpasti sebagai punca kerapnya berlaku masalah ini adalah perancangan pembangunan yang tidak tersusun dan tahap pengetahuan dan kemahiran pengaturcara sistem yang rendah. Kedua-dua faktor ini telah mengakibatkan tempoh pembangunan sistem menjadi lebih panjang dan proses penyelenggaraan kod sumber di dalam sistem menjadi sukar. Maka kajian ini mengambil pendekatan untuk mendapatkan suatu kaedah yang standard dan sistematik bagi mengatasi masalah penyelenggaraan kueri ini. Satu model seni bina penyimpanan dan perkongsian kueri telah dicadangkan dalam kajian ini. Hasil daripada penyimpanan dan perkongsian kueri tersebut, proses penghasilan kueri menjadi lebih mudah dan cepat kerana pengaturcara sistem boleh menggunakan kueri yang telah disediakan dan disesuaikan mengikut keperluan masing-masing. Selain itu, proses ini juga dapat meningkatkan tahap kolaborasi dalam ahli kumpulan pembangunan sistem dan memudahkan proses penyelengaraan kueri. Oleh yang demikian, cadangan model seni bina ini dapat menjadi batu pengukur kepada Jabatan Perkhidmatan Awam Malaysia untuk mengimplementasikannya ke dalam persekitaran pembangunan sistem di organisasi.

1. PENGENALAN

Suatu kueri yang berkualiti boleh diukur berdasarkan output yang dihasilkan samada betul dan tepat serta memenuhi keperluan pengguna. Proses penghasilan kueri yang sistematik dan mantap merupakan perkara penting dalam pembangunan sistem. Perancangan pembangunan sistem yang tidak tersusun menyebabkan organisasi menghadapi pelbagai jenis risiko termasuk tempoh pembangunan sistem yang panjang, sistem yang dibangunkan mempunyai pertindihan fungsi dengan sistem lain, dan pengaturcara sistem perlu meletakkan usaha yang lebih dalam menghasilkan suatu kueri bagi sistem yang mempunyai fungsi sama.

Salah satu faktor yang perlu diambil kira dalam masalah pembangunan sistem di organisasi adalah tahap kemahiran dan pengetahuan pengaturcara yang berbeza antara satu dengan yang lain. Ruang lingkup pengetahuan dan tahap penguasaan pengaturcara sistem terhadap sistem yang sedang dibangunkan amat perlu dititik beratkan supaya proses pembangunan menjadi lancar. Menurut kajian yang dijalankan, terdapat segelintir pengaturcara sistem yang kurang kemahiran teknikal terutama dari aspek pengaturcaraan. Kumpulan minoriti seperti ini memerlukan bimbingan dan panduan daripada ahli pasukan yang lebih berkemahiran dan senior bagi membolehkan mereka mempelajari selok belok struktur pengatrucaraan dengan lebih cepat. Salah satu cara yang difikirkan sesuai bagi mengatasi masalah ini adalah dengan mengadakan konsep kolaborasi dan diterapkan dalam persekitaran pembangunan sistem. Malah, sekiranya konsep kolaborasi sistem yang meliputi perkongsian kueri dan data dapat diterapkan dalam menangani isu seperti ini, pengaturcara sistem dapat menggunakan kueri dan data secara gunasama dan masalah seumpama ini dapat diatasi.

Walaupun ada kajian terdahulu yang telah dilakukan oleh (Arden et al., 2012) bagi mencadangkan kaedah bagi mengatasi masalah yang melibatkan penyimpanan kueri ini, namun, ia tidak dapat memenuhi keperluan di JPA. Ini kerana struktur pembangunan aplikasi di JPA yang unik dan tersendiri serta tidak dapat disesuaikan dengan kaedah standard yang diaplikasikan oleh industri. JPA menguruskan pelbagai sistem aplikasi yang besar dan kompleks dan ia memerlukan kaedah yang bersesuaian. Selain itu, kajian tersebut tidak dapat memenuhi keperluan di JPA kerana ia hanya tertumpu kepada modul dan kelas yang ringkas. Oleh yang demikian, kajian ini melihat perlunya dibangunkan suatu kaedah sesuai yang dapat memenuhi keperluan berikut kepada JPA:

a) Kaedah penyimpanan dan perkongsian kueri yang cekap dan sistematik
b) Kaedah penyimpanan kueri yang dapat menyingkatkan tempoh pembangunan sistem terutama yang melibatkan pembinaan kueri

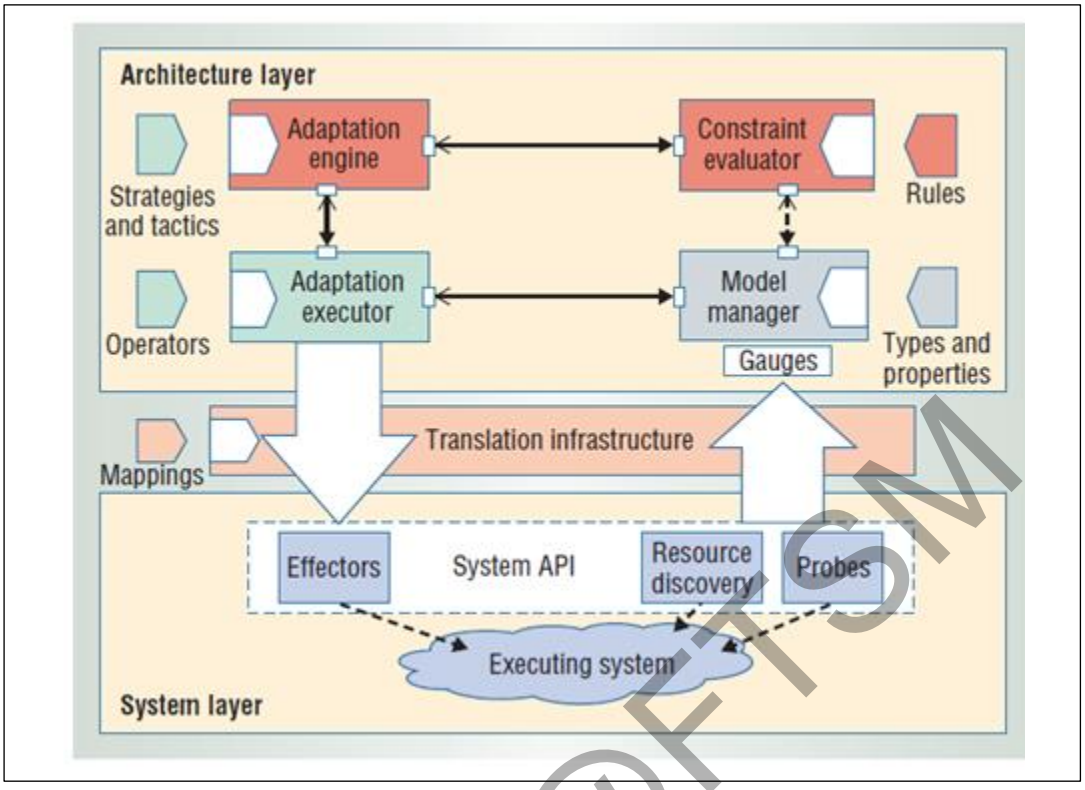
Rentetan daripada isu yang kerap ditimbulkan ini, kajian mencadangkan satu model seni bina penyimpanan kueri yang melibatkan proses penyimpanan dan perkongsian kueri secara sistematik. Cadangan model ini dapat dijadikan rujukan asas oleh pihak JPA dalam usaha membantu mempertingkatkan kualiti pembangunan sistem dan seterusnya meluaskan keupayaan kemahiran pengaturcara sistem.

2. KAJIAN BERKAITAN

2.1. Struktur Kueri Terkini

a) Model adaptasi sendiri Penggunaan Semula Infrastruktur (Garlan, Cheng, Huang, Schmerl, & Steenkiste, 2004)

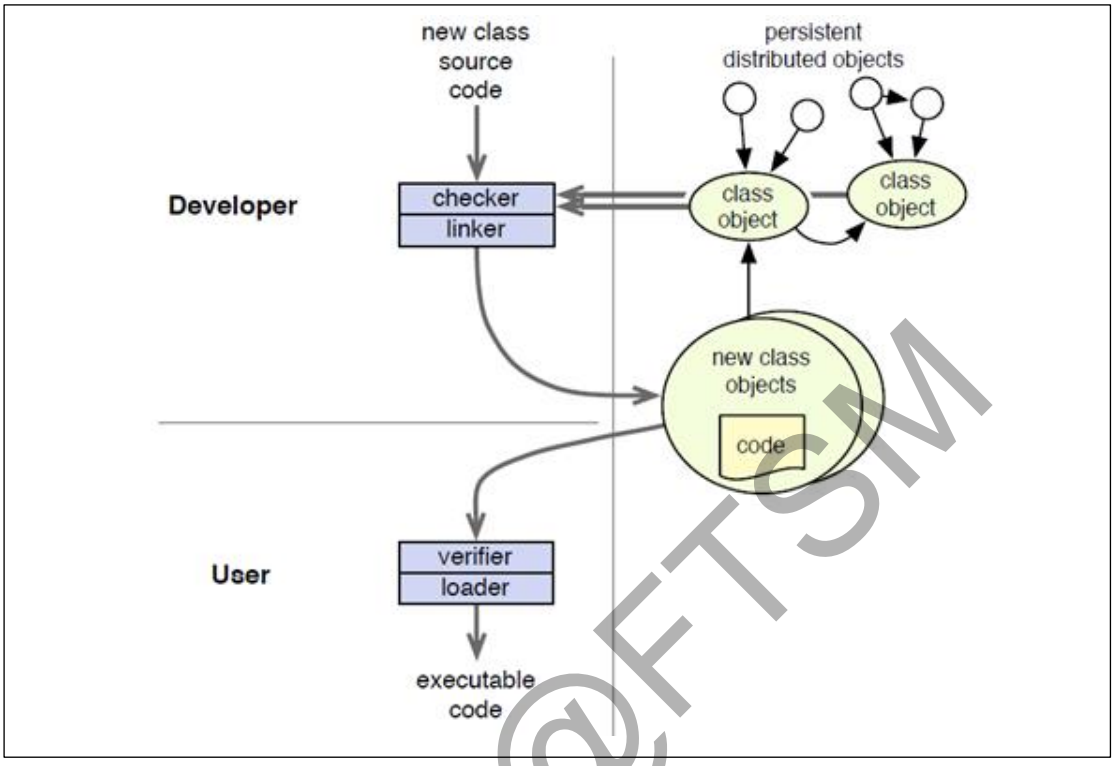
Pemodelan perkongsian perisian telah dijalankan oleh (Garlan et al., 2004). Kajian yang dijalankan oleh mereka adalah berkaitan dengan penggunaan semula perisian dan infrasturktur yang boleh diadaptasi dan disesuaikan dengan persekitaran pembangunan pengguna lain. Dalam model adaptasi sendiri yang dicipta oleh Garlan ini, telah mereka bentuk tiga (3) lapisan iaitu sistem, seni bina dan penterjemah. Komunikasi antara sistem dan seni bina dihubungkan oleh medium penterjemah yang akan memetakan permintaan yang dihantar oleh sistem kepada lapisan seni bina. Konsep ini membenarkan sistem akses dan menggunakan komponen di lapisan seni bina dan menyesuaikan mengikut keperluan di lapisan sistem. Enjin adaptasi akan menentukan proses adaptasi yang dibenarkan bagi mengelakkan berlaku pelanggaran peraturan yang dibenarkan oleh lapisan seni bina. Kajian Garlan ini dapat digunakan kerana seni bina yang dihasilkan berstruktur dan komponen yang digunakan bersesuaian dengan peranan dalam model penyelenggaraan kueri yang dicadangkan.



Rajah 2.1: Model adaptasi sendiri Penggunaan Semula Infrastruktur (Garlan et al., 2004)

b) Model Perkongsian Mobile Kod (Arden et al., 2012)

Kajian yang dilaksanakan oleh (Arden et al., 2012) pula menerangkan bagaimana kod kueri dibina dan disimpan ke dalam storan khusus. Pihak pembangun akan menyimpan kod kueri yang dibina dan menggunakan label kod khusus bagi setiap kod kueri bagi tujuan pengenalan dan memudahkan proses pengenalpastian serta pemeriksaan terhadap kod yang didaftar bagi tujuan simpanan dan akses oleh pengguna lain. Model ini menyimpan kod kueri ke dalam pelbagai class object bagi tujuan keselamatan dan kemudahan capaian dan penyelenggaraan. Hanya kod kueri yang dibenarkankan sahaja akan ditarik melalui satu medium dinamakan new class object. Konsep ini dapat mengelakkan berlaku kesesakan dan sekaligus menekankan aspek perlindungan keselamatan kod kueri.

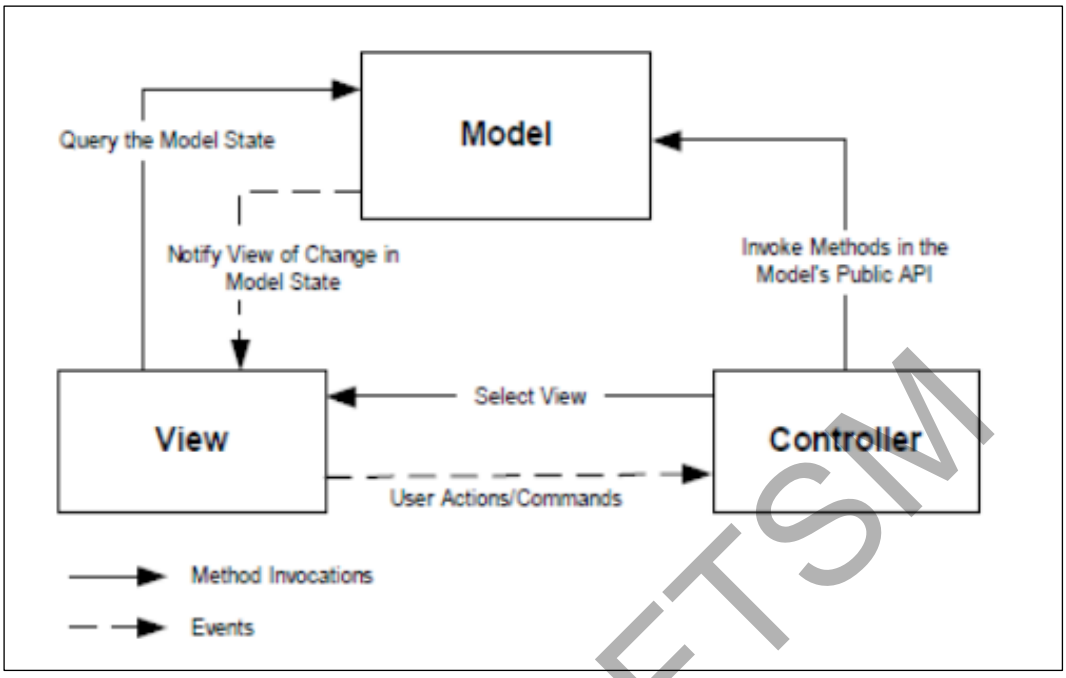


Rajah 2.2 Model Perkongsian Mobile Kod (Arden et al., 2012)

c) Model *Model-View-Controller* (MVC) (Uyun & Rifqi, 2010)

(Uyun & Rifqi, 2010) telah menjalankan kajian yang menggunakan konsep Model-View-Controller (MVC) dalam pembangunan sistem informasi berasaskan web. Kajian ini dilakukan untuk menerangkan bagaimana teori MVC ini berfungsi dan manfaat yang boleh diperolehi daripada penggunaannya. MVC adalah konsep yang diperkenalkan oleh (Reenskaug, 2003). Ia menggunakan pendekatan berperingkat yang mengasingkan kod dalam fungsi untuk kelas yang berbeza.

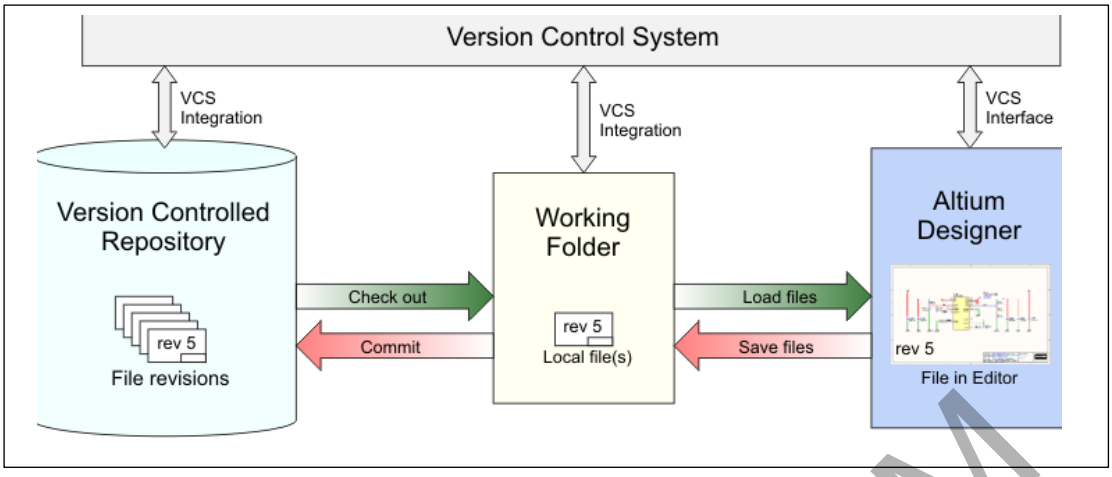
Pengkaji ini telah mengenalpasti kebaikan penggunaan konsep MVC ini kerana ia memudahkan pembangunan aplikasi, lebih modular, mudah diselenggara, dapat digunakan semula (SATHISH C.G., 2004) turut bersetuju dengan kelebihan konsep MVC ini dari aspek penggunaan semula kod. Selain itu, konsep MVC inidapat meningkatkan keupayaan aplikasi untuk berkembang. Ini dapat dibuktikan sekiranya aplikasi yang dibangunkan mempunyai capaian yang perlahan kepada pangkalan data, maka kita hanya perlu membuat peningkatan perkakasan yang terlibat dengan pangkalan data sahaja tanpa perlu melibatkan perkakasan lain. Ini kerana, kelebihan menggunakan konsep MVC yang menyusun setiap komponen secara berasingan dan tidak bergantung antara satu dengan yang lain untuk melaksanakan pengoperasiannya.



Rajah 2.3 *Model-View-Controller* (MVC) (Uyun & Rifqi, 2010)

d) Konsep *Concurrent Versioning System* (CVS) (Margaret Rouse, n.d.)

Kajian ini turut mengkaji berkenaan konsep *Concurrent Versioning System* (CVS) yang popular digunakan dalam sistem pengoperasian Linux dan Unix seperti yang dinukil oleh penulis (Margaret Rouse, n.d.). Konsep diperkenalkan bertujuan untuk membenarkan pengaturcara menyimpan dan mencapai semula pengekodan yang dihasilkan dalam versi yang pelbagai ke dalam repositori. Capaian tidak terhad kepada seorang pengaturcara sahaja, malahan konsep ini membenarkan ahli kumpulan untuk sama-sama berkongsi menggunakan pengekodan yang dihasilkan dan berupaya untuk mengawal versi fail yang dikemaskini dan mengalami perubahan di dalam repositori tersebut. CVS berperanan dengan membenarkan hanya satu fail asal sahaja yang dikemaskini dan sebarang pegemaskinian atau perubahan fail oleh pengguna lain perlu melalui proses "commit" bagi memberi notifikasi kepada pengguna lain tentang berlakunya pengemaskinian atau perubahan. CVS biasaya digunakan bersama program yang dinamakan *Revision Control System* (RCS) yang berperanan melakukan proses pengurusan versi yang dilakukan. RCS akan menjalankan fungsi merekodkan sebarang perubahan yang dilakukan oleh pengaturcara dan menyimpan fail tersebut di dalam repositori. Fungsi konsep CVS juga hampir sama dengan konsep *Version Control System.*



Rajah 2.3 Konsep *Concurrent Versioning System* (CVS) (Margaret Rouse, n.d.)

Hasil dapatan daripada kajian literatur yang diterangkan seperti di atas, dapatlah disimpulkan bahawa proses penyimpanan dan perkongsian kueri merupakan suatu cara yang ideal dan praktikal untuk diaplikasikan di pelbagai platform, teknologi dan bahasa pengaturcaraan. Pengimplementasiannya boleh diadaptasi melalui kombinasi idea daripada model dan konsep *Concurrent Versioning System* (CVS), *Model-View-Controller* (MVC) dan yang digunakan oleh pengkaji terdahulu yang telah terbukti keberkesanannya.

3. CADANGAN MODEL

3.1 Model Seni bina Penyimpanan Kueri

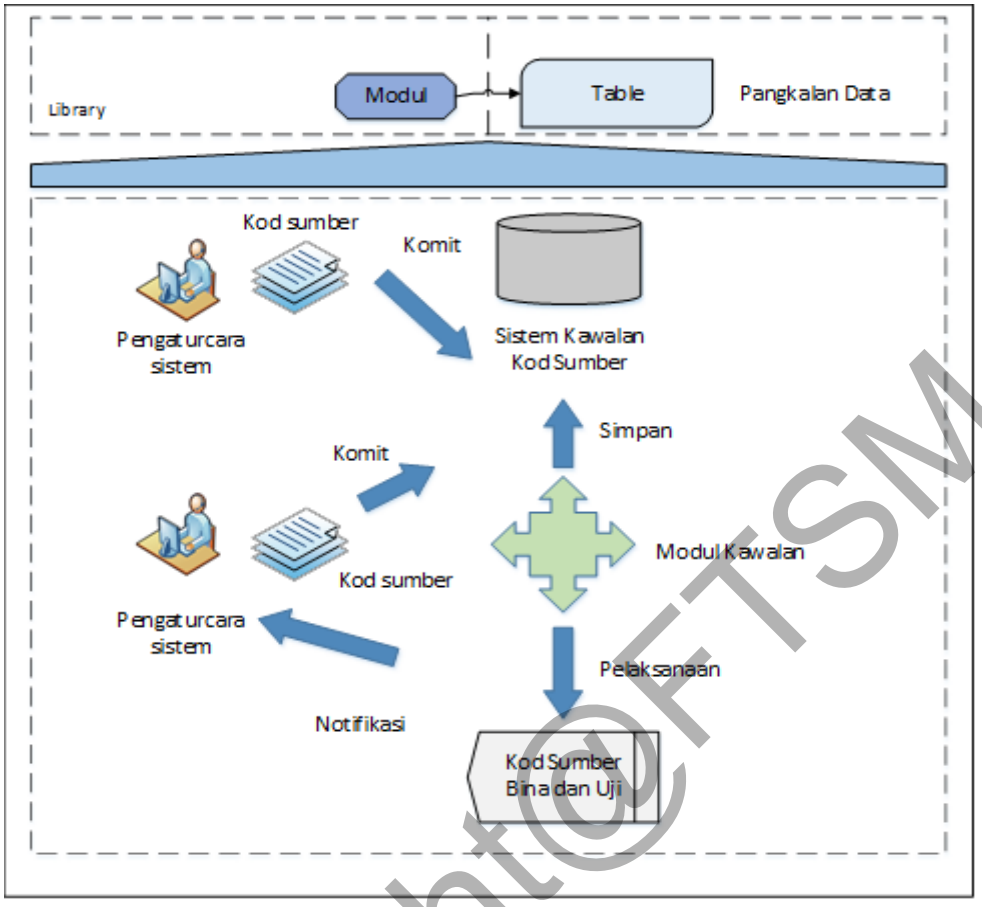
a. Penyimpanan Kueri

Model seni bina dalam Rajah 3.1 menunjukkan cadangan proses penyimpanan kueri yang boleh digunakan dalam perancangan pembinaan kueri di JPA. Seni bina ini diadaptasi daripada reka bentuk yang dihasilkan oleh (Leggett, n.d.) yang menerangkan bagaimana mereka membina kaedah yang boleh dipercayai untuk membina perisian yang sedang dibangunkan, yang berlaku secara berterusan.

Model seni bina penyimpanan kueri yang dicadangkan ini mempunyai lima (5) komponen utama yang berperanan seperti beruikut:

Jadual 3.1: Komponen Seni Bina Penyimpanan Kueri

Komponen	Fungsi
Modul Controller	Ia berfungsi sebagai komponen yang membuat pengesahan terhadap sintaks yang diterima oleh sistem luar untuk mendapatkan akses terhadap modul yang menyimpan kueri yang dihasilkan.
Modul Builder	Modul *Builder* merupakan sub kepada Modul *Controller* di mana ia menjalankan fungsi mengenalpasti modul yang akan terlibat dan di rujuk berdasarkan parameter yang dihantar.
Modul	Ia berfungsi sebagai repositori yang menyimpan kueri yang telah dibangunkan dan ia bersifat dinamik dan boleh disesuaikan dengan persekitaran sistem. Modul ini tidak terhad menyimpan kueri, malahan fungsinya untuk menyimpan fungsi pengekodan.
Table	Dalam seni bina ini, table adalah komponen yang menyimpan data secara berstruktur untuk dirujuk oleh modul dan ia terletak dalam pangkalan data. Kemungkinan juga, dua modul akan merujuk kepada satu *table* yang sama atau satu modul merujuk kepada lebih satu *table* dalam pangkalan data yang sama.
Pangkalan Data	Merupakan komponen yang menyediakan kemudahan pembinaan *table* untuk menyimpan data.
Perkhidmatan/ Protokol lain	Memberi peranan capaian kepada sistem.



Rajah 3.1: Seni Bina Penyimpanan Kueri

Proses penyimpanan kueri yang dicadangkan menjalankan fungsinya seperti berikut:

i. Pengaturcara sistem membangunkan kod sumber kueri dalam persekitaran mereka menggunakan bahasa pengaturcaraan yang telah ditetapkan. Kod sumber kueri ini dibangunkan mengikut keperluan dan fungsi yang akan diguna pakai oleh pengguna lain. Setiap kod sumber kueri yang dibangunkan perlu dikomit sebelum dihantar kepada Sistem Kawalan Kod Sumber. Konsep CVS yang digunakan dalam kajian (Greene, Esterhuizen, & Fischer, 2017) telah diadaptasi di dalam proses ini di mana pengaturcara perlu komit kod atau kueri yang dibina sebelum boleh dihantar kepada Sistem Kawalan Kod Sumber. Ini bagi memastikan, hanya fail asal sahaja yang membuat pengemaskinian terhadap sebarang perubahan terhadap fail tersebut bagi mengelakkan duplikasi terhadap perubahan yang dilakukan.

ii. Sistem Kawalan Kod Sumber akan menerima kod sumber yang dihantar oleh pengaturcara sistem. Sistem Kawalan Kod Sumber akan menghantar kod sumber tersebut kepada Kod Sumber Bina Uji bagi tujuan pengujian dan verisfikasi. Kod Sumber Bina Uji akan melaksanakan pengujian terhadap kod sumber tersebut bagi memastikan kod seumber tersebut berfungsi dengan baik

dan mengeluarkan output yang tepat seperti yang diperlukan. Kod sumber yang berjaya akan dihantar semula kepada Sistem Kawalan Kod Sumber untuk di hantar kepada lapisan produksi yang menempatkan modul dan pangkalan data yang akan diakses oleh sistem lain.

iii. Bagi kod sumber yang tidak berjaya dan menghasilkan ralat, Modul Kawalan akan bertindak mengeluarkan notifikasi kepada pengaturcara sistem untuk memberitahu bahawa kod sumber yang dihasilkan mempunyai ralat dan memerlukan pembaikan. Fail kod sumber akan dihantar juga kepada pengaturcara untuk tindakan selanjutnya.

iv. Kod sumber yang berjaya akan di hubungkan kepada lapisan produksi untuk disimpan ke dalam platform produksi. Pengaturcara bertanggungjawab kepada kod sumber yang telah disimpan di dalam produksi.

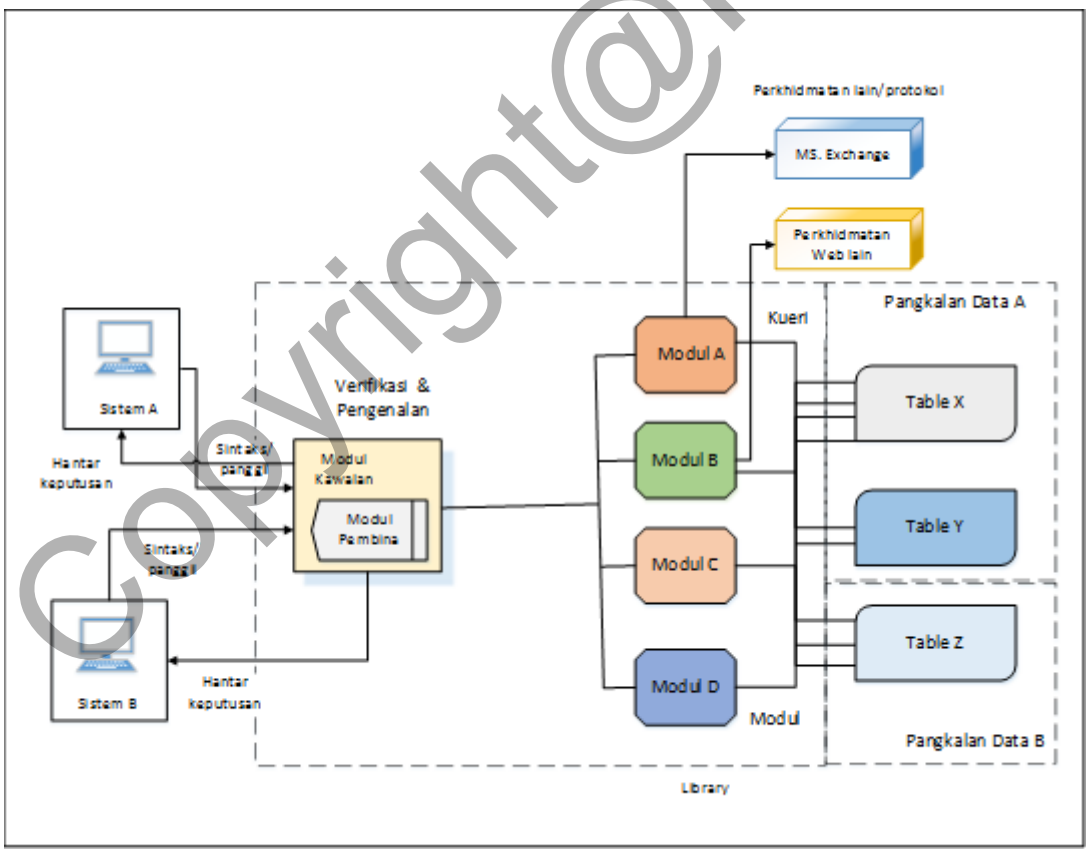
b. Perkongsian Kueri

Model seni bina penyimpanan kueri ini mencadangkan perkongsian kueri di antara pelbagai sistem dalam platform yang sama. Model ini mengambil adaptasi daripada kajian yang dijalankan oleh (Garlan et al., 2004) berkenaan penggunaan semula sumber bagi perisian dan kombinasi idea oleh (Arden et al., 2012) yang mengkaji tentang perkongsian kod mobil dalam domain yang berbeza serta (Uyun & Rifqi, 2010). Antara komponen yang terlibat dalam seni bina ini adalah:

Jadual 3.2: Komponen Seni bina Penyimpanan Kueri

Komponen	**Fungsi**
Modul Controller	Ia berfungsi sebagai komponen yang membuat pengesahan terhadap sintaks yang dihantar oleh sistem untuk mendapatkan akses terhadap kueri yang dibina.
Modul Builder	Modul *Builder* merupakan sub kepada Modul *Controller* di mana ia menjalankan fungsi mengenalpasti modul yang akan terlibat dan di rujuk berdasarkan parameter yang dihantar.
Modul	Ia berfungsi sebagai repositori yang menyimpan kueri yang telah dibangunkan dan ia bersifat dinamik dan boleh disesuaikan dengan persekitaran sistem.

Table	Dalam seni bina ini, *table* adalah komponen yang menyimpan data secara berstruktur untuk dirujuk oleh modul dan ia terletak dalam pangkalan data. Kemungkinan juga, dua modul akan merujuk kepada satu *table* yang sama atau satu modul merujuk kepada lebih satu *table* dalam pangkalan data yang sama.
Pangkalan data	Merupakan komponen yang menyediakan kemudahan pembinaan *table* untuk menyimpan data.
Perkhidmatan/ protokol lain	Memberi peranan capaian kepada sistem.



Rajah 3.2: Seni bina Perkongsian Kueri

4. KESIMPULAN

Secara amnya, penyimpanan dan perkongsian kueri merupakan suatu proses yang dicadangkan bagi membantu dalam proses pembangunan sistem. Model seni bina penyimpanan dan perkongsian kueri yang dicadangkan boleh diimplementasi di persekitaran di Jabatan Perkhidmatan Awam Malaysia bagi mengatasi masalah pembangunan sistem yang berpunca daripada perancangan pembangunan yang tidak tersusun dan tahap pengetahuan dan kemahiran yang tidak sama. Punca masalah ini telah mengakibatkan tempoh pembangunan sistem yang panjang dan sistem sukar untuk diselenggara.

Dalam bidang ilmu, model ini berupaya menjimatkan masa pembinaan kueri semula oleh pengaturcara sistem. Kueri yang disimpan boleh dicapai dan diubah mengikut keperluan semasa kerana ia mempunyai ciri-ciri yang dinamik.

Kajian yang telah dijalankan masih mempunyai ruang untuk diperkembangkan atau dilakukan penambahbaikan pada beberapa aspek tertentu. Antara perluasan kajian yang masih boleh dilakukan adalah seperti berikut:

a) Memfokuskan kepada keselamatan sistem rangkaian secara keseluruhan memandangkan kueri ini boleh dicapai oleh pelbagai sistem dan melibatkan capaian data dari pangkalan data.

b) Kajian juga perlu dilakukan dari aspek penyelenggaraan modul yang sentiasa berkembang dan meningkat bilangannya. Penggunaan kaedah yang lebih sesuai bagi memastikan modul disimpan dan dapat dicapai dengan lebih sistematik dan cepat.

RUJUKAN

Arden, O., George, M. D., Liu, J., Vikram, K., Askarov, A., & Myers, A. C. (2012). Sharing mobile code securely with information flow control. *Proceedings - IEEE Symposium on Security and Privacy*, 191–205. https://doi.org/10.1109/SP.2012.22

Garlan, D., Cheng, S.-W., Huang, A.-C., Schmerl, B., & Steenkiste, P. (2004). Rainbow: Architecture- Based Self-Adaptation with Reusable Infrastructure. *Computer*, 46–54.

Greene, G. J., Esterhuizen, M., & Fischer, B. (2017). Visualizing and exploring software version control repositories using interactive tag clouds over formal concept lattices. *Information and Software Technology*, *87*, 223–241. https://doi.org/10.1016/j.infsof.2016.12.001

Leggett, A. (n.d.). Continuous Integration- Basic Overview and Best Practices. Retrieved from http://blogs.collab.net/cloudforge/continuous-integration-overview-best-practice

Margaret Rouse. (n.d.). Concurrent Versions System (CVS). Retrieved from http://whatis.techtarget.com/definition/Concurrent-Versions-System-CVS

Reenskaug, T. (2003). The Model-View-Controller (MVC) Its Past and Present. *University of Oslo Draft*, (Mvc), 1–16. Retrieved from http://ci.nii.ac.jp/naid/10019460482/

SATHISH C.G. (2004). Architecture, Model View Controller (MVC). Retrieved from http://www.dotnetspider.com/resources/316-Model-View-Controller-MVCarchitecture.aspx

Uyun, S., & Rifqi, M. (2010). Implementation of Model View Controller (Mvc) Architecture on Building Web-Based Information System. *Islam Zeitschrift Für Geschichte Und Kultur Des Islamischen Orients*, *2010*(Snati), 47–50.