Pilihan judul tesis:

1. Pengembangan Kakas Visualisasi dari Graf Kode Program untuk Memahami Eksekusi kode program

* Pengembangan
* Kakas
* Visualisasi Program
* Graf 🡪 lebih fokus lagi pada algoritma tertentu, misalnya khusus deteksi algoritma dijkstra, BFS/DFS (atau bisa ditulis dalam batasan)
* *D3*
* SVG
* Web-Based
* Deteksi algoritma

“Saya mengembangkan kakas ini adalah untuk mengembangkan fitur alat pengajar pemrograman struktur data. Sebagaimana diciptakan papan tulis atau proyektor untuk membantu dalam belajar mengajar. Adanya papan tulis atau proyektor lantas tidak membuat pelajar menjadi pintar, akan tetapi informasi materi yang ingin disampaikan oleh pengajar menjadi lebih mudah dengan perantara papan tulis atau proyektor. Demikian pula dengan kakas ini, informasi yang divisualkan adalah untuk membantu agar mudah tersampaikan kepada pelajar. Bagaimana informasi itu dapat bermanfaat bagi pelajar, itu tergantung bagaimana pengajar menyampaikannya. Sama halnya, apakah teknisi proyektor mengerti bagaimana cara mengajar yang baik dengan proyektor yang diciptakannya? Tentu tidak, karena ia hanya fokus bagaimana membuat fitur-fitur yang canggih dalam sebuah alat proyektor tersebut. Oleh karena itu, saya fokus bagaimana mengembangkan kakas ini dapat mendukung visualisasi struktur data graf dengan secara otomatis mendeteksi adanya graf dalam sebuah kode program.”

Latar Belakang : Peserta didik kesulitan memahami eksekusi graf kode program, belum ada kakas berbasis web untuk graf VP yang mudah digunakan tanpa repot dengan plugin seperti Java Applet dsb.

Metode Pengembangan : Evaluasi heuristik, kuesioner online

Metode Pengujian : Evaluasi keefektifan visualisasi graf

[Kerangka penulisan laporan tesis] – Habibie Ed Dien | 23515043 | TMPB

**Bab I - Pendahuluan**

**I.1 Latar Belakang**

|  |  |
| --- | --- |
| Alinea 1 | Belajar pemrograman berkaitan erat dengan eksekusi kode program |
| Alinea 2 | Hambatan pengajar dalam menjelaskan materi struktur data graf |
| Alinea 3 | Kesulitan dalam struktur data graf (Piteira dan Costa, 2013) |
| Alinea 4 |  |
| Alinea 5 |  |
| Alinea 6 | Kesimpulan terhadap penelitian tesis yang akan dilakukan. |

Alasan lain pengembangan kakas ini adalah karena telah berkembang pesat model belajar secara *online* atau yang saat ini disebut MOOC (*Massive Open Online Course*). Para pengguna internet dapat belajar dari jarak jauh melalui portal web yang disediakan oleh lembaga atau komunitas tertentu. Adanya kakas ini diharapkan dapat membantu sebagai media belajar pemrograman jarak jauh.

Berbagai riset dalam dekade tahun terakhir para peneliti mencari cara untuk mengembangkan kakas guna memahami alur kerja sebuah program. Visualisasi menjadi salah satu teknik yang telah dilakukan, sehingga saat ini muncul istilah *software visualization* (visualisasi perangkat lunak). Visualisasi adalah teknik interaksi terpenting dalam penyampaian informasi (Cetin dan Andrews-Larson, 2016; Sorva, 2012; Sorva dkk., 2013; Gračanin dkk., 2005; Guo, 2013). Melalui indera visual, manusia lebih cenderung menangkap lebih banyak informasi yang diterima dibandingkan melalui indera lainnya (Ware, 2004).

Istilah visualisasi dalam perangkat lunak (Cetin dan Andrews-Larson, 2016) didefinisikan sebagai sebuah seni tipografi, desain grafis, animasi, dan sinematografi melalui interaksi modern antar manusia-komputer. Visualisasi perangkat lunak (*software visualization*/ SV) dibagi menjadi dua bagian besar (Cetin dan Andrews-Larson, 2016; Sorva, 2012; Sorva dkk., 2013), yaitu visualisasi algoritma (*algorithm visualization*/ VA) dan visualisasi program (*program visualization*/ VP). VA berkaitan dengan abstraksi algoritma, sedangkan VP berkaitan dengan eksekusi kode program dan proses struktur data.

Visualisasi adalah proses mengubah data menjadi bentuk visual yang memungkinkan pengguna untuk mengamati data tersebut (Diehl, 2007). Fitur yang tersembunyi di dalam sebuah data memungkinkan untuk ditampilkan secara visual. Namun demikian, eksplorasi dan analisis data masih tetap harus dilakukan terhadap visualisasi tersebut. Saat ini komputer telah menjadi salah satu kakas penting untuk dapat melakukan visualisasi, sehingga membantu pengguna untuk memahami fenomena yang kompleks. Bahkan bidang visualisasi telah menjadi salah satu disiplin ilmu dalam *computer science*.

Visual adalah salah satu subjek penting dalam dunia komputer.

Masalahnya representasi visual graf yang ada belum memenuhi syarat sebagai visual graf sebenarnya.

Visualisasi tidak dapat diukur secara kuantitatif. Metode yang digunakan berdasarkan pendekatan pengalaman interaksi manusia.

Efisien berarti informasi dapat dipahami dengan mudah dan cepat oleh penerimanya. Efektif berarti makna yang terkandung di dalam informasi dapat dipersepsi dengan benar, sehingga tujuan dari penyampaian informasi tersebut dapat tercapai.

Fakta, ide, informasi, angka, statistik, cerita, anekdot, frasa kunci, kutipan, gagasan:

* Kegiatan belajar pemrograman tidak lepas dari eksekusi kode sumber
* Visualisasi sebagai salah satu media belajar pemrograman
* Graf adalah salah satu struktur data
* Perkembangan kakas visualisasi eksekusi kode berbasis web masih jarang

Aku ingin menjelaskan di latar belakang sebagai berikut: dari menceritakan proses kegiatan belajar pemrograman, kesulitan yang ditemukan, hambatan dalam proses mengajar pemrograman bagi seorang pengajar, visualisasi sebagai salah satu media belajar mengajar, kakas yang telah mendukung visualisasi dalam proses belajar mengajar pemrograman, salah satu kesulitan teori dalam pemrograman adalah struktur data graf, kekurangan fitur kakas yang telah ada, dan harapan yang ingin dicapai dalam penelitian ini.

Pertanyaan yang harus dijawab dari judul:

* Mengapa pengembangan kakas ?
* Mengapa visualisasi eksekusi kode untuk graf ?
* Mengapa sebagai media belajar pemrograman ?

Berikut **draft**, nanti masih mau diperbaiki kosa kata dan susunan kalimatnya:

Proses belajar pemrograman tidak lepas dari melakukan eksekusi kode. Belajar pemrograman bagi sebagian pelajar tidak mudah. Selain pemahaman tentang algoritma, kegiatan mengonstruksi sebuah program adalah bagian penting untuk implementasi dari algoritma yang telah dirancang.

Bagi pengajar untuk menjelaskan proses eksekusi kode di kelas terkadang menggunakan papan tulis atau *slide PowerPoint*. Hal ini membutuhkan persiapan ekstra terhadap materi yang perlu disiapkan terkait gambar, alur diagram, atau bagian-bagian kode program yang perlu dijelaskan. Terutama jika materi pemrograman itu memasuki ke tingkat yang lebih rumit, seperti struktur data. Diperlukan media belajar khusus untuk dapat menjelaskan proses eksekusi kode itu. Sehingga pelajar dapat memahami proses yang sebenarnya terjadi di dalam program komputer.

Sebuah penelitian yang telah dilakukan oleh Piteira dan Costa (2013) di Institut Politeknik Setubal, Portugal, menemukan bahwa konsep pemrograman struktur data memiliki tingkat kesulitan yang tinggi bagi sebagian besar pelajar. Hal itu disebabkan konsep pemrograman untuk struktur data merupakan konsep data abstrak yang tidak tampak jelas dalam kode. Penelitian tersebut juga memberikan perhatian khusus terhadap konsep pemrograman seperti pointer dan reference, parameter, abstract data types, error handling dan penggunaan bahasa dalam library.

Graf adalah salah satu struktur data ...

Visualisasi sebagai media belajar pemrograman bukan suatu hal yang baru. Para peneliti telah banyak mengembangkan kakas visualisasi untuk membantu mempelajari algoritma dan pemrograman (Sorva, 2013). Visualisasi merupakan bagian dari GUI, yaitu untuk interaksi antar manusia dengan mesin atau komputer. Karena melalui interaksi visual, manusia lebih cenderung menangkap lebih banyak informasi yang diterima dibandingkan melalui indera lainnya (Ware, 2004). Visualisasi dapat mendukung interaksi yang efisien dan efektif untuk beragam pekerjaan kognitif seperti menganalisis, meringkas, dan menarik kesimpulan atas informasi yang diperoleh.

Pada penelitian yang telah dilakukan oleh Philip Guo (2010), kakas visualisasi eksekusi kode telah dibangun bernama Online Python Tutor (OPT). Kakas berbasis web ini memiliki fitur *embeddable* yang mudah digunakan. Fitur tersebut digunakan untuk melampirkan visualisasi di halaman web lain dengan tujuan tertentu. Kakas ini dapat memvisualisasikan operasi kode program yang sebenarnya terjadi di dalam komputer.

**I.2 Rumusan Masalah** *(apa masalahnya)*

1. Apa strategi pengembangan kakas untuk mendeteksi adanya graf dalam kode sumber sehingga dapat divisualisasikan ?
2. Apa yang harus dilakukan untuk meningkatkan persepsi manusia dalam representasi visual graf ?

**I.3 Tujuan Penelitian** *(Tujuan penyelesaian masalah, bukan manfaat)*

Menghasilkan kakas yang dapat melakukan visualisasi graf dari eksekusi kode.

**I.4 Batasan Masalah**

1. Keefektifan visualisasi graf
2. Usability

**I.5 Metodologi Penelitian**

1. Tinjauan Pustaka dan Eksplorasi

2. Merumuskan Hipotesis

3. Pengumpulan Graf Kode Program

4. Perumusan Model Graf

5. Pengembangan Kakas

6. Evaluasi

**I.6 Sistematika Penulisan**

Bab I Pendahuluan

Bab II Tinjauan Pustaka dan Eksplorasi

Bab III Analisis Masalah dan Perumusan Model Graf

Bab IV Pengembangan Kakas

Bab V Pengujian dan Evaluasi Kakas

Bab VI Penutup

**Bab II - Tinjauan Pustaka dan Eksplorasi**

**II.1 Terminologi Visualisasi Perangkat Lunak**

**II.2 Revolusi Teknologi Web dan Pustaka untuk Visualisasi**

**II.3 Graf dan Model Visualnya**

**II.4 Hasil Eksplorasi Kakas Visualisasi Program**

**II.5 Eksplorasi Kakas *Online Python Tutor***

**II.5.1 Arsitektur Kakas OPT**

**II.5.2 Komponen Kakas OPT**

**II.5.3 *Capturing Execution Trace***

**II.5.4 *Execution Trace Format***

**II.5.5 Fitur Pustaka *Data-Driven Documents* (D3)**

**II.7 Kesimpulan Awal Berdasarkan Studi Literatur dan Eksplorasi**

**Bab III Analisis Masalah**

**III.1 Analisis Tujuan Visualisasi Graf**

Visualization is the representation of an object, situation, or set of information as a chart or other image; The formation of a mental image of something (en.oxforddictionary.com).

Visual information is first processed by large arrays of neurons in the eye and in the primary visual cortex at the back of the brain. Individual neurons are selectively tuned to certain kinds of information, such as the orientation of edges or the color of a patch of light. It is also rapid. If we want people to understand information quickly, we should present it in such a way that it could easily be detected by these large, fast computational systems in the brain. (Ware p.20-21, 2004)

A computer program has two forms: static and dynamic. The static aspect of a program is visible in code, but the dynamic aspect is usually implicit. The hidden nature of program dynamics has been linked to the multitude of misconceptions about programming concepts that students have (Sorva p.61, 2012).

**Understand** (Sorva p.20, 2012) (Thompson dkk., p.3 2008) is defined as ‘constructing meaning from instructional messages, including oral, written, and graphical communications’. In the revised taxonomy, this category includes Interpreting, Exemplifying, Classifying, Summarising, Inferring, Comparing, and Explaining. We interpret this in programming assessment terms to mean:

1. translating an algorithm from one form of representation to another form;
2. explaining a concept or an algorithm or design pattern;
3. presenting an example of concept or an algorithm or design pattern.

**III.2 Analisis Desain Visualisasi Graf**

Asumsi pengguna untuk mencapai tujuan visualisasi

the human eye allows color perception only in the center, no color is perceived in the periphery. The effectiveness of recognizing shapes decreases with the distance from the center owing to the decreasing density of receptors. Movements are perceived effectively in the periphery and usually people react by orienting their focus to the moving objects (Diehl, p.20 2007).

Thus, in the design of a visualization, color should be used for detail information, shape can be used for detail information about a single object but also be used to show the relation between different objects, and motion can be used as a stimulus in the periphery to attract attention and to establish links between remote objects (Diehl, p.20 2007).

In visualization, color is often used to represent values from a discrete or continuous numerical interval. In this case each value is mapped to a color – the value iscolor coded. The following color scales are widely used, because they are very intuitive:

Linear optimal color scales are used when it is important that the perceived difference between colors is proportional to the distance between the values they encode. Whenever possible, software visualization tools should allow the user to choose among different predefined color scales.

Eye-tracking experiments have shown that distribution of the spontaneous attention of a person looking at a screen is not evenly distributed. If we divide the screen into two upper and two lower areas, then the left upper part receives 40%, the right upper part 20%, the left lower part 25%, and

the right lower part only 15% of the user’s attention. Accordingly, important information should be placed in the areas of high attention, and less important information in those of low attention. Like many psychological results, this one depends heavily depends on culture (Diehl, p.20 2007).

**III.3 Analisis Deteksi Graf dalam Kode Program**

**III.3.1 Representasi Graf dalam Kode Program**

**III.3.2 Analisis Kakas Pendukung untuk Deteksi Data Graf**

**III.3.3 Analisis Metode Deteksi Graf**

**III.3.4 Analisis Kakas Pendukung untuk Visualisasi Graf**

**III.4 Penunjang Pengembangan Kakas**

**III.4.1 Kebutuhan Perangkat Lunak**

**III.4.2 Kebutuhan Perangkat Keras**

**Bab IV Pengembangan Kakas**

**IV.1 Proses Visualisasi Data**

**IV.2 Diagram Alur Sistem**

**IV.3 Implementasi Kakas**

**Bab V Evaluasi Visualisasi**

**V.1 Survei pengguna dengan Kuesioner Online**

As stated above, PAV systems are interactive systems, therefore usability is a key property of those systems (Urquiza-Fuentes and Velázquez-Iturbide, p.3 2009).

**V.2 Analisis Hasil Kuesioner**

**Bab VI Kesimpulan dan Saran**

**VI.1 Kesimpulan**

**VI.2 Saran**

**Daftar Pustaka**

Cetin, I. and Andrews-Larson, C. (2016): Learning sorting algorithms through visualization construction, *Comput. Sci. Educ.*, **26**, 27–43.

Diehl, S. (2007): *Software visualization: visualizing the structure, behaviour, and evolution of software ; with 5 tables*, Berlin, Springer.

Gračanin, D., Matković, K. and Eltoweissy, M. (2005): Software visualization, *Innov. Syst. Softw. Eng.*, **1**, 221–230.

Guo, P.J. (2013): Online python tutor: embeddable web-based program visualization for cs education, *Proceeding of the 44th ACM technical symposium on Computer science education*, ACM, 579–584, diperoleh melalui situs internet: http://dl.acm.org/citation.cfm?id=2445368 (accessed 20 July 2016).

Sorva, J. (2012): *Visual program simulation in introductory programming education*, Aalto University publication series Doctoral dissertations, Espoo, Aalto Univ. School of Science.

Sorva, J., Karavirta, V. and Malmi, L. (2013): A review of generic program visualization systems for introductory programming education, *ACM Trans. Comput. Educ. TOCE*, **13**, 15.

Thompson, E., Luxton-Reilly, A., Whalley, J.L., Hu, M. and Robbins, P. (2008): Bloom’s Taxonomy for CS Assessment, ACE ’08, Wollongong, NSW, Australia, Australian Computer Society, Inc., 155–161, diperoleh melalui situs internet: https://dl.acm.org/citation.cfm?id=1379265.

Urquiza-Fuentes, J. and Velázquez-Iturbide, J.Á. (2009): A Survey of Successful Evaluations of Program Visualization and Algorithm Animation Systems, *ACM Trans. Comput. Educ.*, **9**, 1–21.

Ware, C. (2004): *Information visualization: perception for design, 2nd edition*, 2nd edition, San Francisco, Kanada, Elsevier Inc.