

PENERAPAN MULTITHREAD PADA ALGORITMA PENCARIAN SEQUENTIAL FILE DOKUMEN (.DOC)

Cahya Wijayadi¹, Soebandi², Thommy Willay³

^{1,2,3}Teknik Informatika, STMIK Widyadharma, Pontianak

email: ¹wijayadicaHYa@gmail.com, ²soebandi@gmail.com, ³w.thommy@gmail.com

Abstract

Each person would have known and did not hear the familiar words search. Search has the meaning of an action taken to find something important from a set of objects or information. Search results are then used and recycled into a required information. Various search algorithms too common and generally we know two kinds of simple search algorithm is sequential and binary. The author would like to develop a sequential search algorithms with multithreading programming techniques imbued with the hope of a long search process can be shortened. The author uses a study design causal relationship (experimental) is the author of collecting data by studying the literature relating to the search algorithm and multithreading. The author uses modeling Unified Modeling Language (UML) to model the application as an analytical technique. In designing these applications using Microsoft Visual basic.NET author. Applications resulting from this research is the application of search words have been inserted sequential search algorithm and multithreading and efficiency of search speed after using multithreading. From the whole process can be concluded that the application of search words to locate words and as a comparison the efficiency of search in general. The suggestions addressed to the reader is to develop search applications with multithreading said to be better and could be useful as a medium for finding a good word.

Keywords : Multithread, Searches Algorithms, Sequential Search, Documents.

Abstrak

Setiap orang tentu sudah mengenal dan tidak asing lagi mendengar kata pencarian. Pencarian mempunyai makna sebuah tindakan yang dilakukan untuk menemukan sesuatu hal yang penting dari sekumpulan objek atau informasi. Hasil pencarian kemudian digunakan dan diolah kembali menjadi sebuah informasi yang dibutuhkan. Berbagai algoritma pencarian pun banyak ditemukan dan umumnya kita mengenal 2 macam algoritma pencarian sederhana yaitu sekuensial dan binary. Penulis ingin mengembangkan algoritma pencarian sekuensial dengan membubuhi teknik pemrograman *multithreading* dengan harapan lama proses pencarian dapat dipersingkat. Penulis menggunakan desain penelitian hubungan kausal (eksperimental) yaitu penulis melakukan pengumpulan data dengan cara mempelajari literatur-literatur yang berhubungan dengan algoritma pencarian dan *multithreading*. Penulis menggunakan pemodelan *Unified Modelling Language* (UML) untuk memodelkan aplikasi sebagai teknik analisis. Dalam merancang aplikasi tersebut penulis menggunakan *Microsoft visual basic.NET*. Aplikasi yang dihasilkan dari penelitian ini adalah aplikasi pencarian kata yang telah disisipkan algoritma pencarian sekuensial dan *multithreading* dan efisiensi kecepatan pencarian setelah menggunakan *multithreading*. Dari keseluruhan proses penelitian dapat disimpulkan bahwa aplikasi pencarian kata untuk mengetahui letak kata dan sebagai pembanding efisiensi pencarian pada umumnya. Adapun saran yang ditujukan kepada pembaca adalah untuk mengembangkan aplikasi pencarian kata dengan *multithreading* menjadi lebih baik dan dapat berguna sebagai media untuk pencarian kata yang baik.

Kata Kunci : Multithread, Algoritma Pencarian Sequential, Dokumen

1. PENDAHULUAN

Setiap orang tentu sudah mengenal dan tidak asing lagi mendengar kata pencarian. Pencarian mempunyai makna sebuah tindakan yang dilakukan untuk menemukan sesuatu hal yang penting dari sekumpulan objek atau informasi. Hasil pencarian kemudian digunakan dan diolah kembali menjadi sebuah informasi yang dibutuhkan. Dalam menghasilkan informasi yang cepat dan akurat diperlukan algoritma pencarian yang dapat diandalkan. Beberapa metode tersebut yang umumnya kita kenal ialah pencarian terurut dan biner. Pada metode-metode tersebut memiliki kelebihan dan kelemahan, terutama pada pencarian terurut, dimana dapat diketahui pencarian terurut memerlukan waktu lama jika posisi data terlalu jauh. Teknik *multithread* dapat digunakan sebagai solusi permasalahan di atas. *Multithread* merupakan sebuah teknik yang digunakan untuk membuat sebuah aplikasi lebih responsif. Karena *multithread* dapat mengurai proses menjadi sub proses yang bersifat independen sehingga memungkinkan proses memiliki banyak sub proses, kelebihan

ini yang digunakan dalam mempercepat proses. Penggunaan metode pencarian terurut membutuhkan sedikitnya percepatan pencarian. Penerapan *multithread* pada penulisan ini akan difokuskan bagaimana teknik *multithread* dapat mempercepat proses pencarian dari proses pencarian biasanya. Model *multithread* yang digunakan ialah model *many to one* dan *many to many*. Model ini dipilih karena diharapkan pencarian data dapat dilakukan dengan waktu yang lebih cepat di banding dengan biasanya.

Berdasarkan atas informasi diatas, penulis membuat sebuah implementasi dengan menerapkan teknik *multithread* pada metode pencarian terurut dalam aplikasi program yang dibentuk kedalam tugas akhir untuk menyelesaikan studi program sarjana strata satu (S1).

2. METODE PENELITIAN

2.1 Metode penelitian yang dilakukan adalah dengan cara:

2.1.1 Rancangan Penelitian

Dalam penyusunan penelitian ini, penulis menggunakan desain penelitian hubungan kausal (*eksperimental*), yaitu penulis melakukan percobaan dan pengujian terhadap aplikasi *multithread* pada pencarian *sequential* yang dibuat dan dengan cara mempelajari literatur maupun tulisan yang ada hubungannya dengan masalah yang diteliti.

2.1.2 Metode Pengumpulan Data

Metode pengumpulan data untuk penelitian ini dilakukan dengan metode studi pustaka yaitu membaca dan mempelajari literatur-literatur yang berhubungan dengan permasalahan yang diangkat

2.1.3 Teknik Analisis Sistem

Teknik analisis sistem yang digunakan penulis adalah menggunakan *Unified Modeling Language* (UML) untuk menggambarkan sistem yang sedang diterapkan pada aplikasi pencarian kata dan memperoleh masalah-masalah dan kesulitan-kesulitan yang dihadapi.

2.1.4 Teknik Perancangan Sistem.

Teknik perancangan aplikasi yang digunakan adalah dengan teknik pemrograman prosedur dan dengan dijabarkan diagram *Unified Modelling Language* (UML), selanjutnya penulis melakukan perancangan aplikasi *multithread* pada pencarian *sequential* dengan menggunakan VB.Net dan *Microsoft Word 2007* sebagai *Resource*.

2.2 Landasan Teori

2.2.1 Pengertian Thread

A Thread is code to execute in the processor. A Thread belongs to a process and is called the dynamic part of a process atau sebuah *thread* adalah kode yang dieksekusi di dalam prosesor. Sebuah *thread* berasal dari proses dan disebut bagian dinamis dari sebuah proses [1]. *Thread* adalah sebuah alur control dari sebuah proses [2].

2.2.2 Pengertian multithread

While a process with multiple threads is called multithreaded process atau sementara proses dengan beberapa *thread* yang disebut proses *multithreaded* [1]. *Multithread* memungkinkan proses untuk menjalankan lebih dari satu tugas pada satu waktu.[2]

2.2.3 Pengertian synchronization

Synchronization is the most common tool used to create a thread-safe environment. Even single-threaded applications use synchronization on occasion atau sinkronisasi ialah alat yang paling umum digunakan untuk menciptakan lingkungan *thread*-aman. Bahkan aplikasi *single-threaded* menggunakan sinkronisasi pada kesempatan tertentu [1]. *There are often multiple threads simultaneously running on a computer. It may then be necessary to synchronize the threads to avoid collisions* atau sering ada beberapa *thread* secara bersamaan berjalan pada komputer. Kemudian mungkin diperlukan untuk menyinkronkan *thread* untuk menghindari tabrakan [1].

2.2.4 Hubungan thread dan synchronization

There is a need for synchronization of thread in two situation:Thread can cooperate in various situation. For example, a thread can be responsible to perform a task for another thread;A competition between threads occurs want to use the the same resource at the same time. An example of this is two threads that write to the screen simultaneously, which could lead to collisions and messy prints atau Ada kebutuhan untuk sinkronisasi *thread* dalam dua situasi:Thread bekerja sama. Thread dapat bekerja sama dalam berbagai situasi. Misalnya, sebuah *thread* dapat bertanggung jawab untuk melakukan tugas untuk *thread* lain.Thread bersaing untuk sumber daya . Sebuah kompetisi terjadi antara *thread* yang ingin menggunakan sumber daya yang sama pada waktu yang sama. Contohnya adalah dua *thread* yang menulis ke layar secara bersamaan, yang dapat menyebabkan tabrakan dan hasil cetak menjadi berantakan [1].

2.2.5 Metode *synchronization*

The term critical region is important in connection to synchronization of threads. A critical region is a place where collisions may occur; Mutual exclusion is an important concept in connection to synchronization. That thread can get mutual exclusion to a resource is necessary for a computer system to function properly; One way to control access to a resource is to use a semaphore. A semaphore may be described as an integer, and can have value 0,1,2..., n is a number that indicates how many threads are allowed to access resource at the same time Wilayah Kritis Istilah wilayah kritis merupakan hal penting sehubungan dengan sinkronisasi *thread*. Sebuah daerah kritis adalah tempat di mana tabrakan dapat terjadi. *Mutual Exclusion*. Mutual exclusion merupakan konsep penting dalam kaitannya untuk sinkronisasi. *Thread* yang bisa saling *mutual exclusion* untuk sumber daya yang diperlukan untuk sebuah sistem komputer berfungsi dengan baik. *Semaphore*. Salah satu cara untuk mengontrol akses ke sumber daya adalah dengan menggunakan *semaphore*. Sebuah *semaphore* dapat digambarkan sebagai integer, dan dapat memiliki nilai 0,1,2 ..., n adalah angka yang menunjukkan berapa banyak benang yang diizinkan untuk mengakses sumber daya pada saat yang sama [1]. Jika kita menjalankan program pada sistem tanpa *multithread*, maka seluruh waktu didedikasikan untuk proses utama [13].

2.2.6 Keuntungan Program yang Menggunakan *Multithreading*

Keuntungan dari program yang *multithreading* dapat dipisah menjadi empat kategori: Membuat aplikasi yang interaktif menjadi *multithreading* dapat membuat sebuah program terus berjalan meskipun sebagian dari program tersebut diblok atau melakukan operasi yang panjang, karena itu dapat meningkatkan respon kepada pengguna; *Thread* berbagi memori dan sumber daya dengan thread lain yang dimiliki oleh proses yang sama. Keuntungan dari berbagi kode adalah mengizinkan sebuah aplikasi untuk mempunyai beberapa thread yang berbeda dalam lokasi memori yang sama; Dalam pembuatan sebuah proses banyak dibutuhkan pengalokasian memori dan sumber daya. Alternatifnya adalah dengan penggunaan *thread*, karena *thread* berbagi memori dan sumber daya proses yang memilikinya maka akan lebih ekonomis untuk membuat dan *context switch thread*; Keuntungan dari *multithreading* dapat sangat meningkat pada arsitektur *multiprocessor*, dimana setiap *thread* dapat berjalan secara parallel di atas *processor* yang berbeda [1].

2.2.7 *Sequential Search*

Pencarian *Sequential* merupakan salah satu metode pencarian yang paling umum digunakan dengan melakukan pencarian mulai awal himpunan rekaman dan bergerak melalui masing – masing rekaman hingga program pencarian menemukan rekaman yang dicari atau hingga mencapai ujung rekaman [3]. Pada dasarnya, algoritma pencarian beruntun adalah proses membandingkan setiap elemen larik satu per satu secara beruntun, mulai dari elemen pertama, samapai elemen yang dicari ditemukan atau seluruh elemen sudah diperiksa [4].

2.2.8 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak [5]. UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar [6].

2.2.9 *Microsoft Visual Basic .NET 2010*

Visual Basic 2008 adalah salah satu kelompok bahasa pemrograman yang dibuat oleh *Microsoft* dan tergabung dalam satu paket bahasa pemrograman *Microsoft Visual Studio 2008* [7]. *Visual Basic.Net* adalah salah satu dari kumpulan *tools* pemrograman yang terdapat pada paket *Visual Studio NET* [8].

2.2.10 *Microsoft SQL Server 2008*

Microsoft SQL Server adalah sebuah RDBMS (*Relational Database Management System*), yang digunakan untuk menyimpan dan mengolah data [9]. *Microsoft SQL Server* adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi basis data [10].

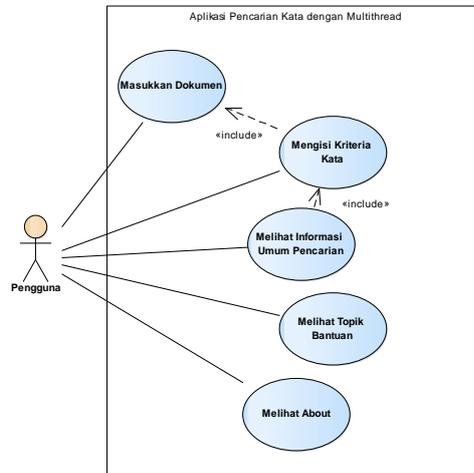
2.2.11 *Microsoft Word 2007*

Microroft Word 2010 terutama berfungsi untuk pengolahan kata, seperti surat, karya tulis, penelitian, tesis dan laporan [11]. *Word 2007* adalah program pengolah kata terbaru yang memiliki desain sangat berbeda dengan versi sebelumnya [12].

3. HASIL DAN PEMBAHASAN

3.1 Gambaran Umum Rancangan Perangkat Lunak Melalui *Use Case Diagram*

UML adalah metodologi untuk mengembangkan sistem OOP yang terdiri dari sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut dan merupakan bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan suatu sistem informasi. Berikut ini adalah diagram *use case* pada aplikasi pencarian kata dengan *multithread*. Gambar 1 menggambarkan diagram *use case* aplikasi pencarian kata dengan *multithread*. Penggunaan aplikasi terdiri atas satu aktor, yaitu aktor pengguna. Aktor pengguna dapat mengakses pada ruang lingkup sistem ini seperti dapat memasukkan dokumen, mengisi kriteria, melihat informasi umum pencarian, melihat topik bantuan dan tentang (*about*).



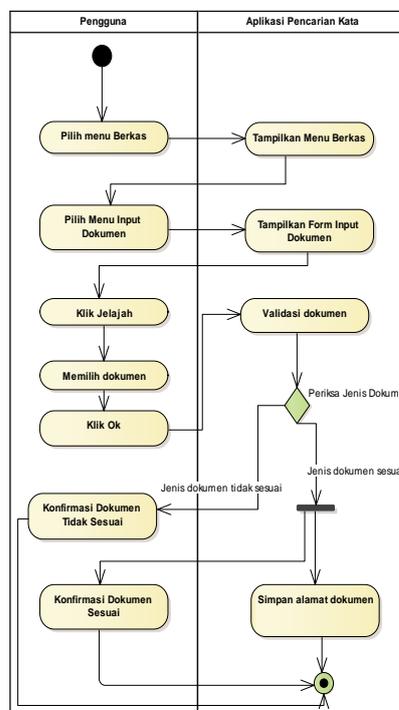
Gambar 1. Use Case Diagram Aplikasi Pencarian Kata Dengan *Multithread*

3.2 Gambar Umum Rancangan Perangkat Lunak Melalui Diagram Aktivitas

Activity diagram merupakan pemodelan untuk menggambarkan proses-proses yang terjadi pada saat aktivitas dimulai hingga aktivitas berhenti. Berikut adalah *activity diagram* yang digunakan dalam perancangan aplikasi:

1.1.1. Diagram Aktivitas Memasukkan Dokumen

Diagram Aktivitas menggambarkan prosedur yang terjadi di dalam aplikasi. Berikut ini adalah diagram aktivitas pada aplikasi pencarian kata dengan *multithread*:



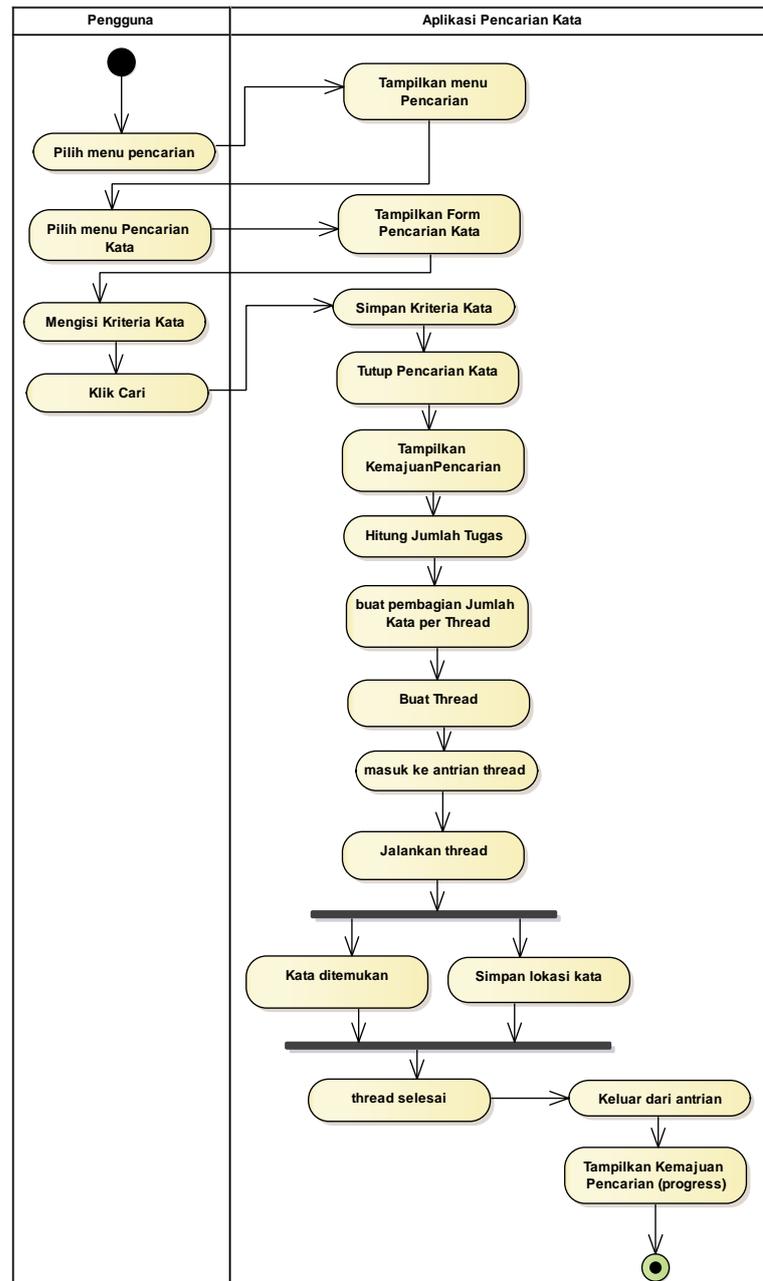
Gambar 2 Diagram Aktivitas Memasukkan Dokumen

Pada Gambar 2 dapat dijelaskan bahwa pada saat input dokumen, maka akan masuk ke menu utama. Setelah masuk ke menu utama maka akan memilih menu berkas dan memilih sub menu Input dokumen.

Kemudian pengguna menekan tombol jelajah untuk memilih dokumen. Dokumen yang diinputkan kemudian diperiksa jenisnya.

1.1.2. Activity Diagram Mengisi Kriteria Kata

Pada Gambar 3 dapat dijelaskan bahwa pada saat memasukkan kriteria kata yang akan dicari maka menu utama akan ditampilkan. Setelah masuk ke menu utama maka akan memilih menu pencarian dan memilih sub menu pencarian kata. Pengguna memasukkan kata yang digunakan pada saat pencarian. Kemudian pengguna dapat memulai pencarian dengan cara menekan tombol cari. dalam proses pencarian ini kata yang dicari sudah disimpan di *database*. Kemudian *form* kemajuan pencarian akan dimunculkan beserta *progress bar* yang menandakan proses sedang berjalan. Dalam proses terdapat beberapa bagian proses yang mendukung proses pencarian yaitu: perhitungan jumlah tugas, pembagian jumlah kata, pembuatan *thread* dan sinkronisasi-nya. Selama proses berjalan pengguna dapat membatalkan proses pencarian dengan menekan tombol batal untuk menghentikan proses pencarian.

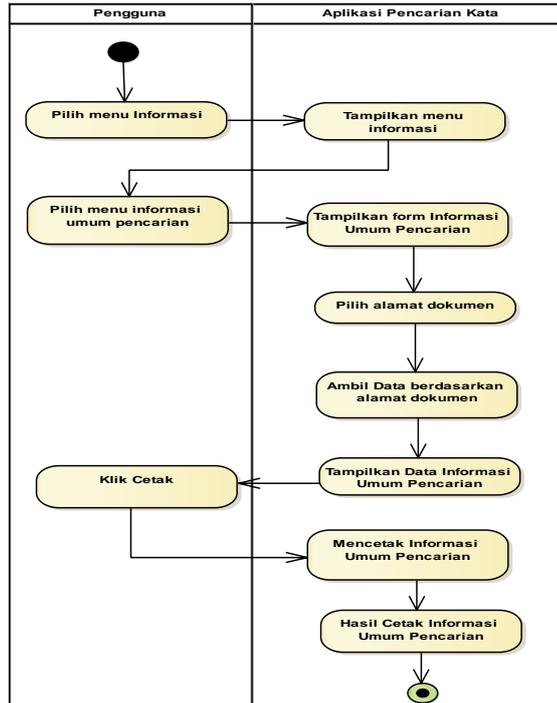


Gambar 3 Activity Diagram Mengisi Kriteria Kata

1.1.3. Activity Diagram Melihat Informasi Umum Pencarian

Pada Gambar 4 dapat dijelaskan bahwa pada saat memunculkan *form* informasi, terlebih dahulu masuk ke menu utama maka akan memilih sub menu, yaitu menu informasi. Di dalam menu informasi maka akan

memilih sub menu informasi tugas pencarian. Setelah informasi tugas pencarian ditampilkan maka pengguna dapat melihat hasil sebelumnya. Kemudian pengguna memilih untuk mencetak hasil keluar.

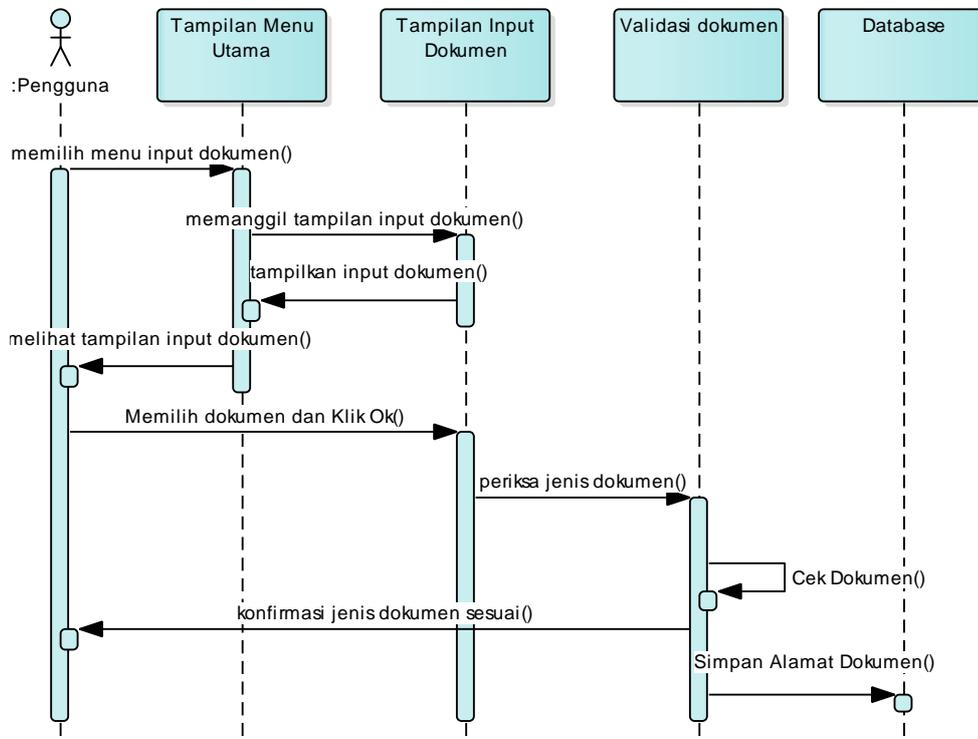


Gambar 4 Activity Diagram Melihat Informasi Umum Pencarian

1.2. Gambar Umum Rancangan Perangkat Lunak Melalui Sequence

Diagram Sequence menggambarkan tahapan yang dilakukan pengguna secara bertahap sesuai urutan. Berikut ini adalah diagram sekuensial pada aplikasi pencarian kata dengan *multithreading*:

1.2.1. Diagram Sekuensial Masukkan Dokumen



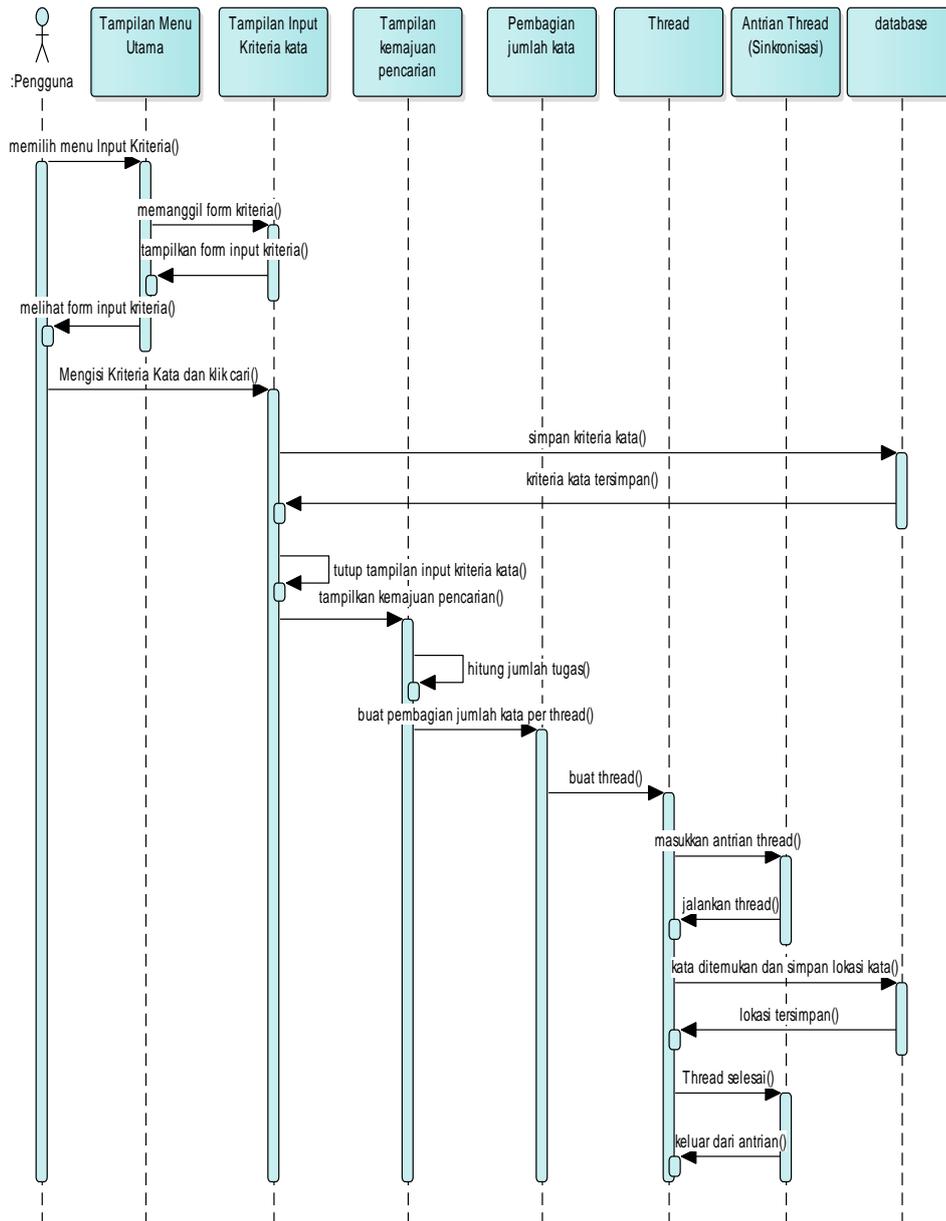
Gambar 5 Diagram Sekuensial Masukkan Dokumen

Pada Gambar 5 dapat dijelaskan bahwa pengguna memilih menu input dokumen pada menu utama kemudian *form* input dokumen ditampilkan kemudian pengguna menekan tombol jelahah. Kemudian kotak dialog *open file* akan ditampilkan selanjutnya. Pengguna memilih dokumen yang akan digunakan sebagai

sumber pencarian dan setelah itu pengguna menekan tombol *open* untuk memasukkan dokumen. Pengguna menekan tombol *ok* untuk menyelesaikan inputan, bersamaan dengan itu aplikasi melakukan pemeriksaan terhadap jenis dokumen. Apabila jenis dokumen bukan berekstensi “.docx” maka akan ditampilkan konfirmasi jenis dokumen belum sesuai. Sebaliknya jika jenis dokumen sesuai maka akan ditampilkan konfirmasi jenis dokumen sesuai. Kemudian informasi dokumen yang dimasukkan disimpan kedalam *database*.

1.2.2. Diagram Sekuensial Masukkan Kriteria Kata

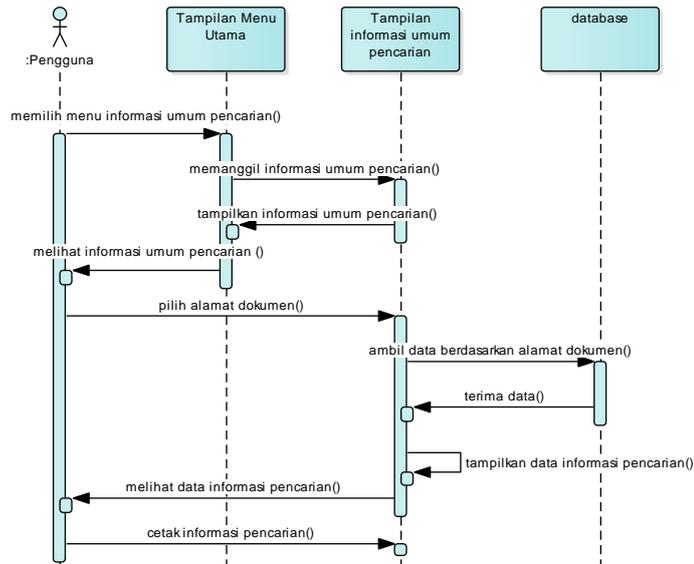
Pada Gambar 6 dapat dijelaskan bahwa pengguna memilih *form* pencarian kata pada menu utama. Kemudian *form* pencarian kata ditampilkan dan pengguna mengisi kriteria kata pada *textbox* yang disediakan kemudian pengguna menekan tombol cari. Pada saat bersamaan *form* pencarian kata ditutup dan memunculkan *form* kemajuan pencarian. Kemudian kriteria kata yang dimasukkan pada *form* pencarian kata di kirim ke bagian-bagian proses yaitu: proses perhitungan jumlah tugas, pembagian jumlah kata, pembuatan *thread*, dan sinkronisasi. Kemudian data hasil pencarian dimasukkan ke dalam entitas informasi umum pencarian yang mewakili tabel-tabel dalam *database*.



Gambar 6 Diagram Sekuensial Masukkan Kriteria Kata

1.2.3. Diagram Sekuensial Informasi Umum Pencarian

Pada Gambar 7 dapat dijelaskan bahwa pengguna memilih menu informasi umum pencarian. Setelah menu informasi umum pencarian dimunculkan maka *form* ini menampilkan data yang berisikan data hasil pencarian sebelumnya yang meliputi pekerjaan pencarian dari awal hingga akhir. Kemudian pengguna mencetak keluar hasil untuk mengetahui lebih jelas kinerja keseluruhan.



Gambar 7 Diagram Sekuensial Informasi Umum Pencarian

1.3. Tampilan Aplikasi

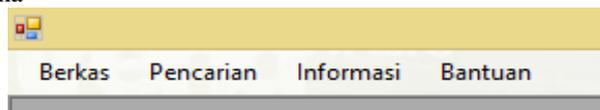
Untuk memulai aplikasi pencarian, pertama-tama pengguna akan ditampilkan sebuah *form splash screen* aplikasi pencarian menggunakan *multithread*.

1.3.1. Tampilan *Splash Screen*



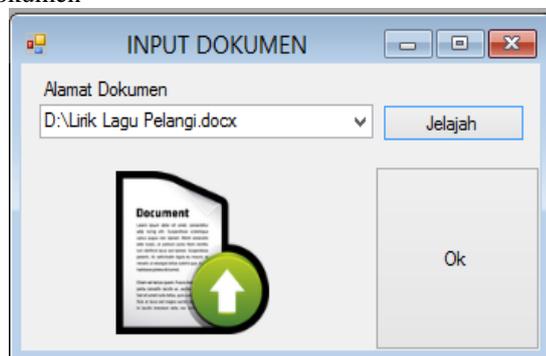
Gambar 8 Tampilan *Splash Screen*

1.3.2. Tampilan Menu Utama



Gambar 9 Tampilan Menu Utama

1.3.3. Tampilan Masukkan Dokumen

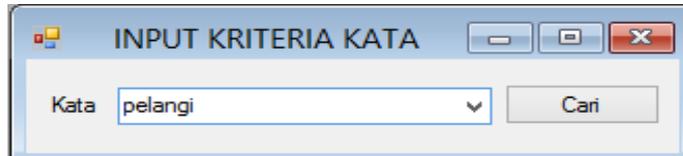


Gambar 10 Tampilan Masukkan Dokumen

Setelah masuk ke dalam menu input dokumen. Pengguna dapat menekan tombol jelajah dan pada saat itu *open dialog* akan ditampilkan. Pada *opendialog* pengguna dapat memilih dokumen apa yang akan digunakan dalam pencarian. Setelah menemukan dokumen yang diinginkan, pengguna dapat menekan tombol *Open* untuk

meneruskan inputan atau *cancel* untuk membatalkan pilihan dan keluar dari *open dialog*. Setelah pengguna menekan tombol *open*, pengguna dapat menekan tombol Ok sebagai aksi terakhir inputan dokumen. *Form* input dokumen akan ditutup setelah konfirmasi dokumen sesuai. Setelah pengguna selesai menginputkan dokumen, informasi mengenai alamat dokumen, ukuran (*size*) dokumen, jumlah kata dalam dokumen dan nama dokumen disimpan kedalam basis data dan pengguna dapat melanjutkan ke tahap selanjutnya yaitu, memilih pencarian kemudian pilih pencarian kata.

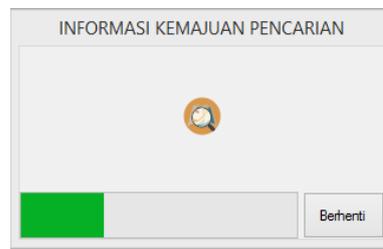
1.3.4. Tampilan Masukkan Kriteria Kata



Gambar 11 Tampilan Masukkan Kriteria Kata

Setelah *form* pencarian kata ditampilkan. Pengguna dapat mengisi kata yang dijadikan kriteria dalam pencarian. Kemudian pengguna dapat menekan tombol cari untuk memulai proses pencarian. Setelah pengguna menekan tombol cari maka *form* ini akan ditutup dan kemudian ditampilkan *form* kemajuan pencarian.

1.3.5. Tampilan Kemajuan Pencarian



Gambar 12 Tampilan Kemajuan Pencarian

Dalam proses pencarian pada *form* kemajuan pencarian, penulis melakukan penerapan *multithread* dalam proses pencarian. Dalam penerapannya penulis memisahkan proses pencarian ke dalam sub proses yang menjadi satu kesatuan dalam proses pencarian yakni:

1.3.6. Tampilan Informasi Umum Pencarian



Gambar 13 Tampilan Informasi Umum Pencarian

Dari informasi yang dimunculkan, penulis memperhatikan kolom *arrive*, *begin* dan *end*, kolom ini diperoleh dari urutan kerja *thread* yang sedang berjalan sebelumnya. Pada *form* di atas ditampilkan hanya satu *thread* yang muncul sehingga waktu yang diperlukan relatif singkat, lamanya satu *thread* berjalan. Dari hasil pengamatan data, Jika dihasilkan perataan lama *thread* sama dengan 0 maka dapat disimpulkan lama proses dari satu *thread* untuk menyelesaikan tugasnya berlangsung seketika itu juga. Sedangkan kolom *arrive* digunakan sebagai indikator apakah sinkronisasi berjalan atau tidak pada banyak *thread*. Untuk proses pencarian yang tidak menggunakan *multithread* terlihat jumlah *thread* dapat dipastikan satu dan nilai kolom *arrive* sama dengan kolom *begin* atau dapat disimpulkan prosesnya tidak menggunakan sinkronisasi. Tujuan diberikan sinkronisasi ialah memberikan jeda pada prosesor dan mengatur penggunaan sumber daya. Sinkronisasi dalam penggunaannya akan menghasilkan efek negatif yakni memangkas sedikit efisiensi dan efek positif dapat menjaga penggunaan memori berlebihan.

4. KESIMPULAN

Pada penulisan penelitian ini, penulis membuat suatu aplikasi yang menerapkan *multithread* dalam prosesnya yang berfungsi sebagai aplikasi pencarian kata. Aplikasi dimaksudkan sebagai representasi dari hasil penerapan *multithread* pada pencarian linear atau sekuensial. Dari uraian implementasi dan analisis serta berdasarkan referensi, maka dapat diambil kesimpulan:

- a. Telah dihasilkan aplikasi pencarian kata yang menerapkan *multithread*.
- b. Aplikasi ini dibuat sebagai representasi dari penerapan *multithread* pada proses pencarian terurut.
- c. Aplikasi ini dibuat agar memudahkan penilaian seberapa besar pengaruh penerapan *multithread* pada proses pencarian terurut dengan tanpa *multithread*.
- d. Peningkatan waktu proses pencarian dengan menerapkan *multithread*
- e. Mengurangi beban pencarian (jumlah pencarian) sehingga lebih efisien.

5. SARAN

Adapun berbagai saran untuk melengkapi kesimpulan yang diambil adalah sebagai berikut:

- a. Untuk menunjang kinerja pencarian kata yang lebih bervariasi maka diperlukan jumlah kata lebih dari satu.
- b. Dari segi tampilan aplikasi ini masih sangat minim, diperlukan banyak perubahan dan penambahan keduanya oleh orang-orang yang memiliki tingkat desain grafis yang baik serta menggunakan grafik yang mewakili hasil kinerja pencarian untuk pengembangannya.
- c. Diharapkan agar pembaca dapat mengembangkan pencarian kata dengan metode lain untuk mencapai efisiensi dan efektivitas yang maksimal.

UCAPAN TERIMA KASIH

Dalam penulisan penelitian ini, penulis telah banyak mendapat bantuan berupa bimbingan, petunjuk, saran maupun dorongan moril dari berbagai pihak, maka pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada seluruh civitas akademika Sekolah Tinggi Manajemen Informatika dan Komputer Widya Dharma Pontianak.

DAFTAR PUSTAKA

- [1] Einar Krogh, *An Introduction to Windows Operating System*, tersedia dalam <http://www.bookboon.com>, diakses pada tanggal 16 maret 2016 jam 08.00.
- [2] Yasin, Verdi. (2012). *Rekayasa Perangkat Lunak Berorientasi Objek*. Mitra Wacana Media.
- [3] Donis Marshall, *Pararell Programming with Microsoft Visual Studio 2010, online book + practice file*. authorization of Microsoft Corporation by: O'Reilly Media, Inc.1005 Gravenstein Highway North Sebastopol, California 95472.
- [4] Nugroho, Adi. (2011). *Perancangan dan Implementasi Sistem Basis Data*. Andi. Yogyakarta.
- [5] Munir, Renaldi. (2009). *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*. Informatika. Bandung
- [6] Sugiarti, Yuni. (2010). *Analisis dan Perancangan UML (Unified Modelling Language) Generated VB.6*. Graha Ilmu. Yogyakarta
- [7] Widodo, Prabowo Pudjo dan Herlawati. (2011). *Pengertian UML*. Informatika Bandung.
- [8] Hendrayudi. (2009). *VB 2008 Untuk Berbagai Keperluan Pemrograman*. PT. Elex Media Komputindo. Jakarta.
- [9] Hidayatullah, Priyanto. (2012). *Visual Basic .Net Membuat Aplikasi Database dan Program Kreatif*. Informatika. Bandung.
- [10] Cybertron Solution dan SmitDev Community. (2010). *Membangun Aplikasi Database dengan Visual Basic 2008 dan SQL Server 2008*. PT. Elex Media Komputindo. Jakarta.
- [11] Priyatno, Duwi. (2010). *Microsoft Office 2010 + Trik Aman Berinternet*. Kanaya Press.
- [12] Harmove, C.Widyo, dan Mario. (2010). *3 in 1 Microsoft office 2007- buku wajib pencari kerja*. Andi Yogyakarta dengan ELCOM.
- [13] Setiawan, Sulhan. (2016). *Teknik Pemrograman dan Multithreading pada Mikrokontroler*. Andi.Yogyakarta.