PERANCANGAN APLIKASI EVALUASI PERFORMA MULTITHREADING PADA PEMBUATAN LAPORAN PENJUALAN DENGAN APLIKASI DESKTOP

Rosalia Hadi

[1]Program Studi Sistem Informasi, Institut Teknologi dan Bisnis STIKOM Bali
Email: rosa@stikom-bali.ac.id

ABSTRAK

Pada jaman sekarang, computer sudah berkembang fungsionalitasnya dalam berbagai hal seperti pencatatan transaksi penjualan. Sekarang teknologi *multithreading* memberikan kemampuan pada computer untuk melakukan tugas secara simultan dan bersamaan. *Multithreading* merupakan bentuk kecil dari sebuah pararel processing yang memungkinkan sebuah pekerjaan bisa dilakukan secara simultan dan bersamaan. Laporan penjualan merupakan salah satu laporan yang wajib ada pada sebuah sistem Point of Sale. Pada saat system membuat laporan penjualan, system akan melakukan query pada database yang kemudian system akan melakukan perulangan pada setiap record yang dihasilkan oleh query untuk melakukan proses pencetakan. Hal ini jika dilakukan dengan metode konvesional yang menggunakansinglethreadakanmenghabiskanwaktu yang sangat lama. Penulis melakukan penelitian berupa rancangan yang akan melakukanev aluasi performa pada metode *multithreading* yang akan diterapkan dalam pembuatan laporan keuangan pada sistem point-of-sale. Terdapatd ua level hirarki *threading* yang digunakan pada metode multi-threading yaitu satu main-threading dan sisanya adalah child-threading.

Kata kunci: pararel *processing, multithreading*, *point-of-sale*.

ABSTRACT

In today's era, computers have developed functionality in various ways such as recording sales transactions. Now multithreading technology provides the ability for computers to perform tasks simultaneously and simultaneously. Multithreading is a small form of parallel processing that allows a job to be carried out simultaneously and simultaneously. A sales report is one of the mandatory reports in a Point of Sale system. When the system makes a sales report, the system will query the database which the system will loop on each record generated by the query to do the printing process. This is done with conventional methods that use single-thread will spend a very long time. The author conducts research in the form of a design that will conduct a performance evaluation on the multithreading method that will be applied in making financial reports on a point-of-sale system. There are two levels of threading hierarchy that are used in multi-threading methods namely one main-threading and the rest are child-threading.

Keywords*: pararel processing, multithreading, point-of-sale.*

1. Pendahuluan

Pada jaman sekarang, computer sudah tidak lagi sebagai sebuah kebutuhan rekreasi (Underwood, 2015). Namun komputer juga telah berkembang fungsionalitas dalam berbagai hal seperti penggunaan pada dunia medis (Bar et al., 2015), hingga pencatatan transaksipenjualan (Lopez et al., 2018). Hal ini dikarenakan komputer pada saat ini memiliki kemampuan yang cepat dengan harga yang terjangkau.

Teknologi computer mengalami perkembangan pesat dan tidak hanya dari sector perangkat keras hingga arsitektur perangkat lunak(Dingsøyr et al., 2012). Munculnya teknologi-teknologi baru dalam arsitektur processor dimulai dari *pipelining* yang

memungkinkan proses *fetch* dan *execute* dilakukan secara bersamaan (Jouppi et al., 2017). Hingga saat ini sudah ada teknologi *multithreading* yang memberikan kemampuan pada computer untuk melakukan tugas secara simultan dan bersamaan (Trott and Olson, 2010).

Multithreading merupakan bentuk kecil dari sebuah *pararel processing* yang memungkinkan sebuah pekerjaan bisa dilakukan secara simultan dan bersamaan (Shen et al., 2016). Pararel processing memungkinkan sebuah perkerjaan dilakukan secara bersamaan dan menurunkan waktu eksekusi yang diperlukan. Antar proses/thread pada pararel processing berdiri sendiri tanpa mempengaruhi proses-proses yang lainnya.

Komputer dalam penggunaan sebagai pencatatan penjual atau Point of Sale System bertugas dalam mencatat proses bisnis yang terjadi pada sebuah took (Buttenheim et al., 2012). Sistem melakukan pencatatan mulai dari proses datangnya barang, hingga proses penjualan barang ke customer. Tentunya semua proses bisnis yang terjadi akan menghasilkan sebuah laporan yang dapat dibaca dan diolah oleh seorang akuntan pada sebuah toko.

Data penjualan yang biasanya tercatat pada system porsinya selalu lebih tinggi jumlah *record*-nya dibandingkan dengan data pembelian. Hal ini dikarenakan biasanya took melakukan pembelian sekali dalam jumlah yang besar, sehingga system akan mencatat pembelian hanya sekali dengan kuantitas barang yang banyak. Berbeda dengan data penjualan yang memiliki jumlah *record* lebih tinggi dari data pembelian. Asumsi dimana toko melakukan pembelian barang sebanyak 100 dalam satu kali pembelian, dimana 100 barang tersebut dijual secara eceran sebanyak 1 barang per transaksi. Asumsi seperti ini menggambarkan bahwa proses bisnis pembelian barang hanya dilakukan sekali dan dicatat sekali, namun penjualan dapat dilakukan lebih dari satu kali dan dicatat sebanyak penjualan yang terjadi.

Laporan penjualan merupakan salah satu laporan yang wajib ada pada sebuah sistem Point of Sale. Pada saat system membuat laporan penjualan, system akan melakukan *query* pada *database* yang kemudian system akan melakukan perulangan pada setiap *record* yang dihasilkan oleh *query* untuk melakukan proses pencetakan, baik *hardcopy* maupun *softcopy*. Hal ini jika dilakukan dengan metode konvesional yang menggunakan *single thread* akan menghabiskan waktu yang sangat lama.

Permasalahannya adalah ketika program melakukan ektraksi data dari database dengan jumlah data yang banyak. Asumsikan jika data yang tersedia adalah sejumlah 10.000 dan untuk memproses satu data membutuhkan waktu sekitar 1 detik. Maka untuk mencetak data dibutuhkan sekitar 10.000 detik atau sekitar 2.78 jam untuk menghasilkan laporan jika dilakukan secara *singlethread*/metode konvensional. Hal ini menyebabkan program tidak dapat berjalan secara responsive dan tidak dapat diandalkan ketika membutuhkan data dalam hitungan kurang dari 1 jam. Terlebih lagi, munculnya efek samping seperti program mengalami hang/freeze saat membuat laporan tersebut karena pekerjaan dilakukan di *main thread*.

Berdasarkan masalah tersebut, penulis meneliti rancangan aplikasi yang akan melakukan evaluasi performa pada metode *multithreading* yang akan diterapkan dalam pembuatan laporan keuangan pada sistem point-of-sale. Diharapkan dengan evaluasi performa ini, dapat menghasilkan desain dan algoritma yang lebih baik pada proses pembuatan laporan penjualan berdasarkan waktu eksekusi yang dibutuhkan dalam menghasilkan satu laporan. Pada uji performanya akan memanfaatkan C# untuk bahasa pemrogramannya, .NET Framework dan MySQL dalam penggunaan databasenya. Usulan penelitian ini diharapkan dapat mengembangkan program yang lebih responsive dengan peningkatan performa yang lebih baik.

2. Metode

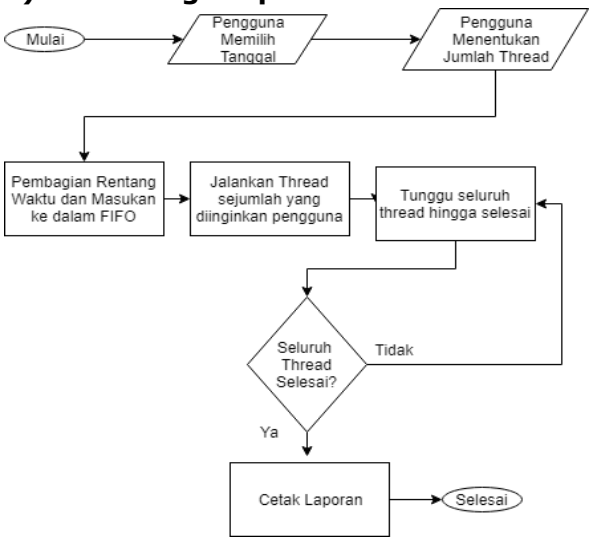
Implementasi system uji coba memerlukan rancangan dasar dari aplikasi yang akan diuji. Terdapat beberapa tahapan yang diperlukan dalam melakukan perancangan aplikasi, diantaranya sebagai berikut:

a) Perancangan aplikasi secara umum.
   Perancangan ini bertujuan untuk merancang aplikasi secara umum, sehingga mendapatkan aliran dari gambaran aplikasi yang dirancang secara keseluruhan
b) Perancangan Diagram Konteks
   Diagram konteks berfungsi sebagai pemberi gambaran ruang lingkup dari proses beserta data yang akan diolah sehingga dapat memberikan gambaran umum dari aplikasi uji performa yang akan dibangun.
c) Perancangan Flowchart
   Flowchart berfungsi untuk menggambarkan alir kerja dari aplikasi uji performa yang akan dibangun.
d) Perancangan Konseptual Database
   Konseptual database berfungsi untuk menggambarkan data-data yang nantinya akan digunakan dalam aplikasi uji performa.
e) Perancangan User Interface
   User interface perlu dirancang untuk memberikan gambaran umum tampilan yang akan digunakan dalam aplikasi uji performa ini.

3. Hasil dan Pembahasan

Berdasarkan tahapan-tahapan yang sudah dijelaskan, perancangan-perancangan aplikasi dibagi dalam beberapa hal sebagai berikut.

a) PerancanganAplikasiSecaraUmum



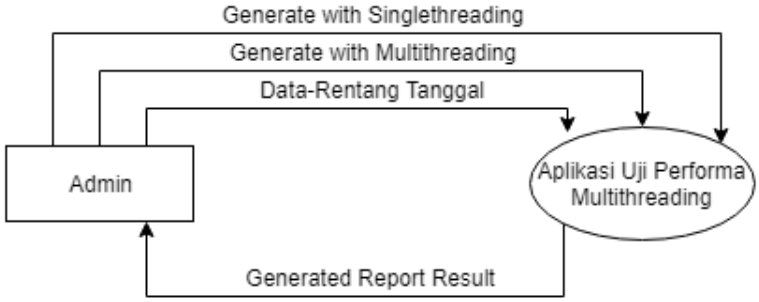
Gambar 1. Diagram alir algoritma secara umum

Secara umum, alur algoritma yang akan diteliti diawali dengan pengguna memilih rentang tanggal laporan yang dikehendaki. Selanjutnya pengguna menentukan jumlah *thread* yang akan digunakan dalam pembuatan laporan. Aplikasi selanjutnya membagi rentan waktu kedalam bagian-bagian kecil kemudian memasukan hasil pembagian tersebut kedalam global variable penyimpanan dengan konsep FIFO (First In First Out). Kemudian aplikasi menjalankan *thread* dan *thread* akan mengambil bagian pekerjaan dari global variable FIFO. Aplikasi kemudian menunggu semua *thread* selesai berjalan

dan setelah semua *thread* selesai berjalan, data ditampilkan oleh aplikasi. Gambar 1 merupakan rancangan algoritma secara umum.

b) Perancangan Diagram Konteks

Aplikasi uji performa ini hanya akan memiliki dua proses utama saat digunakan. Pengguna dapat memilih apakah akan menggunakan metode *pararel-processing* atau menggunakan metode *serial-processing*. Gambar 2 merupakan diagram konteks dari aplikasi uji performa yang akan dibangun.



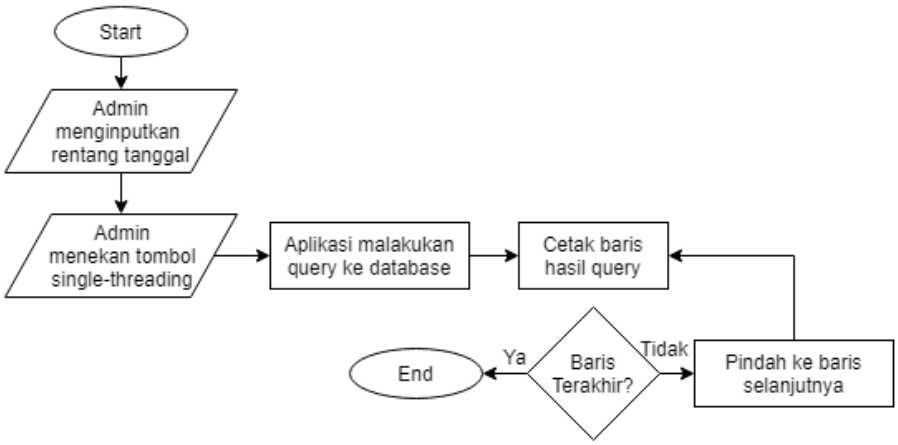
Gambar 2. Konteks Diagram Aplikasi Uji Performa

Terdapat 3 nilai inputan yang dapat diterima oleh aplikasi dari admin. Pertama adalah rentang waktu tanggal yang dipilih oleh admin, kedua adalah metode yang dipilih oleh admin apakah memilih metode *single-threading* atau metode *multi-threading*. Kemudian diakhir aplikasi akan memberikan output berupa laporan yang sudah selesai diproses, baik menggunakan metode *single-threading* maupun *multi-threading*.

c) Perancangan Flowchart

Terdapat 2 proses utama yang akan dibangun dalam aplikasi ini, yaitu proses pembuatan laporan dengan menggunakan *single-threading* metode dan menggunakan *multi-threading* metode. Admin nantinya dapat memilih menggunakan *single-threading* atau *multi-threading* sehingga dapat membandingkan kedua performa dari kedua metode tersebut.

1) Single Threading Method



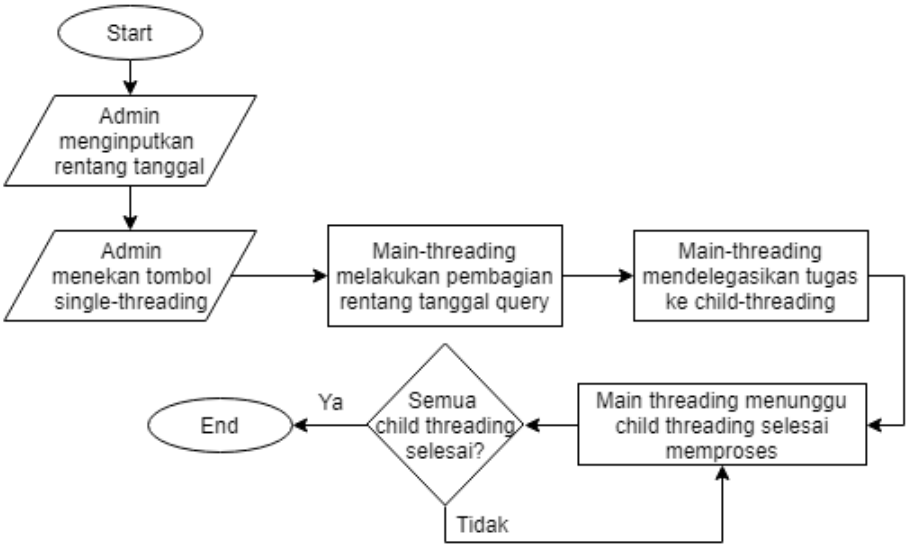
Gambar 3. Aliran Proses pada Metode S*ingle-Threading*.

Pada *single-threading method*, aplikasi awalnya akan menerima inputan berupa rentang tanggal yang akan dipilih yang akan digunakan dalam proses pencetakan laporan. Kemudian aplikasi akan melakukan *query* ke *database* untuk mengambil

keseluruhan data yang akan ditampilkan. Setelah semua data diambil, kemudian semua data akan dicetak kelayar untuk ditampilkan. Gambar 3 merupakan aliran proses yang terjadi pada *single-threading method.*

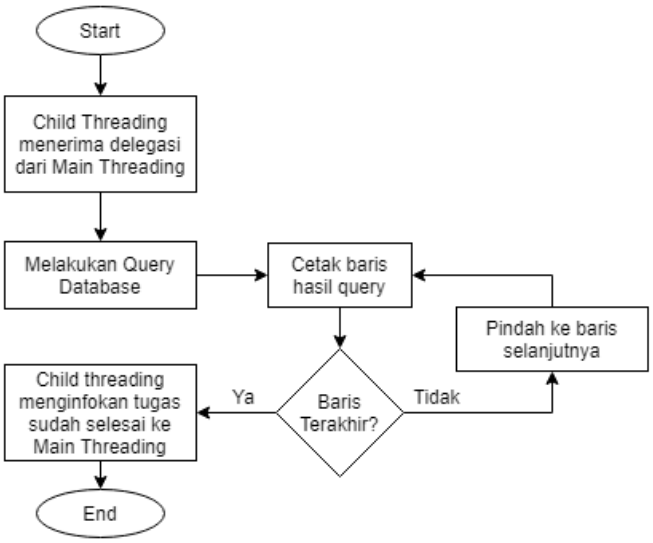
2) Multi Threading Method

Pada metode *multi-threading* terdapat dua hirarki threading, yaitu yang bertindak sebagai *main threading* dan sisanya sebagai *child threading*. *Main threading* berfungsi untuk mengatur dan mengkordinasikan seluruh *child-threading*. *Child threading* nantinya yang akan melakukan processing data baik *query* maupun mencetak ke layar.



Gambar 4. Proses yang Dirancang pada Main-Threading

Pada Gambar 4 yang merupakan aliran proses yang terjadi pada *main threading*. Proses yang terjadi pada *main threading* adalah yang pertama menerima input berupa rentang tanggal yang dipilih oleh admin. Kemudian *main threading* akan membagi rentang waktu dalam beberapa bagian. Masing-masing bagian nantinya akan didelegasikan ke masing-masing *child threading*. Setelah semua bagian telah didelegasikan kepada *child-threading*, *main-threading* akan menunggu semua *child threading* usai menjalankan tugasnya.

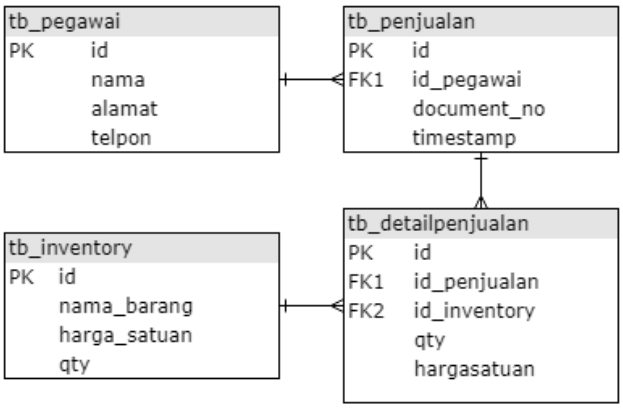


Gambar 5. Proses yang terjadi pada *child threading*

Pada Gambar 5 merupakan proses yang terjadi di *child-threading* dimana pada *child-threading* akan menerima delegasi tugas dari *main-threading*. Proses yang terjadi pada *child-threading* adalah melakukan *query* sesuai dengan delegasi tanggal yang diberikan oleh *main-threading*. Kemudian data yang dihasilkan dari *query* akan dicetak dan ditampilkan ke admin. Setelah usai melakukan tugasnya, *child-threading* akan menginformasikan kepada *main-threading* bahwa tugasnya telah selesai.

d) PerancanganKonseptual Database

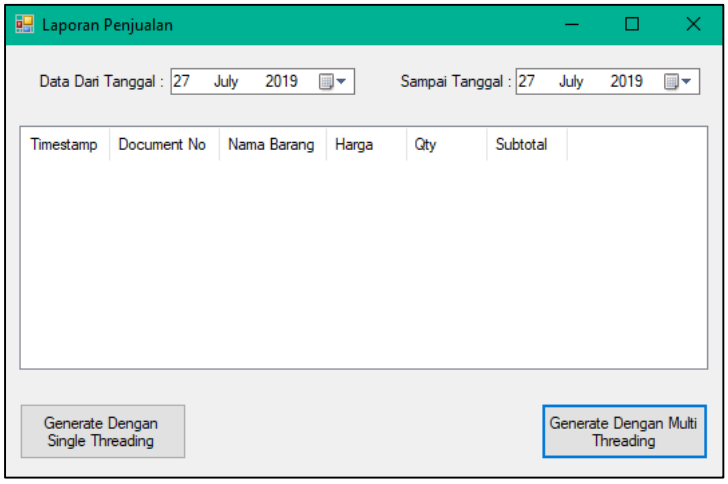
Pada aplikasi uji performa *multi threading* ini, semua data penjualan yang digunakan bersifat *dummy* data. Namun meskipun *dummy data,* data tersebut tetap disimpan dalam sebuah *database*. *Database* yang digunakan hanya memiliki 4 tabel yaitu table penjualan, detail penjualan, master inventori dan master pegawai. Gambar 6 merupakan konseptual database yang digunakan dalam perancangan aplikasi uji performaini.



Gambar 6. Konsptual Database yang Digunakan

e) Perancangan User Interface

Pada gambar7 merupakan rancangan dari *user interface* yang akan digunakan. Hanya terdapat beberapa control penting pada rancangan antarmuka. Terdapat date timepicker untuk memilih tanggal yang menjadi batas rentang dari hingga batas tanggal pencetakan laporan. Selain itu terdapat list view detail yang akan digunakan untuk menampung semua data yang sudah selesai diproses. Rancangan juga dilengkapi dengan dua tombol untuk memilih apakah menggunakan *single threading* atau *multi threading* dalam memproses laporan.



Gambar 7. Rancangan desain antar muka aplikasi

4. Simpulan

Dari penelitian yang dilakukan, maka dapat disimpulkan aplikasi yang dirancang digunakan untuk melakukan evaluasi performa multithreading pada pembuatan laporan transaksi penjualan berbasiskan aplikasi desktop. Aplikasi yang dirancangan memiliki dua proses yaitu membuat laporan dengan mode *single-threading* dan *multi-threading*.Terdapat dua level hirarki *threading* yang digunakan pada metode *multithreading* yaitu satu *main threadiing* dan sisanya adalah *child threading*.Database yang dirancang terdiri dari empat tabel yang dikhususkan untuk menampung data penjualan.

Daftar Rujukan

BAR, Y., DIAMANT, I., WOLF, L., LIEBERMAN, S., KONEN, E. & GREENSPAN, H. Chest pathology detection using deep learning with non-medical training. 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), 2015. IEEE, 294-297.

BUTTENHEIM, A. M., HAVASSY, J., FANG, M., GLYN, J., KARPYN, A. E. J. J. O. T. A. O. N. & DIETETICS 2012. Increasing supplemental nutrition assistance program/electronic benefits transfer sales at farmers' markets with vendor-operated wireless point-of-sale terminals. 112**,** 636-641.

DINGSØYR, T., NERUR, S., BALIJEPALLY, V. & MOE, N. B. 2012. A decade of agile methodologies: Towards explaining agile software development. Elsevier.

HINES, C. & YOUSSEF, A. Machine Learning Applied to Point-of-Sale Fraud Detection. International Conference on Machine Learning and Data Mining in Pattern Recognition, 2018. Springer, 283-295.

JAILIA, M., KUMAR, A., AGARWAL, M. & SINHA, I. Behavior of MVC (Model View Controller) based Web Application developed in PHP and. NET framework. 2016 International Conference on ICT in Business Industry & Government (ICTBIG), 2016. IEEE, 1-5.

JOUPPI, N. P., YOUNG, C., PATIL, N., PATTERSON, D., AGRAWAL, G., BAJWA, R., BATES, S., BHATIA, S., BODEN, N. & BORCHERS, A. In-datacenter performance analysis of a tensor processing unit. 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), 2017. IEEE, 1-12.

LOPEZ, M. E., HEINE, W. A., MCKINSTRAY, K. D. & KEREN, G. 2018. System, method, and computer program for dynamically reconciling a distributor invoice with a retailer receiving invoice for products sold under multiple UPCs and in multiple quantity units. Google Patents.

LOW, B. W., OOI, B. Y. & WONG, C. S. Scalability of database bulk insertion with multi-threading. International Conference on Software Engineering and Computer Systems, 2011. Springer, 151-162.

SHEN, H., WEI, X., LIU, H., LIU, Y. & ZHENG, K. Design and implementation of an LTE system with multi-thread parallel processing on OpenAirInterface platform. 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), 2016. IEEE, 1-5.

SOLIS, D. & SCHROTENBOER, C. 2018. C# and. NET Core. *Illustrated C# 7.* Springer.

TROTT, O. & OLSON, A. J. J. J. O. C. C. 2010. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. 31**,** 455-461.

UNDERWOOD, C. J. S. 2015. Snapshots issue 8: Playing computer games for recreation. 8, 1.