

Implementasi Optical Character Recognition (OCR) Pada Masa Pandemi Covid-19

^{*1}Ali Firdaus, ^{*2}M. Syamsu Kurnia, ^{*3}Tia Shafera, ^{*4}Wahyu Istalama Firdaus

^{*1,2,3} Jurusan Teknik Komputer Politeknik Negeri Sriwijaya - Palembang

^{*4}Faculty of Islamic Studies, International University of Aprica - Sudan Nort Aprica

^{*1}alifirdaus1970@gmail.com, ^{*2}syamsu@polsri.ac.id, ^{*3}tia@polsri.ac.id,

^{*4}wahyuistalama02@gmail.com

Abstrak

Banyak dokumen yang harus diproses dan dipindai secara digital, terutama di masa pandemi ini, dokumen fisik seringkali harus diolah kembali dengan menggunakan aplikasi pengolah kata. Proses ini menguras banyak waktu terutama bila dokumen tersebut rusak, hilang dan tidak memiliki salinannya. Hal ini dapat diatasi dengan menggunakan algoritma Optical Character Recognition (OCR). OCR mengkonversi gambar huruf menjadi karakter ASCII yang dikenali komputer. File gambar relatif lebih besar dari pada file teks ASCII-nya saja. Dari permasalahan yang ada, peneliti mencoba untuk menemukan solusinya dengan membuat program aplikasi OCR untuk mempermudah proses tersebut. Program OCR yang dibuat berhasil digunakan untuk men-scan atau memindai suatu dokumen fisik jika dokumen tersebut hilang atau rusak, dan dapat menghemat waktu pembuatan, pemrosesan dan pengetikan dokumen.

Kata kunci : Algoritma, Optical Character Recognition, ASCII

Abstract

Many documents must be processed and scanned digitally, especially during this pandemic, physical documents often have to be reprocessed using word processing applications. This process takes a lot of time especially if the document is damaged, lost and does not have a copy. This can be overcome by using the Optical Character Recognition (OCR) algorithm. OCR converts letter images into computer-recognized ASCII characters. Image files are relatively larger than their ASCII text files. From the existing problems, the researchers tried to find a solution by creating an OCR application program to simplify the process. The created OCR program is successfully used to scan or scan a physical document if the document is lost or damaged, and can save time on document creation, processing and typing.

Keywords: Algorithm, Optical Character Recognition, ASCII

1. PENDAHULUAN

Kantor Pelayanan Perbendaharaan Negara (KPPN) Berperan sebagai Kuasa Bendahara Umum Negara (BUN) di daerah. Tentu ini membuat tugas dan fungsi KPPN Palembang sendiri sangatlah penting. KPPN Palembang termasuk dalam klasifikasi KPPN Tipe A1 Provinsi yang mempunyai 3 (tiga) tugas diantaranya :

1. Melaksanakan kewenangan perbendaharaan dan bendahara umum negara,
2. Penyaluran pembiayaan atas beban anggaran, dan
3. Penata usahaan penerimaan dan pengeluaran anggaran melalui dan dari kas negara berdasarkan peraturan perundang-undangan

Tugas-tugas tersebut dilaksanakan secara sistematis dengan menggunakan prosedur, alat dan ketentuan yang sesuai.

Optical Character Recognition (OCR) adalah algoritma proses konversi gambar huruf menjadi karakter ASCII (*American Standard Code for Information Interchange*) yang dikenali oleh komputer. Gambar huruf yang dimaksud dapat berupa hasil scan dokumen, hasil print-screen halaman web, hasil foto, dan lain-lain [1][8][10].

Salah satu kebutuhan mengapa perlu konversi gambar huruf menjadi karakter ASCII adalah karakter ASCII memiliki kapasitas penyimpanan yang lebih kecil. Contohnya, suatu paragraf di-*printscreen* dan disimpan dalam format *png*. Paragraf tersebut juga di-*copy* dan disimpan dalam format *txt*. Untuk file gambar, memiliki size 42KB, sedangkan untuk file teks, memiliki size 1KB. Dari sini bisa terlihat bahwa file gambar akan selalu relatif lebih besar ketimbang menyimpan langsung teks ASCII-nya saja [2][7].

OCR adalah sistem yang sudah lama dikembangkan. Tahun 1914, Emanuel Goldberg telah mulai membuat sistem OCR yang berfungsi untuk telegram dan alat baca untuk orang tunanetra. Sistem OCR terus dikembangkan hingga kini sehingga dapat menghasilkan akurasi yang lebih baik bahkan dalam situasi-situasi yang dimana karakter sulit untuk dikenali. Pengaplikasian OCR pada program memerlukan *Library*. *Library* memungkinkan algoritma pembaca dapat di-*compile* oleh program yang mengaplikasikan OCR tersebut.

Bahkan pengaplikasian secara lanjut memungkinkan program (dan secara tidak langsung, komputer) untuk melakukan proses yang lebih rumit, contohnya translasi ke bahasa asing, pencarian, sistem baca otomatis untuk orang tunanetra, input data, pengenalan karakter seperti plat nomor, pengetesan CAPTCHA, atau masalah teks lainnya [12][13][14]. Terdapat dua macam *Optical Character Recognition*, antara lain: *Offline* dan *Online Optical Character Recognition* [3][9].

Didalam penelitian ini penulis mencoba untuk membuat program aplikasi yang dapat mempermudah proses tersebut. Dengan cara cara membuat program yang memanfaatkan *Optical Character Recognition* (OCR) untuk menyelesaikan masalah yang terdapat pada Kantor Pelayanan Perbendaharaan Negara (KPPN) Palembang.

2. METODE PENELITIAN

2.1. Pengembangan Java

Banyak pihak telah membuat IDE (*Integrated Development Environment* - Lingkungan Pengembangan Terintegrasi) untuk *Java*. IDE pada dasarnya adalah dasar dari pembuatan program *Java* dan menyederhanakan pembuatan program.

Berikut adalah beberapa contoh IDE:

- *Dr. Java.*
- *Eclipse JDT.*
- *OracleJDeveloper.*
- *NetBeans.*
- *IntelliJ IDEA.*
- *Xinox JCreator.*

2.2. NetBeans

NetBeans adalah suatu serambi pengembangan perangkat lunak yang ditulis dalam bahasa pemrograman *Java*. Serambi Pada *NetBeans*, pengembangan suatu aplikasi dan program dapat dilakukan dimulai dari setelan perangkat lunak modular bernama *modules*. (*NetBeans*, 2020) [4]

Semula, aplikasi *NetBeans* IDE ini diperuntukkan bagi pengembangan dalam *Java*. Namun, aplikasi ini juga mendukung program-program pembuatan bahasa lain secara khusus seperti *PHP*, *C/C++* dan *HTML5*. *NetBeans* adalah alat lintas serambi serta penerapannya dijalankan pada *Microsoft Windows*, *Mac OS X*, *Linux*, *Solaris* dan serambi-serambi lainnya yang mendukung *JVM* yang sepadan[9].

2.3. Optical Character Recognition (OCR)

Pengenalan Karakter Optik (bahasa Inggris: *Optical Character Recognition*, yang biasa disingkat OCR) adalah sebuah perangkat lunak yang mengubah teks dalam format berkas citra atau gambar ke dalam format teks yang bisa dibaca dan disunting oleh aplikasi komputer (misalnya: *Notepad* dan *Microsoft Word*). Berkas teks berformat citra tersebut didapatkan dengan cara memindai atau memfoto sebuah buku, manuskrip, tulisan di papan pengumuman, ataupun materi kuliah di papan tulis dsb. Sedangkan alat yang digunakan untuk memindai adalah pemindai (*scanner*) atau kamera baik kamera DSLR ataupun kamera di ponsel pintar [2][5][10][11].

Sebagai perangkat lunak yang kompleks, OCR diterapkan di berbagai bidang seperti untuk membaca data secara otomatis dan langsung memasukkannya ke dalam basis data, contohnya seperti dalam pemindaian passport, nota transfer bank, kertas berharga, untuk pengenalan pelat kendaraan bermotor dari video atau gambar yang tertangkap kamera, dan yang paling marak adalah untuk preservasi konten buku-buku sumber referensi utama dan manuskrip bersejarah. Dalam beberapa kasus, OCR berkontribusi sangat besar dalam proses pembangunan perpustakaan digital.

Sejarah OCR bisa dirunut sejak tahun 1809 saat peranti membaca untuk aplikasi orang buta dan pembacaan telegraf dikembangkan. Kemudian berlanjut pada penemuan mesin yang mampu mengubah karakter tercetak ke dalam kode standard telegraf buatan Emanuel Goldberg pada tahun 1914. Mesin ciptaan Goldberg ini diklaim sebagai asal muasal teknologi OCR. Di waktu yang bersamaan, Edmund Fourier mengembangkan *Optophone* yang merupakan mesin pemindai jinjing yang mampu menghasilkan bunyi sesuai dengan karakter khusus yang tercetak di dokumen. Kembali ke penemuan Goldberg, di akhir tahun 1920an tepatnya pada tahun 1927, ciptaannya yang dinamai Mesin statistik (*statistical machine*) merupakan sistem pengenalan kode optik yang digunakan untuk pencarian arsip mikrofilm [10].

Melompat ke dekade 1990an, OCR banyak dimanfaatkan oleh perpustakaan-perpustakaan untuk mendigitalkan surat kabar bersejarah. Proyek digitalisasi buku-buku bersejarah dan sumber referensi primer ini mulai menjamur memasuki abad 21 karena didukung oleh perkembangan pesat di bidang perangkat keras, perangkat lunak dan Internet. Ini memungkinkan *WebOCR*, sebuah perangkat lunak daring yang diluncurkan oleh *Expervision* -- beroperasi di lingkungan komputasi *Cloud* dan aplikasi perangkat bergerak (*mobile applications*) [10].

OCR memiliki beberapa tahapan utama yang terdiri dari:

1. Pra-Pemrosesan

Tahap prapemrosesan ini bertujuan untuk mendapatkan karakter tunggal dari sebuah teks terpindai dalam kondisi yang bagus dan bersih sehingga memudahkan proses pengenalannya. Menurut Bieniecki dkk, prapemrosesan diawali dengan normalisasi kondisi teks dengan cara menghilangkan derau seperti noktah dan koreksi orientasi citra teks, tahap binerisasi, serta segmentasi. Tahap ini jika dilakukan dengan benar akan meningkatkan rasio akurasi pengenalan karakter [2][11].

2. Tahap Ekstraksi Fitur

Tujuan dari ekstraksi fitur adalah untuk menemukan atribut pola-pola karakter yang terpenting dan berbeda dari karakter lainnya agar bisa diklasifikasikan. Peran manusia adalah menentukan dan menyeleksi fitur yang memungkinkan proses pengenalan yang efisien dan efektif. Pertanyaannya adalah lalu apa saja yang bisa dijadikan fitur bagi himpunan aksara atau alfabet dalam sistem penulisan bahasa tertentu [11].

3. Proses Pengenalan

Saat pola-pola karakter telah terpetakan ke dalam nilai vektor, maka permasalahan berikutnya adalah bagaimana mengelompokkan karakter yang memiliki nilai vektor yang sama

atau hampir sama. Permasalahan ini diselesaikan dengan cara klasifikasi. Rangkaian proses klasifikasi nilai vektor inilah yang dikenal sebagai proses pengenalan karakter. Dengan demikian proses pengenalan karakter berurusan dengan dan berada ditataran representasi karakter[2][3].

4. Paska-Proses

Setiap sistem OCR yang dibangun dengan algoritma tercanggih-pun sering kali membuat kesalahan, dalam arti tidak semua karakter yang dibaca dikonversikan ke karakter padanannya. Untuk itulah tahap paska-proses pencocokan karakter dilakukan untuk meningkatkan akurasi pengenalan karakter. Sistem paska-proses ini dikenal juga sebagai proses koreksi karena modul ini bertugas untuk mengoreksi kesalahan yang sering dilakukan di tataran kata (Nicomsoft, 2012). [5]

2.4. Tesseract

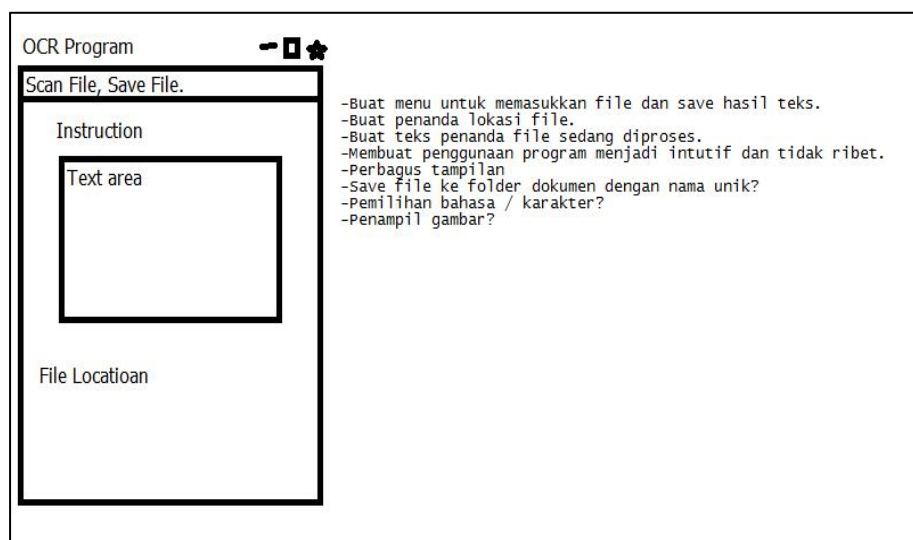
Dalam perangkat lunak komputer, *Tesseract* adalah mesin Pengenal Karakter Optik (*Optical Character Recognition* - OCR) gratis. *Tesseract* pada awalnya dikembangkan sebagai perangkat lunak berpemilik di *Hewlett-Packard* antara tahun 1985 hingga 1995. Setelah sepuluh tahun tanpa perkembangan apapun yang terjadi, *Hewlett Packard* dan UNLV merilis *Tesseract* sebagai sumber terbuka pada tahun 2005. *Tesseract* saat ini sedang dikembangkan oleh *Google* dan dirilis di bawah Lisensi *Apache, Version 2.0*. *Tesseract* dianggap salah satu perangkat lunak mesin OCR bebas yang paling akurat yang tersedia pada tahun 2006. (Kay, 2007).[2][3][6]

3. HASIL DAN PEMBAHASAN

3.1. Membuat Rancangan Program

Rancangan dalam hal ini adalah proses penetapan langkah-langkah dalam pembuatan program dan pentuan bagaimana hasil dari pembuatan program tersebut. Bila dilakukan dengan baik, rancangan dapat mempercepat proses pembuatan program dengan cara memberikan fokus dan tujuan untuk dicapai. Rancangan dapat dilakukan secara sederhana maupun lebih kompleks tergantung dengan keinginan perancang tersebut.

Berikut adalah contoh rancangan sederhana, berbentuk *to-do-list* dan desain sederhana program yang dibuat [12][14].

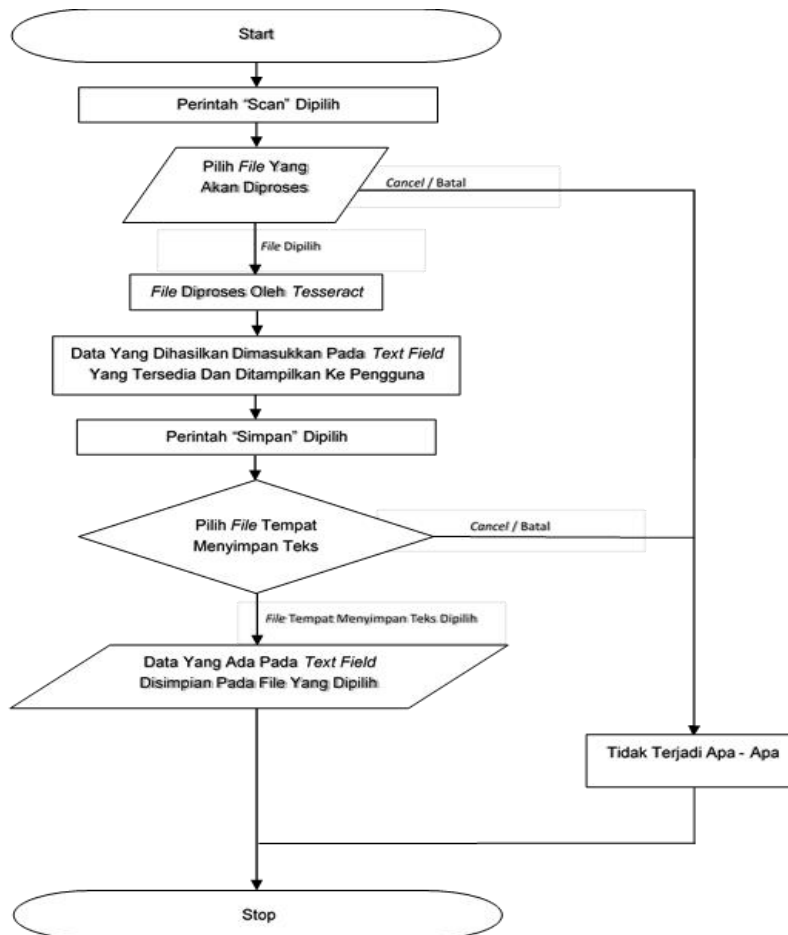


Gambar 3.1. Rancangan Sederhana Untuk Program

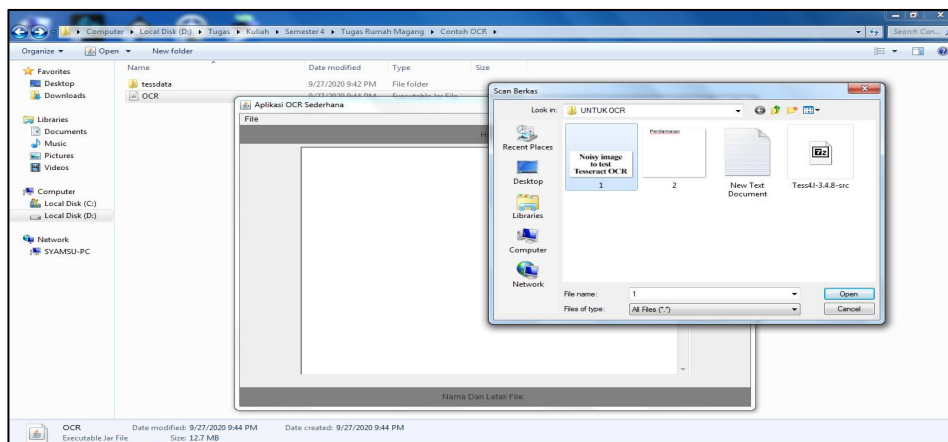
3.2. Pengimplementasian Rancangan Pada Program / Pembuatan

Rancangan diimplementasikan dengan cara mengembangkan source code dengan bahasa pemrograman *Java*. *Java* memiliki banyak fitur dan instruksi yang memungkinkan berbagai aksi dapat dilakukan oleh pengguna. Hal ini ditambah dengan pengkodean yang tidak terlalu sulit membuat *source code Java* relatif mudah dikembangkan dan banyak fitur dapat dimuat kedalamnya [13][14].

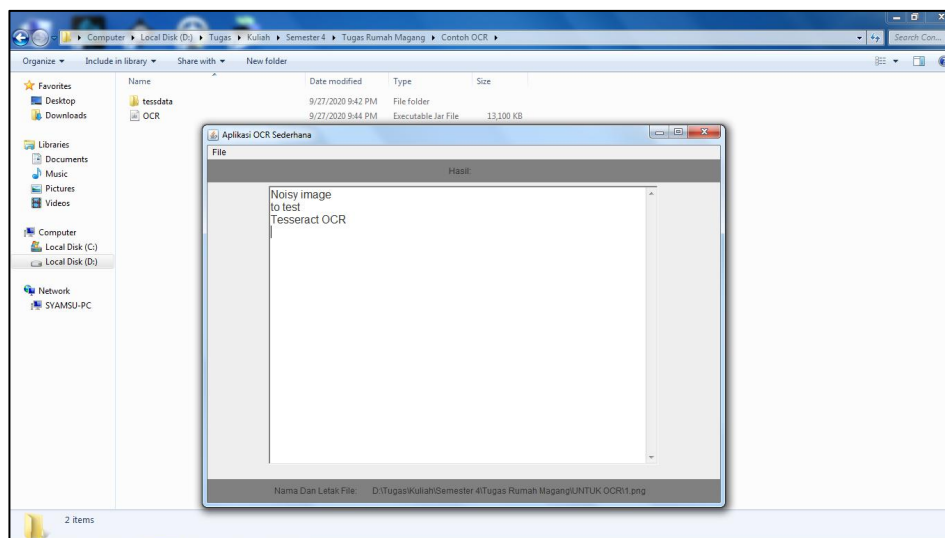
Berikut adalah hasil dari *program* digambarkan dengan *Flowchart*.



Gambar 3.2. Flowchart Program



Gambar 3.3 Jendela Pemilihan Berkas Muncul Setelah Perintah "Scan" Dipilih



Gambar 3.4 Tampilan Program Setelah Berkas Dipilih

4. KESIMPULAN

Berdasarkan uraian pada bab sebelumnya, maka penulis dapat menyimpulkan bahwa: Program OCR (*Optical Character Recognition*) dapat digunakan untuk men-scan atau memindai suatu dokumen fisik jika dokumen tersebut hilang atau rusak pada Kantor Pelayanan Pembendaharaan Negara (KPPN) Palembang. Dan Program OCR (*Optical Character Recognition*) dapat menghemat waktu pembuatan, pemrosesan dan pengetikan dokumen.

5. SARAN

Untuk mendapatkan hasil yang maksimal sebaiknya diperbanyak melakukan percobaan pada banyak jenis objek, karena kesalahan dapat terjadi di proses pasca-pembuatan program. Program ini belum sempurna dan diharapkan dapat dikembangkan lebih lanjut.

UCAPAN TERIMA KASIH

Pada kesempatan ini Kami ucapkan terima kasih kepada semua pihak yang telah memberikan kontribusi di dalam penelitian ini baik moril atau materil. Terima kasih juga kepada Jurnal JUPITER yang telah bersedia untuk mempublish Jurnal Penelitian ini.

DAFTAR PUSTAKA

- [1] Yusuf, Muhammad. 2016. *Bagian 1 – Cara Install JDK (Java Development Kit)*. <https://www.kodingindonesia.com/cara-install-jdk/>, (Diakses pada 26 September 2020)
- [2] Phangtriasu, Michael Reynaldo. 2017. Optical Character Recognition (OCR). <https://mti.binus.ac.id/2017/07/03/optical-character-recognition-ocr/>, (Diakses pada 12 Februari 2021)
- [3] Rao, V., Sasrty, A., Chakracarthy, A., & Kalyanchakravarthi, P. (2016). Optical Character Recognition Technique Algorithms. *Journal of Theoretical and Applied Information Technology*, 83(2), 275-282.

- [4] Apache NetBeans. 2020. *NetBeans IDE 8.1 Information*. <https://netbeans.org/community/releases/81/>, (Diakses pada 26 September 2020)
- [5] Nicomsoft. 2012. *Optical Character Recognition (OCR) – How it works*. <https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/>, (Diakses pada 2 Februari 2021)
- [6] Kay, Anthony. 2007. *Tesseract: an Open-Source Optical Character Recognition Engine*. <https://www.linuxjournal.com/article/9676>, (Diakses pada 2 Februari 2021)
- [7] Wilson, Leslie B. (1993). *Comparative Programming Languages*, Second Edition. Addison-Wesley. ISBN 0-201-56885-3.
- [8] Sobri, Muhammad. *Pengantar Teknologi Informasi - Konsep dan Teori*. ISBN 9789792963120.
- [9] Silberschatz, Abraham (1994). *Operating System Concepts*, Fourth Edition. Addison-Wesley. ISBN 0-201-50480-4.
- [10] Mohammad, F., Anarase, J., Shingote, M., & Ghanwat, P. (2014). *Optical Character Recognition Implementation Using Pattern Matching*. *International Journal of Computer Science and Information Technologies*, 2088-2090.
- [11] Kemal, David dkk. 2018. *Optical Character Recognition (OCR) menggunakan Tesseract dan Penerapannya pada Industri Digital di Indonesia*. <https://mti.binus.ac.id/2018/12/26/optical-character-recognition-ocr-menggunakan-tesseract-dan-penerapannya-pada-industri-digital-di-indonesia/>, (Diakses pada 6 Februari 2021)
- [12] Knuth, Donald E. (1997). *The Art of Computer Programming, Volume 1*, 3rd Edition. Boston: Addison-Wesley. ISBN 0-201-89683-4.
- [13] Knuth, Donald E. (1997). *The Art of Computer Programming, Volume 2*, 3rd Edition. Boston: Addison-Wesley. ISBN 0-201-89684-2.
- [14] Knuth, Donald E. (1997). *The Art of Computer Programming, Volume 3*, 3rd Edition. Boston: Addison-Wesley. ISBN 0-201-89685-0.