Bahasa Algo

by Tedy Setiadi

Submission date: 02-Nov-2020 05:46AM (UTC+0700)
Submission ID: 1432993879
File name: Pembangunan_Bahasa_Algo.docx (636.29K)
Word count: 1761
Character count: 12232

PEMBANGUNAN BAHASA ALGO (BAHASA UNTUK PEMBELAJARAN ALGORITMA PEMROGRAMANA)

Drs. Tedy Setiadi, MT.
Fakultas Teknik Industri, Universitas Ahmad Dahlan

ABSTRAK Proses menulis program menggunakan bahasa pemrograman tidak sekedar hanya menulis suatu instruksi untuk dikerjakan oleh komputer melainkan untuk memecahkan suatu permasalahan yang disebut dengan algoritma. Belajar algoritma jauh lebih sulit daripada belajar bahasa pemrograman. Disisi lain, notasi algoritma berbeda dengan notasi bahasa pemrograman sehingga tidak sedikit pemrograman pemula kesulitan dalam mentraslantasikann algoritma ke dalam bahasa pemrograman. Tujuan penelitian ini adalah membangun bahasa pemrogaman dengan notasi algoritma yang tidak perlu lagi mentranslasikannya ke bahasa pemrograman yang sudah ada (semisal C atau Pascal).

Penelitian ini dimulai dengan mendefinisikan spesifikasi bahasa yang akan dibangun, membuat *recognize* bahasa dalam bentuk *grammar Definition Language* enggunakan *software* bantu ProGrammar, mengimplementasikan bahasa tersebut menjadi *interpreter* menggunakan Visual Basic 6.0 dan tahap akhir dengan menguji program aplikasi bahasa pemrograman ALGO yang berupa *interpreter* dan setelah diuji oleh pengampu mata kuliah algoritma pemrograman dinyatakan layak diterapkan untuk membantu mahasiswa dalam belajar Algoritma darei suatu persoalan pemrograman.

Kata kunci : Bahasa, Algoritma, *Grammar, Interpreter*.

A. PENDAHULUAN

Algoritma adalah bagian yang penting dalam ilmu komputer dan informatika. Banyak cabang ilmu komputer yang di acu dari tcrminologi algoritma salah satunya adalah mata kuliah Algoritma dan Pemrograman. Algoritma dan Pemrograman. Algoritma dan Pemrogaman merupakan salah satu cabang ilmu yang memanfaatkan algoritma untuk menyelesaikan masalah dan memberikn solusi yang nantinya akan diimplementasikan ke dalam berbagai bahasa pemrograman.

Penulisan algoritma independen dari spesifikasi bhasa pemrograman dan komuter yang mengeksekusinya[5]. Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman, sehingga siapa pun dapat membuta notasi algoritma yang berbeda. Hal yang penting mengenai notasi tersebut adalah mudah dibaca dan dimengerti. Kekonsistenan terhadap notasi perlu diperhatikan untuk menghindari kekeliruan karena tidak ada notasi baku dalam penulisan algoritma[4].

Ada beberapa notasi yang digunakan dalam penulisan algoritma. Notasi 1, menyatakan langkah-langkah algoritma dengan untaian kalimat deskriptif. Dengan notasi bergaya kalimat ini deskripsi setiap langkah dijelaskan dengan bahasa yang gamblang.

Proses diawali dengan kata kerja seperti ‘baca’, ‘hitung’, ‘bagi’, ‘ganti’, dan sebagainya, sedangkan pernyataan kondisional dinyatakan dengak ‘jika...maka...’. notasi ini bagus untuk algoritma yang pendek, namun untuk masalah yang algoritmanya besar, notasi ini jelas tidak efisien. Selain itu, pengkonversian notasi algoritma ke notasi bahasa pemrograman cenderung relatif sukar.

Notasi berikutnya menggunakan diagram alir (*flow chart*). Diagram alir populer pada awal-awal era pemrograman dengan komputer (terutama dengan bahasa *Basic Fortran,* dan *Cobol*). Sampai saat ini diagram alir masih digunakan orang. Diagram alir lebih menggambarkan alirn instruksi di dalam program secara visual ketimbang memperlihatkan struktur program. Kotak persegi panjang menyatakan proses, sedangkan pernyataan kondisional dinyatakan dengan bentuk intan (*diamond*). Seperti halnya pada notasi pertama, notasi algoritma dengan diagram alir cocok untuk masalah yang kecil dan tidak cocok untuk masalah yang besar karena membutuhkan berlembar-lembar kertas. Selain itu, pengkoversian notasi algoritma ke notasu bahasa pemrograman cenderung relatif sukar.

Notasi ketiga menggunakan *pseudo-code*. *Pseudocode* (*pseudo* artinya semu atau tidak sebenarnya) adalah notasi yang menyerupai notasi pemrograman tingkat tinggi, khususnya Bahasa *pascal dan C*. Hasil pengamatan memperlihatkan bahwa bahasa pemrograman umumnya mempunyai notasi yang hampir mirip untuk beberapa instruksi, seperti notasi *if-then else, repeat-until, read, write,* dan sebagainya.

Berdasarkan pengamatan tersebut, maka beberapa penulis buku algoritma, mendefinisikan notasi algoritma yang disebut *pseudocode* itu. Tidak seperti bahasa pemrograman yang direpotkan dengan tanda titik koma (*semicolon*), indeks, format keluaran, kata-kata khusus dan sebagainya, sembarang versi *pseudocode* dapat diterima asalkan perintahnya tidak membingungkan.

Agar dapat dijalankan oleh komputer, program dalam notasi algoritma harus ditranslasikan (diterjemahkan) ke dalam notasi bahasa pemrograman yang dipilih sehingga untuk membuat sebuah program diperlukan perancangan algoritma yang kemudian ditranslasikan ke dalam sebuah bahasa pemrograman. Hal ini sering membingungkan para mahasiswa yang mengambil mata kuliah Algoritma dan Pemrograman untuk mentranslasikan algoritma yang telah dibuat ke dalam bahasa pemrograman dikarenakan perbedaan notasi algoritma dengan notasi bahasa pemrograman.

Berdasarkan masalah diatas maka dalam penelitian ini akan dibangun **BAHASA ALGO** dengan harapan dapat memudahkan mahasiswa untuk memahami algoritma yang telah dibuat dan dapat dikompile tanpa harus ditranslasikan ke bahasa pemrograman lain.

B. LANDASAN TEORI

Penelitian sebelumnya [6] telah dibahas tentang pembuatan bahasa pemrograman dengan pendekatan bahasa Indonesia yang dapat menangani fungsi dan prosedur, dapat

menangani tipe data sederhana seperti *string, integer, float* dan *boolean,* pemilihan *if-else* dan *Select-case,* perulangan *for* dan *while* serta *life scope* variabel. Pada penelitian tersebut belum bisa menangani pemilihan *Repeat-until,* pengiriman parameter fungsi dan prosedur secara *reference,* belum dapat menangani kesalahan logika seperti pembagian dengan 0 serta salah menginputkan nilai ke dalam variabel karena perbedaan tipe data.

Dari penelitian tersebut akan dikembangkan bahasa pemrograman dengan pendekatan notasi algoritma dengan tujuan untuk mempermudah dalam pembelajaran Algoritma dan Pemrograman. Dalam penelitian ini sudah menangani perulangan *Repeat-until,* pengiriman parameter fungsi dan prosedur secara *reference,* serta dapat menangani kesalahan logika seperti pembagian dengan 0 serta salah menginputkan nilai ke dalam variabel, karena perbedaan tipe data.

ALGORITMA

Algoritma berisi langkah-langkah penyelesaian masalah. Langkah-langkah tersebut dapat ditulis dalam notasi apapun, asalkan mudah dibaca dan dimengerti, karena memang tidak ada notasi baku dalam penulisan algoritma. Tiap orang dapat membuat aturan penulisan dan notasi algoritma sendiri. Agar notasi algoritma mudah ditranslasi ke dalam notasi bahasa pemrograman, maka sebaiknya notasi algoritma tersebut berkoresponden dengan notasi bahasa pemrograman secara umum.

Penulisan algoritma mempunyai aturan sendiri dan setiap algoritma akan selalu terdiri dari tiga bagian yaitu:

a. Judul (*Header*)
b. Kamus
c. Algoritma
d.

Judul { Komentar mengenai Algoritma seperti cara kerja program, Kondisi awal dan kondisiakhir dari algoritma }
Kamus/ Deklarasi { Pada bagian ini, didefinifikan nama konstanta, nama variable, nama prosedur dan nama fungsi }
Algoritma/ Deskripsi { Pada bagian ini algoritma dituliskan. Semua teks yang dituliskan tidak diantara tandakurung kurawa akan dianggap sebagai notasi algoritma yang akan berpengaruhterhadap kebenaran algoritma }

Listing 1. Bagian-bagian algoritma

C. METODE PENELITIAN

Metode [neleitian perangkat lunk ang dihgunakan adalah *waterfall*,. Adapun tahapanya sebgai bertikut:

1. Analis sistem

   Dalam tahapan ini akan dtentukan spesifikasi bahasa pemrograman yg akan dibangun.

2. Perangcanga Sistem

   a. Perancangan Bahasa

      Perancangan atuan-aturan bahasa yang akan dijadikan bahasa pemrograman. Perancangan-perancaga meliputi,

      1) Pembuatan diagram keadaan.

         Pembutan diagram keadaan (*State Trantiton Diagram*) untuk mendapatkan simbol terminal (token) yang digunakan dalam analisis leksikal dengan menggunakan JFLAP 2.0 Beta.

      2) Pembuatan aturan-aturan Produksi

         Pembuatan aturan-aturan produksi memanfaatkan JFLAP 2.0 Beta yang mampu menkorversi diagram transisi ke dalam bentuk aturan produksi bahasa.

   b. Perancangan Interpreter

      Perancanan interpreter diperlukan untuk memberikan arahan dalam proses pembuatan interpreter. Perancangan-perancangn ini meliputi.

      1) Perancangan anlisis semantik

         Perancangan analisis semantik dilakukan dengan metode *syntax direct translation*.

      2) Perancangan tabel simbol

         Perncangn tabel smbol merupakan ebuah perancangan struktur data yang cocok dengan kebutuhan yang diperlukan selama proses interprestasi berlangsung dan aturan bahasa yang dirancang.

      3) Perancangan IDE (*Intergrated Development Environment*)

         Perancangan IDE meliputi perancangan interface dan perancangan editor program.

3. Implementasi

   Mengimplementasikan rancangan bahasa tersebut mejadi sebuh bahasa pemrograman dengan menggunakan ProGrammar 1.2a dan pembuatan sebuah *interpreter* untuk bahsa tersebut dengan menggunakan Microsoft Visual Basic 6.0.

4. Pengujian

   a. Black Box Test

      Pengujian ini dilakukan oleh dosen pengampu mata kuliah algoritma pemrograman dengan menguji fungsi/fitur dari spesifikasi sistem yang telah ditetapkan sebelumnya.

   b. Alfa Test

      pengujian sistem yang dilakukan oleh para pemakai sehingga dapat diperoleh tanggapan dari pemakai tentang program tersebut, baik dari segi format, tampilan maupun tingkat keramahan programnya. J ika sebagian besar pemakai menyatakan baik dan masukkan maupun keluaramya sesuai dengan

harapan maka sistem tersebut dianggap baik. Pemakai sistem ini adalah mahasiswa yang mengambil mata kuliah algoritma dan pemrograman.

D. HASIL DAN PEMBAHASAN

Spesifikasi dari bahasa ALGO adalah sebagai berikut :

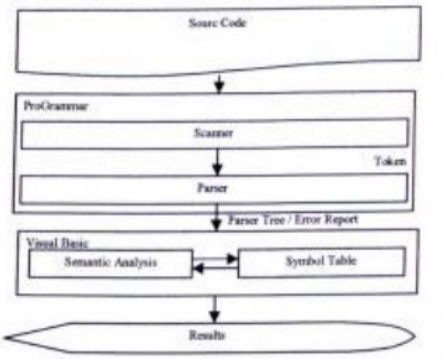
1. *Grammar* yang dibangun menggunakan notasi algoritma
2. dapat menangani input dari keyboard dan menampilkan output di layar
3. mempunyai pendeklarasian variabel, konstanta, fungsi dan prosedur.
4. mampu menangani tipe data *Integer, Real, String* dan *Boolean* dan konstanta.
5. Terdapat fungsi-fungsi bawaan yaitu min, max, avg, sqr, sqrt, ln dan log.
6. menyediakan struktur pemilihan dengan if-then-else dan case.
7. mcnyediakan struktur pengulangan dengan for ,while dan repeat until.
8. menyediakan fasilitas komentar program.
9. dapat menangani kesalahan leksik dan sintaks.

Perancangan Sistem

Tahapan perancangan dilakukan kegiatan-kegiatan di bawah ini.

1. Perancangan tata bahasa ALGO
   a. Blok
      Blok menandakan awal dan akhir dari serangkaian statement-statement, di dalam bahasa ALGO blok diawali oleh keyword “{” dan diakhiri keyword “}”.
   b. Bagian-bagian utama program
      Bagian utama program adalah umtan dan posisi dari nama algoritma, deklarasi dan deskripsi. Urutan ketiga bagian tersebut tidak boleh terbalik atau dihilangkan salah satunya. Nama algoritma bersifat identifier.
   c. Bagian Deklarasi Program
      Bagian Deklarasi diawali oleh keyword “dek” ayng berisi berisi deklarasi variabel, konstanta, fungsi dan prosedur. Variabel, konstanta, fungsi dan prosedur tidak harus ada.

2. Perancangan Interpreter
   Cara kerja ProGrammar dan Visual Basic diilustrasikan seperti gambar di bawah ini.



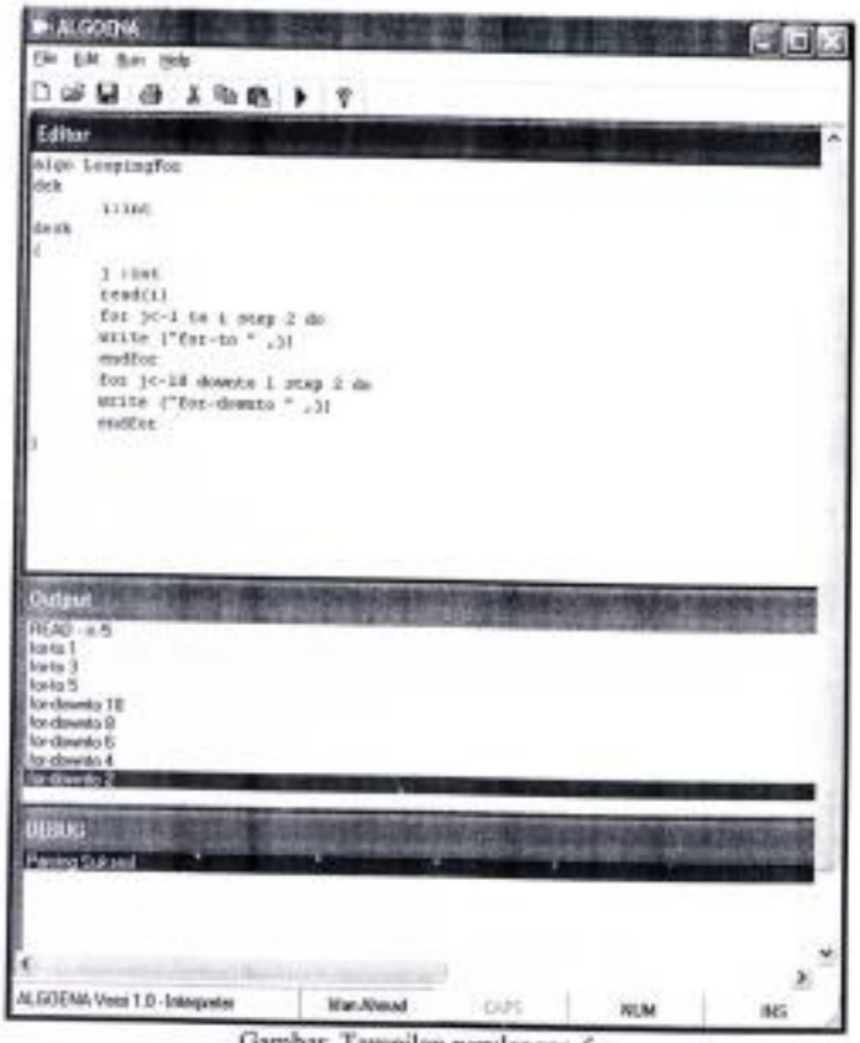
Bagan cara kerja ProGrammar dan Visual Basic pada pembuatan *Interpreter* ALGO

Implementasi Sistem

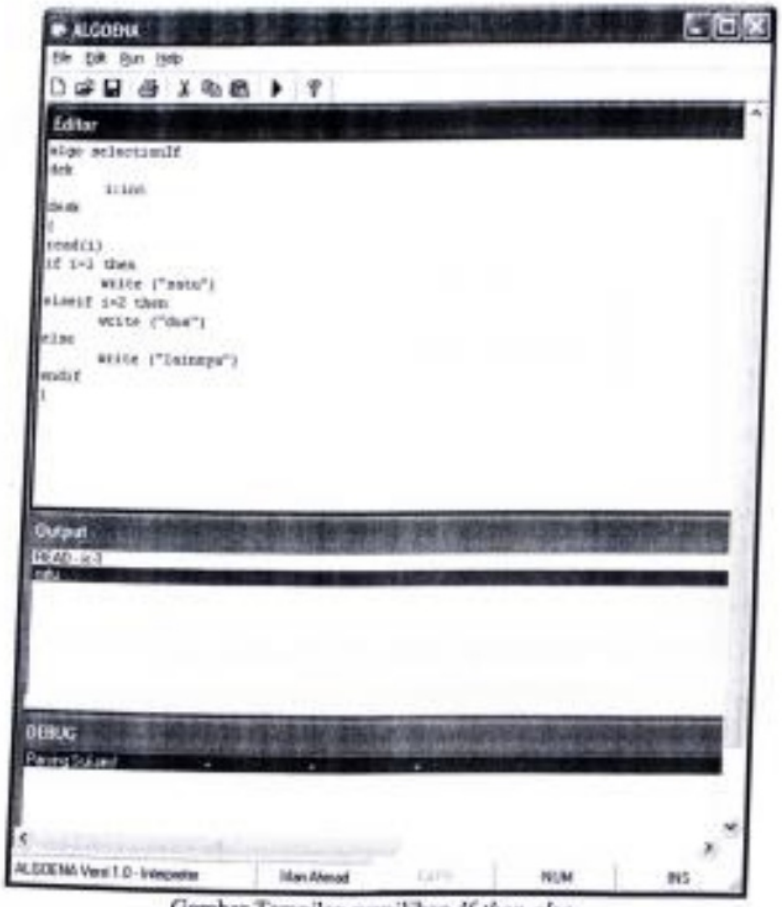
Setelah user memberikan kode sumber dan menjalankannya, kode sumber akan di-parsing oleh ProGrammar. Jika kode sumber sesuai dengan tata bahasa ALGO, maka ProGrammar akan memberikan output berupa pohon *parsing* yang akan dianalisis oleh bagian semantik dan mengeluarkan output sesuai hasil pemaknaan yang dilakukan oleh bagian semantik. Contoh kode sumber yang benar dan hasilnya diperlihatkan pada gambar berikut ini.

Gambar. Tampilan dengan kode sumber yang benar

Gambar Tampilan penggunaan fungsi bawaan ALGO

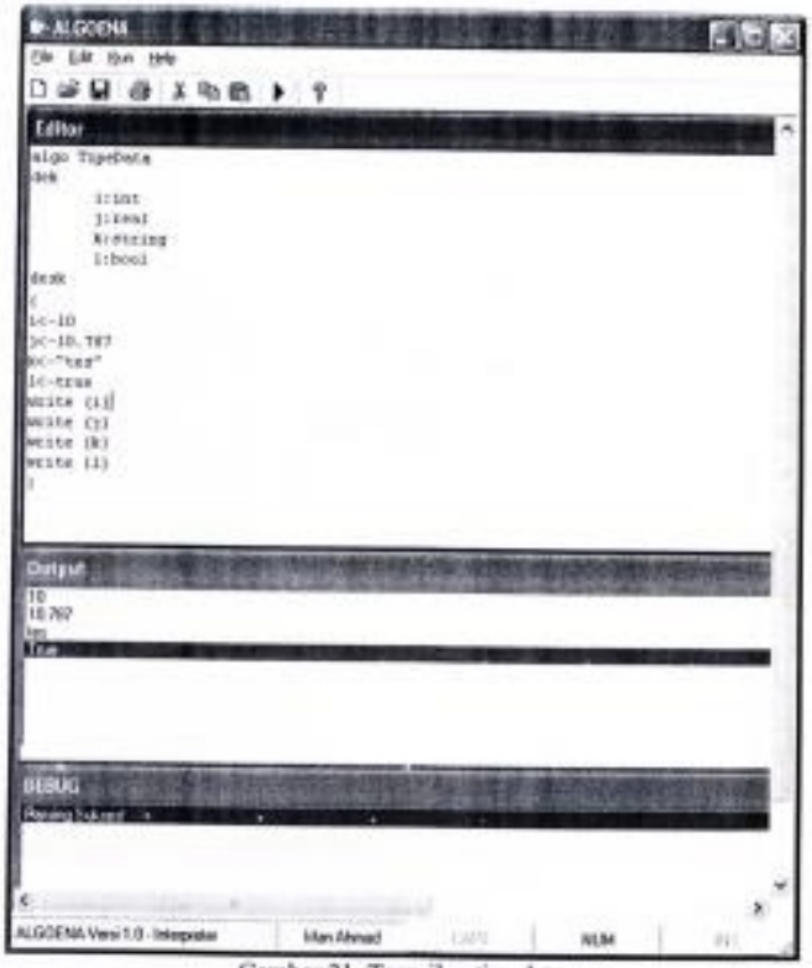


Gambar Tampilan perulangan *for*



Gambar Tampilan pemilihan *If-then-else*

Gambar 20. Tampilan pemilihan *case*



Gambar 21. Tampilan tipe data

Jika sumber tidak sesuai dengan tata bahasa ALGO, maka akan diberikan pesan kesalahan sintakis yang telah disediakan ProGrammar. Contoh kode sumber yang memiliki kesalahan sintakis diperlihatkan pada gambar berikut ini.

Gambar. Tampilan dengan kesalahan sintak

E. KESIMPULAN

Berdasarkan hasil penelitian maka dapat disimpulkan :

1. Telah berhasil dibangun sebuah perangkat lunak (software) bahasa pemrograman yang diberi nama Bahasa ALGO dengan pendekatan notasi algoritma yang dapat digunakan sebagai software bantu dalam pembelajaran algoritma.
2. Aplikasi bahasa pemrograman ini telah diuji coba dengan responden dosen pengampu mata kuliah algoritma pemrograman dengan hasil yang layak digunakan.

SARAN

Saran yang berkaitan dengan sistem dapat dikcmbangkan lagi untara lain:

1. Kemampuan interpreter untuk menangani tipe data kompleks seperti *array* dan *user defined data type.*
2. Kemampuan bahasa untuk mendukung *object oriented programming*.

3. Kemampuan *intepreter* untuk menangani *error* sintak yang lebih baik dan akurat dikarenakan pada penelitian ini untuk pe yang disediakan nanganan error hanya memanfaatkan fasilitas yang disediakan oleh *software* bantu ProGrammar.

F. DAFTAR PUSTAKA

Aho, Alfred V., Ravi Sethi and Jeffrey D. Ullman., 1986. *Compilers, Principles, Techniques and Tools*, Addison-Wesley, Reading, Massachusetts.

Hariyanto, Bambang lr., MT, 2004. *Teori Bahasa Otomata, dan Komputasi serta Terapannya*, Penerbit Informatika, Bandung

Horowitz, Ellis, 1999. *Fundamentals of Computer Algorithms*, Galgotia, NewDelhi

Kristanto, Andri., 2003. *Algoritma dan Pemrograman dengan C++*, Penerbit Graha Ilmu, Yogyakarta

Munir, Rinaldi., 2003. *Algoritma dan Pemrograman Edisi Kedua Revisi*, Buku 1, lnformatika, Bandung

Setiadi Tedy dan Musbadie, Arie., 2006. *Perancangan dan Implementasi Bahasa Pemrograman KINUY (Bahasa Pemrograman Pribumi)*, Teknik lnformatika UAD, Yogyakarta

Pujiyono, Wahyu., 2004. *Diktat Kuliah Algoritma dan Pemrograman*, Program Studi Teknik Informatika, *Yogyakarta*

Schroer, Friedrich Wilhelm. 2005. *The GENTLE Compiler Construction System*, Metarga, Berlin

Utdirartatmo, Firrar., 2005. *Teknik Kompilasi*, Graha Ilmu, Yogyakarta

Wahana Komputer Semarang, 2000, *Pemprograman Visual Basic 6.0*, Penerbit Andi, Yogyakarta

http://en.wikipedia.org/wiki/ALGOL_68

http://www.epeapers.com, Niemann, Thomas, *A Compact Guide To Lex & Yacc*

htgp://www.gregpub.com/algol.html, *The ALGOL Programming Language*

http://www.programmar.com, *ProGrammar Definition Language 1.0* Reference, Norken Technology,

http://www.programmar.com, *ProGrammar Parser Development Toolkit*, Norken, Technology,

http://scifac.ru.ac.za/compilers/, P.D. Terry. 1996. *Compilers and Compiler Generators an introduction with C++*, Rhodes University

http://www.seasys.demon.co.uk/SSP82h/Algol68.html

http://www.thefreecountry.com, *Compailer Construction*.

http://www.wiryana.pandu.org/?id=7 , Wiryana, *Compailer Construction*

Bahasa Algo

ORIGINALITY REPORT

18%	15%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Muhammad Khoiruddin Harahap, Nurul Khairina. "Pencarian Jalur Terpendek dengan Algoritma Dijkstra", SinkrOn, 2017 Publication	2%
2	media.neliti.com Internet Source	2%
3	christantiono.blogspot.com Internet Source	2%
4	sadidauwibi.blogspot.com Internet Source	1%
5	hanifsaeful.wordpress.com Internet Source	1%
6	eprints.akakom.ac.id Internet Source	1%
7	boesta.blogspot.com Internet Source	1%
8	vdocuments.site Internet Source	1%

9	teknik.umk.ac.id Internet Source	1%
10	Submitted to Thomas Jefferson Senior High School Student Paper	1%
11	213.135.146.124 Internet Source	1%
12	education-programmer.blogspot.com Internet Source	1%
13	www.acs.org.au Internet Source	1%
14	pusat-ilmu-umum.blogspot.com Internet Source	1%
15	www.cs.nyu.edu Internet Source	<1%
16	smai.fti.mercubuana-yogya.ac.id Internet Source	<1%
17	lecturer.ukdw.ac.id Internet Source	<1%
18	widuri.raharja.info Internet Source	<1%
19	ocw.upj.ac.id Internet Source	<1%

20	journal.uad.ac.id Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		