

Menentukan 1 Syawal pada Kalender Masehi berdasarkan Algoritma Brute Force

Willy, 13512065

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13512065@itb.ac.id

Abstract—Sejak ribuan tahun yang lalu, kalender telah diciptakan oleh manusia. Di dunia ini terdapat berbagai macam sistem kalender. Tanpa kalender, manusia tidak punya acuan pasti dan semua kegiatannya bergantung pada kondisi alam. Dalam pembuatan kalender, ada beberapa acuan yang digunakan oleh manusia seperti pergerakan bulan, pergerakan bumi, pergerakan matahari, ataupun kombinasinya. Sekarang, kita mengenal istilah Kalender Masehi dan Kalender Hijriyah. Hari Raya Idul Fitri atau yang sering disebut lebaran adalah hari raya yang dirayakan 020umat Muslim pada 1 Syawal di kalender Hijriyah. Bulan Syawal adalah bulan ke-9 di kalender Hijriyah.

Index Terms—Brute Force, Syawal, Lebaran, Idul Fitri.

I. PENDAHULUAN

Dalam kehidupan sehari-hari, kita mengenal adanya bulan Januari, Februari, Maret sampai Desember. Bulan tersebut adalah penamaan bulan yang ada di kalender Masehi.

Kalender Masehi adalah kalender dengan aturan penanggalan sesuai dengan acuan revolusi Bumi terhadap Matahari. Kalender Masehi yang kita gunakan sekarang juga disebut Kalender Gregorian. Kalender Gregorian berlaku sejak 15 Oktober 1582 sampai sekarang, sedangkan sebelum tanggal 15 Oktober 1582, orang-orang bukan menggunakan istilah Kalender Gregorian, melainkan Kalender Julian.

Persamaan Kalender Julian dan Kalender Gregorian adalah sama-sama mempunyai aturan penomoran tahun yang sama. Perbedaannya Kalender Julian dan Kalender Gregorian terletak pada aturan penentuan tahun kabisat.

Pada umumnya 1 tahun pada Kalender Masehi mempunyai 365 hari. Akan tetapi, tahun kabisat pada Kalender Masehi adalah tahun dimana jumlah hari dalam 1 tahun sebanyak 366 hari.

Kalender Julian mempunyai aturan penentuan tahun kabisat sebagai berikut: **Jika suatu tahun habis dibagi 4, maka tahun tersebut merupakan tahun kabisat.** Contoh: tahun 1248 dan tahun 1400 adalah tahun kabisat menurut penanggalan Kalender Julian, sedangkan tahun 1333 dan tahun 1521 bukan tahun kabisat menurut penanggalan Kalender Julian.

Kalender Gregorian mempunyai aturan penentuan tahun kabisat sebagai berikut: **Jika suatu tahun habis dibagi 100, maka tahun tersebut harus juga habis dibagi 400 baru disebut tahun kabisat. Jika suatu tahun tidak habis dibagi 100, maka tahun tersebut harus habis dibagi 4 baru disebut tahun kabisat.** Contoh: tahun 2000 dan tahun 2012 adalah tahun kabisat menurut penanggalan Kalender Julian, sedangkan tahun 1700, 1800, 1900 dan tahun 1945 bukan tahun kabisat menurut penanggalan Kalender Julian.

Mengapa terjadi perubahan aturan penanggalan pada bulan Oktober 1582 dari Kalender Julian ke Kalender Gregorian? Telah disebutkan sebelumnya bahwa Kalender Julian dan Kalender Gregorian mempunyai aturan penanggalan sesuai dengan acuan revolusi Bumi terhadap Matahari.

Penanggalan Kalender Julian menggunakan perhitungan bahwa lamanya revolusi Bumi terhadap Matahari adalah 365,25 hari. Oleh sebab itu, aturan tahun kabisat dibuat 4 tahun sekali karena dianggap rata-rata waktu 4 tahun sama dengan 365,25 hari $((365+365+365+366)/4 = 365,25)$.

Ternyata revolusi Bumi terhadap 365,25 kurang presisi atau akurat. Oleh sebab itu, digunakanlah aturan penanggalan Kalender Gregorian.

Penanggalan Kalender Gregorian menggunakan perhitungan bahwa lamanya revolusi Bumi terhadap Matahari adalah 365,2425 hari (lebih presisi dari 365,25 hari). Oleh sebab itu, aturan tahun kabisat dibuat 4 tahun sekali sekali dan tahun yang kelipatan 100 tetapi bukan kelipatan 400 tahun bukan tahun kabisat karena dianggap rata-rata waktu 400 tahun sama dengan 365,2425 hari $((365,25*400 - 3)/400 = 365,2425)$.

Pengumuman perubahan aturan dari Kalender Julian ke Gregorian, tepat dilakukan tanggal 4 Oktober 1582, dan untuk menyesuaikan posisi Matahari akibat perhitungan revolusi yang sebelumnya kurang presisi, maka diputuskanlah keesokan harinya bukan 5 Oktober 1582, tetapi 15 Oktober 1582 (kehilangan 10 hari).

II. TEORI DASAR

2.1 Algoritma Brute Force

^[1]Algoritma Brute Force adalah penyelesaian suatu

permasalahan dengan pendekatan yang cukup jelas (*straightforward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah dan definisi konsep yang dilibatkan. Algoritma brute force ini bukanlah algoritma yang cerdas ataupun efisien, karena sifat dari algoritma brute force ini adalah mengenumerasi semua kemungkinan yang ada secara sistematis, kemudian setiap kemungkinan tersebut dievaluasi sampai mendapatkan solusi sesuai dengan kriteria yang kita ingin inginkan.

2.1.1 Karakteristik Algoritma Brute Force

^[1]Algoritma Brute Force mempunyai karakteristik sebagai berikut:

1. Seperti yang telah disebutkan sebelumnya, algoritma brute force ini umumnya tidak cerdas dan tidak mangkus karena membutuhkan langkah yang banyak dalam menyelesaikan sebuah permasalahan. Hal ini menjadi masalah jika ukuran input semakin besar, maka waktu ataupun langkah yang dibutuhkan untuk menyelesaikan akan meningkat.
2. Untuk masalah yang cukup sederhana, biasanya orang lebih mempertimbangkan Algoritma Brute Force karena pengaruh efisiensi algoritma untuk masalah yang sederhana (inputnya kecil) tidak berbeda jauh. Selain itu, dalam memprogram, algoritma ini lebih mudah diimplementasi dibandingkan dengan algoritma yang lebih efisien.
3. Algoritma Brute Force pada umumnya kurang disukai karena harus mengenumerasi lebih banyak langkah untuk mendapatkan suatu solusi dibandingkan algoritma-algoritma lain pada umumnya.
4. Algoritma Brute Force bukanlah algoritma yang efisien, akan tetapi algoritma ini biasanya dapat diterapkan di sebagian besar permasalahan yang ada.
5. Algoritma Brute Force biasanya digunakan sebagai perbandingan terhadap algoritma tertentu dengan tujuan untuk mengetahui apakah algoritma tersebut efisien atau tidak. Contoh: untuk menghitung nilai a^n , penyelesaian dengan Algoritma Brute Force membutuhkan kompleksitas waktu $O(n)$ sedangkan penyelesaian dengan Algoritma Divide And Conquer membutuhkan kompleksitas waktu $O(\log n)$. Dengan demikian, dapat diambil kesimpulan bahwa Algoritma Divide and Conquer cukup efisien dibandingkan Algoritma Brute Force.

2.1.2 Kekuatan Algoritma Brute Force

Algoritma Brute Force mempunyai beberapa kekuatan sebagai berikut:

1. Mudah dimengerti karena kesederhanaan algoritma tersebut.
2. Mudah diimplementasi karena penyelesaian permasalahannya tidak ribet dan cenderung *straightforward*
3. Algoritma ini dapat menyelesaikan hampir semua permasalahan yang ada.

2.1.3 Kelemahan Algoritma Brute Force

Kelemahan dari algoritma ini adalah algoritma ini cukup sederhana dan umumnya metode penyelesaiannya mudah ditebak sehingga solusinya cenderung tidak lebih efisien dibandingkan algoritma lainnya

2.2 Kalender Hijriyah

Kalender Hijriyah adalah kalender dengan aturan penanggalan sesuai dengan acuan rotasi dan revolusi Bulan terhadap Bumi. 1 Kali revolusi Bulan terhadap Bumi kira-kira 29,5 hari. Oleh sebab itu kita bisa lihat dari table di bawah bahwa jumlah hari selang-seling Antara 30 hari dan 29 hari. Adanya 1 hari di tahun kabisat karena adanya koreksi dari revolusi Bulan yang tidak tepat 29,5 hari, melainkan 29 hari 12 jam 44 menit 2,5 detik. Jadi, setiap 1 bulan mengalami selisih $((44+2,5/60)/60/24 = 0,0305844$ hari. Jadi jika 30 tahun (360 bulan), maka mengalami selisih $360 * 0,0305844 = 11$ hari. Inilah alasan mengapa setiap 30 tahun, ada **11 kali** tahun kabisat. Aturan penentuan tahun kabisat di Kalender Hijriyah berbeda dengan aturan penentuan tahun kabisat di Kalender Masehi. Jika tahun Hijriyah tersebut dibagi 30 bersisa 2, 5, 7, 10, 13, 16, 18, 21, 24 26, 29, maka tahun tersebut adalah tahun kabisat. 1 Muharram 1 Hijriyah jatuh pada tanggal 16 Juli 622 Masehi. Pada penulisan makalah ini, penulis menggunakan jumlah hari dalam tiap bulan seperti tabel dibawah:

Bulan	Jumlah Hari
Muharram	30
Safar	29
Rabiul Awal	30
Rabiul Akhir	29
Jumadil Awal	30
Jumadil Akhir	29
Rajab	30
Syaban	29
Ramadhan	30
Syawal	29
Dzulkaidah	30
Dzulhijjah	29(basit)/30(kabisat)

Tabel 2.1. Jumlah hari dalam Kalender Hijriyah

Jumlah hari dalam kalender Hijriyah adalah 354 hari untuk tahun basit dan 355 hari untuk tahun kabisat.

Kalender ini mempunyai lama 1 daur sebanyak 30 tahun Hijriyah ($= 19*354+11*355 = 10631$ hari).

2.3 Kalender Masehi

Pada penulisan makalah ini, penulis menggunakan jumlah hari dalam tiap bulan seperti tabel dibawah:

Bulan	Jumlah Hari
Januari	31
Februari	28(basit)/29(kabisat)

Maret	31
Akhir	30
Mei	31
Juni	30
Juli	31
Agustus	31
September	30
Oktober	31
November	30
Desember	31

Tabel 2.2. Jumlah hari dalam Kalender Masehi

Jumlah hari dalam kalender Masehi adalah 365 hari untuk tahun basit dan 366 hari untuk tahun kabisat.

Kalender ini mempunyai lama 1 daur sebanyak 400 tahun Masehi ($= 400 \times 365,25 - 3 = 146097$ hari).

2.4 Julian Date (JD)

Julian Date adalah angka yang menandakan jumlah hari sejak tanggal 1 Januari -4712 (atau 4713 SM). Contoh: Julian Date untuk tanggal 1 Januari 2014 adalah 2456672, artinya jumlah hari antara 1 Januari -4712 dan 1 Januari 2014 adalah 2456672 hari.

III. ALGORITMA

A. Algoritma menghitung Julian Date

Cara untuk mendapatkan Julidan Date tanggal d, bulan m, tahun y pada kalender Masehi adalah:

1. Cari JD 1 Januari pada tahun yang sama, yaitu tahun y. JD tersebut adalah $= \text{roundup}(4712+y) \times 365,25$.
2. Kemudian tambahkan JD pada langkah 1 dengan **jumlah hari** dari 1 Januari tahun y ke tanggal y, bulan m, tahun y.

Setelah itu, lakukan koreksi sebagai berikut:

1. Jika tanggal d, bulan m, tahun d adalah setelah atau sama dengan 15 Oktober 1582, koreksi dengan kurangi JD yang didapatkan dengan **10 hari** karena hilangnya 5 Oktober 1582 – 14 Oktober 1582.
2. Jika tahun y adalah setelah atau sama dengan tahun 1700, lakukan koreksi dengan mengurangi JD sebanyak jumlah tahun yang habis dibagi 100 tetapi tidak habis dibagi 400 dengan perhitungan mulai dari tahun 1700 sampai tahun y. (karena ketika kita menghitung JD 1 Januari tahun y, kita mengalikan 365,25 hari yang adalah rata-rata tiap 4 tahun). Koreksi dilakukan ketika sudah berlaku Kalender Gregorian. Contoh: jika tahun 2014, maka kurangi **3 hari** (tahun 1700, 1800, 1900).

B. Algoritma menghitung 1 Syawal

Cara untuk menghitung 1 Syawal pada tahun y di Kalender Masehi adalah sebagai berikut:

1. Hitung JD 16 Juli 622 dengan cara di bagian IIIA. JD yang kita dapatkan adalah JD 1 Muharram 1H (karena 16 Juli 622 = 1 Muharram 1H)

2. Perbedaan jumlah hari dari 1 Muharram dan 1 Syawal adalah 266 hari, maka JD yang didapatkan di tahap sebelumnya ditambah dengan 266 hari. Sekarang, JD yang kita dapatkan adalah JD pada tanggal 1 Syawal 1H.
3. Hitung JD 1 Januari tahun y di bagian IIIA.
4. Hitunglah JD pada tanggal 1 Syawal tahun y' dimana y' adalah tahun berapapun pada kalender Hijriyah yang membuat JD 1 Syawal tahun y' adalah JD paling kecil yang nilainya lebih besar dari JD yang didapatkan di nomor 3. Contoh: Jika JD yang di nomor 3 adalah 1948974 dan JD 1 Syawal 1H adalah 1948706, maka JD 1 Syawal 2H ($1948706+354=1949060$) adalah JD paling kecil yang nilainya lebih besar dari 1948974. Contoh lainnya adalah Jika JD yang di nomor 3 adalah 1949340 dan JD 1 Syawal 1H adalah 1948706, maka JD 1 Syawal 3H ($1948706+354+355=1949415$) adalah JD paling kecil yang nilainya lebih besar dari 1949340.
5. Hitunglah selisih JD dengan cara JD yang didapatkan di nomor 4 kurang JD yang didapatkan di nomor 3.
6. 1 Januari tahun y di kalender Masehi ditambah dengan jumlah hari yang didapatkan di nomor 5 adalah tanggal 1 Syawal. Contoh: Pada tahun 2014, hasil yang didapatkan dari nomor 5 adalah 209, maka 209 hari setelah 1 Januari 2014 adalah 29 Juli 2014, artinya tanggal 29 Juli 2014 tersebut adalah tanggal 1 Syawal di Kalender Hijriyah.
7. Ada kemungkinan terjadi 2 kali tanggal 1 Syawal di suatu tahun Masehi yang sama. Jika hasil yang didapatkan di nomor 5 ditambah dengan jumlah hari dalam 1 tahun Hijriyah (apakah kabisat atau tidak bergantung pada tahun berapa Hijriyah di tahun Masehi y) kurang dari jumlah hari di tahun y, maka terjadi 2 kali tanggal 1 Syawal di tahun y tersebut.

IV. CONTOH IMPLEMENTASI

Berikut kita implementasikan algoritma pada bagian 3 pada perhitungan 1 Syawal di tahun Masehi 2014.

1. Pertama-tama, kita mencari JD 16 Juli 622M. JD pada tanggal 16 Juli 622M / 1 Muharram 1H = 1948440
2. JD 1 Syawal 1H = $1948440 + 266 = 1948706$.
3. JD 1 Januari 2014 = $2456672 - 10 - 3 = 2456659$.
4. Perhatikan tabel dibawah

Tanggal Hijriyah	JD
1 Syawal 1H	1948706
1 Syawal 2H	1949060
1 Syawal 3H	1949415(tambah 355 hari karena tahun 2H adalah tahun Masehi)
dst	dst
1 Syawal 1434H	2456513
1 Syawal 1435H	2456868
1 Syawal 1436H	2457222

Tabel 4.1. Tanggal Hijriyah vs JD

- Jadi, JD paling kecil yang lebih besar 2456659 adalah JD pada tanggal 1 Syawal 1435H (JD = 2456868).
- Kurangi JD yang didapatkan pada nomor 4 dengan JD yang didapatkan pada nomor 3.
Selisih = 2456868 – 2456659 = 209.
 - Dari hasil nomor 5, artinya, 1 Syawal 1435H jatuh pada 209 hari setelah 1 Januari 2014, yaitu **29 Juli 2014**.
 - Jumlah hari pada tahun 1435H adalah 354 hari. Jumlah hari pada tahun 2014M adalah 365 hari. Karena 209 + 354 >= 365, maka tanggal 1 Syawal hanya terdapat 1 kali pada tahun 2014.

V. KODE PROGRAM

Berikut adalah implementasi algoritma dalam bahasa pemrograman c++ dengan input tahun Masehi dan output 1 Syawal yang jatuh pada kalender Masehi:

```
#include <cstdio>
#include <iostream>
#include <cmath>
using namespace std;

typedef struct {
    int day;
    int month;
    int year;
}date;

typedef struct {
    int bedaHari;
    int yHijriyah;
} tb;

bool kabisatHijriyah (int year) {
    return ((year%30==2) ||
(year%30==5) || (year%30==7) ||
(year%30==10) || (year%30==13) ||
(year%30==16) || (year%30==18) ||
(year%30==21) || (year%30==24) ||
(year%30==26) || (year%30==29));
}

int numberOfDaysHijriyah (int month,
int year) {
    if (month == 12 &&
kabisatHijriyah(year)) return 29;
    if (month%2) return 30; else
return 29;
}

bool kabisatGregorian(int year) {
    if (year%4 == 0) {
        if (year%100 == 0) return
(year%400 == 0);
        else return true;
    } else return false;
}

bool kabisatJulian(int year) {
    return (year%4==0);
}
```

```
}

int numberDaysInAYear (int year) {
    if (year < 1582) {
        if (kabisatJulian(year))
return 366;
        else return 365;
    } else {
        if
(kabisatGregorian(year)) return 366;
        else return 365;
    }
}

int numberOfDaysMasehi (int month, int
year) {
    if (month == 2) {
        if (year < 1582) {
            if
(kabisatJulian(year)) return 29;
            else return 28;
        } else {
            if
(kabisatGregorian(year)) return 29;
            else return 28;
        }
    } else
        if (month == 1 || month ==
3 || month == 5 || month == 7 || month
== 8 || month == 10 || month == 12)
            return 31;
        else return 30;
}

int JD(date d) {
    if (d.year == 1582 && d.month ==
7) {
        if (d.day >=5 && d.day
<=14) { //tidak ada tanggal 5 okt 1582
- 14 okt 1582
            throw "tanggal tidak
valid";
        }
    }
    int numOfJD =
ceil(((4712+d.year)*365.25)); //jumlah
JD sampai 1 januari tahun tersebut

    if (d.month > 1) {
        for (int i = 1; i <
d.month; ++i) { //jumlah sampai
tanggal 1 bulan tersebut
            numOfJD +=
numberOfDaysMasehi(i, d.year);
        }

        numOfJD += (d.day - 1); //total
JD.

        /* koreksi akibat pengaruh
kalender julian ke gregorian */
}
```

```

        if (d.year > 1582) {
            numOfJD -= 10; //adanya
            pengurangan 10 hari pada bulan oktober
            1582
        } else if (d.year == 1582) {
            if (d.month > 10) {
                numOfJD -= 10;
            } else if (d.month == 10)
        {
            if (d.day >= 15) {
                numOfJD -= 10;
            }
        }

        if (d.year > 1700) {
            for (int i = 1700; i <
d.year; i += 100) {
                if (i % 400) { //tidak
                habis dibagi 400
                    --numOfJD;
                }
            }
        }
        return numOfJD;
    }

    tb diffDayFrom1Jan (int yearMasehi) {
        date dM = {1,1,yearMasehi};
        int JDM = JD (dM);
        date dH = {16,7,622};
        int JDH = JD (dH); //dpt JD 1
        Muharram 1H
        int yH = 1; //tahun 1 H

        for (int i = 1; i < 10; ++i) {
            JDH +=
            numberOfDaysHijriyah (i, yH);
        } //dpt JD 1 Syawal 1H

        while (true) {
            if (JDH >= JDM) {
                //printf("JDH =
%d\n", JDH);
                tb t = {JDH -
                JDM, yH};
                return t;
            } else {
                JDH +=
                kabisatHijriyah (yH) ? 355 : 354;
                ++yH;
            }
        }

        date getAbsoluteDay (int numberOfDay,
int year) {
            date dtemp = {1,1,year};
            while
            (numberOfDaysMasehi (dtemp.month,
dtemp.year) < numberOfDay) {
                numberOfDay -=
                numberOfDaysMasehi (dtemp.month,
dtemp.year);

```

```

            ++dtemp.month;
        }
        dtemp.day += numberOfDay;
        return dtemp;
    }

    int main() {
        int y;
        do {
            cout << "Masukkan tahun
Masehi (minimal tahun 622M): ";
            cin >> y;
            if (y < 622) cout <<
            "Masukkan tahun salah!!" << endl;
        } while (y < 622);
        if (y == 622) {
            cout << "1 Syawal 1H
terdapat pada: 16-7-622" << endl;
        } else {
            int beda =
            diffDayFrom1Jan(y).bedaHari;
            int yH =
            diffDayFrom1Jan(y).yHijriyah;
            date d =
            getAbsoluteDay(beda, y);
            cout << "1 Syawal " << yH
            << "H terdapat pada: " << d.day << "-"
            << d.month << "-" << d.year << endl;

            if (beda +
            (kabisatHijriyah(yH) ? 355 : 354) <
            numberDaysInAYear(y)) {
                beda +=
                (kabisatHijriyah(yH) ? 355 : 354);
                date d1 =
                getAbsoluteDay(beda, y);
                cout << "1 Syawal "
                << yH+1 << "H terdapat pada: " <<
                d1.day << "-" << d1.month << "-" <<
                d1.year << endl;
            }
        }
        return 0;
    }

```

V. HASIL OUTPUT PROGRAM

Berikut adalah tabel hasil program jika diinput tahun antara tahun 2000-2014 beserta perbandingan dengan tanggal yang ditetapkan di dunia nyata:

Tahun	Output Program	Kenyataan ^[6]
2000	8 Januari 2000 28 Desember 2000	8 Januari 2000 28 Desember 2000
2001	17 Desember 2001	17 Desember 2001
2002	6 Desember 2002	6 Desember 2002
2003	26 November 2003	26 November 2003
2004	14 November 2004	14 November 2004
2005	3 November 2005	4 November 2005
2006	24 Oktober 2006	24 Oktober 2006
2007	13 Oktober 2007	13 Oktober 2007
2008	2 Oktober 2008	2 Oktober 2008

2009	21 September 2009	21 September 2009
2010	10 September 2010	10 September 2010
2011	31 Agustus 2011	31 Agustus 2011
2012	19 Agustus 2012	19 Agustus 2012
2013	8 Agustus 2013	8 Agustus 2013
2014	29 Juli 2014	28 Juli 2014

Bandung, 18 Mei 2014



Willy
13512065

VI. KETIDAKPASTIAN DENGAN DUNIA NYATA

Pada keadaan sebenarnya, bulan Ramadhan mungkin saja hanya 29 hari. Dalam hal ini, penulis mengambil asumsi bulan Ramadhan selalu 30 hari. Oleh sebab itu, mungkin terjadi kesalahan sebesar 1 hari karena faktor tersebut. Contohnya pada tahun 2014, secara perhitungan, 1 Syawal berada pada tanggal 29 Juli 2014, akan tetapi pada kalender tertulis tanggal 28 Juli 2014.

Pada penulisan makalah ini, penulis hanya mencoba membantu memaparkan penghitungan 1 Syawal berdasarkan algoritma brute force, bukan mempengaruhi pihak apapun menentukan tanggal 1 Syawal untuk tahun-tahun ke depannya.

VII. UCAPAN TERIMA KASIH

Dalam makalah ini, penulis mengucapkan terima kasih kepada banyak pihak yang telah membantu menyelesaikan penulisan tersebut, khususnya kepada orang tua yang selalu memberi dukungan, Tuhan Yang Maha Esa, Ibu Masayu Leylia Khodra dan Bapak Rinaldi Munir sebagai dosen mata kuliah IF2211 (Strategi Algoritma), teman-teman Teknik Informatika 2012 dan semua pihak yang tidak bisa disebutkan satu per satu.

DAFTAR REFERENSI

- [1] Munir, Rinaldi. *Diktat Kuliah IF2211 Strategi Algoritma*. 2009. Bandung: Teknik Informatika ITB.
- [2] <http://dedyjsn.wordpress.com/2008/03/24/algoritma-brute-force/> diakses pada 18 Mei 2014 jam 20.00 WIB
- [3] <http://ramadhanakurnia.blogspot.com/2013/06/sejarah-dan-ketentuan-tahun-hijriyah.html> diakses pada 18 Mei 18.00 WIB
- [4] <http://hidayahdakwah.blogspot.com/2013/07/perbedaan-perhitungan-kalender-hijriah.html> diakses pada 18 Mei 20.30 WIB
- [5] Karttunen, H, dkk. *Fundamental Astronomy 5th Edition*. 2007. New York: Springer Berlin Heidelberg.
- [6] <http://www.timeanddate.com/holidays/indonesia/idul-fitri> diakses pada 19 Mei 01.00 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.