

Algoritma mengukur keberadaan pensyarah Universiti Teknologi MARA

Khairul Nizam Abd Halim

Fakulti Sains Komputer dan Matematik, Universiti Teknologi MARA Pahang,
26400 Bandar Tun Razak Jengka, Pahang, Malaysia
khairulnizam.uitm@gmail.com

Abstrak: UiTM mensyaratkan jam keberadaan pensyarahnya di pejabat atau di kampus perlu cukup 38.5 jam seminggu. Bagi mengukur jam keberadaan pensyarah, mesin kad perakam waktu digunakan. Masalah yang timbul terhadap sistem ini adalah kesukaran para pensyarah juga pengurusan Universiti Teknologi MARA (UiTM) mengira jumlah jam keberadaan dengan tepat dan rasional. Ini kerana sistem kehadiran ini tidak dilengkapi dengan fungsi pengiraan jam keberadaan 38.5 jam seminggu. Di dalam kertas penyelidikan ini fungsi pengiraan jam keberadaan 38.5 jam seminggu diperkenalkan. Fungsi ini digunakan bagi menghasilkan model sistem kehadiran yang baru juga prototaip sistem kehadiran berkomputer yang diberi nama *UiTM Time Card Calculator*. Algoritma fungsi pengiraan jam keberadaan 38.5 jam seminggu diuji menggunakan set data yang dicipta khusus untuk membuktikan ianya cekap dan mampu melaksanakan fungsi yang ditetapkan. Hasil pengujian mendapati ia berjaya menyelesaikan masalah pengiraan dan dapat memberi kelebihan membuat pengiraan dengan tepat dan rasional disamping mengutamakan masa keberadaan yang fleksibel. Dengan ini, diharap para pensyarah dapat menjaga kualiti serta integriti sebagai seorang pensyarah, dan algoritma ini dapat menjadi pencetus kepada kecemerlangan akademik di universiti.

Kata Kunci: algoritma, keberadaan, pensyarah, sistem kehadiran

1. Pengenalan

Pekeliling Akademik Universiti Teknologi MARA mensyaratkan jam keberadaan pensyarah di pejabat atau di kampus perlu cukup 38.5 jam seminggu (Pekeliling Timbalan Naib Cancellor, 2001, 2003). Bagi mengukur jam keberadaan pensyarah, mesin kad perakam waktu digunakan. Masalah yang timbul terhadap sistem ini adalah kesukaran para pensyarah juga pengurusan Universiti Teknologi MARA (UiTM) mengira jumlah jam keberadaan dengan tepat dan rasional. Ini kerana sistem kehadiran ini tidak dilengkapi dengan fungsi pengiraan jam keberadaan 38.5 jam seminggu.

Terdapat banyak jenis sistem kehadiran daripada yang berjenis tradisional seperti tandatangan, kad biasa, kad magnetik, kad bercip dan kad RF sehinggalah berjenis moden yang menggunakan biometrik (Zhang Yongqiang, Liu Ji, 2006). Semua sistem kehadiran ini tidak mempunyai fungsi pengiraan jam keberadaan 38.5 jam seminggu. Ini kerana fungsi ini menumpu kepada pengiraan kehadiran yang piawai iaitu bermula jam 8 pagi sehingga jam 5 petang dan ianya bukan untuk mengukur masa keberadaan pensyarah yang menuntut masa keberadaan yang lebih fleksibel disamping memenuhi kehendak UiTM melalui pekelingnya. Di dalam kertas penyelidikan ini fungsi pengiraan jam keberadaan 38.5 jam seminggu diperkenalkan dan ianya berjaya menyelesaikan masalah pengiraan. Ianya dapat memberi kelebihan membuat pengiraan dengan tepat dan rasional disamping mengutamakan masa keberadaan yang fleksibel.

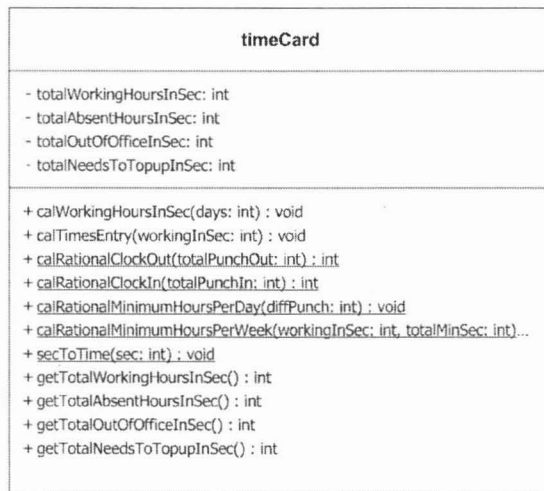
Bagi menghasilkan fungsi pengiraan jam keberadaan 38.5 jam seminggu untuk sistem kehadiran, kertas penyelidikan ini meletakkan objek kajian iaitu untuk (1) menghasilkan algoritma mengukur keberadaan pensyarah UiTM, (2) menilai algoritma berkenaan, (3) menghasilkan model sistem kehadiran yang baru untuk pensyarah UiTM.

2. Metodologi

Penyelidikan ini mempunyai dua bahagian metodologi, iaitu bahagian pertama untuk mereka bentuk model sistem kehadiran menggunakan fungsi mengira jam keberadaan 38.5 jam seminggu. Seksyen 2.1, 2.2, 2.3 dan 2.4 adalah merujuk kepadanya. Bahagian kedua pula untuk menguji model berkenaan menggunakan set data yang dicipta khusus untuk membuktikan ia cekap dan mampu melaksanakan fungsi yang ditetapkan. Seksyen 2.5 adalah merujuk kepadanya. Bagi memudahkan pengujian terhadap model, satu prototaip sistem kehadiran dibangunkan berdasarkannya. Prototaip ini diberi nama *UiTM Time Card Calculator*. Prototaip ini boleh diperolehi secara percuma di <http://www.khairulnizamabdhalim.blogspot.com/2014/06/uitm-time-card-calculator.html>.

2.1 Diagram kelas untuk fungsi mengira jam keberadaan 38.5 jam seminggu

Rajah 1 adalah diagram kelas yang menggunakan pendekatan reka bentuk *Unified Modeling Language* (UML) adalah untuk fungsi (kelas) mengira jam keberadaan 38.5 jam seminggu. Kelas ini digunakan untuk pembangunan model sistem kehadiran pensyarah UiTM. Kelas ini mampu menghasilkan kelas *instances* seperti di Rajah 2. Operasi-operasi (*member function*) di dalam kelas ini membantu menghasilkan data seperti (1) jumlah jam bekerja seminggu, (2) jumlah jam untuk hari bercuti biasa, bercuti sakit atau bekerja luar pejabat, (3) jumlah jam keluar pejabat yang bersifat peribadi, dan (4) baki jam memenuhi syarat keberadaan 38.5 jam seminggu. *Person 1* dan *Person 2* adalah contoh *instances* yang boleh dihasilkan daripada reka bentuk diagram kelas di Rajah 1.



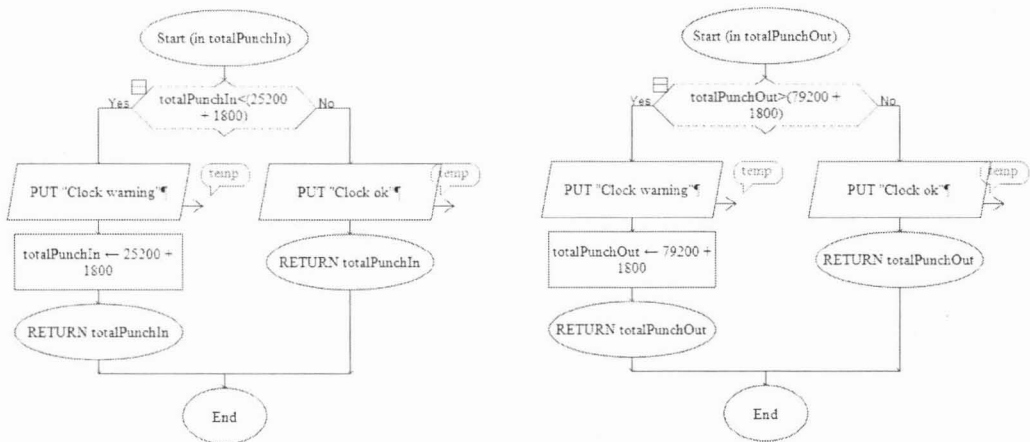
Rajah 1 Diagram kelas mengira keberadaan 38.5 jam seminggu

timeCard : Person 1	timeCard : Person 2
- totalWorkingHoursInSec: 144000 - totalAbsentHoursInSec: 0 - totalOutOfOfficeInSec: 540000 - totalNeedsToTopupInSec: -3600	- totalWorkingHoursInSec: 139052 - totalAbsentHoursInSec: 27720 - totalOutOfOfficeInSec: 540000 - totalNeedsToTopupInSec: -40250
+ calWorkingHoursInSec(days: int) : void + calTimesEntry(workingInSec: int) : void + calRationalClockOut(totalPunchOut: int) : int + calRationalClockIn(totalPunchIn: int) : int + calRationalMinimumHoursPerDay(diffPunch: int) : void + calRationalMinimumHoursPerWeek(workingInSec: int, totalMinSec: int)... + secToTime(sec: int) : void + getTotalWorkingHoursInSec() : int + getTotalAbsentHoursInSec() : int + getTotalOutOfOfficeInSec() : int + getTotalNeedsToTopupInSec() : int	+ calWorkingHoursInSec(days: int) : void + calTimesEntry(workingInSec: int) : void + calRationalClockOut(totalPunchOut: int) : int + calRationalClockIn(totalPunchIn: int) : int + calRationalMinimumHoursPerDay(diffPunch: int) : void + calRationalMinimumHoursPerWeek(workingInSec: int, totalMinSec: int)... + secToTime(sec: int) : void + getTotalWorkingHoursInSec() : int + getTotalAbsentHoursInSec() : int + getTotalOutOfOfficeInSec() : int + getTotalNeedsToTopupInSec() : int

Rajah 2 Contoh kelas instances

2.2 Algoritma operasi rasional masa masuk dan masa balik

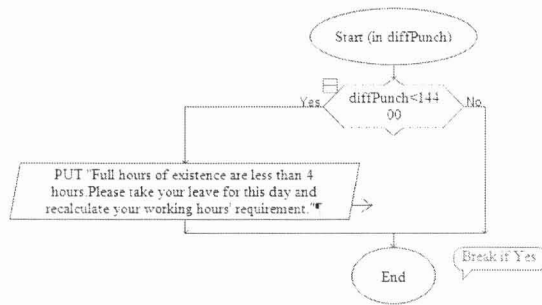
Rajah 3 adalah merujuk kepada algoritma operasi (member function) rasional masa masuk dan masa balik. Algoritma ini hanya membenarkan masa keberadaan bermula dari pukul 7.30 pagi sehingga 10.30 malam.



Rajah 3 Algoritma operasi rasional masa masuk dan masa balik

2.3 Algoritma operasi rasional jam minimum sehari

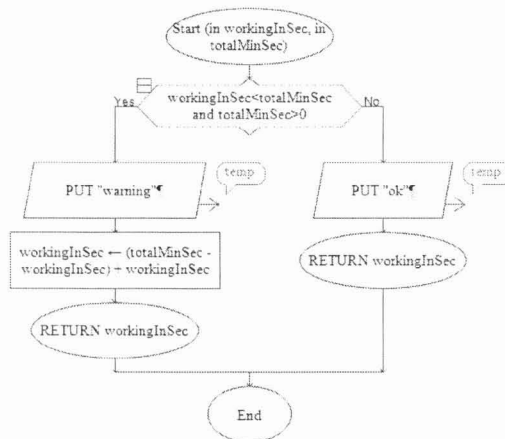
Rajah 4 merujuk kepada algoritma operasi rasional jam minimum sehari. Algoritma ini bertanggung jawab menolak pengiraan jam keberadaan pada hari berkenaan sekiranya jam bekerja kurang daripada 4 jam. Algoritma ini akan menasihati pengguna supaya mengambil cuti rehat sekiranya perkara ini berlaku. Proses pengiraan akan dibuat semula selepas itu. Algoritma ini direka bentuk berdasarkan pekeliling pentadbiran UiTM (Pekeliling Pendaftar, 2010).



Rajah 4 Algoritma operasi rasional jam minimum sehari

2.4 Algoritma operasi rasional jam minimum seminggu

Rajah 5 merujuk algoritma operasi rasional jam minimum seminggu. Algoritma ini akan memastikan setiap hari bekerja perlu mempunyai jumlah jam keberadaan minimum sebanyak 4 jam. Sekiranya jumlah hari bekerja seseorang pensyarah adalah 5 hari, mereka tidak boleh mencapai jumlah jam keberadaan 38.5 jam terlalu awal, sebagai contoh mencapai 38.5 jam pada hari ke-2 dengan waktu masuk kerja bermula 8 pagi dan waktu balik kerja pada 10.30 malam sepanjang 2 hari berturut-turut. Algoritma ini akan memastikan pensyarah datang bekerja dan perlu mempunyai jam keberadaan minimum harian sebanyak 4 jam sehari. Algoritma ini direka bentuk berdasarkan pekeliling pentadbiran UiTM (Pekeliling Pendaftar, 2010).



Rajah 5 Algoritma operasi rasional jam minimum seminggu

2.5 Set data pengujian

Jadual 1 adalah set data yang dicipta khusus untuk membuktikan model sistem kehadiran mengira jam keberadaan dengan cekap dan mampu melaksanakan operasi yang ditetapkan.

Jadual 1 Set data untuk menguji algoritma

KES	Hari 1			Hari 2			Hari 3			Hari 4			Hari 5		
	M	B	P	M	B	P	M	B	P	M	B	P	M	B	P
Kes1	06:32	17:55	30	07:45	23:45	45	04:05	22:40	60	10:12	17:45	33	14:23	22:05	55
Kes2	08:45	17:12	0	08:45	12:00	0	07:45	17:32	30	08:30	10:30	120	07:30	18:32	30
Kes3	08:05	22:20	30	08:00	22:20	0	07:35	17:00	30	08:00	12:00	0	07:50	12:00	0

M = Jam masuk kerja

B = Jam balik kerja

P = Jam keluar pejabat / kampus (minit)

Kes 1 = Menguji algoritma operasi rasional masa masuk dan masa balik

Kes 2 = Menguji algoritma operasi rasional jam minimum sehari

Kes 3 = Menguji algoritma operasi rasional jam minimum seminggu

Bagi memudahkan pengujian terhadap model, prototaip sistem kehadiran yang diberi nama *UiTM Time Card Calculator* digunakan sebagai alat pengujian. Algoritma operasi yang telah dibangunkan dan digunakan di dalam *UiTM Time Card Calculator* diuji menggunakan set data seperti di Jadual 1.

3. Keputusan dan perbincangan

Hasil pengujian terhadap algoritma yang menggunakan set data di Jadual 1 ditunjukkan di seksyen 3.1. Hasil pengujian menunjukkan algoritma yang dibina mampu mengatasi masalah yang ada dan dapat memberi kelebihan membuat pengiraan dengan tepat dan rasional di samping mengutamakan masa keberadaan yang fleksibel. Di seksyen 3.2 dicadangkan model sistem kehadiran yang baru yang boleh mengira jam keberadaan pensyarah UiTM.

3.1 Keputusan pengujian

Rajah 6 menunjukkan hasil pengujian terhadap algoritma operasi rasional masa masuk dan masa balik. Merujuk set data di Jadual 1 (kes 1), jika dilihat pada hari ke-3, algoritma ini berjaya menyelesaikan waktu masuk kerja pada pukul 4.30 pagi dan waktu balik kerja pada pukul 22.40 malam. Waktu-waktu ini adalah waktu yang tidak rasional untuk dikira, algoritma ini akan membuat pelarasan waktu iaitu waktu masuk 4.30 pagi kepada 7.30 pagi dan waktu balik 22.40 malam kepada 22.30 malam.

```

<< DAY 3 >>
Punch In (24 hours format) 00:00           = 04:05
Punch Out (24 hours format) 00:00         = 22:40
Not available at the office / campus (in minutes) = 60
-----
# The time entered (punch in) is lower the limit (Cut to 07:30)
# The time entered (punch out) is over the limit (Cut to 22:30)
# Total minimum hours (4 hours) in a day are OK
# Total minimum hours in a week failed and need adjustment.
Punch In (24 hours format) = 7:30:0
Punch Out (24 hours format) = 22:30:0
Needs to top up (38:30:0) = 8:0:0
-----
    
```

Rajah 6 Keputusan ujian terhadap algoritma operasi rasional masa masuk dan masa balik

Rajah 7 menunjukkan hasil pengujian terhadap algoritma operasi rasional jam minimum sehari. Merujuk set data di Jadual 1 (kes 2), jika dilihat pada hari ke-2, algoritma ini berjaya menyelesaikan masalah ketidakcukupan waktu kerja iaitu kurang daripada 4 jam sehari. Algoritma ini akan menasihati pengguna supaya mengambil cuti rehat seperti di Rajah 7. Proses pengiraan akan dibuat semula selepas itu.

```

<< DAY 2 >>
Punch In (24 hours format) 00:00 = 08:45
Punch Out (24 hours format) 00:00 = 12:00
Not available at the office / campus (in minutes) = 0

```

```

# The time entered (punch in) is OK
# The time entered (punch out) is OK

```

WARNING:

```

# The total minimum requirement of four hours is not met
# Please take your leave for this day
# Please recalculate your working hours' requirement

```

Rajah 7 Keputusan ujian terhadap algoritma operasi rasional jam minimum sehari

Rajah 8 menunjukkan hasil pengujian terhadap algoritma operasi rasional jam minimum seminggu. Merujuk set data di Jadual 1 (kes 3), jika dilihat pada hari ke-1 dan ke-2, algoritma ini berjaya menyelesaikan masalah ketidakcukupan jam keberadaan minimum untuk hari yang berikutnya. Asas algoritma ini adalah memastikan pensyarah perlu mempunyai jam keberadaan minimum 4 jam sehari. Di Rajah 8, jika sistem kehadiran tidak mempunyai algoritma ini, kemungkinan besar pensyarah berkenaan akan cukup jam keberadaan 38.5 jam seminggu hanya dalam tempoh 2 hari bekerja. Masalah ini akan mengundang salah faham di mana pensyarah boleh membuat interpretasi yang salah di mana mereka tidak perlu hadir bekerja untuk hari berikutnya kerana jam keberadaan 38.5 jam seminggu telah dicapai. Jika interpretasi ini dilakukan maka para pensyarah akan melanggar Pekeliling Pendaftar, 2010.

Rajah 8 menunjukkan algoritma operasi rasional jam minimum seminggu dapat dilaksanakan. Algoritma akan membuat pelarasan pada hari ke-2 di mana pernyataan *Needs to top up (38:30:0)* memastikan 3 hari berikutnya perlu mempunyai jam keberadaan yang cukup iaitu 4 jam sehari. Algoritma ini mampu memastikan pensyarah datang bekerja dan jam keberadaan minimumnya sebanyak 4 jam sehari dapat dicapai.

```

-----
UNIVERSITI TELNOLOGI MARA
TIME CARD CALCULATOR (Weekly): Version KN.3.0
-----
Number of day/s (leave, MC, outstation) = 0
Required hours (presence at the office) = 38:30:0 (5 day/s)

<< DAY 1 >>
Punch In (24 hours format) 00:00 = 08:05
Punch Out (24 hours format) 00:00 = 22:20
Not available at the office / campus (in minutes) = 30

# The time entered (punch in) is OK
# The time entered (punch out) is OK
# Total minimum hours (4 hours) in a day are OK
# Total minimum hours in a week are OK

Punch In (24 hours format) = 8:5:0
Punch Out (24 hours format) = 22:20:0
Needs to top up (38:30:0) = 24:45:0

-----

<< DAY 2 >>
Punch In (24 hours format) 00:00 = 08:00
Punch Out (24 hours format) 00:00 = 22:20
Not available at the office / campus (in minutes) = 0

# The time entered (punch in) is OK
# The time entered (punch out) is OK
# Total minimum hours (4 hours) in a day are OK
# Total minimum hours in a week failed and need adjustment.

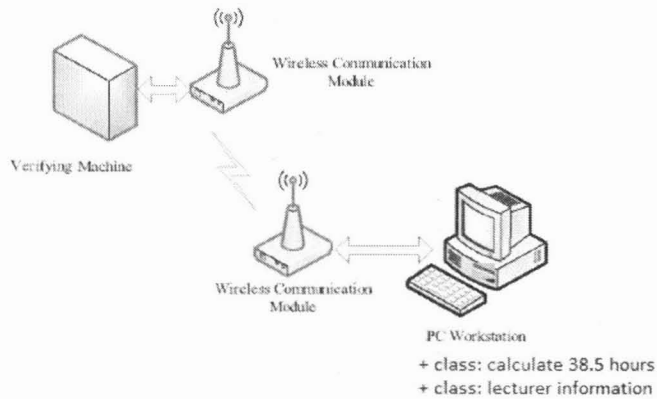
Punch In (24 hours format) = 8:0:0
Punch Out (24 hours format) = 22:20:0
Needs to top up (38:30:0) = 12:0:0
-----

```

Rajah 8 Keputusan ujian terhadap algoritma operasi rasional jam minimum seminggu

3.2 Organisasi model sistem kehadiran

Rajah 9 menunjukkan cadangan organisasi sistem kehadiran pensyarah UiTM di mana *PC Workstation* akan mempunyai 2 fungsi sistem yang utama (1) *class: calculate 38.5 hours* (2) *class: lecturer information*. *class: calculate 38.5 hours* akan melakukan kerja-kerja pengiraan jam keberadaan pensyarah iaitu 38.5 jam seminggu. Manakala *class: lecturer information* akan melakukan kerja-kerja memperincikan maklumat pensyarah. Kesemua data ini akan disimpan di dalam pangkalan data di *PC Workstation* dan boleh dijadikan laporan untuk pengurusan UiTM nanti. *Verifying machine* boleh merujuk kepada penggunaan teknologi seperti kad kod bar, kad magnetik, kad bercip, kad RF atau biometrik.



Rajah 9 Organisasi sistem kehadiran

4. Kesimpulan dan kerja masa depan

Algoritma yang diperkenalkan di dalam kertas penyelidikan ini berjaya menyelesaikan masalah pengiraan keberadaan pensyarah. Ia dapat memberi kelebihan membuat pengiraan dengan tepat dan rasional di samping mengutamakan masa keberadaan yang fleksibel. Algoritma ini boleh digunakan untuk model sistem kehadiran pensyarah UiTM khususnya dan universiti lain secara umumnya, jika keperluannya sama seperti UiTM. Dengan ini model serta prototaip yang terdapat di dalam kertas penyelidikan ini boleh diperkembangkan penyelidikannya untuk bahagian implementasi seperti yang dicadang di seksyen 3.2 di masa depan nanti. Diharap dengan adanya penyelidikan awal berkaitan penghasilan algoritma mengukur jam keberadaan pensyarah ini dapat menyumbang ke arah peningkatan kualiti serta integriti sebagai seorang pensyarah. Algoritma ini juga diharap dapat menjadi pencetus kepada kecemerlangan akademik di universiti.

5. Rujukan

Pekeliling Timbalan Naib Cancellor (2001). Pekeliling Timbalan Naib Cancellor (Akademik) bil.10/2001.
 Pekeliling Timbalan Naib Cancellor (2003). Pekeliling Timbalan Naib Cancellor (Akademik) bil.5/2003.
 Yongqiang, Z., & Ji, L. (2006). The Design of Wireless Fingerprint attendance system. International Conference on Communication Technology (ICCT '06). doi: 10.1109/ICCT.2006.341990.
 Pekeliling Pendaftar (2010). Pekeliling Pendaftar bil.7-2010, 2010.