BAB III

PENCARIAN SOLUSI OPTIMAL PADA GRAPH OR DENGAN ALGORITMA A*

3.1. Bentuk Umum Algoritma A*

Sebelum membahas algoritma A* terlebih dahulu akan dijelaskan pengertian dan bentuk algoritma A, sebab pada dasarnya algoritma A* mempunyai prosedur pencarian yang sama dengan algoritma A.

Algoritma A bekerja dengan cara memeriksa simpul-simpul dari graph OR dari suatu persoalan, dimana masing-masing simpul menyatakan keadaan yang mungkin terjadi dari persoalan tersebut. Suatu simpul pada graph OR menyatakan keadaan awal yang disebut simpul awal atau keadaan tujuan yang disebut simpul tujuan atau pengembangan suatu keadaan dari persoalan yang direpresentasikan.

Sedangkan tujuan dari algoritma A adalah menemukan solusi optimal yaitu lintasan atau path terpendek yang menghubungkan simpul awal ke simpul tujuan. Path ini disebut dengan path solusi optimal.

Setiap simpul pada graph OR memuat fungsi evaluasi f yang terdiri dari jumlahan fungsi g dan h (persamaan 2.4.1). Pada setiap simpul n berlaku :

$$f(n) = g(n) + h(n)$$

dengan :

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation: (http://eprints.undip.ac.id)

g(n) = harga dari PP_{s-n}, yaitu petunjuk path berarah dari s ke n, g(s) = 0.

h(n) = perkiraan dari $h^*(n)$, yaitu panjang path terpendek dari n ke τ, dengan $h(\tau) = 0$, $\tau \in \Gamma$.

Dalam pencarian solusi optimal algoritma A menggunakan dua buah himpunan simpul, yaitu :

1. OPEN = himpunan simpul-simpul yang telah dihasilkan tetapi belum diperiksa.
2. CLOSED = himpunan simpul-simpul yang telah diperiksa.

Prosedur algoritma A adalah sebagai berikut :

1. Tempatkan simpul awal s pada OPEN.
2. Jika OPEN kosong, keluar dengan kegagalan.
3. Bergerak kembali dari OPEN dan tempatkan suatu simpul n yang fungsi f-nya minimal pada CLOSED.
4. Jika n suatu simpul tujuan, keluar dengan sukses dan solusi diisi dengan petunjuk path dari n ke s.
5. Sebaliknya perluas simpul n, kembangkan semua penggantinya dan letakkan petunjuknya kembali ke n. Untuk setiap pengganti n´ dari n :
   a. Jika n´ tidak berada pada OPEN atau CLOSED, perkirakan h(n´) dan jumlahkan f(n´) = g(n´) + h(n´) dimana g(n´) = g(n) + c(n,n´) dan g(s) = 0.
   b. Jika n´ berada pada OPEN atau CLOSED, arahkan petunjuk sepanjang path yang menghasilkan g(n´) terendah.
   c. Jika n´ simpul petunjuk yang dibutuhkan dan

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation: (http://eprints.undip.ac.id)

ditemukan pada CLOSED maka buka kembali simpul tersebut.

6. Kembali ke langkah 2.

Catatan bahwa untuk sembarang simpul x, y ∈ OPEN berlaku, jika $f(x) = f(y) < f(n)$ ∀ n ∈ OPEN, maka A akan memilih simpul x untuk diperiksa apabila $g(x) > g(y)$.

Jika $h \equiv 0$ dan $g \equiv d$ yaitu kedalaman simpul pada graph OR, maka algoritma A identik dengan breadth first search. Sekarang apabila algoritma A menggunakan fungsi heuristik h yang memenuhi :

$$h(n) \leq h^*(n) \quad \forall n \qquad \ldots\ldots (3.1.1)$$

maka algoritma A disebut dengan algoritma A^*.

Karena algoritma A^* juga merupakan algoritma A, maka penjelasan sebelumnya yang berlaku untuk algoritma A juga berlaku untuk algoritma A^*.

Di bawah ini merupakan contoh pencarian oleh algoritma A^* pada graph OR yang simpul - simpulnya menyatakan ruang keadaan tertentu seperti pada gambar 20. Misalkan A = simpul awal, P = simpul tujuan dan pada masing-masing simpul terdapat fungsi heuristik h yang merupakan perkiraan jumlah gerakan (garis) yang dilalui dari simpul tersebut ke simpul P, dimana jumlah gerakan antara dua simpul yang adjacent adalah satu.

Jika B successor dari A, maka $g(B) = 1$.

Jika B ancestor dari P, maka h(B) merupakan perkiraan jumlah gerakan dari B ke P.

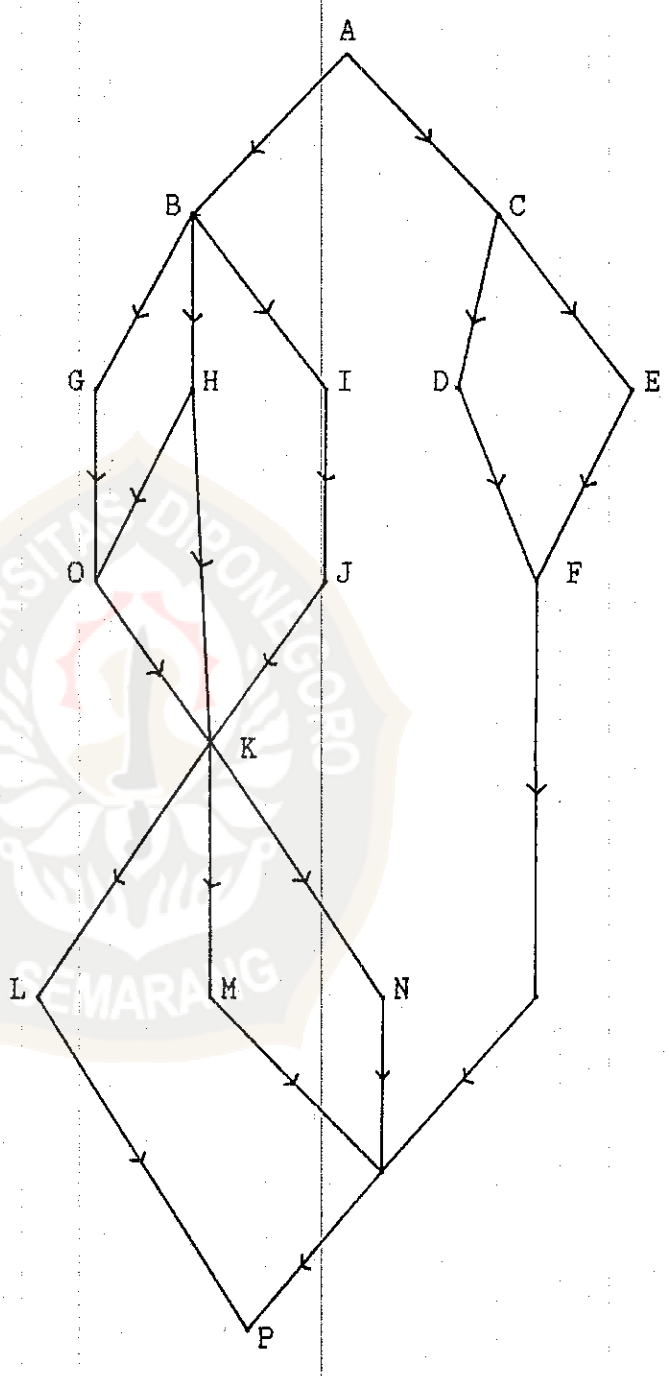
This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation: (http://eprints.undip.ac.id)

Berdasarkan persamaan 2.4.1 berlaku :

$$f(n) = g(n) + h(n) \quad \forall\ n.$$

Misaslkan :

h(A) = 5	h(I) = 1
h(B) = 4	h(J) = 1
h(C) = 3	h(K) = 1
h(D) = 2	h(L) = 1
h(E) = 2	h(M) = 2
h(F) = 3	h(N) = 2
h(G) = 4	h(O) = 3
h(H) = 3	h(P) = 0



gambar 20.

Diberikan simbol PS_n untuk menyatakan petunjuk simpul n ke simpul parent-nya yang dilalui oleh path dengan g(n) terendah.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Pada tahap 1 :

OPEN = [A] ; CLOSED = [ ]

A satu-satunya simpul pada OPEN, maka A dikembangkan dan menghasilkan B, C.

PS_B = BA ; f(B) = g(B) + h(B) = 1 + 4 = 5

PS_C = CA ; f(C) = g(C) + h(C) = 1 + 3 = 4

Pada tahap 2 :

OPEN = [B, C] ; CLOSED = [A]

f(C) < f(B), maka C dikembangkan dan menghasilkan D, E.

PS_D = DC ; f(D) = g(D) + h(D) = 2 + 2 = 4

PS_E = EC ; f(E) = g(E) + h(E) = 2 + 2 = 4

Pada tahap 3 :

OPEN = [B, D, E] ; CLOSED = [A, C]

f(D) = F(E) < f(B) dan misalkan D yang dipilih, maka dikembangkan dan menghasilkan F.

PS_F = FD ; f(F) = g(F) + h(F) = 3 + 3 = 6

Pada tahap 4 :

OPEN = [B, E, F] ; CLOSED = [A, C, D]

f(E) < f(B) < f(F), maka E dikembangkan dan menghasilkan F. Ternyata F telah berada pada CLOSED.

g(F) = 2 jika path ke F melalui E, sama dengan g(F) = 2 jika path ke F melalui D, sehingga PS_F ditetapkan sama dengan sebelumnya yaitu PS_F = FD.

Pada tahap 5 :

OPEN = [B, F] ; CLOSED = [A, C, D, E]

f(B) < f(F), maka B dikembangkan dan menghasilkan G, H, I.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

PS_G = GB ; f(G) = g(G) + h(G) = 2 + 4 = 6

PS_H = HB ; f(H) = g(H) + h(H) = 2 + 3 = 5

PS_I = IB ; f(I) = g(I) + h(I) = 2 + 1 = 3

Pada tahap 6 :

OPEN = [F, G, H, I] ; CLOSED = [A, C, D, E, B]

f(I) terendah, maka I dikembangkan dan menghasilkan J.

PS_J = JI ; f(J) = g(J) + h(J) = 3 + 1 = 4

Pada tahap 7 :

OPEN = [F, G, H, J] ; CLOSED = [A, C, D, E, I]

f(J) terendah, maka J dikembangkan dan menghasilkan K.

PS_K = KJ ; f(K) = g(K) + h(K) = 4 + 1 = 5

Pada tahap 8 :

OPEN = [F, G, H, K] ; CLOSED = [A, C, D, E, I, J]

f(K) < f(H) dan g(K) > g(H), maka K dikembangkan dan menghasilkan L, M, N.

PS_L = LK ; f(L) = g(L) + h(L) = 5 + 1 = 6

PS_M = MK ; f(E) = g(E) + h(E) = 5 + 2 = 7

PS_N = NK ; f(E) = g(E) + h(E) = 5 + 2 = 7

Pada tahap 9 :

OPEN = [F, G, H, L, M, N] ;

CLOSED = [A, C, D, E, I, J, K]

f(H) terendah, maka H dikembangkan dan menghasilkan O, K.

PS_O = OH ; f(O) = g(O) + h(O) = 3 + 3 = 6

K telah berada dalam CLOSED. g(K) = 3 jika path ke K melalui H dan lebih kecil daripada g(K) = 4 jika path ke K melalui J, sehingga :

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

PS_K = KH ; f(K) = g(K) + h(K) = 3 + 1 = 4

f(K) < f(n) ∀ n ∈ OPEN, sehingga K dibuka kembali.

Pada tahap 10 :

OPEN = [F, G, L, M, N, O, K] ;

CLOSED = [A, C, D, E, I, J, H]

f(K) terendah, maka K dikembangkan dan menghasilkan L, M dan N. Simpul-simpul ini telah berada dalam OPEN, dan nilai fungsi evaluasi f-nya diperbarui kembali menjadi :

PS_L = LK ; f(L) = g(L) + h(L) = 4 + 1 = 5

PS_M = MK ; f(E) = g(E) + h(E) = 4 + 2 = 6

PS_N = NK ; f(E) = g(E) + h(E) = 4 + 2 = 6

Pada tahap 11 :

OPEN = [F, G, L, M, N, O] ;

CLOSED = [A, C, D, E, I, J, H, K]

f(L) terendah, maka L dikembangkan dan menghasilkan P.

PS_P = PL ; f(P) = g(P) + h(P) = 5 + 0 = 5

Pada tahap 12 :

OPEN = [F, G, M, N, O, P] ;

CLOSED = [A, C, D, E, I, J, H, K, L]

f(P) terendah dan P merupakan suatu simpul tujuan, sehingga algoritma A^* akan keluar dengan sukses dan solusi diisi oleh petunjuk path dari P ke A, yaitu :

$PP_{P-A} = PS_P, PS_L, PS_K, PS_H, PS_B, PS_A$

= PL, LK, KH, HB, BA

dengan panjang path solusi optimal 5.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

3.2. Sifat - sifat Algoritma A^*

Pada saat pencarian path solusi optimal, algoritma A^* selalu memilih dan memeriksa simpul yang mempunyai harga fungsi evaluasi yang terendah. Simpul - simpul yang tidak berada pada path solusi optimal ada kemungkinan dipilih dan diperiksa dalam proses pencarian, jika suatu saat sebelum proses pencarian berakhir simpul tersebut memiliki nilai fungsi evaluasi yang terendah. Simpul demikian yang tidak berada pada sembarang path solusi optimal disebut dengan simpul - simpul pada track jelek (off track).

Kemudian untuk menetapkan harga sebenarnya dari P_{s-n} yaitu sembarang path dari s ke n dan $P_{n-\tau}$ yaitu sembarang path dari n ke τ, sepanjang path khusus P, dituliskan berturut-turut sebagai $g_p(n)$ dan $h_p(n)$.

Dari keterangan di atas dapat diturunkan sifat- sifat fungsi evaluasi f yang digunakan oleh algoritma A^* dalam pencarian solusi optimal. Jika s suatu simpul awal dan n sembarang simpul yang dihasilkan oleh A^*, maka berlaku sifat - sifat sebagai berikut :

1. Untuk setiap path P, berlaku :

$$g_p(n) \geq g^*(n) \quad \text{dan} \quad h_p(n) \geq h^*(n) \qquad \ldots\ldots (3.2.1)$$

2. Jika g dan h sama dengan nilai optimalnya, berlaku :

$$f^*(n) = g^*(n) + h^*(n) \qquad \ldots\ldots (3.2.2)$$

3. $f^*(s) = h^*(s) = g^*(\tau) = f^*(\tau) = C^* \quad \forall\, \tau \in \Gamma^* \qquad (3.2.3)$

C^* = Nilai path solusi optimal.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation.
(http://eprints.undip.ac.id)

4. Jika n sembarang simpul pada path solusi optimal dari s ke τ, berlaku :

$$f^*(n) = C^* \qquad n \in P^*_{s-\Gamma} \qquad \ldots\ldots (3.2.4)$$

5. Jika n terletak pada off track, berlaku :

$$f^*(n) > C^* \qquad n \notin P^*_{s-\Gamma} \qquad \ldots\ldots (3.2.5)$$

Bukti :

1. $g^*(n)$ adalah harga path terpendek dari s ke n, dan $h^*(n)$ adalah harga path terendah dari n ke Γ.
   Sehingga untuk setiap path P, dengan P_{s-n} dan $P_{n-\tau}$ sepanjang $P \in P_{s-\Gamma}$, berlaku :
   $g_p(n) \geq g^*(n)$ dan $h_p(n) \geq h^*(n)$.

2. Jika g dan h sama dengan nilai optimalnya, berarti terdapat path P^*_{s-n} dan $P^*_{n-\tau}$ sepanjang path khusus $P \in P^*_{s-\Gamma}$ yang memenuhi :

   $$g_p(n) = g^*(n) \text{ dan } h_p(n) = h^*(n)$$

   sehingga,

   $$f(n) = g(n) + h(n)$$
   $$f(n) = g_p(n) + h_p(n)$$
   $$f(n) = g^*(n) + h^*(n) = f^*(n)$$

   sebab $f^*(n)$ merupakan nilai optimal semua path solusi yang melalui n.

3. Dari sifat 2, berlaku :
   $f^*(n) = g^*(n) + h^*(n)$, jika g dan h sama dengan nilai optimalnya.
   Substitusikan s ke persamaan tersebut, diperoleh

   $$f^*(s) = g^*(s) + h^*(s).$$

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Karena $g^*(s) = 0$ maka $f^*(s) = h^*(s)$.

Begitu pula untuk $\tau \in \Gamma^*$ diperoleh

$$f^*(\tau) = g^*(\tau) + h^*(\tau).$$

Karena $h^*(\tau) = 0$ maka $f^*(\tau) = g^*(\tau)$.

Diketahui bahwa :

$g^*(\tau)$ = harga path terendah dari s ke $\tau \in \Gamma$,

$h^*(s)$ = harga path terendah dari s ke $\tau \in \Gamma$,

yang tidak lain adalah harga path solusi optimal (C^*),

Sehingga

$f^*(s) = h^*(s) = g^*(\tau) = f^*(\tau) = C^*, \quad \forall \tau \in \Gamma^*$.

4. $n \in P^*_{s-\Gamma}$ secara tidak langsung menyatakan bahwa terdapat path solusi P melalui n seharga C^*, tepatnya sepanjang path ini $g_\rho(n) + h_\rho(n) = C^*$. Dengan menggunakan syarat optimal dari g^* dan h^* pada sifat 1, yaitu $g_\rho(n) \geq g^*(n)$ dan $h_\rho(n) \geq h^*(n)$, diperoleh :

$$g^*(n) + h^*(n) \leq C^*.$$

Andaikan terdapat path P' yang memenuhi

$$g_{\rho'}(n) + h_{\rho'}(n) < C^*,$$

maka ini kontradiksi dengan syarat optimal dari C^*. Sehingga

$$g^*(n) + h^*(n) = C^*$$

atau

$$f^*(n) = C^*.$$

5. Andaikan $f^*(n) \leq C^*$ atau $g^*(n) + h^*(n) \leq C^*$, maka di sana terdapat suatu path solusi P' yang melalui n sedemikian sehingga $g_{\rho'}(n) + h_{\rho'}(n) \leq C^*$, yang memenuhi P' sebagai path solusi optimal dan ini

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation: (http://eprints.undip.ac.id)

kontradiksi dengan asumsi bahwa n adalah suatu simpul pada off track.

Pengandaian diingkar, sehingga

$$f^*(n) > C^* \quad \forall\, n \notin P^*_{s-\Gamma}$$ ■

Bila graph OR yang merupakan ruang pencarian solusi suatu persoalan adalah berhingga maka pada graph tersebut berlaku sifat-sifat algoritma A^* sebagai berikut :

1. Algoritma A^* selalu berakhir (terminate).

   Bukti :

   Jumlah simpul yang menyusun path pada graph berhingga adalah berhingga dan setiap simpul hasil pengembangan A^* akan menambah mata rantai baru pada path tersebut melalui tree. Setiap ditambahkan lintasan baru mewakili suatu path pasti pada akhirnya akan berhenti (habis). ■

2. Algoritma A^* adalah komplit yaitu berakhir dengan solusi apabila solusi tersebut ada.

   Bukti :

   Suatu kegagalan dari A^* dihasilkan hanya bila OPEN ditemukan kosong sebelum path solusi $P_{s-\tau}$ ditemukan bila $P_{s-\tau}$ ada. Jika berlaku kegagalan berarti di sana ada simpul $n' \in P_{s-\tau}$ pada OPEN (paling sedikit satu simpul) yang ditemukan dan dikembangkan tanpa menghasilkan pengganti yang baru. Ini kontradiksi dengan asumsi bahwa n' terletak pada path solusi yang paling sedikit mempunyai satu pengganti (kecuali simpul tujuan) yang juga terletak pada path solusi. ■

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

3.3. Jaminan untuk Solusi Optimal

Karena Algoritma A^* merupakan algoritma A yang fungsi heuristiknya memenuhi persamaan 3.1.1, yaitu :

$$h(n) \leq h^*(n) \qquad \forall\, n,$$

maka diasumsikan definisi di bawah ini dipenuhi untuk penjelasan selanjutnya.

Definisi 3.3.1 :

Suatu fungsi heuristik h dikatakan dapat diterima (admissible) jika

$$h(n) \leq h^*(n) \qquad \forall\, n.$$

Lemma 3.3.1 :

Pada setiap saat sebelum A^* berakhir, terdapat suatu simpul n' dari $P^*_{s-\tau}$ pada OPEN dengan $f(n') \leq C^*$.

Bukti :

Pertimbangkan sembarang path optimal $P^*_{s-\tau} \in P^*_{s-\Gamma}$ dengan

$P^*_{s-\tau} = s, n_1, n_2, \ldots, n', \ldots, \tau$

dan misalkan n' simpul OPEN yang dangkal pada $P^*_{s-\tau}$ (pada OPEN terdapat minimal satu simpul sampai $\tau \in$ CLOSED).

Karena semua pendahulu dari n' berada pada CLOSED dan path $s, n_1, n_2, \ldots, n'$ optimal maka petunjuk sekarang ditetapkan ke n' sepanjang $P^*_{s-n'}$, karena itu $g(n') = g^*(n')$.

Dengan menggunakan syarat admissible dari h diperoleh

$$f(n') = g^*(n') + h(n')$$

$$\leq g^*(n') + h^*(n') = f^*(n')$$

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation: (http://eprints.undip.ac.id)

atau $f(n') \leq f^*(n')$ ..... (3.3.1)

dan karena $n' \in P^*_{s-\tau}$ dari persamaan (3.2.4) berlaku $f^*(n') = C^*$.

Sehingga $f(n') \leq C^*$ ..... (3.3.2).

Lemma 3.3.2 (akibat lemma 3.3.1) :

Misalkan n' simpul OPEN yang dangkal pada path optimal $P^*_{s-n'}$ (tidak perlu ke Γ), maka

$g(n') = g^*(n')$ ..... (3.3.3)

menyatakan bahwa A^* siap menemukan petunjuk path optimal ke n' (yaitu n' sepanjang $P^*_{s-n'}$) dan path itu tidak akan diubah selama pencarian.

Bukti :

Langsung mengikuti bukti lemma 3.3.1.

Definisi 3.3.2 :

Suatu algoritma adalah admissible jika algoritma tersebut dijamin menghasilkan suatu solusi optimal bilamana solusi tersebut ada.

Theorema 3.3.1 :

A^* adalah algoritma yang admissible.

Bukti :

Andaikan A^* berakhir pada suatu simpul tujuan $t \in \Gamma$ yang mana $f(t) = g(t) > C^*$.

A^* selalu memeriksa simpul-simpul agar terpenuhi terhadap satu-satunya kriteria akhir, yaitu simpul n yang f(n)-nya terendah, kemudian A^* menyeleksinya untuk perluasan.

Karena itu, bila t terpilih untuk perluasan, berarti

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

t memenuhi

$$f(t) \leq f(n) \qquad \forall\, n \in \text{OPEN}.$$

Dengan cara ini segera sebelum berakhir, semua simpul-simpul pada OPEN memenuhi

$$f(n) > C^* \qquad \forall\, n \in \text{OPEN}.$$

Bagaimanapun ini kontradiksi dengan lemma 3.3.1 yang menjamin keberadaan sekurang-kurangnya satu simpul n´ pada OPEN dengan $f(n') \leq C^*$.

Untuk itu simpul terakhir t harus memenuhi $g(t) = C^*$, dengan cara itu A^* menghasilkan path optimal. ■

3.4. Membandingkan Dua Fungsi Heuristik

Kekuatan fungsi heuristik h ditentukan oleh ketelitian dari perkiraannya. Jika h memperkirakan nilai penyelesaian secara tepat ($h = h^*$) maka A^* hanya akan menjalani simpul-simpul yang terletak sepanjang path optimal.

Jika nilai perkiraan fungsi heuristik h_2 pada semua simpul n yang bukan merupakan simpul tujuan lebih besar daripada perkiraan fungsi heuristik h_1 atau $h_2(n) > h_1(n)$ maka h_2 memberikan informasi heuristik yang lebih menguntungkan (informed) dibandingkan h_1.

Definisi 3.4.1 :

Suatu fungsi heuristik h_2 dikatakan lebih informed daripada h_1 jika keduanya admissible dan memenuhi

$$h_2(n) > h_1(n) \quad \forall\, n \text{ bukan tujuan} \qquad \ldots\ldots \quad (3.4.1)$$

Dengan kata lain penggunaan h_2 pada algoritma A^* dikatakan lebih informed daripada penggunaan h_1.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Selanjutnya akan ditunjukkan bahwa penggunaan heuristik yang lebih informed diperlukan karena secara tidak langsung lebih kuat. Pertama diperlukan beberapa hasil awal.

Theorema 3.4.1 :

Sembarang simpul yang diperluas oleh A^* tidak mungkin mempunyai nilai yang melebihi C^*, yaitu :

$f(n) \leq C^*$ $\forall$ n yang diperluas ..... (3.4.2)

Bukti :

Andaikan n dipilih untuk perluasan dari OPEN dengan $f(n) > C^*$.

Ini kontradiksi dengan lemma 3.3.1 yang menyebutkan bahwa pada OPEN terdapat suatu simpul n' sedemikian sehingga $f(n') \leq C^*$ dan pasti n' akan dipilih untuk perluasan daripada simpul n. ■

Theorema 3.4.2 :

Setiap simpul n pada OPEN yang memenuhi $f(n) < C^*$ akhirnya akan dikembangkan oleh A^*.

Bukti :

Misalkan pada suatu tahap, n ditemukan pada OPEN dengan $f(n) < C^*$. Sedangkan A^* berakhir dengan $f(t) = C^*$ setelah pemilihan t untuk perluasan dari OPEN.

Kenyataannya t yang dipilih dan bukan n dengan salah satu dari dua cara yaitu n dikembangkan sebelum t

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation: (http://eprints.undip.ac.id)

atau f(n) dalam hal ini dimodifikasi agar melebihi C^*. Tetapi f hanya dapat dimodifikasi menurun, sehingga alternatif n dikembangkan sebelum t sebagai satu - satunya kemungkinan. ■

Catatan bahwa baik theorema 3.4.1 maupun theorema 3.4.2 masing-masing memuat syarat perlu dan cukup. Ini karena kenyataannya simpul-simpul $n \in$ OPEN yang $f(n) = C^*$ tidak ditentukan oleh salah satu theorema tersebut, tetapi ditentukan oleh seri aturan keputusan yang digunakan pada fakta-fakta implementasi dari A^*. Terlihat bahwa himpunan simpul-simpul yang memenuhi persamaan ini relatif berukuran kecil, oleh sebab itu dapat diabaikan persamaan yang terjadi dari perhitungan yang banyak.

Theorema 3.4.1 memperkenalkan suatu himpunan simpul-simpul yaitu $S = \{ n : f(n) > C^* \}$ yang siap dikeluarkan dari perluasan. Himpunan S memberi pola sederhana dari ruang penyimpanan tanpa sifat admissible yang mencurigakan sebab sembarang simpul yang terdapat pada S semestinya dapat dengan aman dipindahkan dari OPEN tanpa pengaruh perlakuan dari A^*. Tentu saja tidak diketahui nilai sebenaranya dari C^* tetapi batas atas dari C^* sudah cukup bila oleh suatu nilai tengah (misalnya dengan depth first search) ditentukan suatu path yang menuju suatu tujuan dan path tersebut mempunyai nilai C'. Jelas bahwa $C^* \leq C'$ dan setiap simpul yang memenuhi $f(n) > C'$ pasti merupakan anggota dari S sehingga dapat dijauhkan dari penyimpanan. Jika secara kebetulan n diperbaharui lagi dengan suatu f yang rendah maka n akan diperlakukan sebagai suatu simpul

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

baru yang mungkin diperluas atau tidak. Tetapi syarat perluasan pada theorema 3.4.1 dan 3.4.2 belum menyediakan kriteria yang memadai untuk efesiensi algoritma A^*.

Definisi 3.4.2 :

Suatu path *P* dikatakan C-bounded jika setiap simpul n sepanjang path ini memenuhi $g_\rho(n) + h(n) \leq C$ dan disebut strictly C-bounded jika tepat $g_\rho(n) + h(n) < C$. Untuk mengetahui heuristik yang digunakan dalam pertaksamaan tersebut digunakan notasi C(h)-bounded.

Theorema 3.4.3 :

Syarat cukup agar A^* memperluas suatu simpul n adalah bila di sana terdapat suatu path strictly C^*- bounded *P* dari s ke n.

Bukti :

Andaikan terdapat suatu strictly C^*-bounded *P* dari s ke n yang berakhir pada simpul n dan tidak diperluas.
Misalkan n´ simpul OPEN yang dangkal pada *P*.
Karena semua pendahulu dari n´ ada pada CLOSED maka g(n´) yang diberikan oleh A^* tidak mungkin lebih besar daripada $g_\rho(n')$.
Karena itu

$$f(n') = g(n') + h(n')$$
$$\leq g_\rho(n') + h(n')$$
$$< C^*$$

Pada akhirnya $f(n') < C^*$ dan n´ diputuskan untuk diperluas daripada simpul tujuan yang berakhir dengan

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

f = C^*. Ini kontradiksi dengan asumsi di atas sehingga simpul n´ dari OPEN dapat diperluas pada *P*, karena itu n harus diperluas. ■

Theorema 3.4.4 :

Syarat perlu agar A^* memperluas suatu simpul n adalah di sana terdapat suatu path C^*- bounded dari s ke n.

Bukti :

Jika A^* memperluas suatu simpul n maka n harus berada dalam OPEN dan secara bersamaan $f(n) \leq C^*$ (theorema 3.4.1). Sekarang pertimbangkan petunjuk path *PP* yang diberikan ke n pada saat perluasannya. Setiap pendahulu n´ dari n sepanjang *PP* ditutup karena dipilih untuk perluasan (mungkin lebih dari sekali) pada saat sebelumnya dimana n´ memenuhi

$g'(n') + h(n') \leq C^*$ (theorema 3.4.1).

Nilai g(n´) tidak mungkin lebih besar daripada nilai g´(n´) pada saat perluasan.

Konsekuensinya setiap simpul n´ sepanjang *PP* sekarang memenuhi $f(n') \leq C^*$, artinya bahwa *PP* itu sendiri adalah C^*- bounded. ■

Definisi 3.4.3 :

Suatu algoritma A_1 dikatakan mendominasi (dominate) A_2 jika setiap simpul perluasan oleh A_1 juga diperluas oleh A_2. Secara sama A_1 strictly dominate A_2 jika A_1 dominate A_2 dan A_2 tidak dominate A_1.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Theorema 3.4.5 :

Apabila A_2^* lebih informed daripada A_1^* maka A_2^* dominate A_1^*.

Bukti :

Misalkan A_2^* menggunakan heuristik h_2 dan A_1^* menggunakan heuristik h_1 sedemikian sehingga berlaku $h_2(n) > h_1(n)$ untuk setiap simpul bukan tujuan n dan asumsikan bahwa A_2^* memperluas suatu simpul n.

Dari theorema 3.4.4 diketahui bahwa di sana harus ada suatu path C^*- bounded P dari s ke n jika dinilai dari h_2. Karena $h_2(n') > h_1(n')$ untuk setiap simpul n' sepanjang P maka P adalah strictly C^*- bounded bila dinilai oleh h_1.

Berdasarkan theorema 3.4.3 maka A_1^* juga harus memperluas simpul n. ■

3.5. Heuristik Monoton dan Konsisten

Pada bagian ini akan ditunjukkan bahwa di bawah kondisi tertentu A^* tidak pernah membuka kembali simpul-simpul dari CLOSED, yang konsekuesinya tindakan yang berhubungan dengan perluasan kembali dapat dicegah.

Kondisi yang memungkinkan untuk tidak dilakukan pembukaan kembali suatu simpul pada CLOSED merupakan sifat khusus dari h yang dipenuhi oleh sembarang fungsi heuristik h^*, karena itu diberi nama konsisten. Pada bagian 3.2 terlihat bahwa setiap fungsi h^* terlepas dari kebenaran perkiraan nilai optimal ke Γ, juga memenuhi relasi khusus antara $h^*(n)$ dan h^* yang dievaluasi pada penerus n.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Pada dasarnya persamaan 3.2.4 dan 3.2.5 secara tidak langsung menyatakan bahwa path terpendek yang dipaksa melalui n tidak mungkin bernilai lebih kecil daripada path terpendek tanpa pemaksaan ini, yaitu :

$$g^*(n) + h^*(n) \geq h^*(s) \quad \forall\, n \qquad \ldots\ldots (3.5.1)$$

atau

$$k(s,n) + h^*(n) \geq h^*(s) \qquad \ldots\ldots (3.5.2)$$

Jika n´ adalah sembarang simpul keturunan dari n maka diperoleh :

$$h^*(n) \leq k(n,n') + h^*(n') \quad \forall\, (n,n') \qquad \ldots\ldots (3.5.3)$$

yang secara otomatis dipenuhi untuk simpul n´ yang bukan merupakan keturunan dari n, karena $k(n,n') = \infty$.
Dengan hal ini layak untuk memperkirakan jika proses perkiraan h(n) konsisten maka h(n) akan mewarisi sifat konsisten dari $h^*(n)$ dan memenuhi :

$$h(n) \leq k(n,n') + h(n') \quad \forall\, (n,n') \qquad \ldots\ldots (3.5.4)$$

Definisi 3.5.1 :

Suatu fungsi heuristik h(n) dikatakan konsisten jika persamaan 3.5.4 dipenuhi oleh semua pasangan simpul n dan simpul n´.

Definisi 3.5.2 :

Suatu fungsi heuristik h(n) dikatakan monoton jika memenuhi

$$h(n) \leq c(n,n') + h(n') \quad \forall\, n,n' \mid n' \in SCS(n) \qquad (3.5.5)$$

Sifat monoton mungkin nampak sekilas saja, sebab hanya menghubungkan heuristik pada suatu simpul ke heuristik pengganti terdekatnya.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Suatu bukti sederhana akan meyakinkan bahwa persamaan 3.5.5 secara tidak langsung menyatakan persamaan 3.5.4, oleh sebab dinyatakan theorema berikut.

Theorema 3.5.1 :

Kemonotonan dan kekonsistenan heuristik h adalah dua sifat yang ekuivalen.

Bukti :

(i) bukti jika konsisten maka monoton.

Dari sifat konsisten heuristik h diperoleh

$$h(n) \leq k(n,n') + h(n') \quad \forall\ (n,n').$$

Ini berarti terdapat path optimal $P^*_{n-n'} \in P^*_{n-n'}$ dari simpul n ke n', dengan panjang k(n,n').

Ambil sembarang path tersebut, misalnya :

$$P^*_{n-n'} = n, n_1, n_2, \ldots, l, m, n'.$$

Gunakan induksi sebagai berikut :

- karena $n' \in SCS(m)$ maka P^*_{n-m} optimal sepanjang $P^*_{n-n'}$ sehingga berlaku

$$h(n) \leq k(n,m) + h(m)$$

- karena $m \in SCS(l)$ maka P^*_{n-l} optimal sepanjang P^*_{n-m} sehingga berlaku

$$h(n) \leq k(n,l) + h(l)$$

dan seterusnya sampai diperoleh $n_1 \in SCS(n)$ dan $P^*_{n-n_1}$ optimal sepanjang path optimal $P^*_{n-n_2}$ sehingga berlaku

$$h(n) \leq k(n,n_1) + h(n_1).$$

Karena $n_1 \in SCS(n)$ dan $P^*_{n-n_1}$ optimal, maka berlaku

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

$k(n,n_1) = c(n,n_1)$.

Karena $P^*_{n-n'}$ diambil sembarang, maka

$h(n) \leq c(n,n_1) + h(n_1) \quad \forall\, n_1 \in SCS(n)$

yang memenuhi sifat monoton heuristik h.

(ii) bukti jika monoton maka konsisten.

Andaikan h tidak konsisten, berarti terdapat path optimal $P^*_{n-n'} \in P^*_{n-n'}$ dari n ke n´yang memenuhi :

$h(n) > k(n,n') + h(n')$ (a)

Misalnya

$P^*_{n-n'} = n, n_1, n_2, \ldots, m, n'$

maka pertaksamaan (a) juga menyatakan :

$h(n) > c(n,n_1)+c(n_1,n_2)+ \ldots +c(m,n')+h(n')$ (b)

dengan $n_1 \in SCS(n)$, $n_2 \in SCS(n_1)$, ... , $n' \in SCS(m)$.

Gunakan induksi sebagai berikut :

- karena h monoton maka

$h(m) \leq c(m,n') + h(n') \quad n' \in SCS(m)$

sehingga pertaksamaan (b) menjadi :

$h(n) > c(n,n_1)+c(n_1,n_2)+ \ldots +c(l,m)+h(m)$ (c)

- karena h monoton maka

$h(l) \leq c(l,m) + h(m) \quad m \in SCS(l)$

sehingga pertaksamaan (c) menjadi :

$h(n) > c(n,n_1)+c(n_1,n_2)+ \ldots +h(l)$ (d)

Hal ini terus dilakukan hingga pertaksamaan (d)

$h(n) > c(n,n_1) + h(n_1) \quad n_1 \in SCS(n)$.

Ini kontradiksi dengan sifat monoton heuristik h, sehingga pengandaian diingkar.

Terbukti jika h monoton maka h konsisten. ■

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Theorema 3.5.2 :

Setiap heuristik yang konsisten juga admissible.

Bukti :

Dengan sederhana disubsitusikan simpul n´ pada persamaan 3.5.4 dengan sembarang simpul tujuan $\tau \in \Gamma$, menghasilkan :

$$h(n) \leq k(n,\tau) + h(\tau) \qquad \forall\ n$$

karena $h(\tau) = 0$ dan $k(n,\tau) = h^*(n)$ untuk suatu simpul tujuan $\tau \in \Gamma^*$, maka diperoleh syarat admissible $h(n) \leq h^*(n)$ yang tetap. ■

Sekarang akan ditunjukkan keuntungan meggunakan heuristik konsisten dan monoton.

Theorema 3.5.3 :

Suatu algoritma A^* yang dikendalikan oleh suatu heuristik monoton menentukan path optimal menuju semua simpul yang diperluas, yaitu :

$$g(n) = g^*(n) \quad \forall\ n \in \text{CLOSED} \qquad \ldots\ldots (3.5.6)$$

Bukti :

Asumsikan bahwa A^* memilih perluasan suatu simpul n yang memenuhi $g(n) > g^*(n)$.

Pertimbangkan suatu path optimal P^*_{s-n} dari s ke n.

Jika n satu-satunya simpul OPEN pada P^*_{s-n} maka jelas semua pendahulu dari n sepanjang P^*_{s-n} diperluas dan diperoleh $g(n) = g^*(n)$ (lemma 3.3.2).

Asumsi $g(n) > g^*(n)$ secara tidak langsung menyatakan bahwa P^*_{s-n} berisi paling sedikit satu simpul tambahan pada OPEN dan dimisalkan n´ simpul OPEN yang dangkal pada P^*_{s-n}.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation: (http://eprints.undip.ac.id)

Sekarang akan ditunjukkan n´ dan bukan n yang akan dipilih untuk perluasan.

Lemma 3.3.2 menyatakan bahwa $g(n') = g^*(n')$, oleh karena itu dengan menggunakan sifat konsisten h, diperoleh :

$$f(n') = g^*(n') + h(n') \leq g^*(n') + k(n',n) + h(n)$$

Jumlahan $g^*(n') + k(n',n)$ sama dengan $g^*(n)$ sebab n´ adalah pendahulu dari n sepanjang P^*_{s-n} sehingga :

$$f(n') \leq g^*(n) + h(n)$$

Ternyata asumsi $g(n) > g^*(n)$ juga menyatakan

$$f(n') < f(n)$$

yang menghalangi pemilihan n daripada n´. Karena itu disimpulkan bahwa $g(n) = g^*(n)$. ■

Theorema 3.5.4 :

Sifat monoton secara tidak langsung menyatakan bahwa nilai f pada barisan simpul-simpul yang diperluas oleh A^* tanpa pemotongan.

Bukti :

Misalkan n_2 diperluas tepat setelah n_1.

Jika n_2 terletak pada OPEN selama n_1 diperluas maka $f(n_1) \leq f(n_2)$ dari aturan pemilihan simpul.

Jika n_2 tidak terletak di sana, maka n_2 harus merupakan simpul baru pada OPEN, yaitu suatu pengganti dari n_1 yang memenuhi

$$g(n_2) = g(n_1) + c(n_1,n_2)$$

dan dengan sifat monoton h diperoleh

$$f(n_2) = g(n_1) + c(n_1,n_2) + h(n_2)$$
$$\geq g(n_1) + h(n_1) = f(n_1)$$

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation.
(http://eprints.undip.ac.id)

atau

$$f(n_1) \leq f(n_2)$$

Karena itu barisan nilai-nilai f tidak dapat dipotong. ■

Sifat monoton menyederhanakan pengertian syarat perluasan suatu simpul.

Theorema 3.5.5 :

Jika h monoton maka syarat perlu untuk perluasan simpul diberikan oleh

$$g^*(n) + h(n) \leq C^*$$

dan syarat cukup diberikan oleh

$$g^*(n) + h(n) < C^*$$

Bukti :

Syarat perlu menggunakan kombinasi theorema 3.4.1 dan theorema 3.5.3.

Syarat cukup berdasarkan pada sifat tanpa pemotongan dari $g^* + h$ sepanjang suatu path optimal dari s ke sembarang simpul n yang berganti-ganti, sebab :

Jika n´ suatu simpul yang lebih awal daripada n sepanjang sebarang path optimal P^*_{s-n} maka harus memenuhi

$$g^*(n) = g^*(n') + c(n',n)$$

yang bersama-sama dengan sifat monoton h menghasilkan

$$g^*(n) + h(n) = g^*(n') + c(n',n) + h(n)$$
$$\geq g^*(n') + h(n').$$

Ini secara tidak langsung menyatakan bahwa jika n memenuhi $g^*(n) + h(n) < C^*$, maka semua pendahulunya

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

sepanjang P^*_{s-n} juga harus memenuhi pertaksamaan ini. Karena itu P^*_{s-n} adalah strictly C^*-bounded dan menurut theorema 3.4.3, n harus diperluas. ■

Definisi 3.5.3 :

Suatu algoritma A^*_2 dikatakan mendominasi kuat (largely dominate) A^*_1 jika setiap simpul yang diperluas oleh A^*_2 juga diperluas oleh A^*_1, kecuali mungkin beberapa simpul yang memenuhi

$$h_1(n) = h_2(n) = C^* - g^*(n).$$

Akibat theorema 3.5.5 :

Jika $h_2(n) \geq h_1(n)$ untuk semua n dan keduanya monoton maka A^*_2 yang menggunakan h_2 largely dominate A^*_1 yang menggunakan h_1.

Bukti :

Misalkan n suatu simpul yang diperluas oleh A^*_2 dan tidak oleh A^*_1 maka dari theorema 3.5.5 diperoleh :

$$g^*(n) + h_2(n) \leq C^* \quad \text{dan} \quad g^*(n) + h_1(n) \geq C^*.$$

Karena $h_2(n) \geq h_1(n)$ maka secara tidak langsung menyatakan bahwa

$$h_1(n) = h_2(n) = C^* - g^*(n).$$ ■

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)