ISBN: 978-602-958-488-2

PROSIDING

Seminar Nasional

MATEMATIKA DAN PENDIDIKAN MATEMATIKA

MATEMATIKA: DARI IDEALITAS SAMPAI REALITAS

MALANG, 18 MEI 2013

Aula Rektorat Lt.5 Gedung Ir. Soekarno
UIN Maulana Malik Ibrahim Malang

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKONOLOGI
UIN MAULANA MALIK IBRAHIM MALANG

MASALAH PENEMPATAN m-RATU PADA PAPAN CATUR DENGAN ALGORITMA RUNUT BALIK

Wahyudi[1], Wahyu Hengky Irawan[2]

Jurusan Matematika Universitas Islam Negeri Maulana Malik Ibrahim Malang
e-mail: wahyudibooleng@yahoo.co.id, henky_lily@yahoo.com

ABSTRAK

Algoritma runut balik adalah algoritma pencarian yang berbasis pada DFS (Depth First Search) atau pencarian mendalam dengan tujuan mencari solusi permasalahan secara lebih praktis. Simpul-simpul yang sudah dilahirkan (diperiksa) dinamakan simpul hidup (*live node*). Simpul hidup yang sedang diperluas dinamakan simpul-E atau *Expand Node*. Dalam kajian ini penulis menentukan banyaknya cara penempatan *m*-ratu pada papan catur berukuran *m x m* dengan *m* bilangan ganjil sedemikian hingga tidak ada dua ratu yang saling memakan (seperti dalam permainan catur).

Masalah penempatan m-ratu adalah suatu permasalahan bagaimana meletakkan ratu sebanyak *m* pada papan catur yang berukuran *m x m* sehingga tidak ada dua ratu yang saling memakan (seperti dalam permainan catur).Berdasarkan hasil pembahasan dapat diperoleh bahwa rumus umum untuk *m x m* dengan *m* bilangan ganjil adalah

1. $A_{i,\frac{1}{2}i}$ untuk *i* genap dengan $\frac{1}{2}i \leq j \leq \frac{1}{2}(m\text{-}1)$

   $A_{i,\frac{1}{2}(m-1)+t}$ untuk *i* ganjil dan t adalah bilangan asli dengan $\frac{1}{2}(m\text{-}1)+1 \leq j \leq m$ serta *i* dan t diisi secara serentak.
2. $A_{i,\frac{1}{2}(m-1)+(t+1)}$untuk *i* genap dan t adalah bilangan asli dengan $\frac{1}{2}(m\text{-}1)+(t+1) \leq j \leq m$ serta *i* dan t diisi secara serentak.

   $A_{i,t}$ untuk *i* ganjil dan t adalah bilangan asli dengan $1 \leq j \leq \frac{1}{2}(m\text{-}1)+1$ serta i dan t diisi secara serentak.

Kata Kunci:*backtracking*, *langkah Queen*

ABSTRACT

Backtracking algorithms are search algorithms based on DFS (Depth First Search) or search depth with the aim of finding solutions to problems is more practical. The nodes that are born (review) is called the knot live (live node). The node that is being extended is called node-E or Expand the Node. In this study the author determines the number of ways of placing the m-queen on the chess board size m x m with m odd numbers such that no two queens are consuming (as in the game of chess).

The placement issue m-queen is a problem how to put the queen as much as m on a chess board size m x m so that no two queens are consuming (as in a chess game). Based on the results of discussion can be obtained that the general formula for *m x m* with *m* odd numbers i.e.

1. $A_{i,\frac{1}{2}i}$ for even-numbered with $\frac{1}{2}i \leq j \leq \frac{1}{2}(m\text{-}1)$

   $A_{i,\frac{1}{2}(m-1)+t}$ for i is odd and *t* is *a* real number with $\frac{1}{2}(m\text{-}1)+1 \leq j \leq m$. i and t are filled simultaneously.
2. $A_{i,\frac{1}{2}(m-1)+(t+1)}$for i is even and t is natural numbers with $\frac{1}{2}(m\text{-}1)+(t+1) \leq j \leq m$. i and t are filled simultaneously.

   $A_{i,t}$ for i is odd and t is a real number with $1 \leq j \leq \frac{1}{2}(m\text{-}1)+1$. i and t are filled simultaneously.

Keywords:*backtracking, the Queen*

PENDAHULUAN

Runut-balik (*backtracking*) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih mangkus. Runut-balik, yang merupakan perbaikan dari algoritma *brute-force*, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Dengan metode runut-balik, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan. Akibatnya, waktu pencarian dapat dihemat (Munir, 2004).

Algoritma rubut-balik (*Backtracking*) banyak diterapkan dalam program game. Seperti, permainan, tic-tac-toe, menemukan jalan keluar dalam sebuah labirin, dan masalah-masalah pada bidang kecerdasan buatan (*artificial intelligence*).

Dalam permainan catur sudah dikenal bidak yang memiliki nama dan pergerakan yang berbeda-beda. Bidak-bidak tersebut yaitu poin, knigh,king, dan ratu, dan raja. Semuanya memiliki pergerakan masing-masing dengan karakternya masing-masing. Dan sebuah ratu dapat berjalan bergerak dalam sejumlah petak secara horizontal, vertikal, dan diagonal.

KAJIAN PUSTAKA

1. Ratu dalam Permainan Catur

Ratu adalah buah catur yang paling kuat. Ratu dapat berjalan baik vertikal, horizontal ataupun diagonal kesegala arah. Walaupun didalam catur terdapat raja dan raja tersebut adalah tonggak kerajan tetapi raja berjalan selangkah demi selangkah. Sehingga fungsi queen adalah membantu seorang raja untuk menghadapi musuhnya. Pada awal permainan masing-masing pemain memiliki buah ratu yang terletak disamping raja. Ratu putih terletak di kotak putir, dan sebaliknya ratu berwarna hitam terletak di kotak hitam pula.

Dalam permainan catur, buah catur yang memiliki kemampuan yang terbesar adalah queen, bukan raja. Raja merupakan pemimpin yang hanya memiliki kekuasaan atau buah catru yang paling tinggi derajatnya. Jika raja termakan musuh maka permainan akan berhenti dan ada kekalahan.

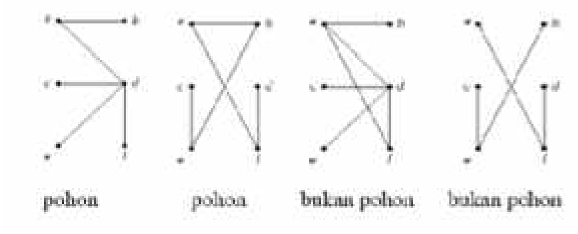
2. Graf

Graf G adalah pasangan himunan (V,G) denagn V adalah himpunan tidak kosong dan berhingga dari objek-objek yang disebut sebagai titik dan E adalah himpunan (mungkin ksoong) pasangan tak beraturan dari titik berbeda di G yang disebut dengan sisi. Himpunan titik di G dinotasikan dengan V(G) dan himpunan sisi dinotasikan dengan E(G). Sedangkan banyaknya unsur di V disebut order dari G dan dilambangkan dengan p(G) dan banyakunsur di E disebut ukuran dari G dan dilambangkan denagn q(G).jika graf yang dibicarakan hanya graf G,maka ordder dan ukuran dari G tersebut cukup ditulis dengan p dan q (Cartrand dan Lesniak, 1986:4)

3. Pohon

Sebuah pohon (bebas) atau (*free*) tree T adalah sebuah graf sederhana yang memenuhi jika v dan w adalah verteks di T, maka terdapat sebuah lintasan sederhana tunggal dari v ke w (Johnsonbaugh, 2004:86).

Diagram pohon (*tree*) adalah sebuah graf tak berarah yang terhubung (*connected*) yang tidak mengandung sirkuit sederhana (Suksmono, 2006:21).



Gambar 2.3 Pohon dan Bukan Pohon

Pohon yang sebuah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah menjauhi dari akar dinamakan pohon berakar (*rooted tree*).

Ada sejumlah aplikasi penting dari pohon, diantaranya:

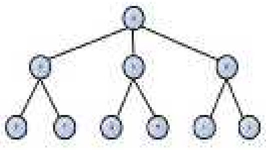
1. Optimasi jaringan dengan pohon pembentang minimum (*minimum spanning trees*)
2. Penyelesaian masalah dengan proses runut-balik pada pohon keputusan (*backtracking in decision trees*).
3. Kompresi data dengan pohon pengkodean Huffman (*Huffman coding trees*)(Suksmono, 2006:24).

4. Runut-Balik (*Backtracking*).

Istilah runut-balik pertama kali diperkenalkan oleh D. H. Lehmer pada tahun 1950. Runut-balik (*backtracking*) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih baik. Runut-balik merupakan perbaikan dari algoritma *brute-force,* secara sistematis mencari persoalan di antara semua kemungkinan solusi yang ada.

Algoritma runut-balik banyak diterapkan untuk program game. Seperti, permainan tic-tac-toe, menemukan jalan keluar dalam sebuah labirin, catur, dan masalah-masalah pada bidang kecerdasan buatan (*artificial intelligence*).

Algoritma backtracking mempunyai prinsip dasar yang sama seperti *brute- force* yaitu mencoba segala kemungkinan solusi. Perbedaan utamanya adalah pada ide dasarnya, semua solusi dibuat dalam bentuk pohon solusi (pohon ini tentunya berbentuk abstrak) dan algoritma akan menelusuri pohon tersebut secara DFS (*Depth First Search*) sampai ditemukan solusi yang layak nama backtracking didapatkan dari sifat algoritma ini yang memanfaat karakteristik himpunan solusinya yang sudah disusun menjadi suatu pohon solusi. Agar lebih jelas bisa dilihat pada pohon solusi berikut:



Gambar 2.5 Pohon Solusi

Misalkan pohon di atas menggambarkan solusi dari suatu persoalan. Jika ingin mencari solusi dari A ke E, maka jalur yang harus ditempuh adalah (A-B-E). Demikian juga untuk solusi-solusi yang lain. Algoritma backtracking akan memeriksa jalur secara DFS, yaitu dari solusi terdalam pertama yang ditemui yaitu solusi E. Jika ternyata E bukanlah solusi yang diharapkan,maka pencarian akan dilanjutkan ke F. Jalur yang harus dilalui untuk bisa mencapai E adalah (A-B-E) dan untuk mencapai F adalah (A-B-F). Kedua solusi tersebut memiliki jalur awal yang sama, yaitu (A-B). Jadi, dari pada memeriksa ulang jalur dari A kemudian B, maka jalur (A-B) disimpan dulu dan langsung memeriksa solusi F. Untuk kasus pohon yang lebih rumit, cara ini dianggap lebih efisien daripada jika menggunakan algoritma *Brute-Force*.

METODE PENELITIAN

Dalam kajian ini penulis menggunakan metode penelitian perpustakaan (*library research).* Adapun langkah-langkahnya sebagai berikut:

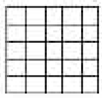
1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti aturan pencarian mendalam (DFS).
2. Simpul-simpul yang sudah dilahirkan dinamakan simpul hidup(*live node*).
3. Simpul hidup yang sedang diperluas dinamakan simpul-E (*Expand-node*).
4. Tiap kali simpul-E diperluas,lintasan yang dibangun olehnya bertambah panjang.
5. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut "diputus" sehingga menjadi simpul mati (*dead node*).
6. Fungsi yang digunakan untukmembunuh simpul-E adalah dengan menerapkan fungsi pembatas (*bounding function*).
7. Simpul yang sudah mati tidak akan pernah diperluas lagi.
8. Jika pembentukan lintasan berakhir dengan simpul mati, maka prose pencarian diteruskan dengan membangkitkan simpul ana yang lainnya.
9. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut-balik ke simpul hidup yang lebih dekat
10. Selanjutnya simpul ini menjadi simpul-E yang baru.
11. Pencarian dihentikan bila kita telah menemukan solusi atau tidak ada lagi simpul hidup untuk runut-balik.
12. Membuat kesimpulan
    Kesimpulan dalam penelitaian ini berupa rumusan umum dalam pencarian penyelesaian penempatan ratu pada papan *m* x *m*
13. Melaporkan
    Langkah terakhir dari kajian adalah menyusun laporan dari penelitian yang telah dilakukan.

PEMBAHASAN

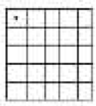
Pada pembahasan ini queen ditepatkan pada papan *m x m,* sehingga tidak ada queen yang saling memakan, dengan menggunakan metode backtracking. Papan *m x m*yang digunakan adalah 5 x 5 ,dan 7 x 7. Quee pada papan disimbolkan dengan (•) dimulai dengan papan 5 x 5 dan cara pengisiannya berdasarkan urutan kolom,dimulai dari kolom 1,kolom 2 dan seterusnya.

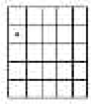
3.1.1 Papan ukuran 5 x 5

1. Papan dengan ukuran 5 x 5, kosong



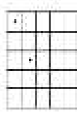
2. Tempat queen pada A11 dan A21

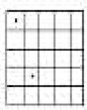


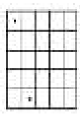


Bagian 1 Bagian 2

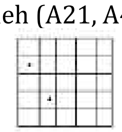
3. Penempatan queen pada bagian 1 diperoleh (A11, A32),(A11,A42) dan (A11,A52)

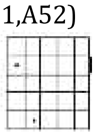




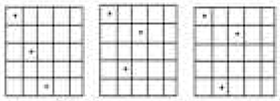


4. Penempatan queen pada bagian 2 diperoleh (A21, A42) dan (A11,A52)

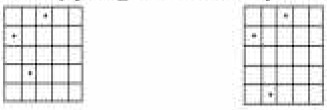




5. Pada bagian 1 penempatan queen pada (A11, A32) yang kolom ke-3 pada A53 ,(A11,A42) yang kolom ke-3 pada A23 dan (A11,A52) yang kolom ke-3 pada A23



Pada bagian 2 Penempatan queen pada (A21, A42) yang kolom ke-3 pada A13 dan (A11,A52) yang kolomke-3 pada A13

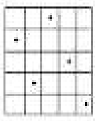


6. Pada bagian 1 penempatan queen pada (A11, A32,A53) yang kolom ke-4 pada A24,(A11,A42,A23) yang kolom ke-4 pada A54 dan (A11,A52,A23) yang kolom ke-4 tidak ada kemungkinan penempatan queen jadi backtracking terakhir pada (A11,A52,A23)

Pada bagian 2 Penempatan queen pada (A21, A42,A13) yang kolom ke-4 pada A34 dan (A11,A52,A13) yang kolomke-4 pada A34.

7. Pada bagian 1 penempatan queen pada (A11,A32,A53,A24) yang kolom ke-5 pada A45 dan pada (A11,A42,A23,A54) yang kolom ke-5 pada A35

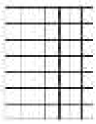
Pada bagian 2 Penempatan queen pada (A21, A42,A13,A34) yang kolom ke-5 pada A55 dan (A11,A52,A13,A34) yang kolom ke-5 tidak ada kemungkinan penempatan queen lagi.



8. Jadi solusi untuk papan 5 x5 yang tepat adalah A21, A42,A13,A34,A55 dan A24,A45, A11,A32,A53

3.1.2 Papan Ukuran 7 x 7

1. Papan dengan 7 x 7, kosong

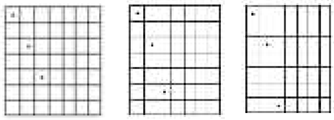


2. Tempat queen pada A11 dan A21

Bagian 1 Bagian 2

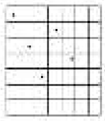
3. Penempatan queen pada A11 kolom ke-2 pada A32 dan A21 kolom ke-2 pada A42

4. Bagian 1penempatan queen pada A11,A32 kolom ke-3 pada A53,A63 dan A73



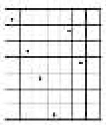
5. Bagian 1 penempatan queen pada A11,A32,A53kolom ke-4 pada A24 dan A74

6. Bagian 1 penempatan queen pada A11,A32,A53,A24 kolom ke-5 pada A45

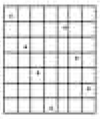


7. Bagian 1 penempatan queen pada A11,A32,A53,A74 kolom ke-5 pada A25 dan A45

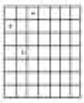
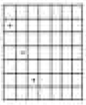
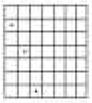
8. Bagian 1 penempatan queen pada A11,A32,A53,A24,A45 kolom ke-6 tidak ada kemungkinan penempatan queen.
9. Bagian 1 penempatan queen pada A11,A32,A53,A74, A25 kolom ke-6 pada A46



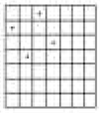
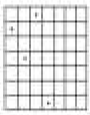
10. Bagian 1 penempatan queen pada A11,A32,A53,A74, A45 kolom ke-7 tidak ada kemungkinan lagi.
11. Bagian 1 penempatan queen pada A11,A32,A53,A74, A25,A46 kolom ke-7 pada A67



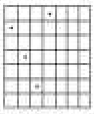
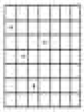
12. Bagian 2 penempatan queen pada A21,A42 kolom ke-3 pada A13, A63 dan A 73

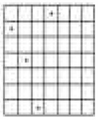
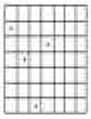
13. Bagian 2 penempatan queen pada A21,A42 ,A13 kolom ke-4 pada A34 dan A74

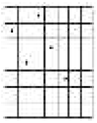
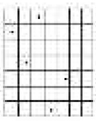
14. Bagian 2 penempatan queen pada A21,A42 ,A63 kolom ke-4 pada A14 dan A34

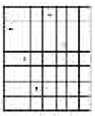
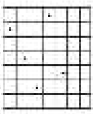
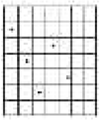
15. Bagian 2 penempatan queen pada A21,A42 ,A73 kolom ke-4 pada A14 dan A34

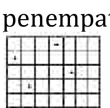
16. Bagian 2 penempatan queen pada A21,A42 ,A13,A34 kolom ke-5 pada A55 dan A21,A42 ,A13,A74 kolom ke-5 pada A55

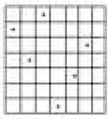
17. Bagian 2 penempatan queen pada A21,A42 ,A63,A14kolom ke-5 pada A35, A55dan A21,A42 ,A63,A34 kolom ke-5 pada A55

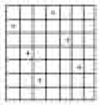
18. Bagian 2 penempatan queen pada A21,A42 ,A73,A14 kolom ke-5 pada A35 dan A21,A42 ,A73,A34 kolom ke-5 tidak kemungkinan lagi penempatan queen.



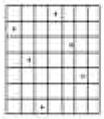
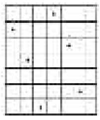
19. Bagian 2 penempatan queen pada A21,A42 ,A13,A34,A55 kolom ke-6 tidak ada kemungkinan penempatan queen lagi dan A21,A42 ,A13,A74,A55kolom ke-6 pada A36



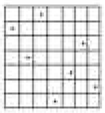
20. Bagian 2 penempatan queen pada A21,A42 ,A63,A14,A35 kolom ke-6 pada A56,, A21,A42 ,A63,A14,A55 kolom ke-6 tidak ada kemungkinan lagi dan A21,A42 ,A63,A34,A55 kolom ke-6 tidak ada kemungkinan lagi



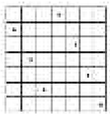
21. Bagian 2 penempatan queen pada A21,A42 ,A73,A14,A35 kolom ke-6 pada A56 dan A66

22. Bagian 2 penempatan queen pada A21,A42 ,A73,A14,A35 kolom ke-6 pada A56 dan A66 pada kolom ke-7 tidak ada kemungkinan penempatan queen lagi.
23. Bagian 2 penempatan queen lagi dan A21,A42 ,A13,A74,A55,A36 kolom ke-7 pada A74



24. Bagian 2 penempatan queen pada A21,A42 ,A63,A14,A35,A56 dan pada kolom ke-7 pada A77



25. Jadi solusi untuk papan 7 x 7 adalah A21,A42 ,A63,A14,A35,A56 dan A11,A32,A53,A74, A25,A46,A67

Lemma

Untuk m x m dengan m bilangan asli ganjil dapat diperoleh rumus umum :

1. *Ai,*$\frac{1}{2}i$ untuk *i* genap dengan $\frac{1}{2}i \le j \le \frac{1}{2}(m-1)$
   Ai, $\frac{1}{2}(m-1)$+t untuk *i* ganjil dan t adalah bilangan asli dengan $\frac{1}{2}(m-1)+1 \le j \le m$ serta *i* dan t diisi secara serentak.
2. *Ai,*$\frac{1}{2}(m-1)+(t+1)$ untuk *i* genap dan t adalah bilangan asli dengan $\frac{1}{2}(m-1)+(t+1) \le j \le m$ serta *i* dan t diisi secara serentak.
   *Ai,*t untuk *i* ganjil dan t adalah bilangan asli dengan $1 \le j \le \frac{1}{2}(m-1)+1$ serta i dan t diisi secara serentak.

Bukti:

1. Akan dibuktikan untuk *i* genap maka $j=\frac{1}{2}i$

i=2	maka j=1
i=4	maka j=2
i=6	maka j=3
i=2k	maka j=k

jadi terbukti i=k maka $j=\frac{1}{2}i$

Akan dibuktikan untuk *i* ganjil $j=\frac{1}{2}(m-1)+t$

m=5+(t-1)2

t=1	m=5	j= 2+t
t=2	m=7	j= 3+t
t=h	m=5+2h-2	$j=\frac{1}{2}(2h+3-1)+t=(h+1)+t$
t=h+1	m=5+2h	$j=\frac{1}{2}(2h+5-1)+t=(h+2)+t)$

jadi terbukti i=2h+5 maka $j=\frac{1}{2}(m-1)+t$

2. Akan dibuktikan *i* genap bahwa $j=\frac{1}{2}(m-1)+(t+1)$

m=5+(t-1)2

t=1	m=5	j= 2+(t+1)
t=2	m=7	j= 3+(t+1)
t=h	m=5-2h-2	$j=\frac{1}{2}(2h+3-1)+t=(h+1)+(t-1)$
t=h+1	m=5-2h	$j=\frac{1}{2}(2h+5-1)+t=(h+2)+(t-1)$

jadi terbukti untuk i=2h+5 maka $j=\frac{1}{2}(m-1)+(t+1)$

Akan dibuktikan untuk i ganjil maka j=t

i=1	maka j=1
i=3	maka j=2
i=5	maka j=3
i=7	maka j=4

Lebih jelasnya dapat dilihat pada Eko Satrio B. Utomo(2010).

PENUTUP

Kesimpulan

Berdasarkan hasil pembahasan, maka dapat diambil kesimpulan,antara lain:

1. *Ai,*$\frac{1}{2}i$ untuk *i* genap dengan $\frac{1}{2}i \le j \le \frac{1}{2}(m-1)$
   Ai,$\frac{1}{2}(m-1)+t$ untuk *i* ganjil dan t adalah bilangan asli dengan $\frac{1}{2}(m-1)+1 \le j \le m$ serta *i* dan t diisi secara serentak.
2. *Ai,*$\frac{1}{2}(m-1)+(t+1)$untuk *i* genap dan t adalah bilangan asli dengan $\frac{1}{2}(m-1)+(t+1) \le j \le m$ serta *i* dan t diisi secara serentak.
   *Ai,*t untuk *i* ganjil dan t adalah bilangan asli dengan $1 \le j \le \frac{1}{2}(m-1)+1$ serta i dan t diisi secara serentak.

DAFTAR PUSTAKA

[1] Chartrand, Gery And Lesniak, Linda.1986. *Graph and Digraph Second Edition.California* : A Division Of Wadsworth,Inc

[2] Munir, Renaldi. 2004. *Algoritma Runut-balik (Backtracking).* Bandung. Informatika

[3] Sukmono,Andriyan B. 2006. *Matematika Diskrit dan Aplikasinya*.itb

[4] Utomo, Eko Satrio B. 2010. *N-Queen Problem dengan Algoritma Backtracking(Rubut-Balik).* Uin Malang: Skripsi, tidak diterbitkan.