



Pembelajaran Teorema-Teorema Klasik Dalam Struktur Aljabar Menggunakan GAP

Ema Carnia¹, Sisilia Sylviani²,
Universitas Padjadjaran
ema.carnia@unpad.ac.id

Abstract

This paper gives an idea of the methods of learning algebra structure using the software GAP (Group, Algorithm, and Programming). Classic theorem is often used in the subject of algebraic structures, including finite abelian group theorem, Lagrange theorem, and isomorphism theorem. Understanding the theorem and the proof is not easy for students. This paper offered a method that can be used by students when studying a theorem through some structured experiments using GAP. Such experiments would eventually lead them to make a conjecture. So hopefully the students can construct an understanding of a theorem, and it will become much easier for them to remember and understand the theorem.

Keywords: classic theorem, abstract algebra, GAP, Algebraic

PENDAHULUAN

Pada paper ini digambarkan salah satu metoda pembelajaran struktur aljabar dengan menggunakan bantuan komputer. Metoda tersebut digunakan untuk mempelajari teorema-teorema klasik (khususnya) dalam mata kuliah struktur aljabar. Teorema klasik merupakan teorema-teorema yang seringkali dijumpai dalam pembelajaran struktur aljabar. Dalam paper ini metode tersebut diilustrasikan dengan tiga contoh dengan menggunakan perangkat lunak GAP (Group, Algorithm, and Programming). Dua contoh pertama didesain sebagai rangkaian latihan pada GAP yang disusun untuk mengarahkan mahasiswa kepada pembuatan konjektur dari suatu teorema. Konjektur tersebut kemudian dapat ditulis secara formal sebagai teorema dan dibuktikan pada saat tatap muka di kelas atau dalam pekerjaan rumah. Contoh ketiga adalah demonstrasi yang dilakukan oleh dosen yang disusun untuk dapat mengarahkan mahasiswa untuk memprediksi suatu teorema.

Dalam pembelajaran struktur aljabar yang tradisional, biasanya dosen menyatakan suatu teorema lalu membuktikannya dengan suatu rangkaian teorema lain (mungkin juga dengan beberapa contoh yang mendemonstrasikan teorema). Hal yang sering dijumpai pada tipe pembelajaran secara klasik tersebut adalah kebanyakan mahasiswa tidak dapat memperoleh gambaran yang konkrit dari teorema yang diberikan. Hal tersebut mengakibatkan mahasiswa mengalami kesulitan untuk memahami teorema tersebut. Pada akhirnya, yang sering terjadi adalah, hasil evaluasi yang diperoleh oleh mahasiswa menjadi rendah.

Berkenaan dengan hal tersebut, dalam paper ini dibahas suatu alternatif metode pembelajaran yang dapat digunakan oleh dosen dalam pembelajaran struktur aljabar. Metode ini diharapkan dapat membuat mahasiswa lebih mudah untuk memperoleh gambaran konkrit dari teorema-teorema yang dijelaskan. Metode pembelajaran yang dilakukan berbasis penemuan. Artinya mahasiswa diarahkan untuk dapat “menemukan” beberapa teorema yang bahkan belum mereka peroleh di dalam kelas.

Metode yang digunakan ini memanfaatkan software GAP untuk mengeksplorasi berbagai latihan yang mengarahkan mahasiswa untuk dapat melihat pola yang dimaksud dalam teorema. Keuntungan dari pendekatan ini adalah mahasiswa dapat mengambil alih “kepemilikan” teorema tersebut, karena mereka yang (seolah-olah) menemukannya sebelum teorema tersebut diberikan kepada mereka. Hal ini dapat membuat proses pembelajaran lebih dapat dinikmati (enjoyable) dan teorema tersebut dapat lebih lama tertanam dalam otak mereka. Yang paling penting adalah proses ini membantu mahasiswa mempunyai pemahaman yang konkrit terhadap teorema. Ditambah lagi dengan system software yang pada beberapa situasi dapat membantu dosen untuk dapat memberikan demonstrasi dan aplikasi dari teorema tersebut.



Namun, kendala yang dapat dihadapi dalam menerapkan cara ini adalah, cara ini memakan waktu yang tidak sedikit. Ditambah lagi terdapat beberapa teorema yang tidak mudah untuk “ditemukan” dengan sendirinya oleh mahasiswa. Namun demikian, dengan menugaskan beberapa pekerjaan rumah dengan menggunakan komputer (daripada memakan banyak waktu di kelas) dan hanya menggunakan teknik ini pada saat penggunaan software memberikan kontribusi yang signifikan untuk pemahaman mahasiswa, software GAP dapat menjadi alat pedagogi yang sangat bermanfaat.

Tiga teorema yang dijelaskan dalam paper ini termasuk teorema yang dapat dengan mudah diarahkan kepada mahasiswa untuk dapat ditemukan. Tujuan penulis adalah untuk mendorong para dosen yang lain untuk dapat menggunakan teknik yang serupa yang dilakukan penulis namun dengan teorema yang berbeda, yang masing-masing dimiliki oleh dosen tersebut. Dua contoh pertama yang dijelaskan dalam artikel ini merupakan rangkaian latihan yang dapat dilakukan mahasiswa pada grup kecil ataupun secara mandiri. Kedua contoh ini dapat diperkenalkan dosen selama KBM di kelas dan kemudian memberikan tugas rumah untuk melengkapinya. Contoh ketiga dapat bekerja dengan baik ketika dosen mendemonstrasikannya dan mahasiswa berpartisipasi dengan merespon pertanyaan-pertanyaan yang diberikan dosen selama melakukan demonstrasi.

CONTOH 1: TEOREMA FUNDAMENTAL GRUP ABELIAN HINGGA

Contoh pertama ini menunjukkan bagaimana mahasiswa dapat “menemukan” satu ide utama yang mengarah ke teorema fundamental grup abelian hingga dengan menggunakan GAP. Berikut ini adalah bunyi dari teorema tersebut.

Teorema 1. Misalkan Z_m dan Z_n adalah grup abelian (hingga), maka $Z_m \oplus Z_n$ isomorfik dengan Z_{mn} jika dan hanya jika m dan n relatif prima.

Mahasiswa dapat membuat konjektur dari fakta tersebut dengan menggunakan bantuan GAP dengan menggunakan perintah *DirectProduct* dan *IsomorphisGroups*. Pertama, mahasiswa melakukan latihan berikut :

1. Cari banyaknya elemen dari masing-masing order dalam $Z_2 \oplus Z_2 \oplus Z_2$, $Z_4 \oplus Z_2$, dan Z_8 .

Pertama, mahasiswa ditugaskan untuk melakukan langkah tersebut dengan cara manual (tanpa bantuan GAP). Karena banyaknya elemen berorder 2 (misal) berbeda dalam masing-masing grup, maka mahasiswa dapat melihat bahwa grup-grup tersebut tidak isomorfik. Kemudian mahasiswa diminta menggunakan GAP untuk memeriksa hal tersebut.

```
gap> z2:=CyclicGroup(2);; z4:=CyclicGroup(4);;
z8:=CyclicGroup(8);;
gap> d1:=DirectProduct(z2,z2,z2);;
gap> d2:=DirectProduct(z4,z2);;
```

Perintah pada baris pertama mendefinisikan $z2$ sebagai grup siklik berorder 2, $z4$ grup siklik berorder 4, dan $z8$ grup siklik berorder 8.

Perintah selanjutnya mendefinisikan $d1$ sebagai grup $Z_2 \oplus Z_2 \oplus Z_2$, dan $d2$ sebagai $Z_4 \oplus Z_2$. Perintah di bawah ini digunakan untuk menunjukkan elemen-elemen dari masing masing grup di atas:

```
gap> Collected(List(Elements(z8),Order));
[[1,1],[2,1],[4,2],[8,4]]
gap> Collected(List(Elements(d1),Order));
[[1,1],[2,7]]
gap> Collected(List(Elements(d2),Order));
[[1,1],[2,3],[4,4]]
```

Output-output di atas menunjukkan banyaknya elemen dalam masing-masing grup beserta ordernya. Sebagai contoh perintah (List(Elements(z8),Order)), menyatakan bahwa terdapat



satu elemen yang berorder 1, satu elemen yang berorder 2, dua elemen yang berorder 4, dan empat elemen dengan order 8 pada grup siklik berorder 8.

Mahasiswa juga dapat langsung memeriksa apakah grup-grup tersebut isomorfik.

```
gap> IsomorphismGroups (d1, d2) ;  
fail  
gap> IsomorphismGroups (d1, z8) ;  
fail  
gap> IsomorphismGroups (d2, z8) ;  
fail
```

Perintah pertama di atas, memeriksa apakah terdapat suatu isomorfisma diantara grup $d1$ dan $d2$. Karena GAP mengeluarkan hasil “fail”, artinya tidak terdapat isomorfisma diantara mereka berdua. Untuk langkah selanjutnya, mahasiswa dapat menggeneralisasi hasil di atas untuk suatu prima p .

2. Periksa apakah $Z_2 \oplus Z_3$ dan Z_6 isomorfik.

Perhatikan kembali bahwa mahasiswa dapat menggunakan perintah GAP yang sama dengan atas dengan sedikit penyesuaian. Penyesuaian yang mereka buat akan membantu mereka memahami teori. Pada langkah ini mahasiswa sudah dapat menyadari bahwa $Z_m \oplus Z_n$ tidak selalu isomorfik dengan Z_{mn} . Langkah selanjutnya akan memberi gambaran yang lebih jelas tentang syarat yang harus dipenuhi agar $Z_m \oplus Z_n$ isomorfik dengan Z_{mn} .

3. Dengan menggunakan GAP, periksa apakah pasangan-pasangan grup berikut isomorfik?

- $Z_4 \oplus Z_{12}, Z_{48}$
- $Z_3 \oplus Z_6, Z_{18}$
- $Z_4 \oplus Z_8, Z_{32}$
- $Z_4 \oplus Z_5, Z_{20}$
- $Z_3 \oplus Z_{12}, Z_{36}$
- $Z_3 \oplus Z_{25}, Z_{75}$

Setelah mengerjakan keenam latihan di atas, mahasiswa sudah dapat membuat suatu konjektur. Perhatikan bahwa latihan 3 ini dapat diselesaikan dengan menugaskan beberapa kelompok pasangan mahasiswa untuk mengerjakannya dan setiap kelompok dapat membandingkan jawabannya.

4. Buat suatu konjektur dari latihan-latihan di atas.

Mahasiswa diminta untuk menuliskan konjektur yang mereka peroleh. Kemudian, dosen memverifikasi konjektur tersebut. Apabila konjektur yang mereka telah sesuai dengan teorema yang ada (atau setiudaknya mendekati), Dosen dapat langsung memberikan teorema tersebut secara formal. Kemudian mahasiswa diminta untuk membuktikan konjektur yang mereka buat. Pembuktian tersebut tentu saja dilakukan dengan bimbingan dari Dosen.

CONTOH 2: TEOREMA ORBIT STABILIZER

Rangkaian latihan ini akan mengarahkan mahasiswa untuk dapat membuat suatu konjektur pencarian order dari grup G yang ditinjau dari order orbit dan stabilizer yang terkait dengan grup tersebut. Perintah $\text{Orbit}(G,s)$ pada GAP memberikan output orbit s di G . Perintah $\text{Stabilizer}(G,s)$ menghasilkan subgroup dari G yang merupakan stabilizer dari s . Pada bagian ini akan dijelaskan latihan-latihan yang dapat mengarahkan mahasiswa untuk menemukan teorema orbit-stabilizer.

Teorema 2. Misalkan G adalah grup permutasi dari himpunan S . Untuk setiap s di S stabilizer s di G adalah subgroup dari G yang didefinisikan $\{g \in G \mid g(s) = s\}$. Orbit dari s pada G



adalah subset dari S yang didefinisikan $\{g(s) | g \in G\}$. Order dari G sama dengan order dari $\text{Stabilizer}(G,s)$ dikali dengan order dari $\text{Orbit}(G,s)$.

Dengan kata lain, teorema tersebut menjelaskan bahwa untuk sebarang s anggota S berlaku $|G| = |\text{Stabilizer}(G,s)| |\text{Orbit}(G,s)|$. Langkah-langkah yang dapat dilakukan adalah

1. Cari banyaknya elemen di $\text{Orbit}(G,s)$ untuk G grup simetri dengan order 3 untuk $s = 1, 2, 3$,
2. Cari banyaknya elemen di $\text{Orbit}(G,s)$ untuk G grup simetri dengan order 4 dan untuk $s = 2, 3, 4$
3. Cari banyaknya elemen di $\text{Orbit}(G,s)$ untuk G grup simetri dengan order 5 dan untuk $s = 2, 3, 4$
4. Ulangi latihan 1 dan 2 untuk grup simetri dengan order 10, 20, 30.

Berikut ini adalah langkah-langkah pada GAP yang dapat dilakukan oleh mahasiswa untuk grup dihedral dengan order 20 dan 100. Perhatikan pada contoh sebelumnya, hanya beberapa perintah GAP yang perlu diperkenalkan kepada mahasiswa.

```
gap>M:=SymmetricGroup(3);
gap>Size(Orbit(M,1));; Size(Orbit(M,2));; Size(Orbit(M,3));;
3
3
3
gap>Size(Stabilizer(M,1));; Size(Stabilizer(M,2));;
Size(Stabilizer(M,3));;
2
2
2
gap>N:=SymmetricGroup(4);
gap>Size(Orbit(N,2));; Size(Orbit(N,4));; Size(Orbit(N,3));;
5
5
5
gap>Size(Stabilizer(N,2));; Size(Stabilizer(N,4));;
Size(Stabilizer(N,3));;
24
24
gap>P:=SymmetricGroup(5);
gap>Size(Orbit(P,2));; Size(Orbit(P,4));; Size(Orbit(P,3));;
4
4
4
gap>Size(Stabilizer(P,2));; Size(Stabilizer(P,4));;
Size(Stabilizer(P,3));;
6
6
6
```

Perintah yang pertama di atas mendefinisikan G sebagai grup dihedral dengan order 20. Perintah selanjutnya meminta GAP untuk memperoleh ukuran dari orbit 1, 2, 3 dan 4 di G . Output GAP menunjukkan bahwa orbit-orbit tersebut memiliki ukuran 10. Perintah selanjutnya meminta GAP untuk menentukan ukuran dari stabilizer dari 1, 2, 3 dan 4 di G . Output GAP menunjukkan bahwa semua stabilizer memiliki ukuran 2. Perintah sisanya merupakan pengulangan untuk grup dihedral dengan order 100.



Setelah contoh-contoh tersebut dilakukan, maka mahasiswa akan melihat bahwa pada grup dihedral dengan order n banyaknya elemen dalam sebarang stabilizer adalah 2, banyaknya elemen dalam sebarang orbit adalah $n/2$, dan bahwa 2 dikali $n/2$ adalah order dari grup (seperti yang telah dinyatakan dalam teorema orbit-stabilizer). Seperti pada contoh sebelumnya pada langkah kelima, daripada membuat mahasiswa mengerjakan kelima grup dihedral tersebut, akan lebih baik jika menugaskan mereka untuk melihat hasil pada grup dihedral yang berbeda, kemudian membandingkan jawaban mereka.

4. Buat suatu konjektur tentang banyaknya elemen dari Stabilizer(G,s) dan pada Orbit(g,s) untuk sebarang s di $\{1,2,3, \dots, n/2\}$ untuk G grup dihedral dengan order n .

Di titik ini, setiap mahasiswa ataupun kelompok mahasiswa dapat ditugaskan untuk mengerjakan grup-grup non dihedral untuk mencari ukuran dari orbit dan stabilizernya. Sebagai contoh, beberapa mahasiswa dapat ditugaskan untuk mengkaji beragam grup simetrik. Grup ini terdefinisi pada GAP dengan menggunakan perintah SymmetricGroup(n) untuk nilai integer positif tertentu. Perintah GAP yang serupa dengan di atas dapat membuat mahasiswa untuk melihat contoh-contoh lain dengan cepat. Selanjutnya dapat dilakukan perbandingan antar data.

5. Dengan bimbingan Dosen, buktikan konjektur yang telah dibuat.
Pada langkah terakhir ini mahasiswa diminta untuk membuktikan secara manual kebenaran dari konjektur yang mereka buat. Hal ini tentu saja dilakukan dengan bimbingan Dosen.

CONTOH 3: TEOREMA PERTAMA ISOMORFISMA GRUP

Contoh ini untuk mengarahkan mahasiswa untuk memahami teorema pertama isomorfisma grup dengan bantuan GAP.

Teorema 3. Misalkan $\varphi: G \rightarrow G'$ merupakan homomorfisma pada dengan kernel K . Maka $G' \approx G/K$.

Tidak seperti contoh-contoh sebelumnya, dimana mahasiswa diharapkan untuk melakukan rangkaian latihan, maka contoh ini lebih baik digunakan sebagai demonstrasi dari dosen. Yaitu, dosen dapat bekerja melalui contoh ini di depan kelas dengan "memancing" mahasiswa melalui pertanyaan-pertanyaan sepanjang demonstrasi untuk memastikan bahwa mahasiswa selalu terlibat. Contoh ini dapat bekerja dengan sangat baik apabila mahasiswa sudah cukup mengenal GAP. Langkah pengerjaan dimulai demonstrasi dengan mendefinisikan salah satu contoh grup. Pada latihan ini yang contoh grup yang digunakan adalah grup simetri S_4 .

```
gap> G:= SymmetricGroup(4) ; ;
```

Selanjutnya definisikan suatu homomorfisma dari S_4 ke S_4 . Homomorfisma tersebut digambarkan di bawah ini. Konstruksi suatu homomorfisma dengan mengkhususkan image untuk sebuah himpunan dari generator-generator S_4 .

```
gap> f:= GroupHomomorphismByImages(G, G, [(1,2),  
(1,2,3,4)], [(1,2), (1,2)]);  
[(1,2), (1,2,3,4)] -> [(1,2), (1,2)]
```

Perintah di atas mendefinisikan secara tunggal homomorfisma $f: S_4 \rightarrow S_4$ yang memetakan generator (12) dan (1234) di S_4 ke (12)(12). GAP akan mengeluarkan output "fail" jika kita mencoba mengkonstruksikan pemetaan yang bukan merupakan homomorfisma. Selanjutnya, kita



dapat “bertanya” pada GAP elemen yang mana yang berada pada kernel dari homomorfisma tersebut.

```
gap> Elements (Kernel (f));  
[ (), (2,3,4), (2,4,3), (1,2)(3,4), (1,2,3),  
(1,2,4), (1,3,2), (1,3,4), (1,3)(2,4), (1,4,2),  
(1,4,3), (1,4)(2,3) ]
```

Perhatikan bahwa GAP menotasikan identitas dari elemen dari suatu grup permutasi dengan (). Pada titik ini, dosen dapat menanyakan pada mahasiswa apakah mereka mengenal kernel tersebut. Dengan memberikan beberapa petunjuk, mahasiswa dapat melihat bahwa kernel ini merupakan grup A_4 . Dosen dapat bertanya pada kelas, apakah yang mereka ketahui tentang grup S_4/A_4 . Petunjuk yang dapat diberikan adalah bahwa S_4/A_4 isomorfik dengan grup yang sudah dikenal mahasiswa. Dengan cara yang serupa, mahasiswa dapat diberi pertanyaan apakah prediksi mereka untuk grup $f(S_4)$. Dosen dapat menguji prediksi mahasiswa dengan menggunakan perintah GAP di bawah ini:

```
gap> Elements (Image (f));  
[ (), (1,2) ]  
gap> IsomorphismGroups (CyclicGroup (2), Image (f));  
[ f1 ] -> [ (1,2) ]  
gap> IsomorphismGroups (CyclicGroup (2),  
FactorGroup (G, Kernel (f)));  
[ f1 ] -> [ f1 ]
```

Perintah pertama meminta elemen-elemen apa saja yang ada di $f(S_4)$. Outputnya menunjukkan bahwa image ini isomorfik dengan Z_2 . Perintah kedua sampai ketiga menguji apakah $f(S_4)$ dan S_4/A_4 masing-masing isomorfik dengan S_4/A_4 (seperti yang diprediksi mahasiswa). Pada output, GAP memberikan nama *f1* untuk generator dari grup siklik order 2 dan generator dari grup factor S_4/A_4 . Pada langkah ini dosen dapat memberikan penjelasan pada observasi yang dilakukan bahwa $f(S_4)$ isomorfik dengan S_4/A_4 .

Demonstrasi dari dosen dapat dilanjutkan dengan membandingkan $f(S_4)$ dan $S_4/Ker(f)$ untuk homomorfisma yang lain yang domainnya S_4 . Dengan menugaskan kepada mahasiswa untuk mencari homomorfisma-homomorfisma tersebut dapat mengarah kepada diskusi yang menarik tentang bagaimana memperoleh suatu pemetaan yang homomorfisma.

Langkah selanjutnya, buat contoh lain homomorfisma f dari S_4 ke S_4 . Misalkan, konstruksi homomorfisma dari generator (12) dan (1234) dari S_4 ke (12) dan (13).

```
gap> f:= GroupHomomorphismByImages (G, G, [(1,2),  
(1,2,3,4)], [(1,2), (1,3)]);  
[ (1,2), (1,2,3,4) ] -> [ (1,2), (1,3) ]  
gap> Elements (Kernel (f));  
[ (), (1,2)(3,4), (1,3)(2,4), (1,4)(2,3) ]
```

Perintah kedua di atas menguraikan elemen-elemen dari kernel dari homomorfisma yang baru. Di sini dosen dapat menugaskan mahasiswa untuk memprediksi grup membandingkan $f(S_4)$ dan $S_4/Ker(f)$. Jika mahasiswa belum dapat menjawab pertanyaan tersebut, dosen dapat memberikan arahan dengan pertanyaan berikut:

1. Ingat jika order dari S_4 adalah 24, dan order dari $Ker(f)$ adalah 4, maka berapakah order dari grup $S_4/Ker(f)$?
Pertanyaan ini akan mengarahkan mahasiswa untuk menyadari bahwa $S_4/Ker(f)$ adalah grup berorder 6. Jika mereka masih belum dapat menjawab pertanyaan tersebut, dosen dapat memberikan petunjuk berupa pertanyaan berikut.



2. Apakah grup factor $S_4/Ker(f)$ abelian?

Pertanyaan ini dapat dijawab dengan melakukan komputasi pada GAP, sebagai contoh,

$(12)ker(f)*(23)Ker(f)=(12)*(23)ker(f)$, dan

$(23)ker(f)*(12)ker(f)=((23)*(12))Ker(f)$

Pada GAP perintah tersebut tertulis sebagai berikut:

```
gap> (1,2)*(2,3)*Elements(Kernel(f));  
[ (1,3,2), (1,4,3), (2,3,4), (1,2,4) ]  
gap> (2,3)*(1,2)*Elements(Kernel(f));  
[ (1,2,3), (2,4,3), (1,4,2), (1,3,4) ]
```

Perhatikan bahwa GAP mengalikan komposisi permutasi dari kiri ke kanan, sebagai contoh $(12)*(23)$ akan menghasilkan (132) . Karena $(12)ker(f)*(23)Ker(f)$ tidak sama dengan $((23)*(12))Ker(f)$ maka grup factor $S_4/Ker(f)$ bukan grup abelian. Selanjutnya Dosen dapat membrikan pertanyaan kepada mahasiswa. Pertanyaan tersebut adalah

3. Grup non abelian dengan order apakah yang pernah dikenal?

Setelelah mahasiswa membuat prediksi mereka, dosen dapat menginputkan dalam GAP.

```
gap> Elements(Image(f));  
[ (), (2,3), (1,2), (1,2,3), (1,3,2), (1,3) ]
```

Perintah di atas menguraikan elemen-elemen di $f(S_4)$, yang jelas merupakan grup simetri S_3 . Sekarang gunakan GAP untuk melihat apakah $f(S_4)$ dan $S_4/Ker(f)$ isomorfik dengan S_3 .

```
gap> IsomorphismGroups(SymmetricGroup(3),  
Image(f));  
[ (1,2,3), (1,2) ] -> [ (1,3,2), (1,3) ]  
gap> IsomorphismGroups(SymmetricGroup(3),  
FactorGroup(G, Kernel(f));  
[ (1,2,3), (1,2) ] -> [ f2, f1*f2 ]
```

Sehingga mahasiswa dapat melihat bahwa contoh lain dimana apakah $f(S_4)$ dan $S_4/Ker(f)$ saling isomorfik. Di sini dosen dapat mengemukakan teorema pertama isomorfisma secara formal. Karena mahasiswa telah melihat contoh yang mengilustrasikan teorema tersebut, maka mereka akan memiliki suatu pemahaman yang lebih konkrit dari teorema pertama isomorfisma daripada dengan tanpa menggunakan GAP.

SIMPULAN & SARAN

Ketiga contoh diatas mengilustrasikan bahwa GAP dapat membantu mahasiswa untuk menemukan teorema struktur aljabar. Masih banyak contoh yang dapat dilakukan. Tujuan dari paper ini adalah untuk menunjukkan mudahnya GAP dapat diintegrasikan dengan perkuliahan struktur aljabar dan bagaimana program tersebut dapat membantu mahasiswa untuk memprediksi banyak teorema-teorema klasik yang ditemui dalam perkuliahan ini. Dalam masing-masing contoh hanya sedikit perintah GAP yang perlu diperkenalkan (sehingga meminimalisir waktu untuk membuat mahasiswa menjadi familiar dengan GAP). Dalam waktu yang sama, perintah GAP disesuaikan oleh mahasiswa selama karena mereka melakukan perintah iterasi (dengan tujuan melihat pola). Penyesuaian ini dibuat oleh mahasiswa dengan pendekatan berbasis penemuan untuk memahami teorema-teorema penting dalam struktur aljabar, suatu pendekatan yang diharapkan menghasilkan pemahaman yang lebih cepat dan lebih mendalam terhadap teorema-teorema tersebut



DAFTAR RUJUKAN

- A. Hulpke, 2011, *Abstract Algebra in GAP*, the Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States, California.
- J. A. Gallian, 2010, *Abstract Algebra with GAP for Contemporary Abstract Algebra* 7th edition, Brooks/Cole, Cengage Learning, Boston.
- Kenneth K. Nwabueze, 2004, *Computers in abstract algebra*, *International Journal of Mathematical Education in Science and Technology*,
- Claus Schuberta, Mary Gfeller & Christopher Donohue, 2013, *Using Group Explorer in teaching abstract algebra*, *International Journal of Mathematical Education in Science and Technology*,